ReSIST: Resilience for Survivability in IST

A European Network of Excellence

Contract Number: 026764

Deliverable D38: Resilient Computing Courseware

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 D38:
Resilient Computing Courseware

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, Paul Ammann, George Candea, Alan Dix, Michael Huth, Ricardo Jimenez-Peris, Johan Karlsson, Marta Patino-Martinez, Amir Pnueli, Luis Rodrigues, Mark Ryan, Kishor S. Trivedi, Udo Voges.

Includes material from: Gregory Abowd, Rajeev Alur, Niels Andersen, R. Anderson, Michael Backes, D. Basin, Russell Beale, R. Beraldi, Jonathan P. Bowen, Giorgio Buttazzo, L. Buttyan, Edmund Clarke, Thierry Coquand, George Couloris, Patrick Cousot, Jean Dollimore, J. Finlay, B. Fisher, Emmanuel Fleury, Greg Gagne, Peter Baer Galvin, S. Goldwasser, Rachid Guerraoui, J. Hammerstein, T.A. Henzinger, J.P. Hubaux, J.C. Hunsaker, C. Johnson, N. Jones, J-P. Katoen, Tim Kindberg, D. Kroening, P. Ladkin, Julia Lawall, M. Lyu, Alberto Marchetti Spaccamela, J. Mattson, U. Maurer, C.J. May, A. Menezes, J. Musa, M.J. Neely, J. Offutt, P. van Oorschot, F. Panieri, Gary Perlman, Karlis Podnieks, Bart Preneel, I. Puaut, Ted Ralphs, R. L. Rivest, Kay Robbins, Larry Rudolph, K. Rush, Theo C. Ruys, Avi Silberschatz, D. Schmidt, Nigel Smart, S. Smith, S. Vanstone, Marcel Waldvogel, Laurie Williams.

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

Contents:

1. Introduction

This Deliverable describes the courseware in support to teaching Resilient Computing in a Curriculum for an MSc track following the scheme of the Bologna process.
The development of the supporting material for such a curriculum has required a rather intensive activity that involved not only the partners in ReSIST but also a much larger worldwide community with the aim of identifying available updated support material that can be used to build a progressive and methodical line of teaching to accompany students and interested persons in a profitable learning process.
The initial activity was related to interviewing all partners in ReSIST and collecting freely available material that could be of interest to the curriculum on Resilient Computing. The 54 collected forms that have been reported in Deliverable D16 contain links to freely available material and the constraints posed for its distribution and use. The 54 forms contained 20 pointers to courseware material, 6 of which are pointing to courseware on slides (in different languages English, French, Italian, Spanish and Hungarian); the majority points to textbooks.
After this collection, a worldwide survey on freely available supporting material has started and has brought to the identification of a set of freely available supporting material mainly based on slides and related to courses offered in several Universities and/or International Organizations. An annotated list of such supporting material is in the Appendix.
The extension to a wider community has brought to collaboration with EWICS TC7 Subgroup on Education and to the organization of a Joint Workshop on Teaching Resilient Computing, reported in Deliverable D16, and to an Open Training and Dissemination Committee Meeting held in Erlangen on May 3, 2007, explicitly dedicated to courseware.
The courseware presented in this Deliverable is composed by the following sets:
1) The set of original ReSIST Courseware (in terms of comprehensive sets of slides) that covers the course of Fundamentals of Dependability in the first semester, all courses of the second semester and the three common courses in the third semester,
2) Freely available courseware material, gathered on the web, authorised by the individual authors, (in terms of sets of slides, textbooks and additional material) for the remaining basic courses in the first semester, and
3) Freely available courseware material gathered on the web, authorised by the individual authors, (in terms of sets of slides, textbooks and additional material) for complementing the material indicated in points 1) and 2).
All this material is on-line on the official ReSIST web site http://www.resist-noe.org/, can be viewed and downloaded for use in a class and constitutes, at our knowledge, the first, almost comprehensive attempt, to build a database of support material related to Dependable and Resilient Computing.
The large group of authors that have been contacted have agreed to maintain updated this material so that its value will not become obsolete with time.

2. Rationale of the Courseware

The Training and Dissemination Committee and the Executive Board had several discussions on the best way of organizing the supporting material, on the style of presentation and on its profitable use for both types of audience: **students** in a MSc curriculum in Resilient Computing or **interested persons** that, through the ReSIST web site and the RKB, may wish to deepen his/her knowledge on specific fields in resilient computing.

