ReSIST: Resilience for Survivability in IST

A European Network of Excellence

Contract Number: 026764

Deliverable D37: Resilient Computimg Curriculum

Report Preparation Date: December 2008
Classification: Public

Contract Start Date: 1st January 2006
Contract Duration: 39 months
Project Co-ordinator: LAAS-CNRS
Partners: Budapest University of Technology and Economics
City University, London
Technische Universität Darmstadt
Deep Blue Srl
Institut Eurécom
France Telecom Recherche et Développement
IBM Research GmbH
Université de Rennes 1 – IRISA
Université de Toulouse III – IRIT
Vytautas Magnus University, Kaunas
Fundação da Faculdade de Ciencias da Universidade de Lisboa
University of Newcastle upon Tyne
Università di Pisa
QinetiQ Limited
Università degli studi di Roma "La Sapienza"
Universität Ulm
University of Southampton

Deliverable D37:
Resilient Computing Curriculum

Co-ordinator: Luca Simoncini

Contributors in ReSIST: Roberto Baldoni, Cinzia Bernardeschi, Robin Bloomfield, Andrea Bondavalli, Christian Cachin, Miguel Correia, Marc Dacier, Felicita Di Giandomenico, Jean-Charles Fabre, Michael Harrison, Mohamed Kaâniche, Karama Kanoun, Chidung Lac, Giuseppe Lami, Jean-Claude Laprie, Istvan Majzik, Paolo Masci, Philippe Palanque, Andras Pataricza, Holger Pfeifer, Michel Raynal, Luca Simoncini, Lorenzo Strigini, Neeraj Suri, Guillame Urvoy-Keller, Paulo Verissimo, Friedrich von Henke, Helene Waeselynck, Marco Winckler.

External contributors: Wolfgang Ahrendt, Ricardo Jimenez-Peris, Johan Karlsson, Udo Voges.

Comments: ReSIST EB Committee, ReSIST T&D Committee.

Contents:

Introduction

The Bologna Declaration, signed on June 19, 1999 [1], and the subsequent documents [2], [3] have identified three big reform areas in higher education:

- Curricular reform: The three cycle system, competence based learning, flexible learning paths, recognition, mobility
- Governance reform: University autonomy, strategic partnerships, quality assurance
- Funding reform: Diversified university income, tuition fees, grants and loans, equity and access, EU funding

and have started the so-called Bologna process that is forecast to be completed in 2010. In particular [1], Universities and other institutions of higher education may:

- Profile their own curricula, in accordance with the emerging post-Bologna environment, in particular through the introduction of bachelor courses in systems where they have not traditionally existed, and **through the creation of master courses** meeting the needs of mobile postgraduate students from around the world;
- **Activate their networks in key areas such as joint curriculum development**, joint ventures overseas or worldwide mobility schemes;
- **Contribute individually and collectively** to the next steps in the process.

In 2003, the Computing Research Association has identified the following Four Grand Challenges in Trustworthy Computing [5]:

- Challenge 1: Eliminate Epidemic Attacks by 2014
- Challenge 2: Enable Trusted Systems for Important Societal Applications
- Challenge 3: Develop Accurate Risk Analysis for Cybersecurity
- Challenge 4: Secure the Ubiquitous Computing Environments of the Future

identifying very strategic fields of activity necessary to fill gaps that limit the introduction of resilience in many complex computer-based critical applications.

Starting January 2006 and lasting three years, the EU has funded a NoE called ReSIST – Resilience for Survivability in IST [4], that collects 18 partners, among the most well-recognized groups in Europe expert in dependability, security and human factors with the following objectives:

- **Integration** of teams of researchers so that the fundamental topics concerning scalable resilient ubiquitous systems are addressed by *a critical mass* of co-operative, multi-disciplinary research.
- **Identification**, in an international context, of the key *research directions (both technical and socio-technical)* induced on the supporting ubiquitous systems by the requirement for trust and confidence in AmI.
- **Production of significant research results** *(concepts, models, policies, algorithms, mechanisms)* that pave the way for scalable resilient ubiquitous systems.
- **Promotion and propagation of a resilience culture in university curricula** and in engineering best practices.

ReSIST has therefore identified in Work Package 3 - Training and Dissemination an activity towards the preparation of a MSc curriculum in Resilient Computing as properly providing a timely and necessary answer to requirements posed by EU.

The decision to dedicate an entire Work Package to educational issues related to the development of a MSc Curriculum in Resilient Computing has been quite timely, if we consider the present activity that is undertaken by the ASysT Institute http://www.asysti.org/issechome.aspx with the iSSEc (integrated Software and Systems Engineering curriculum) Project created in Spring 2007 to develop the Graduate Software Engineering Reference Curriculum (GSwERC) – a new reference graduate SwE curriculum that reflects new understandings in how to build software; how software engineering depends on systems engineering; and how software engineering education is influenced by specific technological domains, such as telecommunications and defense systems. GSwERC is

intended to be suitable for university-level education leading to a Masters Degree in Software Engineering (SwE).

This Deliverable presents the MSc Curriculum in Resilient Computing suggested by ReSIST. It includes the description of the syllabi for all the courses in the two semesters of the first year, those for the common courses in semester 3 in the second year together with an exemplification of possible application tracks with the related courses. This MSc curriculum has been updated and completed taking advantage of a large open discussion inside and outside ReSIST. This MSc Curriculum is on-line on the official ReSIST web site, where all information is available together with all the support material generated by ReSIST and all other relevant freely available support material.

1. Activity in ReSIST for the Curriculum

ReSIST has dedicated a constant and continuous integrated effort towards the identification of the MSc Curriculum in Resilient Computing and towards the identification of the syllabi of the identified courses, the generation of original support material and the link to relevant freely available support material.

The first step of this activity was dedicated to internal interviews to all members in ReSIST for obtaining a snapshot of what is offered inside ReSIST as courses (of all kinds: university, short, industrial courses) so to have a possibly comprehensive view on the following: i) present expertise inside ReSIST, and ii) how to extend to the present curriculum the gathered information. From this activity 54 forms have been collected. They have been reported in Deliverable D16.

A second step, parallel to the first one, was a survey through searching the web, to identify relevant links to Universities and Organizations inside and outside ReSIST that offer Resilience-related courses, to extend the base of knowledge coming from step one. Also the output of this second step has been reported in Deliverable D16.

The third step of this activity has been the organization in the frame of EDCC-6, Sixth European Dependable Computing Conference, held in Coimbra, Portugal on 18-20 October 2006, of a Panel Session on “Education in Dependable and Resilient Computing – Meeting the Needs of the Information Society”, where panellists from Chalmers University, Polytechnic University of Madrid, University of Lisbon and University of Pisa discussed on the topics.

The fourth step was an Open T&D Meeting, held in London on January 31, 2007. At the meeting EWICS-TC7 was invited to coordinate efforts towards a commonly agreed curriculum. In that meeting the skeleton of the present curriculum was outlined, and was decided to call for a Joint ReSIST/EWICS TC-7 Workshop on Teaching Resilient Computing, held in Erlangen on May 2, 2007, reported in Deliverable D16, with the set of presentations publicly available on the ReSIST web site.

On May 3, 2007, an Open T&D Committee Meeting was held in Erlangen to consolidate the outputs of the Workshop.

With the first version of the MSc Curriculum ready, a large dissemination, information and consultation activity has started and has covered the last 18 months of ReSIST. In particular the Curriculum has been presented to both flagship Conferences DSN’07 held in Edinburgh, UK in June 2007 and DSN’08 held in Anchorage, Alaska in June 2008, with a Special Session dedicated to it. The Curriculum has been presented to the IFIP W.G. 10.4 held in conjunction with DSN’07 and DSN’08, at the IFIP W.G. 10.4 held in Natal, Brazil in February 2008, and in a Special Session during EDCC-7 held in Kaunas, Lithuania in May 2008. It has also been presented to the European Computer Science Summit held in Berlin, Germany in September 2007. We consider that, if not all, a very large percentage of the community working in Dependable and Resilient Computing have been informed of this activity in Europe, US, Latin America and in Japan. A large set of comments and suggestions have been received and have been incorporated in the present version of the Curriculum.

The dissemination activity will continue after the end of ReSIST through a direct invitation to Universities who have activities in the field of higher education on Dependable and Resilient

Computing to consider the opportunity to start tracks based on this Curriculum, with the aim of favouring also internationalisation by joint agreements between Universities and the possibility of granting joint Master degrees.

2. Curriculum description

A MSc curriculum in Resilient Computing that covers years 4 and 5 of a University Master track is usually composed by a total of 120 ECTS [6], evenly divided in two years (60 ECTS each year). Even if this is not a common characteristic among European Universities, it was agreed to take as working assumption that a MSc curriculum corresponds to 120 ECTS. It was decided to work on an organization of the curriculum into 4 semesters, two for each year, each tentatively 30 ECTS worth: the 1st semester on Basics and Fundamentals, the 2nd on Methods, Techniques and Tools, the 3rd on Projects in cooperation with industry on specific application fields, and the 4th for Master's Thesis and Dissertation.

There are three phases in the curriculum. In the first phase (2 semesters, 60 ECTS), fundamental knowledge and skills are introduced through modules in: system dependability and security; advanced information security; human factors engineering; distributed and fault-tolerant computing; system validation and assessment, ranging from theoretical bases to methods, techniques and tools. In the second phase (3rd semester, 30 ECTS), the practice of resilient computing is emphasized through modules in high-integrity software development and research skills, followed by a group project on the development and assessment of a real system in specific application domains.

