

Major  
**Modelling, Design and Validation for Communicating Systems**  
( Major MDV )

**Person in charge:**

Dr. Stéphane MAAG

**Objectives:**

Strong technological advances in the fields of (tele)communication networks and services can provide communicating systems that are increasingly complex and allow the portability. These new systems are composed of service enablers (primitives/API) or applications that are integrated in many projects or industrial platforms so-called "of next generation".

Besides, this is one of the reasons why these systems have to find out new concepts both for modelling techniques, the design and validation. These concepts cover a significant portion of the development cycle of these systems. The first model corresponds to urbanism aspects (SOA), but also to high level modelling systems using languages that are said to be "formal". The design then based on modelling studies upstream. Aspects of application portability and collaborative development are essential to the realization of such systems. Then, downstream, industrialization rhymes with validation of the product (software and protocol), crucial step to a successful deployment without a hitch. The expertise of these mechanisms and concepts is now paramount to the success of many actors of diverse fields such as telecommunications, transport, energy, banking and so on.

In this context, the main objective of MDV is to train engineers enabling to provide the following competencies:

- Master the new design models (MDD, MDA, SOA, Agile, ...), model driven architecture, multi-core, software product lines,
- To know the « formal » description techniques for critical and reactive systems, fault models, safety: why and how?
- Master the collaborative process (USVN) and to know how to use the industrial environments such as the IDE (Eclipse, JCreator),
- To design the communicating systems by applying advanced development process (.NET) and following metrology aspects, system portability,
- To design and develop the novel graphic interfaces, HMI, open 3D graphic API (OpenGL ES2 (WebGL)), ergonomics/sociology,
- Master the software and protocol testing process, metrics, risk measurement, safety.

## **Organisation:**

This major consists of six self-consistent modules programmed during the semesters S8 and S9. The first two modules scheduled to S8 are called "for immersion" and are to present the basics of MDV. Then, the first two modules of S9 that define the primary business of MDV come. They are then followed by two modules application oriented. Each of these modules has a total workload of 90h whose 45h or less are made in-face.

## **Program:**

Semester 8 :

- CSC4518 : Software engineering tools
- CSC4519 : Human machine interfaces

Semester 9 :

- CSC5041 : Advanced software engineering
- CSC7302 : Software testing & metrics
- CSC5042 : Advanced modelling for communicating systems
- CSC5043 : Communicating systems quality
- CSC5045 : MDV major's project

**CSC4518****Software engineering tools****Period** : S8 / P3**ECTS** : 4**Language** : English**Organization:**

- Teaching Load / Total Load: 45/90
- Lectures/Exercices/Labs/Final Exam: 18/0/18/0

Courses provide notions and concepts about the tools that are practiced in labs.

**Assessment:**

One graded lab (L) and a project (P) will assess the students.

Final Grade= Average (L,P)

**Objectives :**

- Master software development processes,
- Master software engineering toolkits applied in industry (debugger, profiler, check of memory leak, etc.),
- Master software development industrial environments.

**Keywords :**

Eclipse, Ant, Junit, SubVersion (USVN), Quality Debugger, Profiler, Autotools, .NET

**Prerequisites :**

Basic knowledge of C or C++ and Java

**Course outlines:**

- Integrated development platforms, environments: IDE (Eclipse, JCreator): basic usage for java development,
- Microsoft .NET framework
- Collaborative development tool. SubVersion: main functionalities (import, checkout, commit, update, add, remove, diff) – User Friendly version (USVN)
- Software quality and metrology.
- Basis and mechanisms to software development: AutoTools and ANT
- Software debuggers and profiler: Valgrind, Gprof, DDD...
- Internationalization of application (gettext ...)

**Learning materials and literature :**

- Ant User Manual, <http://www.ing.iac.es/~docs/external/ant/manual/index.html>
- Debugging with DDD, Byron Clark, Computer Science Department, Brigham Young University, 2005 (<http://docs.cs.byu.edu/docs/ddd/>)

- The Java developer's guide to Eclipse, Sherry Shavor, Jim D'anjou, Addison-Wesley, 2003
- Gestion de projet avec Subversion, Ben Collin-Susman, Brian W. Fitzpatrick, C. Michael Pilato, O'Reilly, 2004
- Learning the GNU development tools (Eleftherios Gkiouleka, Marcelo Roberto Jimenez): <http://www.scs.stanford.edu/~reddy/links/gnu/tutorial.pdf>
- Essential Windows Communication Foundation (WCF): For .NET Framework 3.5, Addison-Wesley Professional; 1 edition (February 21, 2008)

**Person in charge:**

Eric Lallet (Eric.Lallet@it-sudparis.eu)

**Lecturers :**

- Eric Lallet
- Dr. Jean-Luc Raffy
- Dr. Paul Gibson

---

**CSC4519 Human machine interfaces****Period : S8 / P4****ECTS : 4****Language : English****Organization :**

- Teaching Load / Total Load : 27/90
- Lectures/Exercices/Labs/Final Exam : 12/0/15/0

Among the courses, six hours are dedicated to the presentation of industrial tools illustrated by the industrials themselves.