These discussions took large advantage by many contributions coming from lecturers not in ReSIST who have expertise in teaching topics in dependable and resilient computing and from valuable suggestions from the panel of reviewers of ReSIST.
For the first type of users (students) there is a mediator (the lecturer of the single course), and the courses must be organized in a progressive teaching/learning process that requires a strong interaction between students and lecturer(s). In addition, each specific lecturer wishes to maintain his/her autonomy in organizing teaching on the basis of his/her expertise and research interests, still following the lines identified in the curriculum.
For the second type of users (interested persons) there is no such mediation and, in absence of a critical selection of material, it is quite likely that he/she is not able to build an individual track of learning out of a huge mass of available material (papers, textbooks, etc.), with consequent difficulty in an effective and efficient individual learning.
It was therefore finally decided to produce a full set of slides as original ReSIST support material for some of the most relevant courses in the curriculum. The courses identified are:
Course 4.7 Fundamentals of Dependability (1st semester)
Course 4.8 Computer Networks Security (2nd semester)
Course 4.9 Resilient Distributed Systems and Algorithms (2nd semester)
Course 4.10 Dependability and Security Evaluation of Computer-based Systems (2nd semester)
Course 4.11 Testing, Verification and Validation (2nd semester)
Course 4.12 Usability and User Centred Design for Dependable and Usable Socio-technical Systems (2nd semester)
Course 4.13 Management of Projects (3rd semester)
Course 4.14 Middleware Infrastructures for Application Integration (3rd semester)
Course 4.15 Software Reliability Engineering (3rd semester)

These courses take advantage of a comprehensive set of slides (original ReSIST support material), the indication of a set of relevant suggested readings, the identification of textbooks (where important) and a set of links to other complementary material useful to have a wider coverage of the topic.

For the other 6 fundamental courses in the 1st semester, it was decided that the freely available material on the web was more than sufficient to cover the teaching needs of any lecturer, while the production of ReSIST material would have either been a mere duplication of material. As well no material has been produced for the examples of Application tracks in the 3rd semester since these are very peculiar to the industrial environment of the country where the curriculum is offered and a greater flexibility has to be maintained, as well as for the possible additional courses and for the seminars in the 4th semester.

All this material is complemented by the other Deliverables developed in ReSIST, like D12 - Resilience-Building Technologies: State of Knowledge, D13 - From Resilience-Building to Resilience-Scaling Technologies: Directions, the set of slides presented at the Summer School in Porquerolles, D39 – Selected Current Practices document, D33- Resilient-Explicit Computing: final, D35 – Resilience Scaling Technologies: final. The integration between the web-site and the RKB adds value to navigation between different information, both educational and informative about the know-how of persons and sites.

2.1 Style of the courseware

The common style identified for the courseware is based on:

- Providing a short description per each course (textual, web-based and RKB-based). These descriptions identify the progressive line of teaching/learning;
- Providing such descriptions with links or pointers to supporting material (papers, sections of textbooks, existing hand-outs or slides). The sequence of this material constitutes the most efficient and effective line for a proficient learning process, and is the most valuable added value that ReSIST provides.

2.2 Organization of the courseware

With this courseware material produced for the entire curriculum in resilient computing (described also in Deliverable D37), the teaching/learning process can be organized as follows:

For students:

- Each lecturer will have a comprehensive set of topics to be taught in each course, and the connections among courses in each semester and between them, with the indication of what is the core material, where to retrieve it or a set of original material that can be directly used in a class;
- Each lecturer will have the autonomy of integrating this set of material with additional supporting material on the basis of his/her experience and/or research activity.

The courseware, in this case, can be organized on a "horizontal" line that is following the sequencing of courses in each semester and progressively span over all the several topics that are included in each semester in the curriculum, following the sequence of semesters.

For interested persons on a specific topic:

The courseware can also be organized "vertically" on the basis of the specific interests of the user. A possible "vertical" organization can be based on the three main areas of Fault Tolerance, Fault Removal and Fault Forecasting, or, if preferred, on more classical lines as Architectural Design, Verification and Validation, Assessment, etc., or on attributes like Security, Safety, Resilience, etc. In this way each person will be able to deepen his/her knowledge with an individual learning line that identifies the core material to walk through. At the end of the learning path, he/she will also be able to untangle the huge mass of available material that is already collected in the web site and use the references in the RKB to provide useful searches on expert sites or work by individuals in the field.

An additional possibility will be the organization of the courseware material on the basis of short general courses for those interested persons who cannot follow a full curriculum, but are interested in the generalities of resilient computing.

All the proposed courseware in this Deliverable has been produced by a joint effort of academic and industrial partners in ReSIST.

3. Description of Courseware per course with the line of teaching and links to supporting material

As detailed in Deliverable D37, the curriculum in Resilient Computing is structured into four semesters over two years. The first semester covers Basics and Fundamentals, offering courses from 4.1 to 4.7 while the second semester covers

Methods, Tools and Techniques, offering courses from 4.8 to 4.12. In the second year, the third semester includes three common courses from 4.13 to 4.15 plus two courses peculiar to each Application tracks. The line of teaching/learning for the courses from 4.1 to 4.15 are described in this Section, as well as those for the two courses for each of the three examples of Application tracks on Resilience in Communication Networks, on Safety Critical Systems and on Resilience in e-Business.

Course *4.1 Advanced Probability and Statistics*

This course provides a comprehensive introduction to probability, stochastic processes, and statistics.