The third phase of the curriculum (4th semester, 30 ECTS) is a six-month individual system development or research project, undertaken with personal supervision of one senior scientist, or in industry, and will be concluded with the preparation and presentation of a Master Thesis.

Following the average indication of the EC that identifies 1 ECTS as worth 25 hours of student work, and that this work has to be flexibly associated to each activity on the basis of hours of lectures, hours of labs, hours of individual study (preparation of exams and of the MSc thesis), the number of ECTS assigned to each course has been distributed with the following rationale:

1) The first two semesters require a rather huge theoretical preparation for the student with a shift between lectures to practical exercise from the first to the second semester. With this approach, since the courses of the first semester are all either 6 or 3 ECTS worth, the courses 6 ECTS worth require 40 hours lectures + 20 hours exercising + 90 hours individual study and the courses 3 ECTS worth 20 hours lectures + 15 hours exercise + 40 hours individual study; in the second semester all courses are 6 ECTS worth and we consider a shift in the required effort as 30 hours lecture + 30 hours exercise + 90 hours individual study. In the second year of the third semester, the common courses are all 3 ECTS worth and we have maintained 20 hours lectures + 15 hours exercise + 40 hours individual study. Obviously the 3 ECTS worth courses in the three Application tracks are more oriented to labs with the suggestion of possible distribution in terms of hours as 15 hours lecture + 20/30 hours exercise + 40/30 hours individual study. The project in cooperation with industry (9 ECTS worth) mainly requires many hours of labs and individual work for a total of 225 hours to which a flexible 6 ECTS worth space for additional courses and/or seminars (strictly related to the project) will require lectures, exercise and individual study for a total of additional 150 hours. During the fourth semester, the main activity will be the preparation of the MSc Thesis (27 ECTS worth) for a total of mainly individual study of 675 hours + a total of 75 hours for seminars and additional courses aimed at the specificity of the MSc Thesis.

We can summarise the total effort (student work) during the first two semesters as:

Lecture hours: 350 hours
Exercise and labs: 270 hours
Individual study: 880 hours

For the 3rd and 4th semesters it is more difficult to provide a distribution of hours, but in an indicative way we can consider:

Lecture hours: 160 hours
Exercise and labs: 215/235 hours
Individual study: 1125/1105 hours

If we sum up the two lists we have:
Total number of hours (total student effort): 3000 hours (consistent with 120 ECTS each 25 hours worth)
Total number of lectures + exercise and lab: 995/1015 hours
Total number of hours of individual study: 2005/1985 hours

2.1 Curriculum Pre-requisites

A student who wants profitably enrol to the MSc Curriculum in Resilient Computing would take advantage from having a basic knowledge in the following fields:

- Discrete Mathematics
- Calculus
- Basic Computer and Network Architectures
- Programming and Data Structures
- Basics of Operating Systems
- Basics of Software Engineering
- Basics of Probability and Statistics

This basic knowledge has to be provided in the first phase of the higher education scheme of the Bologna process.

2.2 Curriculum aims

The aims of the curriculum are:

- To equip students with the skills and knowledge required to develop and assess secure, dependable and resilient computer-based systems
- To provide a qualification enhancing employment prospects in resilient computing
- To develop research skills
- To develop and improve key skills in written and oral communication and in teamwork
- To develop and improve skills in using the literature and information technology resources relevant to resilient computing
- To encourage the development of creativity skills
- To develop skills in critical assessment, analysis and storage of information
- To provide a curriculum which meets the requirements of appropriate professional bodies, thus providing a basis for further professional development and lifelong learning
- To address the relevant professional, legal and ethical issues relevant to the development, assessment and maintenance of resilient systems
- To provide an international perspective on developments in computer resilience.

2.3 Knowledge and understanding

A successful student will have gained and be able to demonstrate:

- Understanding of the theory underpinning dependability, security and resilience in computer-based systems
- Knowledge of major and advanced techniques, methods and tools for assessing information security and system dependability and resilience
- Knowledge of the major and advanced fault tolerance techniques, methods and tools applicable in computer system design
- Understanding of the technologies for the design of trustworthy interactive systems, including human error assessment
- Understanding of the computer aided verification techniques relevant to security in distributed systems
- Understanding of the principles underlying high integrity software development using advanced static analysis and formal techniques
- Understanding of major professional, legal and ethical issues associated with work in secure and dependable computing systems

- Understanding of the international character of contemporary developments in security, dependability and resilience.

2.4 Professional skills

A successful student will:

- Be able to propose, conduct and write up an extended research project involving where appropriate, a literature review, problem specification, design, verification, implementation and analysis
- Be able to design, implement and validate new software for secure, dependable and resilient applications
- Be able to organize and take part in systematic analyses of existing systems
- Have expertise in the use and applicability of up-to-date software development tools
- Be able to assess the main human factors relevant to secure and dependable system operation
- Be able to apply the leading techniques for security in networks and Internet environments, including cryptography and public key infrastructures
- Be able to apply the major methods for assessing system resilience
- Be able to deploy fault tolerance appropriately in system design.

2.5 Key skills

A successful student will have:

- The ability to communicate orally in English in a professional context
- Written communication skills, including an appreciation of the role of peer review of papers, software, proposals and other research and development products
- Information literacy skills, including the ability to use computer-based resources for research in the professional literature and the capacity to undertake critical reviews
- The ability to work as part of a team, including group-based learning, research and development activity
- Creativity skills: recognizing and responding to opportunities for innovation
- Planning and organization skills.

3. Curriculum organization

The curriculum is structured in 4 semesters, 30 ECTS each, over two years, as in the following Tables. The number of ECTS is indicative of the relative weight among the several courses. Courses worth 6 ECTS are taught in parallel, while there is an ordering between courses worth 3 ECTS:

1st Year

1st semester: Basics and Fundamentals (30 ECTS)	**2nd semester: Methods, Techniques and Tools (30 ECTS)**
Courses: • **Advanced Probability and Statistics** (6 ECTS) • **Cryptology and Information Security** (6 ECTS) • **Logic in Computer Science** (6 ECTS) • **Advanced Graph Theory** (3 ECTS) • **Human Factors, Human and Organisational Behaviour** (3 ECTS) • **Fundamentals of Real-Time Systems** (3 ECTS) • **Fundamentals of Dependability** (3 ECTS)	Courses: • **Computer Networks Security** (6 ECTS) • **Resilient Distributed Systems and Algorithms** (6 ECTS) • **Dependability and Security Evaluation of Computer-based Systems** (6 ECTS) • **Testing, Verification and Validation** (6 ECTS) • **Usability and User Centred Design for Dependable and Usable Socio-technical Systems** (6 ECTS)

1st Semester scheduling (time flows from left to right)

Advanced Probability and Statistics	
Cryptology and Information Security	
Logic in Computer Science	
Advanced Graph Theory	**Human Factors, Human and Organisational Behaviour**
Fundamentals of Real-Time Systems	**Fundamentals of Dependability**

2nd Semester scheduling (time flows from left to right)

Computer Networks Security
Resilient Distributed Systems and Algorithms
Dependability and Security Evaluation of Computer-based Systems
Testing, Verification and Validation
Usability and User Centred Design for Dependable and Usable Socio-technical Systems

2nd Year

3rd semester: Projects (in cooperation with industry on specific application fields) (30 ECTS)	**4th semester: Master's Thesis and Dissertation (30 ECTS)**
Courses (common to all application tracks) • **Management of Projects** (3 ECTS) • **Middleware Infrastructures for Application Integration** (3 ECTS) • **Software Reliability Engineering** (3 ECTS) Application track: **Resilience in Communication Networks** Courses (specific for this track): • **IP Networks and Service Resilience** (3 ECTS) • **Resilience of Mobile Applications** (3 ECTS) Application track: **Safety critical Systems** Courses (specific for this track): • **Development Process and Standards for Safety critical Applications** (3 ECTS) • **Architectural Issues and Examples of Systems** (3 ECTS) Application track: **Resilience in e-Business** Courses (specific for this track): • **Enterprise Security** (3 ECTS) • **Computer and Network Forensics** (3 ECTS) Common to all Application tracks: • **Project in cooperation with Industry** (9 ECTS) • **Space for additional Courses** (6 ECTS)	• **Specific Courses and Seminars** (3 ECTS) • **Preparation and Presentation of the Thesis** (27 ECTS)

3rd Semester scheduling (time flows from left to right)

<table>
<tr><td rowspan="3">Management of Projects

Middleware Infrastructures for Application Integration

Software Reliability Engineering</td><td>Appl. Track: Resilience in Communication Networks:
• IP Networks and Service Resilience
• Resilience of Mobile Applications</td></tr>
<tr><td>Appl. Track: Safety critical Systems:
• Development Process and Standards for Safety critical Applications
• Architectural Issues and Examples of Systems</td></tr>
<tr><td>Appl. Track: Resilience in e-Business:
• Enterprise Security
• Computer and Network Forensics</td></tr>
<tr><td colspan="2">Project in cooperation with Industry
Additional Courses</td></tr>
</table>

4th Semester scheduling (time flows from left to right)

<table>
<tr><td>Specific Courses and Seminars</td></tr>
<tr><td>Preparation and Presentation of the MSc Thesis</td></tr>
</table>

4. Courses syllabi

1ST Year – 1st Semester

4.1 Advanced Probability and Statistics (6 ECTS)

The purpose of this course is to provide a methodical advanced background of probability, stochastic processes, and statistics needed to the students to address modelling and assessment issues in resilient computing.

