MAJOR
system Architect for network SeRvices (ASR)

Person in charge: Michel SIMATIC

Objectives:

Computer applications and services are more and more frequently executed on a network of interconnected computers in order to meet availability or performance. This important evolution can be explained by hardware breakthrough: New generation of mobile terminals (like Smartphones), ubiquitous hardware communication capabilities (in house automation, for instance)
In parallel, new applications are developed on top of these hardware (collaborative work, conference management, multimedia, e-commerce, services on mobiles…). Software architecture of these new applications and services differs fundamentally from monolithic applications running on a single machine; they rely on new software technologies (distributed objects, downloadable components, middleware, execution environments for clusters of PCs).
In this context, major ASR is dedicated to the training of system architect for network services, that is engineers having the following skills:

- Design and development of services and applications distributed on networks;
- Practice of basic computer science skills for distribution: System programming, low-level tools for distributed services (sockets, RPC…), distributed algorithms;
- Mastery of high performance computer systems based on multi-core / grids of multi-cores / GPU and related software technologies: Management of these architectures and applications development;
- Practice of middlewares for development of enterprise distributed applications: CORBA, MOM, Web services, J2EE;
- Mastery of "Semantic web" and its applications: ontologies, XML techniques, metadata;
- Knowledge of stakes, concepts and tools of mobile, ubiquitous and pervasive computing.

Organization:

This major takes place in the major courses of Télécom INT curriculum. It is made up of 6 autonomous and coherent teaching units planned during S8 and S9 semesters. Each of them represents for the student a workload of 90 hours, at most 45h being done in face to face, the rest consisting in personal work

Content :

Teaching units are scheduled as follows:

S8 : - CSC4508: Design and development of centralized systems
- CSC4509: Algorithms and communications for distributed applications

S9 : - CSC5001: High performance systems
- CSC5002: Middleware for distributed applications
- CSC5003: Semantic web and applications
- CSC5004: Middleware for mobile, pervasive and ubiquitous computing

During the S9 Semester, a **team project** (CSC5005) will allow students to gain an in-depth expertise on one of the main scientific axes presented in these UVs.

Code: CSC4508 ***Design and development of centralized systems***

Period: S8 – P3 **ECTS:** 4 **Language:** French

Organization: Face to face: 45h Personal work: 45h Total workload: 90h

Courses: 0h Tutorial: 33h Lab: 12h

During tutorials, notions are presented by the lecturer and practiced on computers by students working in groups of two. Labs are made up of 6 hours of experiments done by students working in pairs and 6 hours of graded labs done by students working independently.

Assessment:
The validation of this TU is based on the grading of two exercises (E1) and two graded labs (GL1 and GL2).

Final score = Average (1/4 E1 + 3/4 GL1, GL2)

Objectives:

- Skills in interaction between programs and operating systems
- Understanding and commands of computer-language parsing

Keywords:
Scheduling, threads, software architecture for servers, computer-language parsing

Prerequisites:

- Algorithmic (notions)
- Hardware architectures (notions)
- C Language (good skills)
- Unix usage (good skills)

Course outlines:

- Operating systems concepts; Implementation in the context of Linux
    - Interaction between a multitask system and its processes
    - Memory management
    - Inputs-Outputs
    - Inter-Process Communications (IPC)
    - Synchronization between processes
    - Threads
    - Software architecture for a server
- Notions about parsing ; Use in computer applications
    - Vocabularies, grammars
    - Lexical analysis, syntactic analysis
    - Regular expressions
    - Finite state automata
    - Algorithmic principles for parsing

Learning materials and literature:

- Handouts:
  - « Operating systems concepts and practice » (annotated slides)
  - « Notions on translation, applications in the computer science field » (annotated slides)
- Literature
  - C. Blaess. Programmation système en C sous Linux : signaux, processus, threads, IPC et sockets, 2è Édition. Eyrolles, Paris, Paris, France, 2005.
  - J.-M. Rifflet. La programmation sous UNIX, 3è Édition. Ediscience International, Paris, France, 1995.
  - D. Guine, H.E. Bal, C.J.H. Jacobs, K.G. Langendoen. Compilateurs. Dunod, Paris, France, 2002
  - J.E.F. Friedl. Mastering Regular Expressions, Perl, .NET, Java and more. O'Reilly, 2002.

