PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HARMONIZATION

MASTER TRAINING OFFER

ACADEMIC

Establishment	Faculty / Institute	Department
University of Mohamed Khider Biskra	Faculty of Exact Sciences and natural and life sciences	Computer sciences department

Domain: Mathematics and Computer Sciences

Sector: Computer sciences

Speciality: Software Engineering and Distributed Systems

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

مواءمة

عرض تكوين ماستر

أكاديمي

المؤسسة	الكلية/ المعهد	القسم
جامعة محمد خيضر - بسكرة	كلية العلوم الدقيقة و علوم الطبيعة و الحياة	قسم الإعلام الآلي

الميدان :رياضيات و إعلام آلي

الشعبة : إعلام آلي

التخصص :هندسة البرامج والأنظمة الموزعة

السنة الجامعية:2023-2024

TABLE OF CONTENTS

I – Master’s identity sheet

1 - Formation location:

Faculty: Faculty of exact sciences and natural and life sciences

Department: Computer sciences department

2- External partners *:

- Other partner establishments:

- Businesses and other socio-economic partners:

- International partners:

3 – Context and objectives of the formation

A – Access conditions

IN M1 :

• Academic Degree in Computer Science
• A recognized equivalent title

IN M2 :

- File study

B - Formation objectives

The objectives of this training are to provide the methods and techniques known in the field of Software Engineering for the secure development of distributed systems. Emphasis will be placed, in particular, on modeling, specification, verification and diagnosis activities of embedded, mobile and real-time systems.

Before creating a physical system (hardware or software), it is necessary to go through a design process. This process is made up of several steps. In each, we need to do some specification in order to verify the properties (i.e., requirements). So, we will use specification formalisms and verification techniques.

Once the design is complete, a template must be written. This model will be simulated to remove possible errors. Before physical production, a prototype will be implemented and tested using the prepared tests. These three steps form what we call the critical systems validation process. A real system can fail during operation. A diagnostic process can be applied to determine the faults causing this malfunction. In this training we focus on specification, verification and diagnosis.

C – Profiles and targeted skills *:*

Teachers/trainers mastering the process of validation and diagnosis of hardware and software systems. Thus, training designers by instilling them with the tools used effectively in the industry.

D- Regional and national employability potential

The normal path after completing the Research Master's degree is the preparation of a 3rd cycle doctoral thesis. Students accepted for the Master's degree will be able to integrate into the research laboratories with a view to preparing an LMD doctoral thesis. The Doctorate will thus give them the opportunity to access teaching and research institutions in Higher Education. Recruitment as teacher-researchers and access to preparation for university accreditation with a view to accessing the rank of lecturer; Public research (research organizations and higher education). To public research organizations: CDTA, CDER, etc.

E – Gateways to other specialties

Towards other Masters with equivalent units.

F – Training monitoring indicators

G – Supervisory abilities

30 students for the formation.

4 – Human resources available

A : Teachers working in the specialty

Name, Surname	Graduation Diploma + Speciality	Post-graduation Diploma + Speciality	Grade	Type of intervention	Signature
BENNOUI Hammadi	**Computer sciences engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials, Lab works**	
KAHLOUL Laid	**Computer sciences engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials, Lab works**	
KAZAR Okba	**Computer sciences engineer**	**State Doctorate in computer sciences**	**Prof**	**Lectures, Tutorials**	
BABAHENINI Mohamed Chaouki	**Computer sciences engineer**	**State Doctorate in computer sciences**	**Prof**	**Lectures, Tutorials**	
CHERIF Foudil	**Computer sciences engineer**	**State Doctorate in computer sciences**	**Prof**	**Lectures, Tutorials**	
BITAM Salim	**Computer sciences engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials**	
DJEFFAL AbdElhamid	**Computer sciences engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials, Lab works**	
REZEG Khaled	**Computer sciences engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials**	
SAOULI Rachida	**Computer sciences engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials**	
TERISSA Sadek Labib	**Electronic engineer**	**Habilitation**	**Prof**	**Lectures, Tutorials**	
TIBERMACINE Okba	**Computer sciences engineer**	**Habilitation**	**M.C.A**	**Lectures, Tutorials, Lab works**	
Zernadji Tarek	**Computer sciences engineer**	**Doctorate of sciences in computer sciences**	**M.C.B**	**Lectures, Tutorials, Lab works**	
Kerdoudi Mohamed Lamine	**Computer sciences engineer**	**Habilitation**	**M.C.A**	**Lectures, Tutorials, Lab works**	
BOUKHLOUF Djemaa	**Computer sciences engineer**	**Doctorate of sciences in computer sciences**	**M.C.B**	**Lectures, Tutorials, Lab works**	
TIGANE Samir	**Computer sciences engineer**	**Habilitation**	**M.C.A**	**Lectures, Tutorials, Lab works**	
HEMIDI Zohra	**Computer sciences engineer**	**Doctorate of sciences in computer sciences**	**M.C.B**	**Lectures, Tutorials, Lab works**	
MOHAMEDI Amira	**Computer sciences engineer**	**Magister in computer sciences**	**M.A.A**	**Lectures, Tutorials, Lab works**	
Ramdani Mohamed	**Master in computer sciences**	**LMD Doctorate in computer sciences**	**M.C.B**	**Lectures, Tutorials, Lab works**	

B : External supervision

None

5 – Specific material resources available

A- Educational Laboratories and Equipment: Sheet of existing educational equipment for the practical work of the planned training (1 sheet per laboratory)

Laboratory title: Computing center

N°	Equipment title	Number	Observations
1	PC hp	40	In two rooms
2	PC Other brands	60	In three rooms

Laboratory title: Network Laboratory

N°	Equipment title	Number	Observations
1	Working stations	15	

B- Internship sites and company training:

Training place	Number of students	Stage duration

C- Research laboratories supporting the master's degree:

Head of the laboratory: KAZAR Okba
No. Laboratory approval: Num 242 in 03/04/2013
Date : 05/05/2022 Opinion of the laboratory head:

D- Research projects supporting the master's degree:

Title of the research project	Projet Code	Project start date	Project end date
Développement et Vérification des Systèmes Multi-agents Coopératifs et de leurs Applications	B*01420100023	01/01/2014	31/12/2017

E- Personal work spaces and ICT:

- Intranet space of Mohamed Khider Biskra University.
- Computer center of the computer science department.
- Wireless network (wifi internet) of the IT department:
- Video conference room;
- Library (indicate the number of titles available in the specialty) of 300 titles (books and works) linked to the different areas of computer science in the central library of the university, the faculty library and the department's documentation center.
- Remote teaching room for end of cycle students