Contents

- Introduction
- Discrete random variables
- Continuous random variables
- Expectation
- Conditional distribution and expectation
- Bayesian probability and inference
- Stochastic Processes

Suggested readings:

K. S. Trivedi: **Probability and Statistics with Reliability, Queuing, and Computer Science Applications**, Second Edition, John Wiley & Sons, 2002

Courseware examples and locations where taught:

Slides from the same author, available at http://www.ee.duke.edu/~kst/

4.2 Cryptology and Information Security (6 ECTS)
The purpose of this course is to give an up-to-date treatment of the principles, techniques, and algorithms of interest in information security and cryptography. Emphasis is on fundamental concepts and their practical applications.

Contents
- Introduction
- Information security and cryptology
- Access control and security policies
- One-way functions and pseudorandomness
- Hash functions
- Symmetric-key encryption
- Public-key encryption and digital signatures
- Authentication and identification protocols
- Key agreement, certificates, and public-key infrastructures
- Key management and trust management
- Anonymity and privacy

Suggested readings:
A. J. Menezes, P. C. van Oorschot and S. A. Vanstone: **Handbook of Applied Cryptography,** CRC Press, 1996.
http://www.cacr.math.uwaterloo.ca/hac/
N. Smart: **Cryptography, An Introduction**, McGraw-Hill, 2002.
http://www.cs.bris.ac.uk/~nigel/Crypto_Book/

Courseware examples and locations where taught:
- Saarland University, Cryptography, Michael Backes, http://www.infsec.cs.uni-sb.de/teaching/SS08/Cryptography/
- ETH Zürich, Information Security, David Basin, http://www.infsec.ethz.ch/education/ss08/infsec08
- ETH Zurich, Cryptography, Ueli Maurer, http://www.crypto.ethz.ch/teaching/lectures/Krypto06/
- KU Leuven, Cryptography and Network Security, Bart Preneel, http://www.kuleuven.ac.be/onderwijs/aanbod2006/syllabi/H0244BE.htm
- Univ. Bristol, Introduction to Cryptography, Elisabeth Oswald and Nigel Smart, http://www.cs.bris.ac.uk/Teaching/Resources/COMS30124/
- MIT, Computer and Network Security, Ronald L. Rivest and Shafi Goldwasser, http://courses.csail.mit.edu/6.857/2008/lecture.html

4.3 Logic in Computer Science (6 ECTS)
The goal of the course is to present the fundamental notions of logic that are important in computer science.

Contents
- Propositional logic
  - Natural deduction
  - Induction
  - Semantics
  - Normal form
  - SAT solving
- Predicate logic
  - Natural deduction
  - Semantics
  - Undecidability
  - Expressivity
- Temporal logics
  - Branching time logic
  - Linear time logic
  - Fixed-point characterisation
  - Repetition

Suggested readings:
The course is based mainly on the 3 first chapters of:
M. Huth and M. Ryan: **Logic in Computer Science**, Cambridge University Press
http://www.ewidgetsonline.com/cup/widget.aspx?bookid=51/3mLE/ColK5qnmfcLSyg==&buyNowLink=http://sec.ebooks.com/cambridge-add.asp?I=283471&f=3

As additional reading, one can point to the hypertext-book by
V. Detlovs, K. Podnieks: **Introduction to Mathematical Logic**
http://www.ltn.lv/~podnieks/mlog/ml.htm

Courseware examples and locations where taught:
The course book's webpage, http://www.cs.bham.ac.uk/research/projects/lics/ offers several materials, among them an interactive tutor for each chapter.

A complete set of slides for the whole course, structured in 14 lectures, is available from the web page of the University of Copenhagen's instance of the course (teachers Julia Lawall & Neil Jones), see
- http://www.diku.dk/

Other places where instances of this course are given, and from where additional teaching material can be downloaded, are:
Chalmers University of Technology (teachers Thierry Coquand & Jan Smith)
- http://www.cs.chalmers.se/Cs/Grundutb/Kurser/logcs

University College London (teacher Jonathan P. Bowen)
- http://www.cs.ucl.ac.uk/staff/J.Bowen/GS03/

Many more places where courses are based on the book by Huth&Ryan are listed at
- http://www.cs.bham.ac.uk/research/projects/lics/adoptions.html

4.4 Advanced Graph Theory (3 ECTS)
The purpose of this advanced course is to present the aspects of graph theory beneficial for the design of resilient systems. "Advanced" means here that the basics of graph theory are already known. So, "advanced" is here the "discovery" of concepts that have been recently (or not) introduced in computer science to model problems related to computability, efficiency, or fault-tolerance.

Contents
- Connectivity and traversability: bounded connectivity, regularity, overlay networks
- Graph coloring and graph NP-complete problems
- Topological graph theory: embeddings, genus and maps
- Analytic graph theory: random graphs, Ramsey graphs and the probabilistic approach
- Graph measurements: domination,and tolerance graph
- Small-world networks (on grid and uniform topology, Kleinberg's distribution)

Suggested readings:
S. Even: **Graph Algorithms**, Computer Science Press, 1979.
J.L. Gross and J. Yellen (Eds.): **Handbook of graph theory**, CRC Press, 2003.

Courseware examples and locations where taught:
- Excerpts of the book by Shimon Even can be downloaded at http://www.wisdom.weizmann.ac.il/~oded/even-alg.html

4.5 Human Factors, Human and Organisational Behaviour (3 ECTS)
The purpose of this course is to present human and organisational fundamental concepts and frameworks that influence and determine failures (catastrophic or not) in complex systems and hence the impact on the resilience of the socio-technical system.

Contents
- Cognitive processes for the description and the prediction of human understanding and information processing capabilities
- Human performance cognitive issues (learning, problem-solving)
- Human performance physiological issues (sensation, perception, motor skills, Fitts' law, steering law, …)
- Human error (concepts and classifications)
- Mode confusion and automation surprises

Suggested readings:
C. W. Johnson: **Failure in Safety-Critical Systems. A Handbook of Accident and Incident Reporting**. Available on-line at: http://www.dcs.gla.ac.uk/~johnson/book/ October 2003. 2003. Glasgow, Scotland, University of Glasgow Press.
J. Reason: **Human Error**. 1990. Cambridge University Press.
J. Reason: **Managing the Risks of Organizational Accidents**, 1997, Aldershot, UK, Ashgate.
C. D. Wickens and J. G. Hollands: **Engineering Psychology and Human Performance**. 3rd edition, 1999, Prentice Hall.
J. Rasmussen, M. A. Pejtersen, L. P. Goldstein: **Cognitive Systems Engineering**. New York, USA, John Wiley and Sons, 1994

Courseware examples and locations where taught:
- http://sigchi.org/cdg/index.html (gathering a large set of lectures on HCI and Human factors mainly in the US. This set of courses has been gathered and organised by the ACM Special Interest Group on HCI)
- http://liihs.irit.fr/palanque/Ps/MasterHM-IntroHCIPalanque.pdf

4.6 Fundamentals of Real-time Systems (3 ECTS)
The purpose of this course is to provide a large overview of fundamentals aspects of real-time system architectures and development. This covers scheduling techniques, scheduling analysis including WCET evaluation, design principles of distributed real-time embedded systems, programming distributed real-time applications. Fault tolerance aspects are also addressed, in particular regarding timing faults handling. Examples of real time executive layers are also presented.

Contents
- Introduction to basic concepts
- Reminder of operating systems basic notions
- Scheduling in real-time systems
- WCET analysis and evaluation
- Design principles of distributed RT applications
- Programming distributed RT systems
- Real-time executives and examples

Suggested readings:
A. Silberschatz, P. Baer Galvin, G. Gagne: **Operating Systems Concepts**, John Wiley & Sons, 2008, ISBN 0-470-12872-0.
F. Cottet, J. Delacroix, C. Kaiser, Z. Mammeri: **Scheduling in Real-Time Systems**, Wiley Eds, 2002, ISBN: 0-470-84766-2
G. Buttazzo: **Hard Real-Time Computing Systems**, Second Edition, Series: Real-Time Systems Series, Vol. 23, 2005, XIII, ISBN: 978-0-387-23137-2 Springer, 2005.
H. Kopetz: **Real-Time Systems: Design Principles for Distributed Embedded Applications**, Series: The Springer International Series in Engineering and Computer Science, Vol. 395, 1997, ISBN: 978-0-7923-9894-3
A. Burns and A. J. Wellings: **Real-Time systems and programming languages**, 3rd ed., Addison Wesley, 2001, ISBN 0-201-40365-X

Courseware examples and locations where taught:
These are examples of places where parts of this course are taught, giving emphasis on some aspects of real-time systems.
- Yale University: operating systems concepts. Slides at: http://www.os-book.com/
- University of York (UK): scheduling and programming. See: http://www.cs.york.ac.uk/MSc/Modules/rts.html
- Scuola Superiore Santa Anna Pisa (Italy): scheduling and analysis. Courseware (in Italian) through this page: http://feanor.sssup.it/~giorgio/srt.html
- Universidad Politecnica de Madrid (Spain): real-time and applications. See: http://polaris.dit.upm.es/~jpuente/strl/guia.html
- University of Rennes (France): generic course on real-time. Courseware (in French) through this page: http://www.irisa.fr/caps/people/puaut/puaut.html

4.7 Fundamentals of Dependability (3 ECTS)
The purpose of this course is to give a structured introduction to the concepts of dependability and to the methods and techniques used for dependable design of systems and for scaling to complex resilient systems.