Person in charge: Michel Simatic (Michel.Simatic_AT_it-sudparis.eu) / Dr. François Trahay (francois.trahay_AT_it-sudparis.eu)

Faculty:

- Dr. François Meunier
- Dr. Frédérique Silber-Chaussumier
- Michel Simatic
- Dr. François Trahay

Code : CSC4509	***Algorithms and communications for distributed application***	
Period: S8 – P4	**ECTS:** 4	**Language:** French

Organisation: Face to face: 45h Personal work: 45h Total workload: 90h

Courses: 0h Tutorial: 20h Lab: 25h

Les cours intégrés (CI/TD) consistent en 20h d'introduction à la conception d'applications réparties. Les travaux pratiques se décomposent en 21h de manipulations réalisées en binôme et en 3h (1 TP noté) de manipulations notées réalisées en monôme. L'ensemble des TP permettent de réaliser une étude de cas en binôme, avec un travail complémentaire à réaliser en binôme et un compte rendu (ou rapport) sur le travail complémentaire à rendre en fin de module.
Tutorials are up of 20h of introduction to concepts of distributed applications. Labs are made up of 22 hours of experiments done by students working in pairs and 3 hours of graded labs done by students working independently. There is an additional work to be done in pair with a report to be delivered at the end of the module.

Assessment:
Final grade = 1/2 Graded lab + 1/4 personal work (questionnaires) + 1/4 report

Objectives:

- Master concepts, application programming interfaces and tools that help building services and applications that execute upon the TCP/IP transport layer
- Understand structure of distributed algorithms
- Have faced fundamental problems of distributed algorithms

Keywords:
Distributed algorithms, message exchanges (Sockets, UDP, TCP), remote activity invocation (RPC, RMI), name services (DNS, LDAP), distributed data management (NFS, AFS)

Prerequisites:

- Sequential algorithmic (CSC3002, good skills)
- Imperative programming with C programming language (CSC3002, good skills)
- Object Oriented programming with Java programming language (CSC4002, good skills)
- System call programming (Posix, Linux system calls) (CSC4501, good skills)

Course outlines:

- Distributed Algorithms
    - Basic principles (synchronism/asynchronism, specifications, models...)
    - Overview of distributed systems : Motivation and fundamental characteristics
    - Wave and traversal algorithms, broadcast and echo
    - Group communication
    - Election algorithm, mutual exclusion, deadlock detection and termination detection
- Distributed communication
    - Communication models between remote applications (message passing, remote procedure call, distributed shared memory)

- Unreliable communication using datagram (UDP)
- Reliable full duplex communication (TCP)
- Asynchronous timing model, non-blocking read, multi-client applications, multi-servers

Learning materials and literature:

- Handouts
  - « Distributed algorithms » (online course with simulations)
  - « Architecture of distributed systems » (annotated slides)
  - Oracle online tutorials, series "The Java tutorials"
- Literature
  - A.S. Tanenbaum. Distributed Operating Systems. Prentice Hall, 1995.
  - G. Tel. Introduction to Distributed Algorithms 2nd edition. Cambridge University Press, 2000.
  - - A. Hitchens. JAVA NIO, O'Reilly, 2002

Person in charge: Dr. Denis Conan (Denis.Conan_AT_it-sudparis.eu)

Faculty:

- Pr. Christian Bac
- Dr. Denis Conan

Code: CSC5001	***High performance systems***	
Period: S9 – P1	**ECTS:** 4	**Language:** French

Organization:	Face to face: 45h	Personal work: 45h	Total workload: 90h
	Courses: 30h *(Courses)*	Tutorial: 0h *(Tutorial)*	Lab: 15h *(Lab)*

Labs are divided into two parts. 15 hours with no evaluation are realized in groups of two and 6 hours with evaluation are realized independently.