II – Semester organization sheet for lessons

1- Semester 1 :

Teaching Unit	SHV	Weekly HV				Coeff	Credits	Evaluation Mode	
	14-16 weeks	Lectures	Tutorial	WL	Continuous			Continuous	Exam
Fondamental TU									
FTU1. Advanced systems and architectures	**189h**	**6h**	**3h**	**4h30**	**4h**	**9**	**18**		
Distributed systems	63h	1h30	1h30	1h30	1h30	3	6	50%	50%
Algorithmics and parallel architectures	63h	1h30	1h30	1h30	1h	3	6	50%	50%
Communication networks	63h	3h	-	1h30	1h30	3	6	33%	67%
Methodology TU									
MTU1. Machine Learning and Specification Tools	**105h**	**3h**	**3h**	**1h30**	**2h**	**4**	**10**		
Machine Learning	63h	1h30	1h30	1h30	1h	2	5	33%	67%
Specification Tools	42h	1h30	1h30	-	1h	2	5	33%	67%
Transversal TU									
TTU1. Entrepreneurship and English	**42h**	**1h30**	**1h30**	**-**	**1h**	**2**	**2**		
English 1	21h	-	1h30	-	-	1	1		100%
Entrepreneurship	21h	1h30	-	-	1h	1	1		100%
Total Semester 1	**336h**	**10h30**	**7h30**	**6h**	**7h**	**15**	**30**		

2- Semester 2 :

Teaching Unit	SHV	Weekly HV				Coeff	Credits	Evaluation Mode	
	14-16 weeks	Lectures	Tutorial	WL	Continuous			Continuous	Exam
Fondamental TU									
FTU2. Modeling and Verification	**84h**	**3h**	**1h30**	**1h30**	**2h30**	**6**	**10**		
Calculability	42h	1h30	1h30	-	1h30	3	5	50%	50%
Techniques and Tools of Verification	42h	1h30	-	1h30	1h	3	5	33%	67%
FTU3. Development and Simulation	**84h**	**3h**	**-**	**3h**	**2h30**	**4**	**10**		
Advanced Development Approaches	42h	1h30	-	1h30	1h30	2	5	50%	50%
Modeling and Simulation	42h	1h30	-	1h30	1h	2	5	33%	67%
Methodology TU									
MTU2. Blockchains and RTS	**84h**	**3h**	**-**	**3h**	**2h**	**4**	**8**		
Blockchain Technologies	42h	1h30	-	1h30	1h	2	4	33%	67%
Real-Time Systems	42h	1h30	-	1h30	1h	2	4	33%	67%
Transversal TU									
TTU2. Methodology	**21h**	**1h30**	**-**	**-**	**1h**	**1**	**2**		
Research methodology	21h	1h30	-	-	1h	1	2		100%
Total Semester 2	**273h**	**10h30**	**1h30**	**7h30**	**8h**	**15**	**30**		

3- Semester 3 :

Teaching Unit	SHV	Weekly HV				Coeff	Credits	Evaluation Mode	
	14-16 weeks	Lectures	Tutorial	WL	Continuous			Continuous	Exam
Fondamental TU									
FTU4. High-level Formalisms and development	**126h**	**4h30**	**-**	**4h30**	**3h**	**9**	**18**		
High-level formal languages	42h	1h30	-	1h30	1h	3	6	33%	67%
Rewriting Logics and Their Applications	42h	1h30	-	1h30	1h	3	6	33%	67%
Distributed Application Development	42h	1h30	-	1h30	1h	3	6	33%	67%
Methodology TU									
MTU3. Model, Architecture and Management	**105h**	**4h30**	**1h30**	**1h30**	**3h**	**5**	**10**		
Model Based Diagnosis	42h	1h30	1h30	-	1h	2	4	33%	67%
Service Oriented Architectures	42h	1h30	-	1h30	1h	2	4	33%	67%
Software Project Management	21h	1h30	-	-	1h	1	2	-	100%
Transversal TU									
TTU 3. English	**21h**	**1h30**	**-**	**-**	**1h**	**1**	**2**		
English 2	21h	1h30	-	-	1h	1	2		100%
Total Semester 3	**252h**	**10h30**	**1h30**	**6h**	**7h**	**15**	**30**		

4- Semester 4 :

Domain : Maths - Computer Sciences
Sector : Computer Sciences
Speciality : Software Engineering and Distributed Systems

Internship in a company culminating in a dissertation and a defense.

The S4 semester is reserved for an internship or introductory research work, culminated by a dissertation and a defense..

	SHV	Coeff	Credits
Personal work			
Internship in company			
Seminars			
other	12h/weekly, i.e. 144 hours for the semester	15	30
Total Semester 4	144h	30h	30h

5- Overall summary of the training: (indicate the separate global HV in progress, Tutorials, for the 04 semesters of teaching, for the different types of EU)

TU / HV	FTU	MTU	DTU	TTU	Total
Course	231	147	-	63	441
Tutorials	84	63	-	21	168
Practical work	189	84	-	00	273
Personal work	168	98	-	42	308
other	144	-	-	-	144
Total	798	392	-	126	1316
Credits	86	28	-	6	**120**
% in credits for each TU	71,66%	23,33%	-	5,00%	100%

III - Detailed program by subject

Title of the Master: Software Engineering and Distributed Systems
Semester : S1
Title of the TU: Advanced systems and architectures
Title of the subject: Distributed systems
Credits: 6
Coefficients: 3

Teaching objectives

This course is dedicated to the fundamental aspects of distributed systems and the challenges involved in their design and implementation. Special emphasis will be placed on addressing the lack of global time in asynchronous systems and on basic techniques for designing fault-tolerant systems.

Recommended prior knowledge

Concepts of processes, synchronization, and communication in a centralized system. In the L cycle of the LMD system, students have taken two courses dedicated to these concepts.

Content of the subject:

I- Concept of competition.
- The different interpretations of competition.

II- Time and state in a distributed system.
- Causality and ordering of events in a distributed system;
- Overall state of a distributed system; consistent cuts applications: save-resume algorithms, detection of stable properties;
- Global scheduling by logical application clocks: mutual exclusion, distributed queues;
- Causal scheduling by vector clocks applications: observation, fine-tuning;
- Synchronization of physical clocks

III- Distributed process cooperation
- Virtual ring, insertion and removal protocols, failure management;
- Application election algorithms: group management;
- Termination detection algorithms. Application: distributed garbage collection.