The content of this course provides an introduction to probability theory, and can be used as the core material for a one-semester introductory course on applied probability theory. A major strength is the use of examples and problems as motivation for the probability concepts. Each part also includes sections on the application of the probability concepts to performance and reliability analysis.

"Introduction" provides the motivation and covers the concepts of sample spaces, events and event algebra, probability axioms, combinatorial problems, Bayes rule and Bernoulli trials. On the application side, reliability analysis using block diagrams and fault trees is introduced. Methods of inclusion exclusion, sum of disjoint products and factoring are introduced. Reliability analysis of multistate systems is also presented.

"Discrete Random Variables" covers the functions and distributions for discrete random variables.

"Continuous Random Variables" covers the functions and distributions for continuous random variables. In addition to the exponential distribution, other continuous distributions, such as the Pareto, log-logistic and defective distributions that are used in reliability theory, are presented. A section on functions of normal random variables is given here which is used later on statistical inference. On the application side, there is a section on reliability and failure rate.

"Expectation" covers the topics of moments, transform methods, and inequalities and limit theorems. On the application side, there is a section on computation of Mean Time to Failure for systems with different types of redundancy schemes.

" Conditional Distribution and Expectation" covers the topics of mixture distributions, conditional expectation and random sums. On the application side, there is a section introducing the concept of imperfect fault coverage and its impact on system reliability.

"Stochastic Processes" introduces stochastic processes and then presents the following topics: classification of stochastic processes, Bernoulli process, Poisson process, and renewal processes. On the application side, there are sections devoted to availability analysis and a renewal model of program behaviour.

Textbooks:

The course is based mainly on the 6 first chapters of:

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

Other readings:

S. Ross: **Probability Models for Computer Science**, Academic Press, 2002, San Diego, CA

Links:

The course book's webpage at http://www.ee.duke.edu/~kst/ include all material necessary to teach the course.

Course *4.2 Cryptology and Information Security*
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.

The course presents a broad introduction to information security and cryptology. It focuses on fundamental methods and the principles behind these areas, ranging from mathematical cryptology to formal modelling of security protocols.

The basic concepts of modern cryptology are based on computational difficulty; they are modelled using the language of complexity theory. The course introduces one-way functions, pseudorandom functions and collision-free hash functions and shows some fundamental relations between them. Symmetric-key encryption methods like block-ciphers and stream-ciphers can be derived from these building blocks. The course also mentions practical implementations, like the AES block cipher and the SHA family of hash functions, but does not address their internal.

The most important area of modern cryptology is public-key cryptosystems. The course presents them in an abstract formulation using trap-door one-way permutations and in concrete form using the RSA and ElGamal cryptosystems and the Diffie-Hellman key agreement protocol. In order to lay the foundation for presenting these cryptosystems, the course also introduces some basic notions of algebra and number theory, including the Euler function.

Public-key encryption and hybrid encryption schemes, together with digital signature schemes form the most important part of the course. Methods for secure key distribution in a network, certificates, and public-key infrastructures are also part of the course. The students will gain a sound understanding of the cryptographic mechanisms used in today's computing infrastructure.

The course also presents the intriguing concept of zero-knowledge proofs and their application to secure identification protocols and their role in supporting anonymous yet secure interactions in a network.

Apart from these cryptographic concepts, the course also introduces the alternative approach to model security protocols using logic and formal-language methods (making the so-called Dolev-Yao abstraction). This part enables the students to reason about security protocols using higher-level abstractions than cryptographic proof methods.

Textbooks:
The course can be based on material from the following sources:
N. Smart: **Cryptography, An Introduction** (Second Edition), McGraw-Hill, 2007.
http://www.cs.bris.ac.uk/~nigel/Crypto_Book/
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/

Links:
Course by Michael Backes at Saarland University, http://www.infsec.cs.uni-sb.de/teaching/SS08/Cryptography/
Course by David Basin at ETH Zürich,
http://www.infsec.ethz.ch/education/ss08/infsec08

Course by Elisabeth Oswald and Nigel Smart at University of Bristol,
http://www.cs.bris.ac.uk/Teaching/Resources/COMS30124/
Course by Ronald L. Rivest and Shafi Goldwasser at MIT,
http://courses.csail.mit.edu/6.857/2008/lecture.html

Course *4.3 Logic in Computer Science*
The goal of the course is to present the fundamental notions of logic that are important in computer science.
It presents propositional and predicate logic in a natural deduction style (which is used in many interactive theorem provers) with a suggestive box notation for proofs. Completeness theorems for propositional and predicate logic are discussed, but the proofs are sketched only, rather than given in detail. What is important is that the students understand well the meaning and consequences of these completeness theorems. On the other hand, undecidability of predicate logic is presented (using Post systems).
The course also presents the basis of temporal logic (LTL and CTL) and model checking. Decidability results are mentioned only. The course discusses the fixed-point semantics of CTL, and the main idea behind checking of CTL formulae with fairness.