Contents
- Basic concepts and definitions
- State of the art from statstics
- Threats to dependability
- Fault removal
- Fault forecasting
- Fault tolerance
- Development of dependable systems
- From dependability to resilience

Suggested readings:
J-C. Laprie et al.: **Guide de la sûreté de fonctionnement**, Cepaduès Editions,1995 (in French)
D. P. Siewiorek and R. Swartz: **Reliable Computer Systems, Design and Evaluation,** Third Edition, A K Peters, Ltd., 1998
A. Avizienis, J-C. Laprie, B. Randell and C. Landwehr: **Basic Concepts and Taxonomy of Dependable and Secure Computing,** IEEE Trans. on Dependable and Secure Computing, Vol.1, n.1, Jan.- March 2004, pp. 11-33.
J.C. Laprie: **From Dependability to Resilience***,* 38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks, Anchorage, Alaska, June 2008, Sup. Vol., pp. G8-G9

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/dependability_fundamentals.pdf

Courseware examples and locations where taught:
- Higher National School of Aeronautics and Space (ENSAE), Toulouse. Jean-Claude Laprie
- Higher National School of Electronics, Informatics, and Radiocommunications of Bordeaux. Jean-Claude Laprie
- Higher National School of Electrotechnology, Electronics, Informatics, Hydraulics and Telecommunications, Toulouse. Jean-Charles Fabre

These topics are covered in the MSc-level course on "Fault-tolerant design of computer systems" available at City University as a professional development short course. The slides are not publicly available.

1st Year – 2nd Semester

4.8 Computer Networks Security (6 ECTS)

The purpose of this course is to offer a broad overview of computer network security, not only of security building blocks and approaches but also of the existing threats. The course is organized in four parts: fundamental security concepts, paradigms for secure computing and communication, models for secure computing, and secure systems and platforms. The course builds on the basic cryptographic mechanisms introduced in course 4.2.

Contents

- Fundamental security concepts
- Paradigms for secure computing and communication
    - TCB - trusted computing base
    - Basic cryptography symmetric and asymmetric
    - Authentication and key distribution
    - Access control
    - Secure communication
- Models for secure computing
    - Types of attacks and intrusions
    - Security strategies
    - Using cryptographic protocols
    - Authentication models
    - Key distribution approaches
    - Architectural protection
    - Principles of intrusion detection
    - Secure communication and distributed processing
- Secure systems and platforms
    - SSL - secure sockets layer
    - Network layer security: IPSec

Suggested readings:

P. Verissimo and L. Rodrigues: **Distributed Systems for System Architects**, Kluwer, 2001

C. Kaufman, R. Perlman, and M. Speciner: **Network Security: Private Communication in a Public World**, Second Edition, Prentice Hall

W. Stallings: **Cryptography and Network Security**, 4th Edition, Prentice Hall

W. R. Cheswick, S. M. Bellovin, and A. D. Rubi: **Firewalls and Internet Security: Repelling the Wily Hacker**, Second Edition, Addison Wesley

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/computer_network.pdf

Courseware examples and locations where taught:

- Institut Eurecom, Security applications in networking and distributed systems. R. Molva http://www.eurecom.fr/util/coursdetail.fr.htm?id=23
- Institut Eurecom, Operational Network Security. M. Dacier. http://www.eurecom.fr/util/coursdetail.fr.htm?id=19
- Universidade de Lisboa, Faculdade de Ciências. Security. P. Verissimo

4.9 Resilient Distributed Systems and Algorithms (6 ECTS)
The purpose of this course is to prepare the students to understand and design resilient distributed systems and the algorithms underlying those systems. The course presents fault-tolerant systems and algorithms that tolerate not only accidental faults but also malicious faults, being resilient to a wide range of problems. The emphasis is put on systems that tolerate malicious faults.

Contents
- The case for resilience
- Introduction to fault and intrusion tolerance
    - Brief topics on security and dependability
    - Intrusion tolerance
    - Intrusion forecasting
    - Example intrusion-tolerant networks and architectures
- Resilience building paradigms
    - Intrusion detection
    - Self-enforcing vs. trusted third party protocols
    - Threshold cryptography and secret sharing
    - Byzantine reliable broadcast
    - Byzantine consensus and atomic broadcast
    - Byzantine state machine replication
    - Resilience to attacks and limitations of current I/T paradigms
- Models of resilient systems
    - Intrusion tolerance strategies
    - Advanced modelling concepts for I/T systems
    - Hybrid distributed systems models
    - Review of strategies for construction of I/T subsystems
    - Byzantine protocols on asynchronous fail-uncontrolled models
    - Byzantine protocols on hybrid distributed systems models
- Example resilient systems
    - Maftia
    - Oasis

Suggested readings:
P. Verissimo, M. Correia, N. F. Neves, P. Sousa. **Intrusion-Resilient Middleware Design and Validation**. In *Annals of Emerging Research in Information Assurance, Security and Privacy Services*, H. Raghav Rao and S. Upadhyaya (eds.), Elsevier, to appear, 2008.
P. Verissimo and N. F. Neves and M. Correia. **Intrusion-Tolerant Architectures: Concepts and Design.** In *Architecting Dependable Systems*, R. Lemos, C. Gacek, A. Romanovsky (eds.), LNCS 2677, pp. 3-36, Springer, 2003.
P. Verissimo and L. Rodrigues: **Distributed Systems for System Architects**, Kluwer, 2001
R. Guerraoui and L. Rodrigues: **Introduction to Reliable Distributed Programming,** Springer, 2006.

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/resilient_distributed.pdf

Courseware examples and locations where taught:
- EPFL (Switzerland): slides related to the book by Guerraoui and Rodrigues: http://www.di.fc.ul.pt/~ler/irdp/teaching.htm
- ETH Zurich (Switzerland): slides on security and fault-tolerance in distributed systems: http://www.zurich.ibm.com/~cca/sft08/

4.10 Dependability and Security Evaluation of Computer-based Systems (6 ECTS)
The purpose of this course is to present the main concepts and techniques that are commonly used to evaluate the dependability and security of computing systems. Both accidental and malicious threats are addressed considering model-based and experimental evaluation approaches. Examples of applications and case studies are presented for illustration.

Contents
- Introduction
  - Qualitative and quantitative evaluation
- Definition of quantitative measures
- Quantitative evaluation methods
  - Combinatorial models: reliability block diagrams, fault trees
  - State-based models: Markov chains and Stochastic Petri nets
- Dependability data and measurements
  - Assessment based on field measurements
  - Experimental evaluation based on fault injection
- Case studies
- Evaluation with regard to malicious threats
  - Challenges and state of the art
  - Data collection and analysis based on honeypots

Suggested readings:
D.P. Siewiorek and R. Swartz: **Reliable Computer Systems, Design and Evaluation**, Third Edition, A. K. Peters, Ltd, 1998
K. Trivedi: **Probability and Statistics with Reliability, Queuing, and Computer Science Applications**, 2nd Edition, John Wiley and Sons, New York, 2001. Slides available at http://www.ee.duke.edu/~kst/
M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis: **Modelling with Generalized Stochastic Petri Nets,** John Wiley and Sons. Freely available at: http://www.di.unito.it/~greatspn/bookdownloadform.html
N. Provos, T. Holz: **Virtual Honeypots — From Botnets Tracking to Intrusion Detection,** Addison Wesley, 2007

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/dependability_and_security.pdf

Courseware examples and locations where taught:
Parts of the courseware proposed here are taught at the following schools:
- National School of Civil Aviation (ENAC), Master in Civil Aviation Engineering, Toulouse, France,
- Higher National School of Electronics, Informatics, Hydraulics and Telecommunications (ENSEEIHT), Master in Electrical Engineering and Automation, Toulouse, France
- University of Toulouse, Master in Automation, Decision and Computer Systems.
- Higher National School of Aeronautics and Space (ENSAE), Master in Aeronautics & Computer Science, Toulouse, France
- Higher National School of Electronics, Informatics, and Radiocommunications at Bordeaux (ENSEIRB), Master in Information and Communications Technologies, Bordeaux, France

4.11 Testing, Verification and Validation (6 ECTS)

The purpose of this course is to understand the role of testing, verification and validation in the design and analysis of systems, and to provide an advanced background on related methods and tools.

The course introduces the basic concepts of formal modelling and specification techniques that can be used for verification and validation: model checking, theorem proving, static analysis and abstract interpretation.

The course also presents the fundamentals of software testing. It provides an understanding of testing problems, and covers the major test design techniques. Emphasis is put on the need for rigorous, semi-automated approaches.

Contents

- Model checking
  - Temporal logics as a foundation for model checking
  - Modelling of systems for model checking
  - Standard techniques for model checking, including BDD-based model checking
- Theorem proving
  - Logical foundations
  - Specification and verification with a theorem prover tool:
- Static program analysis
  - Static program analysis definition and main application areas
  - Data-flow analysis
  - Basic elements of abstract interpretation theory
- Software testing
  - Fundamentals of testing: role of testing throughout the software life cycle, test selection and oracle problems, test integration strategy, classification of test methods.
  - Usual structural & functional approaches: control and data flow criteria, predicate coverage, domain testing, model-based testing (e.g., from finite state machines, labelled transition systems).
  - Mutation analysis: principle, examples of usage.
  - Probabilistic test approaches: uniform profile, operational profile, profiles based on structural and functional criteria.

Suggested readings:

T. Kropf: **Introduction to Formal Hardware Verification**, Springer, 1999.

B. Berard, et al.: **System and Software Verification – Model-Checking Techniques and Tools**, Springer, 2001.

C. Hankin, F. Nielson, H. R. Nielson: **Principles of Program Analysis**, Springer, 1999.

A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman: **Compilers: Principles, Techniques, and Tools**, Addison-Wesley, 2006.