IV- Fault tolerance
- Failure hypotheses;
- Specification of coherence: linearization, sequential coherence, causal coherence;
- Primary copy and active duplication;Algorithmes de diffusion fiable et gestion de groupes de processus.

V- Distributed information management

- Principles of distributed object management;
- Implementation: virtual memory, distributed objects;
- Large-scale distribution;
- Cache management, duplication, consistency;
- Applications: P2P systems.

Evaluation Mode: 50 % Exam + 25 % Tutorials + 25 % Lab Work

References

1. R. Guerraoui, L. Rodrigues, ***Reliable Distributed Programming***, Springer, 2006.
2. A. S. Tanenbaum, M. van Steen, ***Distributed Systems - Principles & Paradigms***, Prentice Hall, 2002 .
3. S. Mullender (editor), ***Distributed Systems***, 2nd ed. , Addison-Wesley, 1993.
4. M. Singhal, N. G. Shivaratri, ***Advanced Concepts in Operating Systems***, McGraw-Hill, 1994 .
5. V. C. Barbosa, ***Introduction to Distributed Algorithms***, MIT Press, 1996.
6. C.A.R. Hoare, *Communicating Sequential Process* , Prentice Hall Intern. 2004.
7. A. Silberschatz et J.L. Peterson, ***Operating System Concepts*** , Addison-Wesley, 1983.

Title of the Master: Software Engineering and Distributed Systems
Semester : S1
Title of the TU: Advanced systems and architectures
Title of the subject: Algorithmics and parallel architectures
Credits: 6
Coefficients : 3

Teaching objectives

This course introduces students to parallel architectures and algorithms, specifically vector architectures, systolic architectures, and their associated algorithms.

Recommended prior knowledge

Basic architecture of a Von Neumann machine. Indeed, in the LMD system, students have taken courses dedicated to computer architecture in the Bachelor's degree cycle.

Content of the subject:

Chapiter I. Introduction to parallel architectures

- Motivations for parallelism.
- Sequential and parallel architectures.
- Organization of Parallel Architectures.
- Sources of parallelism.
- Sequential VS Parallel Complexity.
- P-RAM (Parallel - Random Access Machine) model and algorithms:
- Illustration of Coole's algorithm for parallel merge sort
- Evaluation of the Performance of parallel programs.

Chapiter II. Parallel programming models

- SPMD type programming model:
  Presentation of the distributed memory model: OpenMPI,
  Presentation of the shared memory model: OpenMP,
- SIMD type programming model:
  GPGPU (general graphics card processing)

Chapiter III. Interconnection networks

- Static graphs and topologies
- Communication protocols in parallel machines:
  Circuit switching, Store & Forward, Wormhole
- The interconnection of dynamic topologies
- Routing in Crossbar networks
- Case study: Hypercube
  (architecture and routing algorithms)

Chapiter IV. High performance parallel architectures

- Reminder on floating point calculation,
- Vector execution and principle of interleaved memory,
- Vectorization conditions: illustration of the raster product,

- Introduction to BLAS (Level 1),
- Illustration of the LU matrix decomposition algorithm,
- Vector instruction and Vectorization of a scalar product,

Evaluation method:

50 % Exam + 25 % Lab works + 25 % tutorials

References.

[1]Legrand et Y. Robert, Algorithmique Parallèle, Dunod (2004).
[2] Olivier Pironneau, Optimisation des performances et Parallélisme en C/C++ - openMP - MPI - UPC - CUDA –openCL, University of Paris VI
[3] Barbara Chapman, Gabriele Jost, Ruud van der Pas. Using OpenMP Portable Shared Memory Parallel Programming, The MIT Press,Cambridge, Massachusetts London, England (2008).
[4] Jean-Paul Sansonnet. Architectures des machines parallèles CNRS 1992.
[5] A. Legrand et Y. Robert. Algorithmique Parallèle. Dunod (2004).
[6] Philippe MARQUET. Programmation parallèle et distribuée. Université des sciences et technologies de Lille. 2008

Title of the Master: Software Engineering and Distributed Systems

Semester : S1

Title of the TU: Advanced systems and architectures

Title of the subject: Communication networks

Credits: 6

Coefficients: 3

Teaching objectives

The communication networks subject aims to enable the student to:

- Understand and be able to implement various devices and protocols used in both fixed and mobile networks.
- Design, configure, and manage fixed and mobile networks, considering the requirements of applications.

Recommended prior knowledge Networks (Bechlor 2), distributed systems.

Content of the material:

1- Reminder on the link layer

- Concurrent access to support, the CSMA/CD protocol, ...

2- The network layer

- Reminder on the IPv4 protocol
- Routing in IP networks:
  * Distance vector routing protocols
  * Link-state routing protocols.
- The IPv6 protocol
- Management of coexistence between IPv4 and IPv6 networks.
  * Coexistence based on Tunneling.
  * Coexistence based on Gateways.

3- The transport layer

- TCP protocols
- The UDP protocol

4- Quality of service (QoS) in IP networks

- QoS settings
- Network factors that influence the QoS of communications.
- TCP/IP and QoS protocols

5- Introduction to wireless communication networks

- Cellular wireless networks
- Ad Hoc wireless networks

Evaluation Mode : 50 % Exam + 25 % Lab works + 25 % Personal work

References

- Kurose, J. F., & Ross, K. W. (2021). Computer Networking A Top-Down Approach. Peasron Editions. 2021.
- Bonaventure, O., Networking : Principles, Protocols and Practice, 3rd Edition, 2021.
- Lannone, E., Telecommunication networks, O'Reilly Edition, 2017.

Title of the Master: Software Engineering and Distributed Systems

Semester: S1

Title of the TU: Machine Learning and Specification Tools

Title of the subject : Specification Tools

Credits: 5

Coefficients: 2

Teaching objectives.

This material represents the second part of the specification tools material. It aims to present other formalisms which extend finite state automata: Petri Nets and Algebraic Specifications.