Textbooks:
The course is based mainly on the 3 first chapters of:
M. Huth, 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

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

Course *4.4 Advanced Graph Theory*

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 (such as paths, traversals, trees, maximum flow, planarity, basic graph NP-complete problems, etc.) 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). Graph theory has long become recognized as one of the most useful mathematical subjects for the computer science student to master. The approach that is natural in computer science is the algorithmic one; the interest is not so much in existence proofs or enumeration techniques, as it is in finding efficient algorithms for solving relevant problems, or alternatively showing evidence that no such algorithms exist. The course is organized into six sections:

- Connectivity and traversability: bounded connectivity, regularity, overlay networks

The aim of this section is for students to be able to bypass the notion of spanning tree when they have to build and manage a network that covers another one.

- Graph coloring and graph NP-complete problems

The notion of graph coloring is for students to better understand and capture optimality notions related to network structures and be able to know if the problem they have to solve is or is not intrinsically expensive (from a time complexity point of view).

- Topological graph theory: embeddings, genus and maps

The aim is here for the students to be able to associate meaningful “numbers” to the network that support their applications, and exploit these numbers to know noteworthy properties of that underlying network.

- Analytic graph theory: random graphs, Ramsey graphs and the probabilistic approach

Random graphs are useful to model a lot of real networks. Knowing the base results associated with such graphs is consequently a prerequisite for mastering the behaviour of the applications running on such networks.

- Small-world networks (on grid and uniform topology, Kleinberg's distribution)

The small-world phenomenon constitutes now a basic knowledge for the researchers and engineers for whom the underlying network is the object they study. Gossip protocols are among the most famous example of protocols based on small-world results.

Textbooks:

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

Links:

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

Course *4.5 Human Factors, Human and Organisational Behaviour*

The purpose of this course is to present fundamental human and organisational concepts and frameworks that contribute to the “human-in-the-loop” concept in socio-technical systems. This covers three layers of activities:

- **Operator level:** Human information processing, human perception and human motor capabilities. Human error and failures are also addressed.

- **Collaboration level:** Communication schemes, collaboration, task allocation and collaboration breakdowns.
- **Management level:** Organisational aspects of work, task allocation, decision making. Organisational failures and management failures are also covered.

Case studies are used to describe the various levels of failures and a special attention is paid to the notion of task migration and autonomy of systems (design, implementation and assessment of delegating operators' tasks to a system).

The **introduction** addresses the basic concepts of human factors and also positions these human factors aspects within the bigger picture of collaborative activities within an organisation. There is no pre-requisite for the students on this topic as, in most European countries, "licence" curriculum in computer science does not provide courses on this topic.

The **cognitive processes** chapter presents the human cognitive capabilities for information processing and decision making. It is based on various models of human information processing such as [Wickens 99] [Norman 02] and Interacting Cognitive Subsystems (ICS) [Barnard 85]. It also presents modelling techniques for describing human knowledge and evolution of knowledge according to various activities like experience, training and inactivity. It deals with sensory/perception capabilities, decisions processes and memory functioning. The Skills-Rules-Knowledge (SRK) based model [Rasmussen et al. 94] is used to exemplify the impact of training on operators' planning of actions.

The **human performance** chapters (Human performance cognitive issues and Human performance physiological issues) address the issues of performance modelling and performance evaluation of operators/users of systems. The aim is to provide a basis for assessing the performance of the (user, system) couple. This addresses not only cognitive processes [Hick 52] but also motor capabilities such as Fitts's law [Fitts 54] and other human movement prediction laws [Accot & Zhai 97].

The **human error** chapter presents the basics of human-error principles and human-error assessment techniques. This chapter is based on [Reason 90] and [Reason 97] books and presents several models (including the Domino and Swiss cheese models) and various notations and approaches for human-error assessment such as HEART (Human Error Assessment and Reduction Techniques) or IDA (Influence Diagram Approach). It also addresses management issues and errors that can be related to managerial aspects.

The chapter **mode confusion and automation surprises** makes the connection between human factors aspects introduced in this course and system behaviour at the core of other courses. It concludes this course by showing the importance of globally and systemically accounting for human behaviour and system behaviour. The notion of incident and accident analysis and reporting is introduced in this chapter as another connexion point between human factors aspects and system behaviour [Johnson 03].

Textbooks:

[Johnson 03] 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/

[Norman 02] D. Norman: **The design of everyday things**, Basic books, 3rd edition, 2002.
[Rasmussen 94] J. Rasmussen, M. A. Pejtersen, L. P. Goldstein: **Cognitive Systems Engineering**. New York, USA, John Wiley and Sons, 1994
[Reason 90] J. Reason: **Human Error**. 1990. Cambridge University Press.
[Reason 97] J. Reason: **Managing the Risks of Organizational Accidents**, 1997, Aldershot, UK, Ashgate.
[Wickens 99] C. D. Wickens and J. G. Hollands: **Engineering Psychology and Human Performance**. 3rd edition, 1999, Prentice Hall.