P. Cousot, R. Cousot: **Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints**, POPL77, pages 238–252, Los Angeles, California, 1977.

P. Cousot, R. Cousot: **Systematic Design of Program Analysis Frameworks**, POPL79, pages 269–282, San Antonio, Texas, 1979.

B. Beizer: **Software Testing Techniques**, Van Nostrand Reinhold, 1990 (2nd edition)

R. D. Craig, S. P. Jaskiel: **Systematic Software Testing**, Artech House, 2002

P. Ammann, J. Offutt: **Introduction to Software Testing**, Cambridge University Press, 2008

A. Robinson, A. Voronkov (eds.): **Handbook of Automated Reasoning**, **Volume I**, North Holland, 2001

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/testing.pdf

Courseware examples and locations where taught:

- "Computer-Aided Verification", Rajeev Alur at University of Pennsylvania, Philadelphia, USA, http://www.cis.upenn.edu/cis673/, Slides and draft textbook available
- "Deductive Verification of Reactive Systems", Amir Pnueli at The Weizmann Institute of Science, Rehovot, Israel http://www.wisdom.weizmann.ac.il/~amir/Course02a/header.html, Slides available
- "Test and Verification" Emmanuel Fleury, Kim G. Larsen, Brian Nielsen, Arne Skou at Aalborg University, Denmark http://www.cs.auc.dk/~kgl/TOV04/Plan.html, Slides available
- " Theorem Proving and Model Checking in PVS " Edmund M. Clarke and Daniel Kroening at CMU, Pittsburgh, USA http://www.cs.cmu.edu/~emc/15-820A/ Slides available
- "System Validation" Theo C. Ruys at University of Twente http://fmt.cs.utwente.nl/courses/systemvalidation/ Slides available
- "Validation and Verification" J.P. Bowen at University College London http://www.cs.ucl.ac.uk/staff/J.Bowen/GS03/ Slides available
- "Abstract Interpretation" Patrick Cousot, Jerome Clarke Hunsaker at MIT http://web.mit.edu/afs/athena.mit.edu/course/16/16.399/www Slides available
- " Abstract interpretation and static analysis ", David Schmidt at International Winter School on Semantics and Applications, Uruguay, 2003 http://santos.cis.ksu.edu/schmidt/Escuela03/home.html Slides available
- "Introduction to software testing" Paul Ammann and Jeff Offutt at George Mason University http://www.cs.gmu.edu/~offutt/softwaretest/powerpoint/ Slides available

4.12 Usability and User Centred Design for Dependable and Usable Socio-technical Systems (6 ECTS)

The purpose of this course is to introduce the notion of usability of systems and to present user centred development processes that are targeting at usability.

Contents

- Introduction to usability (definition, principles & concepts)
- Ergonomic rules and design guidelines
- Work analysis and task analysis
- Usability evaluation
- User Centred Development processes (implication of users, prototyping approaches)
- Human factors engineering (function allocation)

Suggested readings:

ISO 9241-11:1998 **Ergonomic requirements for office work with visual display terminals** (VDTs) -- Part 11: Guidance on usability
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=16883
J. Nielsen: **Usability Engineering**, Morgan Kaufmann, San Francisco, 1994.
D. Norman: **The design of everyday things**. Basic books, 3rd edition, 2002.

ReSIST Courseware is at: http://resist.isti.cnr.it/files/corsi/courseware_slides/usability.pdf

Courseware examples and locations where taught:

- http://sigchi.org/cdg/index.html (gathering a large set of lectures on HCI and Human factors mainly in the US. This set of courses has been gathered and organised by the ACM Special Interest Group on HCI)
- http://vip.cs.utsa.edu/classes/cs6693s2006/lectures/index.html

2nd Year – 3rd Semester

Common Courses

4.13 Management of Projects (3 ECTS)

The purpose of this course is understanding how to manage a project from initial specification to delivery on the field of systems with resilience requirements.

Contents

- Purpose of project management
- Project management in system engineering
- Project management as a process
- Principal phases of the project management process:
  - Definition of the scope of the project
  - Feasibility evaluation and resources estimation
  - Identification of project interfaces
  - Responsibility allocation
  - Project planning
  - Project monitoring and risk management
  - Project deviation management and impact analysis
  - Project data storage
- Techniques, methods and tools for project management with resilience requirements.
- Requirements engineering activities and their relationships with project management phases.

Suggested readings:

Howard Eisner: **Essentials of Project and System Engineering Management**, Second Edition. John Wiley and Sons, 2002.

James Taylor: **Managing Information Technology Projects,** AMACOM Div American Mgmt Assn 2003.

IEEE Standard 1490-2003: IEEE Guide Adoption of PMI Standard **A Guide to the Project Management Body of Knowledge.**

Mary Beth Chrissis, Mike Konrad, Sandy Shrum: **CMMI Guidelines for Process Integration and Product Improvement**, SEI Series in Software Engineering, 2004.

Ian Sommerville, Pete Sawyer **Requirements Engineering: A Good Practice Guide**. John Wiley and Sons Ed. 1997.

Aybueke Aurum, Claes Wohlin **Engineering and Managing Software Engineering.** Springer Ed. 2005.

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/project_management.pdf

Courseware examples and locations where taught:

- University of Sydney – Course "08PPM0334 : Project management - the complete guide". http://www.cce.usyd.edu.au/cce/subjectcategory.do?id=000223&subject=000236
- Open University – Course "M865: Project Management" http://www3.open.ac.uk/courses/pdfs/M865.pdf

4.14 Middleware Infrastructures for Application Integration (3 ECTS)

The purpose of this course is to offer a broad overview of synchronous and asynchronous middleware technologies that can be used to integrate complex software systems with special emphasis on how to guarantee quality of services in basic middleware operations such as event dissemination and service invocation. The course is organized in two parts: synchronous middleware technologies, including Web Services, J2EE and EJB, and asynchronous middleware technologies including publish subscribe and data distribution service.

Contents

- Basic middleware concepts
- Service Oriented Architecture
    - Remote Procedure Call
    - Remote Method Invocation
    - Remote Service Invocation
    - Interoperability
- Web Services
    - Architecture and basic technologies: XML, SOAP
    - Web Service Definition Language
    - Orchestration and coreography
    - QoS in Web Services: WS-Reliability and WS-Agreement
    - WS implementation in J2EE
- J2EE and EJB platforms
    - Applets, modules and components
    - Developing J2EE application
    - Developing EJB application
    - QoS in J2EE and EJB
- Publish-Subscribe
    - Basic notions
    - Event routing
    - General architecture
- Data distribution service
    - Basic architecture
    - Quality of Service

Suggested readings:

A. Tanenbaum, M. Van Steen: **Distributed Systems** (2nd Edition), Pearson Education, 2007

G. Alonso F. Casati H. Kuno V. Machiraju: **Web services: concepts, architectures and applications**, Springer Verlag

W. Emmerich: **Engineering distributed objects**, John Wiley, 2000

ReSIST Courseware is at: http://resist.isti.cnr.it/files/corsi/courseware_slides/middleware.pdf

Courseware examples and locations where taught:

- Università di Roma "La Sapienza", Distributed System Platforms. R. Beraldi http://www.dis.uniroma1.it/~beraldi
- Università di Bologna, Middleware. F. Panzieri http://courses.web.cs.unibo.it/SistemiMiddleware/MaterialeDiRiferimento

4.15 Software Reliability Engineering (3 ECTS)
The purpose of this course is to give a global overview of approaches to software reliability analysis, evaluation and improvement.

Contents
- Motivations
- Methods for software reliability engineering
  - Data collection, validation and analysis
  - Descriptive statistics
  - Trend analysis (statistical trend tests)
- Software dependability evaluation
  - Reliability growth models
  - Models in stable reliability
  - Dependability benchmarking, for Off-the-Shelf software components
- Software reliability improvement, maturity of the software development process
- Case studies

Suggested readings:
M. Lyu (Ed.): **Handbook of Software Reliability Engineering,** McGraw Hill, 1996 available on line: http://www.cse.cuhk.edu.hk/~lyu/book/reliability/
John Musa: **Software Reliability Engineering: More Reliable Software Faster and Cheaper**, 2nd Edition, September 2004.
K Kanoun, M. R. Bastos Martini, J. Moreira de Souza: **A method for software reliability analysis and prediction, application to the TROPICO-R switching system**, IEEE Transactions on Software Engineering, N° 4, pp. 334-344, April 1991.
J. C. Laprie: **For a product-in-a-process approach to software reliability evaluation**, Third IEEE International Symposium on Software Reliability Engineering (ISSRE'92), Research-Triangle Park (USA), October 7-10 1992, pp.134-138.
John Musa: **Operational Profiles in Software-Reliability Engineering**, IEEE Software 10 (2), pp. 4-32, 1993.
K. Kanoun, M. Kaâniche, J. C. Laprie and S. Metge: **SoRel: a tool for reliability growth analysis and prediction from statistical failure data**, 23rd IEEE International Symposium on Fault-Tolerant Computing (FTCS'23), Toulouse, France, June 22-24, 1993, pp.654-659.
M. Kaâniche, K. Kanoun: **Software failure data analysis of two successive generations of a switching system**, 12th Int. Conference on Computer Safety, Reliability and Security (SAFECOMP'93), Poznan, Poland, 27-29 October 1993, pp.230-239.
K. Kanoun, J. C. Laprie: **Software Reliability Trend Analyses: From Theoretical to Practical Consideration**, IEEE Trans. on Software Engineering, Vol.20, N°9, pp.740-747, September 1994.
K. Kanoun, J.-C. Laprie: **Trend Analysis**, in Handbook of Software Reliability Engineering, Ed. M. Lyu, Mc Graw Hill, Chapter 10, pp. 401-437, 1996. Freely available at: http://www.cse.cuhk.edu.hk/~lyu/book/reliability/
K. Kanoun: **A measurement-based framework for software reliability improvement**, Annals of Software Reliability, Vol.11, N°1, pp.89-106, November 2001.
K. Kanoun, Y. Crouzet, A. Kalakech, A. E. Rugina: **Windows and Linux Robustness Benchmarks With Respect to Application Erroneous Behaviour**, in Dependability Benchmarking for Computer Systems, Chapter 12, pp. 277-254. Editors: Karama Kanoun and Lisa Spainhower, IEEE Computer Society and Wiley, August 2008.