1. **Recommended prior knowledge**
2. **Content of the material:**

    1. Reminder about automata
    2. Concurrence semantics
    3. Petri Nets (PNs)
        a. Definition
        b. Analysis techniques
    4. Extensions of PNs
    5. Introduction to algebraic specification techniques

Evaluation Mode:

67 % Exam + 33 % Tutorials

References

- G.W. Brams, Réseaux de Petri : Théorie et Pratique, Masson, 1983. (ISBN 2-903607-12-5)
- Annie Choquet-Geniet, Les réseaux de Petri : Un outil de modélisation, Éditions Dunod, coll. « Sciences Sup », 7 mars 2006, 240 p.(ISBN 2-10-049147-4)
- René David et Hassane Alla, Du Grafcet aux réseaux de Petri, Paris, Hermès, 1992, 2e éd. (ISBN 2-86601-325-5)
- à noter l'ouvrage en anglais traitant plus spécialement des extensions temporelles et continues : René David et Hassane Alla, Discrete, Continuous, and HybridPetri Nets, Berlin, Springer-Verlag, 2005 (ISBN 3-540-22480-7)

Title of the Master: Software Engineering and Distributed Systems

Semester: S1

Title of the TU: Machine Learning and Specification Tools

Title of the subject : Machine Learning

Credits: 5

Coefficients : 2

Teaching objectives

This course will enable students to understand various types of Artificial Learning, primarily through optimization, to acquire skills in *Machine Learning* and *Deep Learning*.

Recommended prior knowledge

- Programming language: C, python

- Probabilities

Content of the material:

Chapter I. Neural networks

- Machine learning and types of learning,
- Fundamentals of artificial neurons,
- The Perceptron Single-layer and Multi-layer network
- Learning RNs:
  Case of a 2C and multi-C single layer perceptron
- Error gradient backpropagation algorithm

Chapter II. AI and deep learning

- Machine Learning vs. Deep Learning Workflows
- Neural networks and deep learning
- Description of training data
- Overview of Convolutional Neural Networks (CNN):
  The convolution operation and the neural network,
  Control parameters and activation functions,
- Overview of learning VGGnet networks

Chapter III. Bayesian Networks

- Presentation of the RBs
- Conditional independence and d-separation
- Definition of a RB
- Inference schemes in RBs
- Generalized d- separation
- RB learning algorithms:
  Case of incomplete data and known structure
  Case of incomplete data and known structure
- RBs adapted to classification

Chapter IV. Hidden Markov Chains

- The observable Markov model
- The Hidden Markov Model (HMM)
- Algorithms for evaluating the probability of observing a sequence:

- Direct assessment
  Evaluation using forward-backward functions
- Finding the most likely path
  (the Viterbi algorithm)
- Learning HMMs
  Baum-Welch algorithm and re-estimation forms

Evaluation Mode:

67 % Exam + 33 % CC

References :

- A. Cornuéjols, L. Miclet, Y. Kodratoff . Apprentissage artificiel - Concepts et algorithmes. Eyrolles 2002 (1ère edition).
- P. Naïm, P. Wuillemin, P. Leray, O. Pourret, A. Becker. Réseaux bayésiens.
- Eyrolles2007 (3e édition).
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning. Cambridge : MIT press. (Vol. 1).
- Stephen Marsland. MACHINE LEARNING, An Algorithmic Perspective. Second Edition. Chapman & Hall/CRC. 2015.
- Welch, L. R. (2003). Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter , 53(4),* 10- 13

Title of the Master: Software Engineering and Distributed Systems

Semester: S1

Title of the TU: Entrepreneurship and English

Title of the subject: English 1

Credits: 1

Coefficients: 1

Teaching objectives: Allows the student to learn to read, understand and write reports, articles, computer texts, and to present their work in English.

Recommended prior knowledge: A technical English module was followed during 1st cycle of the system (LMD).

Content of the material.

Evaluation Mode.

100% Exam.

References.

Any reference deemed useful.

Title of the Master: Software Engineering and Distributed Systems
Semester: S1
Title of the TU: Entrepreneurship and English
Title of the subject: Entrepreneurship 1
Credits: 1
Coefficients : 1

Teaching objectives.

This course covers the entrepreneurial process. The student discovers how successful entrepreneurs recognize business opportunities, find ideas and organize their resources in order to launch projects or businesses that, while meeting market needs, bring them great benefits. personal satisfactions.

Recommended prior knowledge

Content of the material:
Introduction
Framework for developing course outlines
Overall course overview
Overall overview of Unit 1: Entrepreneurial spirit
Activity 1.1: Entrepreneurial spirit
Activity 1.2: Interview with an entrepreneur
Activity 1.3: Customer satisfaction
Activity 1.4: Francophone entrepreneurs and their contributions
27 Activity 1.5: Summative assessment task - Case study - What nerve!
Overview of Unit 2: Business Ideas and Opportunities
Activity 2.1: Canadian inventions and innovations
Activity 2.2: Business ideas and opportunities
Activity 2.3: Economic and social trends
Activity 2.4: Market research
Activity 2.5: Market research simulation
Overview of Unit 3: Organizing Resources
Activity 3.1: Legal requirements
Activity 3.2: Human and material resources
Activity 3.3: Production plan
Activity 3.4: Marketing (promotion) plan
Activity 3.5: Developing a marketing plan for a new product
Overall overview of Unit 4: Financial plan
Activity 4.1: Financial strategies
Activity 4.2: Financial statements
Activity 4.3: Financing
Activity 4.4: Cash budget
Activity 4.5: Auxiliary plans
Overall Overview of Unit 5: Business Plan
Activity 5.1: Role of a business plan
Activity 5.2: Analysis of a business planActivité 5.3 : Répertoire de ressources

Activity 5.4: Developing a business plan
Activity 5.5: Evaluating and revising a business plan

Evaluation Mode:

100 % Exam

References

1. https://www.cforp.ca/fichiers/esquisses-de-cours/affaires-et-commerce/BDI3C.pdf
2. http://www.fsr.ac.ma/cours/entreprenariat/entreprenariat_cours.pdf
3. http://thierry-verstraete.com/pdf/adreg%2010%20Verstraete%20saporta%20complet.pdf

Title of the Master: Software Engineering and Distributed Systems

Semester: S2

Title of the TU: Modeling and Verification

Title of the subject: Calculability

Credits: 5

Coefficients: 3

Teaching objectives.

Allows the student to learn the basic notions of computer calculation. The most important issues related to decidability and termination are presented.

Recommended prior knowledge

Language theory, Automaton, Logic.

Content of the material:

1 Notions of Effective Computability (Church's Thesis)
1.1 Introduction: CHURCH thesis
1.1.1 Everything is a word
1.1.2 Encoding
1.2 Formalization of computability
1.2.1 TURING Machine approach
1.2.2 Approach using programming languages
1.3 TURING machine
1.3.1 Use of the TURING machine
1.3.2 Examples of TURING machines
1.3.3 Language accepted by a TURING Machine
1.3.4 Language decided by a TURING Machine
1.3.5 Recursive and recursively enumerable languages
1.3.6 Functions computable by a TURING machine
1.3.7 Non-deterministic TURING machines
1.4 Formalization of problems
1.5 Extensions of TURING machines
1.5.1 Multiple ribbon machines
1.5.2 Direct access memory (RAM) machines
1.6 On the termination
1.7 Conclusion

Evaluation Mode.