Assessment:
Final mark = ½ Micro-project + ¼ Evaluated lab 1 + ¼ Evaluated lab 2

Objectives:
- Master cluster- and grid-based architectures
- Design, develop, manage high-performance and high-availability systems
- Design and develop parallel systems

Keywords:
Clusters, SSI, OpenMP, MPI, GPU

Prerequisites:
- Algorithmic (good knowledge)
- C programming language (good practice)
- Unix (good user level practice)
- Threads (good user level practice, as experienced in CSC4508)

Course outlines:
- Parallel programming basics
  - Principles, concepts
  - Hardware architectures
  - Solutions and trends
- Parallel algorithms
  - Communication model
  - Data parallelism (matrix / vector, matrix / matrix product)
  - Task parallelism (sort, irregular applications)
- PC clusters
  - Administration : Installation, Tuning, Tools
  - Single System Image : Principles, Installation, Exploitation
- Development tools
  - OpenMP : Principles, API, Compilation, execution, tuning, parallel application development
  - MPI : Principles, API, Compilation, execution, tuning, parallel application development
  - GPU : Principes, API CUDA, Compilation, execution

    - Hybrid programming: MPI + OpenMP
- Project
- Industrial conferences

Learning materials and literature:

- Handouts
    - « Parallel computing basics » (slides)
    - « Clusters » (slides)
    - « GP GPU » (slides)
    - « OpenMP » (slides)
    - « MPI » (slides)
- Literature
    - A. Grama, A. Gupta, G. Karypis and V. Kumar, Introduction to parallel computing, Addison-Wesley, 2003, 2nd edition
    - OpenMP Application Program Interface, Version 2.5, Public draft, November 2004
    - Marc Snir et al., MPI : The complete reference, MIT Press, 1996

Person in charge: Dr. Frédérique Silber-Chaussumier (frederique.silber-chaussumier_AT_it-sudparis.eu)

Faculty:

- Dr. Elisabeth Brunet
- Dr. Christian Parrot
- Christian Schüller
- Dr. Frédérique Silber-Chaussumier
- Dr. François Trahay

Code: CSC5002	***Middleware for distributed applications***	
Period: S9 – P2	**ECTS:** 4	**Language:** French

Organization: Face to face: 45h Personal work: 45h Total workload: 90h

Courses: 25h30 Tutorial: 3h Lab: 16h30

In this course, students achieve their learning process thanks to lectures, labs, projects, and industrial conferences.

Assessment:
Most of the technologies and middleware studied in this course are more deeply experienced during a project. The achievement of this project is subject to an assessment through a written report (R) and an oral presentation (P) which includes a demo.
Middleware not experimented during the project is assessed with a lab (L).

Final score = Average (L, R, P)
The course is validated if and only if the final score is greater than or equal to 10/20.

Objectives:

- Know design frameworks for distributed applications
- Know the different software techniques for designing distributed applications: synchronous requests, asynchronous requests, 3-tier architecture, service persistency, Service Oriented Architectures (SOA), component assembly, service orchestration
- Master technologies for producing enterprise distributed applications: CORBA, MOM, Web Services, J2EE, BPEL, SCA
- Develop enterprise distributed applications

Keywords:
Middleware, distributed application, service, component, CORBA, J2EE, SOA, Web Services, SCA

Prerequisites:

- Application modeling and implementation with UML and Java (seen in Télécom INT, in CSC4002 course),
- Databases (seen in Télécom INT, in CSC4001 course)
- Fundamentals of distributed applications and distributed algorithms (seen in CSC4509)

Course outlines:
The course is divided into three main parts: Fundamentals for middleware, component-based middleware, and web services.