ReSIST Courseware is at:
http://resist.isti.cnr.it/files/corsi/courseware_slides/software_reliability_eng.pdf

Courseware examples and locations where taught:
- OpenSeminar, Software Reliability Engineering, John Musa, Laurie Williams http://openseminar.org/se/courses/41/modules/206/index/screen.do

Resilience in Communication Networks Application Track

4.16.1 IP Networks and Services Resilience (3 ECTS)

Todays' society unavoidably depends on Internet Protocol, the networking protocol suite used in most Internet sites. The large number of security issues and vulnerabilities threaten the confidence users need to have in the networks and services based on this technology, to which they entrust a growing portion of their daily activities' functioning, both in their private and in their professional life. An exhaustive presentation of these services' resilience (threats and vulnerabilities' identification, countermeasures analysis, …) forms the core programme of this course. An exploding IP service, VoIP, will be the focus of the second part of this lecture because of its widespread use, and the declared intention of Telcos to replace circuit-switched voice, known to be resilient, with packet-switched voice, a less secure but economical solution, both within the enterprise and at home.

Contents:

- E-mail and Web services vulnerabilities and countermeasures
- FTP threats: banner grabbing and enumeration, brute force password guessing, bounce attacks, …
- IP network scanning and VPN security issues
- VoIP protocols and architecture
- Threats to VoIP communication systems
- Validation of existing security infrastructure
- Securing techniques: confirmation of user identity, active security monitoring, logically segregate network traffic, IETF encryption solutions

Suggested readings:

D.C. McNab: **Network Security Assessment: Know Your Network,** O'Reilly, 2004
T. Porter and J. Kanclirz Jr.: **Practical VoIP Security,** Syngress, 2006

Courseware examples and locations where taught:

- ETHZürich: http://www.infsecmaster.ethz.ch/courses/course_contents#system "*Network security*"
- University of Cambridge:
  http://ciutesting.com/ciu/msc-telecom.htm "*Network Security*" (in Semester 2)
  http://www.cambridgeuniv.org.uk/msc_in_telecom.html "*Security and optimisation*"
- University of Maryland: http://www.telecom.umd.edu/current/coursedescriptions
  - ENTS 650: *Network Security*
  - ENTS 689I: *Network Immunity*
- Georges Mason University: http://telecom.gmu.edu/tcom_catalog.html#TCOM 501
  - TCOM 545: *Reliability and Maintainability of Networks*
  - TCOM 548: *Security and Privacy Issues in Telecommunications*
  - TCOM 556: *Cryptography and Network Security*
  - TCOM 562: *Network Security Fundamentals*
  - TCOM 662: *Advanced Secure Networking*
  - TCOM 663: *Operations of Intrusion Detection and Forensics*
  - ECE 543: *Cryptography and Computer Network Security*
  - INFS 762: *Information Security Protocols* (formerly known as ISA 662 *Internet Security Protocols*)
  - INFS 766: *Internet Security Protocols*
  - INFS 767: *Secure Electronic Commerce*
- Queen Mary University of London:
  http://www.elec.qmul.ac.uk/study/courses/elem014.html
  "Security & Authentication" (ELEM014)

- University of Sunderland:
  http://www.sunderland.ac.uk/study/course/867/msc_telecommunications_engineering.
  php "*Advanced Network Security*"
- Swinburne University of Technology:
  http://courses.swinburne.edu.au/Subjects/ViewSubject.aspx?mi=300&id=5620
  "Network Security and Resilience" (HET 317)

4.17.1 Resilience of Mobile Applications (3 ECTS)

Security and privacy protection are strong requirements for the widespread deployment of wireless technologies for commercial applications. It is particularly true for mobile computing devices (PDAs, smartphones, …) with focus on multimedia applications. Also, due to the nature of wireless media, dynamic network topology, resource constraints, and lack of any base station or access point, security in ad-hoc networks is more challenging than with cabled networks, justifying the study of secure protocols used for this purpose. By combining computing and communications with the surrounding physical environment through information collection using various sensors, pervasive computing eases their transparent use in day-to-day activities. The inherent disadvantages of slow, expensive connections, frequent line disconnections, limited host bandwidth, and location dependent data make pervasive computing more vulnerable to various security-related threads: requirements and deployment techniques for this type of computing form the last topic covered by this course.

Contents:

- Securing access to wireless networks
- IEEE 802.11 and Bluetooth networks vulnerabilities
- Security requirements for mobile multimedia network applications
- Network protocols (SIP, SRTP) for secure multimedia streaming services
- Security protocols for ad-hoc networks
- Pervasive computing applications: security, privacy and trust

Suggested readings:

L. Buttyan, and J.-P. Hubaux: **Security and cooperation in wireless networks**, Cambridge University Press, 2007 – the corresponding slides can be found at http://secowinet.epfl.ch/index.php?page=slideshow.html

K.N. De Randall, and C.L. Panos (Eds.): **Wireless Security: Models, Threats, and Solutions,** McGraw-Hill Professional, 2002

G. Karmakar, and L.S. Dooley (Eds.): **Mobile Multimedia Communications: Concepts, Applications and Challenges,** Idea Group Inc, 2007

Courseware examples and locations where taught:

- ETHZürich: http://www.syssec.ethz.ch/education/sown "*Security of wireless networks*"
- University of Cambridge:
  http://ciutesting.com/ciu/msc-telecom.htm "*Network Security*" (in Semester 2)
  http://www.cambridgeuniv.org.uk/msc_in_telecom.html "*Security and optimisation*"
- University of Maryland: http://www.telecom.umd.edu/current/coursedescriptions
  - ENTS 650: *Network Security*
  - ENTS 689I: *Network Immunity*
- Georges Mason University: http://telecom.gmu.edu/tcom_catalog.html#TCOM 501
  - TCOM 545: *Reliability and Maintainability of Networks*
  - TCOM 548: *Security and Privacy Issues in Telecommunications*
  - TCOM 556: *Cryptography and Network Security*
  - TCOM 562: *Network Security Fundamentals*
  - TCOM 662: *Advanced Secure Networking*
  - TCOM 663: *Operations of Intrusion Detection and Forensics*

- ECE 543: *Cryptography and Computer Network Security*
- INFS 762: *Information Security Protocols* (formerly known as ISA 662 *Internet Security Protocols*)
- INFS 766: *Internet Security Protocols*
- INFS 767: *Secure Electronic Commerce*

• Queen Mary University of London:
http://www.elec.qmul.ac.uk/study/courses/elem014.html
"Security & Authentication" (ELEM014)

• University of Sunderland:
http://www.sunderland.ac.uk/study/course/867/msc_telecommunications_engineering.php "*Advanced Network Security*"

• Swinburne University of Technology:
http://courses.swinburne.edu.au/Subjects/ViewSubject.aspx?mi=300&id=5620
"Network Security and Resilience" (HET 317)

4.18.1 Project in cooperation with Industry (9 ECTS)

The purpose of the project is to provide experience to the student on researching on a real world topic. It is in cooperation with a leading industry or administration and its content will depend on them.

4.19.1 Additional Course(s) (6 ECTS)

These additional courses and or seminars are tightly related to the project performed by the student and are not detailed in this document.

Safety Critical Application Track

4.16.2 Development Process and Standards for Safety Critical Applications (3 ECTS)

The goal of this course is to provide an overview on the development process to attempt when designing and developing safety critical applications and to make aware of the standards which are important from a general perspective as well as in the specific application areas.