50 % Exam +25% Personal work+ 25 % Tutorials.

References

1) http://www.enseignement.polytechnique.fr/informatique/INF423/uploads/Main/chap8-good.pdf
2) **http://www.liafa.jussieu.fr/~asarin/calc2k3/calcul_cours.pdf**
3) **http://www.math.ru.nl/~terwijn/publications/syllabus_fr.pdf**

Title of the Master: Software Engineering and Distributed Systems
Semester: S2
Title of the TU: Modeling and Verification
Title of the subject: Techniques and Verification Tools
Credits: 5
Coefficients: 3

Teaching objectives.

Allows the student to learn critical systems verification techniques and certain tools used in industry.

Recommended prior knowledge

Content of the material:

Part I: Model-checking
Chapter 1: Enumerative Model Checking
Chapter 2: Symbolic model checking
Chapter 3: Symbolic trajectory evaluation
Chapter 4: Abstraction and refinement
Chapter 5: Bounded model checking and SAT
Chapter 6: Tools: SMV and SPIN
Part II: Methods of deduction
Chapter 7: Automatic proof of theorems
Chapter 8: Systems for proving theorems
Chapter 9: Tools: PVS and ITP
Mode d'évaluation :

67 % Exam + 33 % Tests

References

Vérification de logiciels : Techniques et outils du model-checking, Philippe Schnoebelen, vuibert informatique, 1999

Model Checking, E. Clarke etOrnaGrumberg, MIT, 1999.

PVS : Combining specification, proof checking, and model checking, S.Owre, 1996

Title of the Master: Software Engineering and Distributed Systems

Semester : S2

Title of the TU: Development and Simulation

Title of the subject: Advanced Development Approaches

Credits: 5

Coefficients: 2

Teaching objectives.

This course aims to present model engineering (or model-driven engineering (MDE)), model-driven architecture (MDA), and reverse engineering and reconstruction of software architecture.

Recommended prior knowledge

OO Design, XML, and UML,

Content of the material:

Lectures

Chapter 1: Introduction to Model Engineering, General Definitions
Chapter 2: MDA Framework
Chapter 3: Meta-modeling and Meta-Modeling Tools EMF, GMF, Papyrus, ArgoUML, MoDisco, etc.
Chapter 4: OCL constraint language
Chapter 5: Model transformations
Chapter 6: Reverse Engineering and Reconstruction of Software Architecture

Lab Works

First model manipulations, with EMF
Meta-modeling
Manipulating OCL constraints.
Model Transformations
Reverse Engineering

Evaluation Mode.

50 % Exam +25% de Personal Work+ 25 % Lab Work.

References

1) Anneke Kleppe, Jos Warmer, Wim Bast, **MDA Explained: The Model Driven Architecture™: Practice and Promise**, Addison Wesley 2003
2) Alan W. Brown **Model driven architecture: Principles and practice**,Softw Syst Model Volume 3, Issue 4, pp 314–327 (2004)
3) David S. Frankel, **Model Driven Architecture Applying MDA to Enterprise Computing**, John Wiley Press, USA, 2002
4) Eric Cariou, « **Ingénierie des modèles** » cours Master 2ème année, Université de Pau et des Pays de l'Adour , http://ecariou.perso.univ-pau.fr/cours/idm.html
5) FRANKEL, David S. **Model driven architecture: applying MDA to enterprise computing**. John Wiley 1st edition (January 28, 2003)
6) DUCASSE, Stéphane et POLLET, Damien. **Software architecture reconstruction: A process-oriented taxonomy**. IEEE TSE, 2009, vol. 35, no 4, p. 573-591.

Title of the Master: Software Engineering and Distributed Systems
Semester : S2
Title of the TU: Development and Simulation
Title of the subject: Modeling and Simulation
Credits: 5
Coefficients: 2

Teaching objectives.

The objective of this subject is to introduce simulation techniques and their theoretical foundations.

Recommended prior knowledge

Probability and statistics, numerical analysis

Content of the material:
1. Reminder of the laws of probability;
2. Stochastic Process;
3. Markov chains;
4. Queues;
5. The Monte Carlo method

Evaluation Mode:

67 % Exam + 33 % Tests

References

1) https://perso.univ-rennes1.fr/bernard.delyon/simu.pdf
2) http://www.cmap.polytechnique.fr/~leonard/Proba2/proba-version2.pdf

Title of the Master: Software Engineering and Distributed Systems

Semester : S2

Title of the TU: Blockchains and RTS

Title of the subject: Real-Time Systems

Credits: 4

Coefficients: 2

Teaching objectives

Introduce the student to the concept of real-time systems. Incriminate the specificities and constraints of this type of system. Learn how to develop a real-time application.

Recommended prior knowledge

Notions of processes, synchronization and communication in systems.

Content of the material.

I - Real-time systems.
II- RTS software engineering.
III- Advanced real-time languages.
IV- Real-time executives.
V: General information on embedded systems
Introduction to embedded systems
Architecture of embedded systems
Specific embedded systems
VI: Design of embedded software
Design of embedded software
Hardware design.
Integration of verification into the design flow.
VII: Formal methods for functional verification
Assisted proof
Model-Checking
Static analysis

Evaluation Mode.

67 % Exam + 33 % Work Lab.

References.

1) 1. Alain DarseOil, Pascal Pillot, **Le temps réel en milieu industriel**; Edition DUNOD, 1991.
2) Model Checking, E. Clarke etOrnaGrumberg, MIT, 1999.
3) Embedded System: Architecture, Programming and Design, Raj Kamal, McGraw-Hill Education, 2003.
4) Embedded System:handbook, Richard Zurawski, 2006.
5) PVS : Combining specification, proof checking, and model checking, S.Owre, 1996
6) Sites Officiels des outils UPPAAL et MathWorks.

Title of the Master: Software Engineering and Distributed Systems

Semester : S2

Title of the TU: Blockchains and RTS

Title of the subject: Blockchain Technologies

Credits: 4

Coefficients: 2

Course objective:

This course aims to introduce blockchain technology to students. This presentation will cover both aspects: theoretical and practical. Blockchain technology can be seen as a technology based on three pillars: algorithmic aspect and data structure, cryptographic aspect and finally distributed and condescended systems aspect. The course consists of providing a refresher on cryptography techniques then beginning the presentation of the blockchain and its algorithms, after this presentation concrete case studies will be treated such as: Bitcoin, Etherum, etc. The course is compensated by a practical work which should allow students to program the concepts presented in the course.