Papers:

[Barnard 85] Barnard, P.J. (1985). **Interacting Cognitive Subsystems: A psycholinguistic approach to short term memory**. In A. Ellis (Ed.), *Progress in the Psychology of Language, Vol. 2*, London: Lawrence Erlbaum Associates, 197-258.
[Fitts 54] Paul M. Fitts (1954). **The information capacity of the human motor system in controlling the amplitude of movement**. Journal of Experimental Psychology, volume 47, number 6, June 1954, pp. 381-391. (Reprinted in Journal of Experimental Psychology: General, 121(3): 262--269, 1992).
[Hick 52] W. E. Hick. **On the rate of gain of information**. Quarterly Journal of Experimental Psychology, 4:11-26, 1952.
[Accot & Zhai 97] Johnny Accot and Shumin Zhai (1997). **Beyond Fitts' law: models for trajectory-based HCI tasks**. Proceedings of ACM CHI 1997 Conference on Human Factors in Computing Systems, pp. 295-302.

Links:

Links to places where parts of this course are taught:

- University of Hawaii (US): Human error. See: http://panko.cba.hawaii.edu/HumanErr/Index.htm
- University of California San Diego (US): cognitive engineering. Syllabus description through this page http://hci.ucsd.edu/102c/
- MIT (US): Human Memory and Learning. Fall 2002. See: http://ocw.mit.edu/OcwWeb/Brain-and-Cognitive-Sciences/
- Ohio State University (US): An introduction to joint man-machine cognitive systems. See courseware at http://csel.eng.ohio-state.edu/courses/ise573/
- MIT (US): Human Memory and Learning. See: http://ocw.mit.edu/OcwWeb/Brain-and-Cognitive-Sciences/

Course *4.6 Fundamentals of Real-Time Systems*

The purpose of this course is to provide a large overview of fundamental 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.
The **introduction** addresses basic concepts concerning real-time applications, real-time environment, and basic notions for real-time task scheduling. Although we assume that students have some basic knowledge and practice in operating systems, a reminder of operating system features should be done, leading the students to acquire basic notions of real-time kernels and microkernels, basic services such as tasks scheduling, resource management, synchronisation, etc.
Scheduling in real-time systems is the first major chapter of the course. Several very good textbooks are available and cover all aspects of real-time scheduling, like

Giorgio Buttazzo's book [Buttazzo 2005], but also [Cottet *et al.* 2002]. The sections of this chapter address the principles and the algorithms for aperiodic and periodic task scheduling, fixed-priority and dynamic priority approaches. This section includes the scheduling of independent tasks but also dependent tasks. Problems related to synchronisation and scheduling, i.e., tasks precedence relationship and tasks sharing critical resources, are part of this section together with solutions (e.g., PIP, PCP).
WCET analysis (Worst Case Execution Time analysis) is a major issue to set up an important parameter of schedulability tests. Material can be found in particular in a Special Issue on worst-case execution-time analysis of the Real-Time System journal (May 2000) [RTS 2000]. Beyond the introduction and the objective of WCET evaluation, the chapter must cover both dynamic WCET analysis (including measures, explicit test cases, symbolic state space exploration) but also static WCET analysis (notion of basic blocks, control-flow graphs and call graphs, program flow analysis) [Colin *et al.* 2003]. Due to the complexity of today's processor, low-level analysis must be addressed, i.e., caching effects, branch prediction [Colin & Puaut 2000], etc. Examples of measures and tools have to be presented.
Design Principles for real-time application, in particular distributed embedded applications, are discussed in this chapter and based on Kopetz's book [Kopetz 1997]. Clearly real-time systems are distributed (notion of real-time distributed environment). The notion of global time is thus a major issue to be addressed before all in this course. The subsection of this chapter should include modelling of real-time systems and then address fault tolerance issues. Another important part of this chapter concerns real-time communication and networks. Among other examples, the time-triggered protocols and architecture can be presented.
Programming distributed real-time systems is clearly a major part of the course that certainly leads to practical classes and student projects. This chapter can usefully be based on several books from Alan Burns and Andy Wellings, e.g., [Burns & Wellings 2001, Wellings 2004]. Beyond, the requirements for programming real-time systems, the chapter addresses first of all concurrent programming. It also covers real-time facilities and deadline scheduling, facilities for interacting with special purpose hardware. Last but not least, it also addresses error-handling facilities in programming languages, and more generally reliability of real-time systems, including fault-tolerance, atomic actions, etc. In practice, illustration can be provided with Real-Time POSIX systems and Real-Time Java.
Finally, **Real time executive layers** must be presented and to some extent used for practical classes and exercises. The students will discover major examples of real-time operating systems and kernels (like RT-LINUX, VxWorks, LynxOS, QNX, etc.), but also real-time middleware (like RT Java, RT-CORBA). Some short overview of open adaptive real-time executives should be addressed, i.e., ExoKernel, Think, 2K, etc.