**Assessment:**

Individual labs (L) and projects (P) will assess the students.

Final Grade= Average(L,P)

**Objectives :**

- Master development techniques for graphic interfaces dedicated to software engineering,
- Study and Design Human-Machine-Interfaces,
- Techniques and tools to focused on 3D graphic API, OpenGL ES2, WebGL,
- Ergonomics and sociology aspects for HMI,
- Master frameworks used in the industry,

**Keywords :**

HMI, Graphic interfaces, TK, Open-GL, WebGL, 3D graphic API

**Prerequisites :**

Knowledge of a procedural or object-oriented programming language

**Course Outlines :**

- The tool box Tk (Tool kit) that allows creating graphical user interfaces.
- Introduction to open-GL ES2, WebGL – Labs applying these new concepts.
- Study and use of 3D graphic API to design novel graphic interfaces,
- Study of ergonomics and sociology aspects for HMI.

**Learning materials and literature:**

- *Tcl/Tk Reference Guide*, <http://www.slac.stanford.edu/~raines/tkref.html>
- *The TkLib*, <http://wiki.tcl.tk/13901>
- *OpenGL Programming Guide: The Official Guide To Learning OpenGL, Version 2*, Dave Shreiner, Mason Woo, Jackie Neider, 2005, Addison-Wesley

- "Graphical applications with Tcl & Tk", E. Johnson M&T books, 1997

**Person in Charge :**

Dr Jean-Luc Raffy ( [jean-luc.raffy@it-sudparis.eu](mailto:jean-luc.raffy@it-sudparis.eu) )

**Lecturers :**

- Dr Jean-Luc Raffy, TELECOM SudParis
- Dr Anis Laouiti, TELECOM SudParis

---

**CSC5041      Advanced software engineering****Period :** S9 / P1**ECTS :** 4**Langue :** English**Organization :**

- Teaching Load / Total Load : 45/90
- Lectures/Exercices/Labs/Final Exam : / / /

**Assessment:**

Continual assessment    Written examination

**Objectives :**

The students will learn how to evaluate and compare novel software engineering techniques, apply them to solve real-world problems, and to integrate them in a coherent fashion.

**Keywords :**

Services, Features, Domains, Patterns, Aspects, Product Lines, Specialised Self-\* Systems, Parallel Execution, Agile, MDD, MDA.

**Prerequisites :**

Object Oriented Programming, Basics of Software Engineering, Mathematical Foundations

**Course outlines:**

- Service Oriented Development and Architectures
- Features and feature interactions
- Domain Analysis and Domain Specific Languages
- Aspect Oriented Analysis/Design and Implementation
- Software Product Lines
- Specialised Self-\* Architectures and Systems
- Design and programming for Multi-core Systems

**Learning materials and literature :**

- Cloud Computing, A Practical Approach, by Toby Velte, Anthony Velte, and Robert C.
- Service-Oriented Architecture: Concepts, Technology, and Design, by Thomas Erl
- Aspect Oriented Software Development by Robert E. Filman, Tzilla Elrad, Siobhan Clarke, and Mehmet Aksit
- Applied Software Product-Line Engineering by Kyo C. Kang, Vijayan Sugumaran, and Sooyong Park
- The Art of Multiprocessor Programming by Maurice Herlihy and Nir Shavit

---

**Person in charge:**

J Paul Gibson (Paul.Gibson@it-sudparis.eu)

**Lecturers:**

- J Paul Gibson

---

**CSC7302      Software testing & metrics****Period :** S9 / P2**ECTS :** 4**Langue :** English**Organization :**

- Teaching Load / Total Load : 30/90
- Lectures/Exercices/Labs/Final Exam : / / /

**Evaluation :**

Continual assessment  
Written examination

**Objectives :**

- Understand the basic concepts of software testing with emphasis on test metrics
- Perform effective and efficient structural testing of your software
- Integrate and test the various units and components of your software system
- Perform effective and efficient functional testing of software
- Select the appropriate tests to regression test your software after changes have been made
- Plan, track and control the software testing effort

**Prerequisites :**

Object Oriented Programming, Basics of Software Engineering, Mathematical Foundations

**Program :**

- Software Testing and Metrics – The Basics
- Structural White-box testing
- Functional (Black Box) Testing
- Regression Testing
- Integration Testing
- Test Planning, Management and Metrics
- Software Quality Metrics

**Learning materials and literature:**

- Testing Object-Oriented Systems: Models, Patterns, and Tools (The Addison-Wesley Object Technology Series) by Robert V. Binder
- Automated Software Testing: Introduction, Management, and Performance by Elfriede Dustin, Jeff Rashka, John Paul, Addison-Wesley Professional

---

**Coordinator :**

Jean-Luc Raffy (jean-luc.raffy@int-edu.eu)

**Lecturer :**

Jean-Luc Raffy

---

**CSC5042      Advanced modelling for communicating Systems****Period:** S9 / P3**ECTS :** 4**Langue :** English**Organization :**

- Teaching Load / Total Load : 45/90
- Lectures/Exercices/Labs/Final Exam : / / /

**Evaluation :**

- Written examination

**Objectives :**

To study how to choose the language and the formal modelling method according to the system type to be modelled.