Content of the material.

Chapter 1: Introduction: information technology revolution with blockchain
Chapter 2: Reminder on cryptography
What is cryptography?
Types of cryptography: symmetric, antisymmetric
The different cryptography algorithms
Chapter 3: Block-chain technology
What is a blockchain
Why blockchain technology
Element constituting a blockchain
Consensus in blockchains
Chapter 4: concrete case study
Blockchain: Bitcoin
The blockchain: Etherum
Blockchain: NFT
Future of blockchain

Evaluation Mode.

67 % Exam + 33 % Lab Work.

References :

Paar, C., & Pelzl, J. (2009). Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media.
Xiao, P. (2019). Practical Java Programming for IoT, AI, and Blockchain. John Wiley & Sons.
Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. PloS one, 11(10), e0163477.

Title of the Master: Software Engineering and Distributed Systems
Semester : S2
Title of the TU: Methodology
Title of the subject: Research methodology
Credits: 2
Coefficients:1

Teaching objectives

The objectives of the research methodology subject are to provide students with the tools allowing them to write written reports and in particular their work reflecting their professional mission and their master's thesis during Semester 4.

Recommended prior knowledge
Course: Scientific writing

Content of the material:

1. Concepts of scientific research methodology
2. Formulation of the research question and research design
3. Review of the literature from a critical point of view
4. Scientific research methods; quantitative and qualitative method
5. Analysis, interpretation, and reporting of research results
6. Writing and presenting a research project

Evaluation method.
100 % Exam

References

1. Blaxter, L. Hughes, C. & Tight, M. (1998) How to Research *Buckingham: Open University Press*

Denscombe, M. (2002) Ground Rules for Good Research *Maidenhead: Open University Press*

Title of the Master: Software Engineering and Distributed Systems
Semester: S3
Title of the TU: High-level Formalisms and Development
Title of the subject: High-level formal languages
Credits: 6
Coefficients: 3

Teaching objectives

This course aims to present a set of high-level formalisms that can be used for the specification, modeling and verification of non-classical systems. By non-classical systems we mean mobile systems, dynamic systems or reconfigurable systems. We consider that these systems share a common property which is a dynamic in their structure leaving this structure modifiable during their executions. The course covers high-level PNs: colored PNs with their different versions, process algebras and their extensions, and finally models based on bigraphs.

Content of the material:

Chapter 1: Informal Introduction: Mobile Systems, Reconfigurable Systems and Dynamic Systems
Chapter 2: High-level PNss
Colored PNs and their extensions
Temporal and timed PNs and their extensions
Reconfigurable and dynamic PNs
Chapter 3: Process algebras
CCS vs CSP algebra
Pi-calculus and its extensions
Chapter 4: bigraphs

Evaluation Mode.

67 % Exam + 33 % Tutorials.

References :

Jensen, K., & Kristensen, L. M. (2015). Colored Petri nets: a graphical language for formal modeling and validation of concurrent systems. Communications of the ACM, 58(6), 61-70.
Milner, R., Parrow, J., & Walker, D. (1992). A calculus of mobile processes, i. Information and computation, 100(1), 1-40.
Milner, R. (2008). Bigraphs and their algebra. Electronic Notes in Theoretical Computer Science, 209, 5-19.

Title of the Master: Software Engineering and Distributed Systems
Semester: S3
Title of the TU: High-level Formalisms and Development
Title of the subject: Rewriting Logics and Their Applications
Credits: 6
Coefficients : 3

Teaching objectives

Allows the student to learn rewriting logic which is a unifying logic where the student learns to write semantics for critical systems in order to apply certain analyzes to them. This module is the basis of the “Semantics of mobile systems” modules.

Content of the material:

1. Rewrite logic
2. Equational specifications and their analysis.
3. Modeling distributed systems using rewriting logic
4. Modeling concurrent objects in rewriting logic.
5. Application: Modeling of communications protocols in Maude
6. Case study.

Evaluation Mode:

67 % Exam + 33 % Tests

References

Maude : Specification and Programming in Rewriting Logic, Computer Science Laboratory, SRI International, 2005

Title of the Master: Software Engineering and Distributed Systems
Semester: S3
Title of the TU: High-level Formalisms and Development
Title of the subject: Distributed Application Development
Credits: 6
Coefficients: 3

Teaching objectives.

Allows the student to learn distributed application development techniques.

Content of the material:

Chapter 1: Introduction. Communication technologies
Chapter 2: Attributes of Distributed Systems/Applications
Chapter 3: Types of distributed applications/systems
Chapter 4: Distributed Application Architectures
Chapter 5: Interaction models
Chapter 6: Development of distributed applications
- 6.1 Communications protocols
- 6.2 Distributed objects
- 6.3 Message Guidance

Chapter 7: Application Servers

Evaluation Mode.
67 % Exam + 33 % Lab Work.

References

1. Andrew S. Tanenbaum and Maarten Van Steen. Distributed Systems: Principles and Paradigms. Prentice Hall, 2001.
2. Jie Wu. Distributed Systems Design. CRC Press LLC, 1999
3. Richard Monson-Haefel and David Chappell. Java Message Service. O'Reilly& Associates, Inc., Sebastopol, CA, USA, 2000.
4. Nancy A. Lynch. Distributed Algorithms. Morgan Kaufmann, San Francisco, CA, USA, 1997.

Title of the Master: Software Engineering and Distributed Systems
Semester: S3
Title of the TU: Model, Architecture and Management
Title of the subject: Model-Based Diagnosis
Credits: 4
Coefficients: 2

Teaching objectives.

Allows the student to understand the notion of diagnosis of physical systems (software or hardware) and the reasoning techniques in this area. An example will be given as a case study.

Content of the material:

Chapter 1: Pragmatic definitions
Chapter 2: Diagnostic Approaches
    Heuristic based approaches
    Model-based approaches
Chapter 3: Model-based approaches
    Consistency-based diagnosis
    Abduction-based diagnosis
Chapter 4: Reasoning techniques for diagnosis
Chapter 5: Distributed diagnostics
Chapter 6: Case study

Evaluation Mode.