Text books:

[Silberschatz 2008] A. Silberschatz, P. Baer Galvin, G. Gagne: **Operating Systems Concepts**, John Wiley & Sons, 2008, ISBN 0-470-12872-0.
[Buttazzo 2005] G. Buttazzo: **Measuring the Performance of Schedulability Tests**, Journal of Real-Time Systems, Springer Netherlands, Volume 30, N° 1-2, May, 2005, pp.129-154.
[Kopetz 1997] 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.

[Burns & Wellings 2001] A. Burns, A. Wellings: **Real-Time Systems and Programming Languages (Third Edition): Ada 95, Real-Time Java and Real-Time POSIX**, Addison Wesley, March 2001, 611 p., ISBN: 0201729881.
[Wellings 2004] A. Wellings: **Concurrent and Real-Time Programming in Java**, John Wiley & Sons Inc., October 2004, 431 p., ISBN-13: 9780470844373.

Papers:

[Colin & Puaut 2000] A. Colin, I. Puaut. **Worst Case Execution Time Analysis for a Processor with Branch Prediction**. Real-Time Systems, Special issue on worst-case execution time analysis, 18(2): 249-274, April 2000.
[Colin *et al* 2003] A. Colin, I. Puaut, C. Rochange, P. Sainrat. **Calcul de majorants de pire temps d'exécution: état de l'art**. Techniques et Sciences Informatiques (TSI), 22(5): 651-677, 2003.
[RTS 2000] **Special issue on worst-case execution-time analysis**, Real-Time Systems, Volume 18, Issue 2-3 (May 2000), ISSN:0922-6443, Editors John A. Stankovic (Univ. of Virginia, Charlottesville, USA), Wolfgang A. Halang (FernUniv. Hagen, Germany), Kim-Fung Man (City Univ. of Hong Kong, Hong Kong), Peter Puschner (Technische Univ. Wien, Vienna, Austria) and Alan Burns (Univ. of York, York, U.K.)

Links:

Links to places where parts of this course are taught:

- 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

Course *4.7 Fundamentals of Dependability*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/dependability_fundamentals.pdf

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. The course is composed of 8 chapters. The first chapter is devoted to the basic concepts and gives the corresponding definitions. The second chapter draws up the stete of the art via statistics. The third chapter addresses the threats to dependability. The three succeeding chapters address the means for dependability: fault removal, fault forecasting, and fault tolerance. The seventh chapter brings together the previous chapters into the development of dependable systems. The last chapter introduces resilience, in the context of ubiquitous computing systems.

- **Basic concepts and definitions.** Attributes of, means for, threats to dependability. Primary attributes: reliability, availability, safety, integrity, confidentiality, maintainability. Secondary attributes: robustness, survivability, resilience, accountability, authenticity, non-repudiability. Relationship between dependability and security, and between dependability and resilience.

- **State of the art from statistics.** Importance of dependability in current information infrastructures. Examples of widely publicised failures, evolution of dependability achievements over time, economic impact of failures.
- **Threats to dependability.** Faults, errors, failures. Definitions and classification. Failure pathology. Respective importances based on statistics (frequency, consequences) of the various classes (physical, development, interaction; accidental, malicious)
- **Fault removal.** Classification of verification approaches: reviews and inspections, abstract interpretation, theorem proving, model checking, symbolic execution, testing. The cost of verification. Statistics on classes of faults uncovered during development and during operation
- **Fault forecasting.** Stable vs. evolutive (growing, decreasing) dependability. Measures of dependability. Classification of evaluation approaches: modelling (FMECA, reliability diagrams, fault trees), measurements (operational testing). The case of safety-critical systems.
- **Fault tolerance.** Basic primitives: error detection (concurrent, pre-emptive), system recovery by error treatment (rollback, rollforward, compensation) and fault treatment (diagnosis, isolation, reconfiguration, reinitialization). Basic schemes: detection-and-recovery, masking-and-recovery. Solid vs. elusive and transient fault tolerance. Distributed fault tolerant systems, and the consensus problem. The notion of coverage and its importance. Examples of fault tolerant systems (Tandem families). Development fault tolerance. Validation of fault tolerance, especially via fault-injection.
- **Development of dependable systems.** Requirements and specifications. Risk analysis and criticality definition. Relationship between a) criticalities, and b) approaches and methods for dependability.
- **From dependability to resilience**. Fault and change tolerance. The ubiquitous systems context: dynamic componentisation. Technologies for resilience.