Contents

- System life cycle
- Development process
- Documentation
- Tools
- V-model
- Spiral model
- Certification and licensing
- Legal frame
- Generic standards
- Application specific standards (e.g. health, nuclear, automotive)
- Safety critical systems development methodologies, tools, languages
- Safety and security
- Security threats
- Formal methods
- Risk management

Suggested readings:

F. Redmill (ed.): **Dependability of Critical Computer Systems - 1 and 2**, ISBN 1-85166-203-0 and ISBN 1-85166-381-9.
P. Bishop (ed.): **Dependability of Critical Computer Systems – 3, Techniques Directory**, ISBN 1-85166-544-7.
BSI **IT Security Guidelines**, Bundesamt für Sicherheit in der Informationstechnik 2007. http://www.bsi.bund.de/gshb
Generic standards:
IEC 61508 “Functional safety of electrical/electronic/programmable electronic safety-related systems”, Parts 0 – 7 (especially Part 3: “Software requirements”)
ISO/IEC 12207:1995 “Information technology – Software life cycle processes”
IEC 61713:200006 “Software dependability through the software life-cycle processes – Application guide”
ISO/IEC 27001:2005 **“**Information technology -- Security techniques -- Information security management systems – Requirements”
ISO/IEC 27002:2005 "Information Technology – Code of Practice for Information Security Management"
ISO/IEC 27005:2008 “Information technology -- Security techniques -- Information security risk management
ISO/IEC 15408-1:2005 “Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general mode”
ISO/IEC 15408-2:2008 “Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional components”
ISO/IEC 15408-3:2008 “Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance components”
ISO/IEC 18045:2008 “Information technology -- Security techniques -- - Methodology for IT security evaluation”
Sector specific standards:
IEC 60880 Ed. 2.0: “Nuclear Power Plants - Instrumentation and control systems important to

safety - Software aspects for computer-based systems performing category A functions”
ISO 14971:2007 “Medical devices – Application of risk management to medical devices”
IEC 60601-1-4:1996 “Medical electrical equipment – Part 1-4: General requirements for safety; Collateral standard: Programmable electrical medical systems
IEC 62304:2006 “Medical device software -- Software life cycle processes”
RTCA DO-178B “Software Considerations in Airborne Systems and Equipment Certification”
UK MoD 00-55:1997 “Requirements for safety related software in defence equipment”
EN 50128:2001 “Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems”
MISRA-C++: "Guidelines for the Use of the C++ Language in Critical Systems", ISBN 978-906400-03-3 (paperback), ISBN 978-906400-04-0 (PDF), June 2008.
MISRA-C2: "Guidelines for the Use of the C Language in Critical Systems", ISBN 0 9524156 2 3 (paperback), ISBN 0 9524156 4 X (PDF), October 2004.
MISRA: "Development Guidelines for Vehicle Based Software", ISBN 0 9524156 0 7, November 1994.

4.17.2 Architectural Issues and Examples of Systems (3 ECTS)
The purpose of this course is to provide an overview on different architectures and their comparison in relation to dependability attributes. Examples of real systems and their implementation show the importance of architecture decisions.

Contents:
- Redundancy and diversity
- Fault tolerant systems
- Fault tolerance strategies
- Failure detection
- Hierarchical systems
- Distributed systems
- Networks
- Embedded systems
- Time triggered architecture
- Communication protocols
- Synchronization
- Dependability of protocols and architectures
- Safety vs security
- Security threats
- Real systems

Suggested readings:
J. L. Hennessy and D. A. Patterson: **Computer Architecture: A Quantitative Approach**, 2nd Edition, Morgan Kaufmann Publishing Co., Menlo Park, CA.
D. A. Patterson and J. L. Hennessy: **Computer Organization and Design. The Hardware - Software Interface,**. Morgan Kaufmann Publishers, San Francisco, CA

4.18.2 Project in cooperation with Industry (9 ECTS)
The purpose of the project is to provide experience to the student on researching on a real world topic. It is in cooperation with a leading industry or administration and its content will depend on them.

4.19.2 Additional Course(s) (6 ECTS)
These additional courses and or seminars are tightly related to the project performed by the student and are not detailed in this document.

Resilience in e-Business Application Track

4.16.3 Enterprise Security (3 ECTS)

This course addresses the security of e-business and cyber environments from an end-to-end perspective. The information security methodologies of inspection, protection, detection, reaction, and reflection are addressed in detail. Principle of survivability and information assurance will be presented in a technologically independent way. Layered network defense structures will be then illustrated. Methods of risk analysis/assessment and "best practices" associated with evaluating, implementing, and administering hardware and software-based firewalls and Intrusion Detection Systems (IDSes). Finally the course will address the problem of governance of Enterprise security and compliance management related to constantly evolving regulations.

Contents

- Information security methodologies
- Principles of survivability and information assurance
- Layered network defense and security metrics
- Security inside telco operators
- Designing, evaluating and implementing firewalls
- Intrusion Detection Mechanisms
- Risk analysis/assesment in security for e-business
- Best practices in security for E-business
- Governance and compliance of Enterprise security

Suggested readings:

R. C. Newman: **Enterprise Security, Prentice Hall**, 2002
P. J. Ortmeier: **Security Management**, **Prentice Hall**, 2004

Courseware examples and locations where taught:

- University of Melbourne, Strategic Security Management. Atif Ahmad. http://disweb.dis.unimelb.edu.au/staff//atif/home.htm
- IBM research, Zurich: http://www.zurich.ibm.com/csc/security/compliance.html

4.17.3 Computer and Network Forensics (3 ECTS)

Computer and network forensics studies cyber-attack prevention, planning, detection, and response with the goals of counteracting cybercrime, cyberterrorism, and cyberpredators, and making them accountable. The topics covered in this course include fundamentals of computer and network forensics, forensic duplication and analysis, network surveillance, intrusion detection and response, incident response, anonymity and pseudonymity, cyber law, computer security policies and guidelines, court report writing and presentation, and case studies.

Contents:

- Digital forensics: an overview
- Forensics basics and criminalistics
- Basics of computer networks and operating systems
- Advanced topics in computer and network forensics
- Cases studies (e.g., intrusion and online frauds detection, steganography & steganalysis, anonymity/pseudonymity/P3P, cyber law, security and privacy policies and guidelines)

Suggested readings:

Brian Carrier: **File System Forensic Analysis**, Addison-Wesley, 2005
Chris Prosise and Kevin Mandia: **Incident Response: Investigating Computer Crime,** Berkeley, California: Osborne/McGraw-Hill, 2001
Warren Kruse and Jay Heiser: **Computer Forensics: Incident Response Essentials,** Addition-

Wesley, 2002
Phillips, Nelson, Enfinger, and Stuart: **Guide to Computer Forensics and Investigations**, Course Technology, 2006

Courseware examples and locations where taught:

- Iowa State University, Network and computer forensics. Yong Guan. http://home.eng.iastate.edu/~guan/course/CprE-536/index.html
- University of Idaho Network and computer forensics.Jim Alves-Foss. http://www.cs.uidaho.edu/CS447.html

4.18.3 Project in cooperation with Industry (9 ECTS)

The purpose of the project is to provide experience to the student on researching on a real world topic. It is in cooperation with a leading industry or administration and its content will depend on them.

4.19.3 Additional Course(s) and/or Seminars (6 ECTS)

These additional courses and or seminars are tightly related to the project performed by the student and are not detailed in this document.

2nd Year – 4th Semester

The courses and/or seminars are specific to the topic of the Thesis and are not detailed in this document.

4.20 Specific Courses and/or Seminars (3 ECTS)

4.21 MSc Thesis Preparation and Presentation (27 ECTS)

5. Conclusions

The effort made with the identification of this MSc Curriculum in Resilient Computing will continue after the end of ReSIST with dissemination to European Universities that may be interested in starting Master tracks on this topic. To this aim a Steering Committee has been nominated. It is composed by: Tom Anderson – Newcastle University, UK, Algirdas Avizienis – Vytautas Magnus University, Lithuania, Hugh Glaser – University of Southampton, UK, Jean-Claude Laprie – LAAS-CNRS, Toulouse, France, Brian Randell – Newcastle University, UK and Luca Simoncini – University of Pisa, Italy.

References in the text

[1] http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf
[2] http://ec.europa.eu/education/policies/educ/bologna/bergen.pdf
[3] http://ec.europa.eu/education/policies/educ/bologna/report06.pdf
[4] http://www.resist-noe.org/
[5] http://www.cra.org/reports/trustworthy.computing.pdf
[6] http://ec.europa.eu/education/programmes/socrates/ects/index_en.html

References to all suggested readings

- A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman: **Compilers: Principles, Techniques, and Tools**, Addison-Wesley, 2006.
- M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis: **Modelling with Generalized Stochastic Petri Nets,** John Wiley and Sons.
- G. Alonso F. Casati H. Kuno V. Machiraju: **Web services: concepts, architectures and applications**, Springer Verlag
- P. Ammann, J. Offutt: **Introduction to Software Testing**, Cambridge University Press, 2008.
- A. Avizienis, J-C. Laprie, B. Randell and C. Landwehr: **Basic Concepts and Taxonomy of Dependable and Secure Computing,** IEEE Trans. On Dependable and Secure Computing, Vol.1, n.1, Jan.-March 2004
- B. Beizer: **Software Testing Techniques**, Van Nostrand Reinhold, 1990 (2nd edition)
- B. Berard, et al.: **System and Software Verification – Model-Checking Techniques and Tools**, Springer, 2001
- P. Bishop (ed.): **Dependability of Critical Computer Systems – 3, Techniques Directory**, ISBN 1-85166-544-7
- BSI "IT Security Guidelines", Bundesamt für Sicherheit in der Informationstechnik 2007. http://www.bsi.bund.de/gshb
- A. Burns and A. J. Wellings: **Real-Time systems and programming languages**, 3rd ed., Addison Wesley, 2001, ISBN 0-201-40365-X
- G. Buttazzo: **Hard Real-Time Computing Systems**, Second Edition, Series: Real-Time Systems Series, Vol. 23, 2005, XIII, 425 p., ISBN: 978-0-387-23137-2 Springer, 2005
- L. Buttyan, and J.-P. Hubaux: **Security and cooperation in wireless networks,** Cambridge University Press, 2007
- B. Carrier, **File System Forensic Analysis**, Addison-Wesley, 2005
- W. R. Cheswick, S. M. Bellovin, and A. D. Rubi: **Firewalls and Internet Security: Repelling the Wily Hacker**, Second Edition, Addison Wesley
- M. B. Chrissis, M. Konrad, S. Shrum: **CMMI Guidelines for Process Integration and Product Improvement**, SEI Series in Software Engineering, 2004
- F. Cottet, J. Delacroix, C. Kaiser, Z. Mammeri: **Scheduling in Real-Time Systems**, Wiley Eds, 2002, 266p. ISBN: 0-470-84766-2
- P. Cousot, R. Cousot: **Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints**, POPL77, pages 238–252, Los Angeles, California, 1977.
- P. Cousot, R. Cousot: **Systematic Design of Program Analysis Frameworks**, POPL79, pages 269–282, San Antonio, Texas, 1979.
- R. D. Craig, S. P. Jaskiel: **Systematic Software Testing**, Artech House, 2002
- V. Detlovs, K. Podnieks: **Introduction to Mathematical Logic** http://www.ltn.lv/~podnieks/mlog/ml.htm
- K.N. De Randall, C.L. Panos (Eds.): **Wireless Security: Models, Threats, and Solutions**, McGraw-Hill Professional, 2002
- Howard Eisner: **Essentials of Project and System Engineering Management**, Second Edition. John Wiley and Sons, 2002
- W. Emmerich: **Engineering distributed objects**, John Wiley, 2000
- S. Even: **Graph Algorithms**, Computer Science Press, 1979
- J.L. Gross and J. Yellen (Eds.): **Handbook of graph theory**, CRC Press, 2003
- R. Guerraoui and L. Rodrigues: **Introduction to Reliable Distributed Programming,** Springer, 2006.
- C. Hankin, F. Nielson, H. R. Nielson: **Principles of Program Analysis**, Springer, 1999
- J. L. Hennessy and D. A. Patterson: **Computer Architecture: A Quantitative Approach**, 2nd Edition, Morgan Kaufmann Publishing Co., Menlo Park, CA
- M. Huth and M. Ryan: **Logic in Computer Science**, Cambridge University Press