67 % Exam + 33 % Tests

References

Fundamentals of Model-based Diagnosis, J. de Kleer, 2003
A theory of diagnosis from first principles, 1987

Title of the Master: Software Engineering and Distributed Systems
Semester: S3
Title of the TU: Model, Architecture and Management
Title of the subject: Service Oriented Architectures
Credits: 4
Coefficients: 2

Teaching objectives.

The objective of this subject is to provide the student with an introduction to Service Oriented Architectures, web services, how to design, create them and compose web services. Web services have become basic elements in the creation and integration of software applications.

Recommended prior knowledge

Web application development, Object Oriented Programming, XML

Content of the material:
椮瓈□侼□僡茬俊錞□錞俸錞侼凂□疛灋□□□珎錞 ⍰珎錞□□沺
⍰凂灋□□□□□□珎□錞珎□疛□沺珎錞
⍰凂□□□玜鉚珎□沺□侼茬鉚□僡珎
⍰凂俊□□珎□□茬蒨□□玜□鉚□□蒨
詯凂椮□侼疛珎□□□□沺珎錞
椮瓈□侼□僡茬俊錞□□錞俸 椮!"⍰!珎#疛□珎錞 ⍰珎錞□□沺
⍰凂瓈侼錞□□珎□錞珎□疛□沺珎□□珎□鉚俸珎□□□珎茬□□錞珎□□ !瓈"
⍰凂#珎僡□錞□□珎□□珎錞□錞珎□疛□沺珎錞□□!#"
⍰凂$⍰⍰沺珎錞錞侼錞□%蒨□珎□錞□□瓈$&"
椮瓈□侼□僡茬俊錞□□□錞俸錞僡$⍰⍰珎錞 ⍰珎錞珎□凂□□灋□珎錞 #疛珎錞%⍰珎錞□□沺
⍰凂!珎□疛□沺珎錞□□珎□□□□□蒨珎錞
⍰凂!珎□疛□沺珎錞□□茬茬鉚□沺□□'錞
侼□茬僡□俊錞□□錞錞錞錞沺□凂□□灋□珎錞□□□錞錞錞&蒨詯
椮瓈□侼□僡茬俊錞□錞俸錞椮!疛灋□□□珎錞 ⍰錞□#珎□珎錞□□錞□凂□疛灋□□□珎錞
⍰錞'!疛灋(!疛疛□"⍰疛□錞 ⍰珎錞沺□珎
⍰凂(蒨)⍰⍰⍰⍰⍰⍰珎□□沺□□□沺□蒨□□錞□*侼珎錞□□俸侼□□!+
⍰凂俊□□珎□'⍰沺珎□珎□□□%茬鉚蒨%珎□□□□□□□俸侼□□!+
⍰凂$⍰鉚珎□□珎錞□錞□□□□□□□錞□□珎錞□!+錞
詯凂□$侼玜鉚□沺□□□□□珎□□□蒨沺□侼疛珎□□珎□□珎錞□!+錞
椮瓈□侼□僡茬俊錞□□錞俸 椮!"⍰!珎□(!疛錞 ⍰珎錞沺□珎
⍰凂,-茬珎錞□□珎□沺□%茬□錞□□□□錞
⍰凂,珎沺.⍰⍰鉚□僡□珎錞□□□錞茬□□□玜鉚珎錞
椮瓈□侼□僡茬俊錞□□□錞俸錞)⍰⍰⍰%!⍰⍰⍰"⍰疛□錞 ⍰珎錞沺□珎
⍰凂&蒨□.⍰⍰⍰鉚□僡□珎錞□□珎□□蒨疛珎鉚□茬茬珎%珎□□□□珎錞□!+錞
詯凂/-沺鉚珎□□珎□疛□珎□□侼□□蒨疛珎鉚□茬茬珎%珎□□□□珎□!+錞
椮瓈□侼□僡茬俊錞□錞俸錞侼凂□疛灋□□□珎錞 ⍰珎錞□□沺
⍰凂灋□□□□□□珎□錞珎□疛□沺珎錞
⍰凂□□□玜鉚珎□沺□侼茬鉚□僡珎
⍰凂俊□□珎□□茬蒨□□玜□鉚□□蒨