Textbooks:

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

N. Ferguson and B. Schneider: **Practical Cryptography,** John Wiley &Sons, 2003

Papers:

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

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

Links:

- At LAAS-CNRS, courseware available at http://www.laas.fr/TSF/courses/N6K-ENAC_Global_2006.pdf from Jean-Claude Laprie, and
- at http://www.laas.fr/TSF/courses/SEC2007-slides.pdf from Jean-Charles Fabre offers digests covering the contents of the course.
- At EPFL, slides can be found for Principles of Dependable Systems by G. Candea: http://dslab.epfl.ch/courses/pods/winter06-07/index.html

Course *4.8 Computer Networks Security*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/computer_network.pdf
The purpose of the 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 content follows a spiral of increasing difficulty and detail. The first part of the courses motivates the rest of the course. It starts by presenting why network security is a problem, what are the desirable security properties of a system, and the way a hacker can attack systems.
Part 2 introduces the main security paradigms, or building blocks: the TCB, basic cryptography and authentication / key distribution (revision from the previous course on cryptography), access control, and secure communication. The cryptographic topics build on the basic mechanisms introduced in course 4.2.
Part 3 uses the security paradigms introduced in part 2 to build more complex models of secure computing. It starts with a presentation of types of attacks and intrusions, introduces security strategies and follows with other topics up to architectural protection (e.g., with firewalls) and intrusion detection.
The last part goes one step further the paradigms and models, and presents in detail two important security frameworks and protocols: SSL and IPSec.
The courseware slides follow the sequence of the contents of the course. The main support text is Part IV of the book by Verissimo and Rodrigues. The other three recommended books provide additional material.

***Textbooks**:*
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

Links:
- 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

Course *4.9 Resilient Distributed Systems and Algorithms*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/resilient_distributed.pdf
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.
The slides provided as courseware follow the contents of the course. The main bibliography of the course includes two books that provide generic background on

architecting and designing secure and fault-tolerant distributed systems and algorithms [Verissimo and Rodrigues 2001, Guerraoui and Rodrigues 2006] and two book chapters that are specifically about *resilient* distributed systems and algorithms [Verissimo et al 2008, Verissimo et al 2003].
The topics covered in the course can be clustered in four main parts. After a brief introduction (the case for resilience), the first part (introduction to fault and intrusion tolerance) deals with a set of fundamental concepts that are essential for the students to understand what are and how are architected resilient distributed systems. These notions are fundamental for the students to be able to understand and learn the topics in the other parts.
The second part presents the main paradigms for resilience building: intrusion detection, self-enforcing vs. trusted third party protocols, threshold cryptography, Byzantine algorithms, resilience to attacks. This part goes beyond the mere concepts and introduces the building blocks of resilient systems.
The third part goes one step further by presenting models of resilient systems. It takes the concepts and paradigms introduced in the two previous parts, uses them to present strategies for building resilient systems, and delves into the details of the two main strategies: the use of Byzantine algorithms on fail-uncontrolled models, and the use of Byzantine algorithms on hybrid system models.
The final and fourth part puts together the contents of the three previous parts and presents some examples of resilient systems built using these strategies.

Text books:

[Verissimo and Rodrigues 2001] P. Verissimo, L. Rodrigues: **Distributed Systems for System Architects**, Kluwer, 2001.
[Guerraoui and Rodrigues 2006] R. Guerraoui, L. Rodrigues: **Introduction to Reliable Distributed Programming,** Springer, 2006.

Papers:

[Verissimo *et al* 2008] 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 Shambhu Upadhyaya (eds.), Elsevier, to appear, 2008.
[Verissimo *et al* 2003] P. Verissimo, N. F. Neves and M. Correia. **Intrusion-Tolerant Architectures: Concepts and Design**, In R. Lemos, C. Gacek, and A. Romanovsky, editors, Architecting Dependable Systems, volume 2677 of Lecture Notes in Computer Science, pages 3–36. Springer-Verlag, 2003.

Links:

- Univ. Lisboa: site of the book [Verissimo and Rodrigues 2001] http://www.navigators.di.fc.ul.pt/dssa/
- EPFL (Switzerland): slides related to the book [Guerraoui & Rodrigues 2006]: 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/

Course *4.10 Dependability and Security Evaluation of Computer-based Systems*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/dependability_and_security.pdf
This course presents 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.

The course is structured into six chapters.

The first chapter gives the **motivation for evaluating the dependability and security** of computing systems, using qualitative and quantitative evaluation approaches.
The second chapter defines the **metrics generally used for quantifying the dependability attributes** (reliability, availability, safety, performability, MTTF, MTTR, MUT, MDT, etc.).
The third chapter is devoted to the **modelling techniques** that are commonly used **to evaluate system level dependability metrics** based the knowledge of the system architecture and the failure and repair rates associated to its components. Two main modelling techniques are presented: combinatorial (Reliability block diagrams, fault trees) and state-based (Markov chains, Stochastic Petri nets and their extensions).
The fourth chapter briefly discusses key issues related to the collection and analysis of the **dependability data** based on field measurements and controlled experiments (fault injection, etc.).
The fifth chapter presents two real life **case studies** illustrating the application of model based-approaches to support the comparative assessment and selection of system architectures.
Finally, the sixth chapter is devoted to the **evaluation of computer systems with regards to malicious threats**. In particular, it first presents the main challenges to be addressed and state-of-the art techniques and criteria that are traditionally used to assess the security of computer systems. Then, this chapter focuses on experimental approaches based on honeypots that are aimed at collecting and analysing real life attacks observed on the Internet.