- Fundamentals for middleware
  - First overview of architecture solutions (MDE/MDA, SOA) and middleware (RMI, CORBA, OSGI, J2EE, Web Services, REST)
  - Patterns for distributed middleware
- Middleware for synchronous requests (RMI, CORBA, Web Services)
  - Study of middlewares for synchronous requests

    - Interoperability protocols (IIOP and SOAP)
    - Bases for implementing server applications
- Messages-oriented middleware (MOM)
    - Study of a MOM (JMS)
- Component-based middleware with J2EE
    - Main concepts of component-oriented middleware (containers, extra-functional properties)
    - Extra-functional properties and services illustrated for the J2EE middleware (naming service, notification service, transaction service, persistency service, security service)
    - EJB components
- Web Services
    - Orchestration of web services (BPEL)
    - Service web composition (SCA service component architecture)
- Project
- Industrial conferences

Learning materials and literature:

- Handouts:
    - « Middleware basics » (slides)
    - « Component-based middlewares, J2EE » (slides)
    - « Services Web » (slides)
- Literature
    - Gerald Brose, Andreas Vogel, and K. Duddy. JavaTM Programming with CORBA TM : Advanced Techniques for Building Distributed Applications (3rd edition). Jonh Wiley & sons, Inc., USA, January 2001.
    - Douglas C. Schmidt, Michael Stal, Hans Rohert, and Frank Buschmann: Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects, John Wiley and Sons, 2000.
    - Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju : Web Services: Concepts, Architecture and Applications, Springer-Verlag, New York, 2004.
    - Richard Monson-Haefel, Entreprise Java Beans, 4th Edition, O'Reilly Media, Inc, 2004.
    - http://www-inf.it-sudparis.eu/cours/middleware

Person in charge: Pr. Chantal Taconet (Chantal.Taconet_AT_it-sudparis.eu)

Faculty:

- Dr. Sophie Chabridon
- Dr. Denis Conan
- Pr. Chantal Taconet
- Dr. Samir Tata

Code: CSC5003	***Semantic Web and Applications***	
Period: S9 – P3	**ECTS:** 4	**Language:** French

Organization: Face to face: 45h Personal work: 45h Total workload: 90h

Courses: 21h Tutorial: 7h30 Lab: 16h30

This teaching unit includes 21 hours of courses and 16.5 hours of labs. Some selected courses are delivered by invited faculty. Labs are made up of 15 hours of experiments done by students working in a group of 2 people and 1.5 hours of graded labs done by students working by themselves.

Assessment:
The validation of this teaching unit is based on a graded lab and a synthesis of research articles which gives rise to a report.
Final score = Average (graded lab, report)

Objectives:

- Know "Semantic webs"
- Master definition and management of an ontology
- Define and manage XML and metadata technologies

Keywords:
Semantic Web, Web of data, Web 2.0, Ontologies, Metadata, RDF/RDFS, OWL

Prerequisites:

- SQL (notions)
- XML (notions)
- Java (notions)

Course outlines:

- Introduction to the semantic Web
  - Current Web versus future Web
  - Semantic Web principles
  - Semantic Web architecture
  - New usages of Web 2.0
- Ontologies for the semantic Web
  - ontology notion
  - ontology building, representation, exploitation
  - Review of some of the active projects and initiatives (protégé, jena, ontobroker…)
- Semantic Web languages
  - XML and XML-Schema
  - RDF and RDF-Schema
  - OWL
- Semantic Web applications

    - Rich Sites Summary feed (RSS)
    - FOAF (The Friend of a Friend) project
- Industrial conferences

Learning materials and literature:

- Handout
    - « Semantic web and applications » (slides)
- Literature
    - Philippe Laublet, Jean Charlet et Chantal Reynaud, "Introduction au web sémantique", Information Interaction Intelligence, N° hors-série de la revue I3, Ed.: Cépaduès (juin 2005), ISBN: 2854286669
    - A. Sheth, "Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics", in Interoperating Geographic Information Systems. M. F. Goodchild, M. J. Egenhofer, R. Fegeas, and C. A. Kottman (eds.), Kluwer, Academic Publishers, 1998, pp. 5-30.
    - Tim Berners-Lee, James Hendler and Ora Lassila, "The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities", Scientific American, May 2001.
    - http://www.w3.org/RDF/
    - OWL Web Ontology Language, W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-owl-features-20040210/
    - Ontology Development Tutorial by Natasha Noy, http://protege.stanford.edu/publications/ontology_development/ontology101.shtml

Person in charge: Pr. Amel Bouzeghoub (Amel.Bouzeghoub_AT_it-sudparis.eu)

Faculty:

- Dominique Bouillet
- Pr. Amel Bouzeghoub
- Pr. Bruno Defude
- Dr. Alda Gancarski
- Dr. Claire Lecocq

Code: CSC5004	***Middleware for mobile, ubiquitous and pervasive computing***	
Period: S9 – P4	**ECTS:** 4	**Language:** French

Organization:	Face to face: 45h	Personal work: 45h	Total workload: 90h
	Courses: 0h *(Lectures)*	Tutorial: 24h *(Tutorial)*	Lab: 21h *(Lab)*

During tutorials, notions are presented by members of the ACMES team of laboratory CNR UMR 5157 SAMOVAR. They are immediately put in practice by manipulating the presented concepts. The courses are complemented by conferences by experts from the industry

Assessment:
This course is validated by a programming exam which makes a synthesis of the concepts presented during this teaching unit.

Objectives:

- Apprehend the issues of mobile, ubiquitous and pervasive computing;
- Understand the scientific and technical stakes of the new ways of communicating between people, machines, mobile terminals and sensors;
- Acquire the basic middleware concepts and tools in the domain of mobile, ubiquitous and pervasive computing;
- Improve capabilities for research and development.

Keywords:
Middleware, mobile computing, ubiquitous computing, pervasive computing.

Prerequisites:

- Good knowledge of Java language and object modelization (seen in Telecom Sudparis CSC4002)
- Good knowledge of distributed computing: Communication primitives (sockets, RPC...) and distributed algorithms (leader election, mutual exclusion, deadlock detection, termination detection, group communication...) (seen in Telecom SudParis with CSC4509)
- Good knowledge of general purpose middleware: CORBA, J2EE, Web services… (seen in Telecom SudParis with CSC5002)

Course outlines:
The contents of this course is designed so that students can discover and practise concepts and tools for mobile, ubiquitous and pervasive computing. The main subjects are based on the current active research studies performed by the MARGE team of the CNRS UMR 5157 SAMOVAR laboratory. The Labs take place in lab "Ambient computing and mobiles", equipped with sensors (movements, temperature…), NFC-enabled phones, UMPC, webcams, Google/Android phones, home automation…
Here follows the program:

- Introduction to middleware for mobile, ubiquitous and pervasive computing: Mobile environments; constrained environments (embedded systems and sensors);

- Programming models for ambient computing: using sensors, event-based programming, software-agent-based programming;
- Middleware services: Context awareness and adaptation; context management; software caching; consistency; session continuity; mobile multiplayer games;
- Peer-to-peer, distributed hash table;
- Study of technologies of ambient systems: X10 protocol, Bluetooth communication, UPNP, SIP…
- Conferences: Seminars given by invited experts and researchers.

Learning materials and literature:

- Handouts
    - D. Belaïd, Services integration (slides)
    - D. Conan, Context management (annotated slides)
    - B. Defude, Peer-to-peer (slides)
    - S. Leriche
        - Sensors and sensor networks (slides)
        - Software mobility (slides)
        - OSGI, a service platform for pervasive systems (slides)
    - M. Simatic, Technologies for NFC programming on mobiles (slides)

Person in charge: Dr. Sébastien Leriche (Sebastien.Leriche_AT_it-sudparis.eu)

Faculty:

- Dr. Djamel Belaïd
- Pr. Bruno Defude
- Dr. Sébastien Leriche
- Michel Simatic

Code: CSC5005	**ASR major's project**	
Period: S9	**ECTS:** 8	**Language:** French

Organization: Face to face: 27h Personal work: 198h Total workload: 225h

ASR major's project is done during the whole semester 9. Each student must make a project with one or two other students. Planning time slots are dedicated to the project. Meetings 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:

- Portable WebID identity provider based on Node.js
- Research engine for Point of Interest (POI)
- Program transformation: Generating a multi-GPU MPI + HMPP program from a GPU program with HMPP directives
- Video game with numerous players and a single (wide) screen
- Multi-scale discovery and signalization

Person in charge: Michel Simatic (michel.simatic_AT_it-sudparis.eu)

Faculty:

All lecturers of ASR major