- IEEE Standard 1490-2003: IEEE Guide Adoption of PMI Standard **A Guide to the Project Management Body of Knowledge**
- ISO 9241-11:1998 **Ergonomic requirements for office work with visual display terminals** (VDTs) -- Part 11: Guidance on usability http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=16883
- W. Johnson: **Failure in Safety-Critical Systems. A Handbook of Accident and Incident Reporting**. Available on-line at: http://www.dcs.gla.ac.uk/johnson/book October 2003. 2003. Glasgow, Scotland, University of Glasgow Press
- M. Kaâniche, K. Kanoun: **Software failure data analysis of two successive generations of a switching system**, 12th Int. Conference on Computer Safety, Reliability and Security (SAFECOMP'93), Poznan, Poland, 27-29 October 1993, pp.230-239
- K Kanoun, M. R. Bastos Martini, J. Moreira de Souza: **A method for software reliability analysis and prediction, application to the TROPICO-R switching system**, IEEE Transactions on Software Engineering, N° 4, pp. 334-344, April 1991
- K. Kanoun, M. Kaâniche, J. C. Laprie and S. Metge: **SoRel: a tool for reliability growth analysis and prediction from statistical failure data**, 23rd IEEE International Symposium on Fault-Tolerant Computing (FTCS'23), Toulouse, France, June 22-24, 1993, pp.654-659
- K. Kanoun, J. C. Laprie: **Software Reliability Trend Analyses: From Theoretical to Practical Considerations**, IEEE Transactions on Software Engineering, Vol.20, N°9, pp.740-747, September 1994
- K. Kanoun, J.-C. Laprie: **Trend Analysis**, in Handbook of Software Reliability Engineering, Ed. M. Lyu, Mc Graw Hill, Chapter 10, pp. 401-437, 1996. Freely available at: http://www.cse.cuhk.edu.hk/~lyu/book/reliability/
- K. Kanoun: **A measurement-based framework for software reliability improvement**, Annals of Software Reliability, Vol.11, N°1, pp.89-106, November 2001
- K. Kanoun, Y. Crouzet, A. Kalakech, A. E. Rugina: **Windows and Linux Robustness Benchmarks With Respect to Application Erroneous Behaviour**, in Dependability Benchmarking for Computer Systems, Chapter 12, pp. 277-254. Editors: Karama Kanoun and Lisa Spainhower, IEEE Computer Society and Wiley, August 2008
- G. Karmakar, and L.S. Dooley (Eds.): **Mobile Multimedia Communications: Concepts, Applications and Challenges**, Idea Group Inc, 2007
- C. Kaufman, R. Perlman, and M. Speciner: **Network Security: Private Communication in a Public World**, Second Edition, Prentice Hall
- H. Kopetz: **Real-Time Systems: Design Principles for Distributed Embedded Applications**, Series: The Springer International Series in Engineering and Computer Science, Vol. 395, 1997, 356 p., ISBN: 978-0-7923-9894-3
- J- T. Kropf: **Introduction to Formal Hardware Verification**, Springer, 1999
- W. Kruse and J. Heiser: **Computer Forensics: Incident Response Essentials,** Addition-Wesley, 2002
- C. Laprie et al.: **Guide de la sûreté de fonctionnement**, Cepaduès Editions, 1995 (in French)
- J. C. Laprie: **For a product-in-a-process approach to software reliability evaluation**, Third IEEE International Symposium on Software Reliability Engineering (ISSRE'92), Research-Triangle Park (USA), October 7-10, 1992, pp.134-13
- J.C. Laprie: **From Dependability to Resilience***,* 38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks, Anchorage, Alaska, June 2008, Sup. Vol., pp. G8-G9
- M. Lyu (Ed.): **Handbook of Software Reliability Engineering,** McGraw Hill, 1996 available on line: http://www.cse.cuhk.edu.hk/~lyu/book/reliability/
- D.C. McNab: **Network Security Assessment: Know Your Network**, O'Reilly, 2004
- J. Menezes, P. C. van Oorschot and S. A. Vanstone: **Handbook of Applied Cryptography,** CRC Press, 1996
- J. Musa: **Operational Profiles in Software-Reliability Engineering**, IEEE Software 10 (2), pp. 4-32, 1993

- J. Musa: **Software Reliability Engineering: More Reliable Software Faster and Cheaper**, 2nd Edition, September 2004
- R. C. Newman: **Enterprise Security,** Prentice Hall, 2002
- J. Nielsen: **Usability Engineering**, Morgan Kaufmann, San Francisco, 1994
- D. Norman: **The design of everyday things**. Basic books, 3rd edition, 2002
- D. A. Patterson and J. L. Hennessy: **Computer Organization and Design. The Hardware - Software Interface**, Morgan Kaufmann Publishers, San Francisco, CA
- Phillips, Nelson, Enfinger, and Stuart: **Guide to Computer Forensics and Investigations**, Course Technology, 2006
- P. J. Ortmeier: **Security Management**, Prentice Hall, 2004
- T. Porter and J. Kanclirz Jr.: **Practical VoIP Security**, Syngress, 2006
- C. Prosise and K. Mandia: **Incident Response: Investigating Computer Crime,** Berkeley, California: Osborne/McGraw-Hill, 2001
- N. Provos, T. Holz: **Virtual Honeypots — From Botnets Tracking to Intrusion Detection,** Addison Wesley, 2007
- J. Rasmussen, M. A. Pejtersen, L. P. Goldstein: **Cognitive Systems Engineering**. New York, USA, John Wiley and Sons, 1994
- J. Reason: **Human Error**. 1990. Cambridge University Press
- J. Reason: **Managing the Risks of Organizational Accidents**, 1997, Aldershot, UK, Ashgate
- F. Redmill (ed.): **Dependability of Critical Computer Systems - 1 and 2**, ISBN 1-85166-203-0 and ISBN 1-85166-381-9
- A. Robinson, A. Voronkov (eds.): **Handbook of Automated Reasoning**, **Volume I**, North Holland, 2001
- D. P. Siewiorek and R. Swartz: **Reliable Computer Systems, Design and Evaluation,** Third Edition, A K Peters, Ltd., 1998
- A. Silberschatz, P. Baer Galvin, G. Gagne: **Operating Systems Concepts**, John Wiley & Sons, 2008, ISBN 0-470-12872-0
- N. Smart: **Cryptography, An Introduction**, McGraw-Hill, 2002
- W. Stallings: **Cryptography and Network Security**, 4th Edition, Prentice Hall
- A. Tanenbaum, M. Van Steen: **Distributed Systems** (II Edition), Pearson Education, 2007
- J. Taylor: **Managing Information Technology Projects,** AMACOM Div American Mgmt Assn 2003
- K. S. Trivedi: **Probability and Statistics with Reliability, Queuing, and Computer Science Applications**, Second Edition, John Wiley & Sons, 2002
- D. Wickens and J. G. Hollands: **Engineering Psychology and Human Performance**. 3rd edition, 1999, Prentice Hall
- P. Verissimo and L. Rodrigues: **Distributed Systems for System Architects**, Kluwer, 2001
- P. Verissimo, N. F. Neves and M. Correia. **Intrusion-Tolerant Architectures: Concepts and Design.** In *Architecting Dependable Systems*, R. Lemos, C. Gacek, A. Romanovsky (eds.), LNCS 2677, pp. 3-36, Springer, 2003
- P. Verissimo, M. Correia, N. F. Neves, P. Sousa. **Intrusion-Resilient Middleware Design and Validation**. In *Annals of Emerging Research in Information Assurance, Security and Privacy Services*, H. Raghav Rao and S. Upadhyaya (eds.), Elsevier, to appear, 2008