諳洽樇□偞符玪□□□汕玪鍔
樇璷□偞□塿莶後鍔□□鍔俸 樇!"□!玪#符□玪鍔 □玪鍔□□汕
□洽璷偞鍔□□玪□鍔玪□符□汕玪□□玪□鎯俸玪□□□玪莶□□鍔玪□□ !璷"
□洽#玪塿□鍔□□玪□□玪鍔□鍔玪□符□汕玪鍔□□!#"
□洽$□□汕玪鍔鍔偞鍔□%菁□玪□鍔□□璷$&"
樇璷□偞□塿莶後鍔□□□鍔俸鍔塿$□□玪鍔 □玪鍔玪□洽□□灠□玪鍔 #符玪鍔%□玪鍔□□汕
□洽!玪□符□汕玪鍔□□玪□□□□□菁玪鍔
□洽!玪□符□汕玪鍔□□莶莶鎯□汕□□'鍔
偞□莶塿□後鍔□□鍔鍔鍔鍔汕□洽□□灠□玪鍔□□□鍔鍔鍔&菁諳
樇璷□偞□塿莶後鍔□鍔俸鍔樇!符灠□□□玪鍔 □鍔□#玪□玪鍔□□鍔□洽□符灠□□□玪鍔
□鍔'!符灠(!符符□"□符□鍔 □玪鍔汕□玪
□洽(菁)□□□□□□玪□□汕□□□汕□菁□□鍔□*偞玪鍔□□俸偞□□!+
□洽後□□玪□'□汕玪□玪□□□%莶鎯菁%玪□□□□□□□□俸偞□□!+
□洽$□鎯玪□□玪鍔□鍔□□□□□□□鍔□□玪鍔□!+鍔
諳洽□$偞玸鎯□汕□□□□玪□□□菁汕□偞符玪□□玪□□玪鍔□!+鍔
樇璷□偞□塿莶後鍔□□鍔俸 樇!"□!玪□(!符鍔 □玪鍔汕□玪
□洽,-莶玪鍔□□玪□汕□%莶□鍔□□□□鍔
□洽,玪汕.□□鎯□塿□玪鍔□□□鍔莶□□□玸鎯玪鍔
樇璷□偞□塿莶後鍔□□□鍔俸鍔)□□□%!□□□"□符□鍔□玪鍔汕□玪
□洽&菁□.□□□鎯□塿□玪鍔□□玪□□菁符玪鎯□莶莶玪%玪□□□□玪鍔□!+鍔
諳洽/-汕鎯玪□□玪□符□玪□□偞□□菁符玪鎯□莶莶玪%玪□□□□玪□!+鍔
樇璷□偞□塿莶後鍔□鍔俸鍔偞洽□符灠□□□玪鍔 □玪鍔□□汕
□洽灠□□□□□□玪□鍔玪□符□汕玪鍔
□洽□□□玸鎯玪□汕□偞莶鎯□塿玪
□洽後□□玪□□莶菁□□玸□鎯□□菁
諳洽樇□偞符玪□□□□汕玪鍔
樇璷□偞□塿莶後鍔□□鍔俸 樇!"□!玪#符□玪鍔 □玪鍔□□汕
□洽璷偞鍔□□玪□鍔玪□符□汕玪□□玪□鎯俸玪□□□玪莶□□鍔玪□□ !璷"
□洽#玪塿□鍔□□玪□□玪鍔□鍔玪□符□汕玪鍔□□!#"
□洽$□□汕玪鍔鍔偞鍔□%菁□玪□鍔□□璷$&"
樇璷□偞□塿莶後鍔□□□鍔俸鍔塿$□□玪鍔 □玪鍔玪□洽□□灠□玪鍔 #符玪鍔%□玪鍔□□汕
□洽!玪□符□汕玪鍔□□玪□□□□□菁玪鍔
□洽!玪□符□汕玪鍔□□莶莶鎯□汕□□'鍔
偞□莶塿□後鍔□□鍔鍔鍔鍔汕□洽□□灠□玪鍔□□□鍔鍔鍔&菁諳
樇璷□偞□塿莶後鍔□鍔俸鍔樇!符灠□□□玪鍔 □鍔□#玪□玪鍔□□鍔□洽□符灠□□□玪鍔
□鍔'!符灠(!符符□"□符□鍔 □玪鍔汕□玪
□洽(菁)□□□□□□玪□□汕□□□汕□菁□□鍔□*偞玪鍔□□俸偞□□!+
□洽後□□玪□'□汕玪□玪□□□%莶鎯菁%玪□□□□□□□□俸偞□□!+
□洽$□鎯玪□□玪鍔□鍔□□□□□□□鍔□□玪鍔□!+鍔
諳洽□$偞玸鎯□汕□□□□玪□□□菁汕□偞符玪□□玪□□玪鍔□!+鍔
樇璷□偞□塿莶後鍔□□鍔俸 樇!"□!玪□(!符鍔 □玪鍔汕□玪
□洽,-莶玪鍔□□玪□汕□%莶□鍔□□□□鍔
□洽,玪汕.□□鎯□塿□玪鍔□□□鍔莶□□□玸鎯玪鍔
樇璷□偞□塿莶後鍔□□□鍔俸鍔)□□□%!□□□"□符□鍔 □玪鍔汕□玪
□洽&菁□.□□□鎯□塿□玪鍔□□玪□□菁符玪鎯□莶莶玪%玪□□□□玪鍔□!+鍔
諳洽/-汕鎯玪□□玪□符□玪□□偞□□菁符玪鎯□莶莶玪%玪□□□□玪□!+鍔

Chapter 1: Introduction: Problem & Motivation of SOA

1. Service concept

2. Description of Service Oriented Architecture

3. Principles of Service Oriented Design

4. Strategic Benefits of an SOA

Chapter 2: Web Services and Components of an SOA

1. Definition and characteristics of Web Services

2. Interface and implementation of a Web service

3. Components of a SOA (Service Provider, Registry and Clients)

4. Operating principle of an SOA

Chapter 3: Web Services Standards Stack

1. HTTP and XML
2. The WSDL language
3. SOAP protocol
4. Publishing and discovering web services in UDDI

Chapter 4: Composition of Web Services

1. Types of compositions (orchestrations and choreographies of services)
2. Service Composition Technologies and Languages: XLANG, WSFL, BPEL, BPML, BPMN, WS-CDL,....
3. BPEL: standard web services composition language

Chapter 5: Web Services Development

1. Web services development methodologies: techniques, tools
2. Development of SOAP services
3. Development of REST APIs
4. Migration to an SOA: Objectives, generic activities, migration techniques, service identification metrics.

Evaluation Mode.

67 % Exam + 33 % Tests

References

Hye-young Paik, Angel LagaresLemos, Moshe Chai Barukh, Boualem BenatallahAarthiNatarajan**Web Service Implementation and Composition Techniques** Springer 2017

James Bean, "**SOA and Web Services Interface Design, Principles, Techniques and Strandards**", Morgan Kaufmann Publishersis an imprint of Elsevier, 2010

Michael P. Papazoglou, "**Web Services: Principles andTechnology**", Prentice Hall, 2008

Thomas Erl**SOA Principles of Service Design** . Prentice Hall . 2007

Thomas Erl**Service-Oriented Architecture: Concepts, Technology, and Design,** Prentice Hall 2005.

Nicolai M. Josuttis**SOA in Practice: The Art of Distributed System Design** O'Reilly 2007

Will Iverson **Real World Web Services,** O'Reilly 2004

RICHARDSON, Leonard, AMUNDSEN, Mike, AMUNDSEN, Michael, et al. **RESTful Web APIs: Services for a Changing World.** " O'Reilly Media, Inc.", 2013.

Title of the Master: Software Engineering and Distributed Systems
Semester: S3
Title of the TU: Model, Architecture and Management
Title of the subject: Software Project Management
Crédits : 2
Coefficients : 1

Teaching objectives.

The objective of this course is to give students a methodology allowing them to successfully carry out a large-scale software project. Their objective will be to successfully complete a project based on rigorous methodologies that have proven their worth.

Recommended prior knowledge

Classic Software Engineering Life Cycle

Content of the material:

I – Requirement analysis,
II - Specifications,
III - Planning,
IV - Resource management,
V - Code quality,
VI - Software Architecture,
VII - Test,
VIII- Tasks and Roles in development

Evaluation Mode.

100 % Exam.

References.

1. B. Boehm: "**Software Engineering Economics**", Prentice Hall, 1981.

2. R.S. Pressman: "**Software Engineering: A Practitioner's Approach**", McGraw-Hill Inc., 1981.
Toute référence jugée utile.

Title of the Master: Software Engineering and Distributed Systems

Semester: S3

Title of the TU: English

Title of the subject: English 2

Crédits : 2

Coefficients : 1

Teaching objectives.

Allows the student to learn to read, understand and write reports, articles, computer texts, and to present their work in English.

Evaluation Mode.

100% Exam.

References.

Any reference deemed useful.