**Keywords :**

Formal modelling, verification, reactive systems, security properties, Simulator/Animator 3D.

**Prerequisites :**

First order logic.

**Program :**

The critical systems and its characteristics.

The PROMELA language for modelling the communicating reactive systems and the associated tools for properties verification.

The EB3 language for modelling the information systems and the associated tools (Atelier B, Brama)

EB3 Model checking

Applications of these modelling techniques in the industry.

**Learning materials and literature :**

Gerard J. Holzmann. The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley, 2003.

Abrial, J. R. (1996). The B-Book: Assigning Programs to Meanings. Cambridge University Press.

ATELIER B, <http://www.atelierb.eu/php/documents-fr.php>

BRAMA, <http://www.brama.fr/index-en.php>

**Coordinator :**

Amel Mammar (amel.mammar@it-sudparis.eu )

---

**Lecturers :**

- Amel Mammar
- Frédéric Gervais (Université Paris Est)
- Industrials from Siemens and RATP.

---

**CSC5043      Communicating systems quality****Period : S9 / P4****ECTS : 4****Langue : English****Organization :**

- Teaching Load / Total Load : 36/90
- Lectures/Exercices/Labs/Final Exam : 10,5/ 4,5 / 21 /

Among the courses, industrials will also present their own testing or specification tools.

**Assessment :**

A graded lab (L) and a written work (W).

Final mark= Moy(2L+W)

**Objectives :**

- Knowledge of Formal Description Techniques.
- Knowledge on common industrial tools to verify and test real systems.
- Develop reliable and efficient software
- Master SDL specification for protocol and services
- Experience in testing methods.
- Knowledge on new specification and testing techniques provided by research laboratories.

**Keywords :**

Validation, Test, Protocols, SDL, TTCN3.

**Prerequisites :**

- Phases in the software/protocol development life cycle.
- Knowledge of a programming language such as C or Java.

**Program :**

- Introduction to the formal techniques for the protocols specification.
- SDL language presentation (architecture, behavior and data types). Use of an industrial tool.
- Labs to specify real protocols and services.
- Introduction to the conformance testing problematic.
- Study of various techniques to automatically generate test sequences.
- Formal analysis of a protocol specification and automatic generation of real test sequences. These test sequences are thereafter applied to real implementations.

- Properties such as security policies will be tested.
- Standard testing architecture and industrial test production – TTCN3

**Learning materials and literature:**

- Course materials: The SDL language, Stéphane Maag
- Validation of Communications Systems with SDL: The Art of SDL Simulation and Reachability Analysis, *L.Doldi*, 310 pages - John Wiley & Sons - May 2003
- Ingénierie des protocoles et qualité de services, *A.Cavalli et al.*, Hermes, Lavoisier, 2001
- Design and Validation of Computer Protocols, *G.J. Holzmann*, Prentice Hall, 1991 (available at the TSP library)
- An Introduction to TTCN-3, Wiley (July 22, 2005)

**Person in charge:**

Dr. Stéphane Maag (Stephane.Maag@it-sudparis.eu)

**Lecturers:**

- Dr Stéphane Maag
- Dr. Fatiha Zaïdi (Univ. Orsay)
- Emmanuel Alibert (IBM)
- Christophe Gaston (CEA/LIST)

---

**CSC5045 MDV major's project****Period : S9****ECTS : 8****Language : English****Organisation:**

- Scheduled Meeting / Total Load: 20h/225h

MDV major's project is done during the whole semester 9. Each student must make a project with two or three other students. Planning time slots are dedicated to the project. Meeting with project manager take place about every other week.

Three types of projects are proposed to students: Experimentation projects, study projects for companies or administrations, research projects.

**Assessment:**

The validation of this project is based on the writing of a report (R) and an oral presentation (P).

Final score = Average (R, P)

**Samples of subjects:**

- Implementation of a WAP service for transportation;
- Implementation on Palm OS of localisation services;
- Specification of telecommunication services and feature interaction detection;
- Routing protocol specification in a mobile ad hoc network (OLSR)
- Prototype of a learning database for an e-learning system with XML-RDF;
- Development of a graphical simulator for the AGL XEDT;
- JAVA implementation of a «multicast FTP» application.

**Person in charge:**

Stephane Maag (stephane.maag@it-sudparis.eu)

**Faculty:**

All lecturers of MDV major