Textbooks:
General background
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

Dependability modelling
J. Arlat, K. Kanoun, J-C. Laprie: **Dependability modelling and evaluation of software fault-tolerant systems**, IEEE Transactions on Computers, Special Issue on Fault Tolerant Computing, 39 (4), pp.504-513, 1990.
J. Bechta-Dugan, S. J. Bavuso and M. A. Boyd: **Dynamic fault-tree models for fault-tolerant computer systems**, IEEE Transactions on Reliability, 41, pp.363-377, 1992
J-C. Laprie, K. Kanoun: **X-Ware reliability and availability modeling**, IEEE Transactions on Software Engineering, 18 (2), pp.130-147, 1992

N. Fota, M. Kaâniche, K. Kanoun: **Dependability Evaluation of an Air Traffic Control Computing System**, 3rd IEEE International Computer Performance & Dependability Symposium (IPDS-98), (Durham, NC, USA), pp. 206-215, IEEE Computer Society Press, 1998. Published in. Performance Evaluation, Elsevier, 35(3-4), pp.253-73, 1999
K. Kanoun, M. Borrel, T. Morteveille and A. Peytavin: **Modeling the Dependability of CAUTRA, a Subset of the French Air Traffic Control System**, IEEE Transactions on Computers, 48 (5), pp.528-535, 1999
I. Mura and A. Bondavalli: **Markov Regenerative Stochastic Petri Nets to model and evaluate the dependability of phased missions**, IEEE Transactions on Computers, 50 (12), pp.1337-1351, 2001
W. H. Sanders and J. F. Meyer: **Stochastic activity networks: Formal definitions and concepts**, Lectures on Formal Methods and Performance Analysis. Lecture Notes in Computer Science 2090, pp.315-343, Springer-Verlag, 2001
M. Kaâniche, K. Kanoun and M. Rabah: **Multi-level modelling approach for the availability assessment of e-business applications**, Software: Practice and Experience, 33 (14), pp.1323-1341, 2003
C. Betous-Almeida and K. Kanoun: **Construction and Stepwise Refinement of Dependability Models**, Performance Evaluation, 56, pp.277-306, 2004
M. Kaâniche, P. Lollini, A. Bondavalli, K. Kanoun: **Modelling the resilience of large and evolving systems**, in International Journal on Performability engineering, vol.4, n°2, pp. 153-168, 2008.

Experimental measurements
J. Arlat, A. Costes, Y . Crouzet, J.-C. Laprie, D. Powell: **Fault Injection for Dependability Evaluation of fault-tolerant systems**", IEEE Transactions on Computers, 42(8), pp. 443-455, 1993
P. Koopman, J. DeVale: **The exception handling effectiveness of POSIX Operating Systems**", IEEE Trans. on Software Engineering, vol. 26, n°9, 2000
Eric Marsden, Jean-Charles Fabre, Jean Arlat: **Dependability of CORBA Systems: Service Characterization by Fault Injection**, SRDS-2002, pp. 276-85, 2002
R. Iyer, Z. Kalbarczyck : **Measurement-based Analysis of System Dependability using Fault Injection and Field Failure Data**, Performance 2002, LNCS 2459, pp.290-317, 2002
D. P. Siewiorek et al. : **Reflections on Industry Trends and Experimental Research in Dependability**, IEEE transactions on Dependable and Secure Computing, vol.1, n°2, April-June 2004.

Security assessment
B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid and D. Gollmann: **Towards Operational Measures of Computer Security**, Journal of Computer Security, 2, pp.211-229, 1993
R. Ortalo, Y. Deswarte and M. Kaâniche: **Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security**, IEEE Transactions on Software Engineering, 25 (5), pp.633-650, 1999
D. M. Nicol, W. H. Sanders and K. S. Trivedi: **Model-based Evaluation: From Dependability to Security**, IEEE Transactions on Dependable and Secure Computing, 1 (1), pp.48-65, 2004.
F. Pouget, M. Dacier, V.H. Pham: **Leurré.com: on the advantages of deploying a large scale distributed honeypot platform**, E-Crime and Computer Conference (ECCE'05),29-30 March 2005, Monaco

E. Alata, V. Nicomette, M. Kaâniche, M. Dacier and M. Herrb: **Lessons Learned from the Deployment of a High-Interaction Honeypot**", in Sixth European Dependable Computing Conference (EDCC-6), (Coimbra, Portugal), pp.39-44, IEEE Computer Society, 2006

Links:

- The course book's webpage at http://www.ee.duke.edu/~kst/ include material related to the course.
- Another relevant link is to: http://www.di.unito.it/~greatspn/bookdownloadform.html on modelling with generalized SPN.

Course *4.11 Testing, Verification and Validation*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/testing.pdf
The purpose of this course is an introduction to testing, verification and validation in the design and analysis of systems, and to provide an advanced background on related methods and tools. The course stresses the development of practical skills based on a solid theoretical foundation.

The methods and techniques to be presented in the course are those that have proved to be relevant in practice and successful in supporting modelling, specification, analysis and verification of software. These include practical formal techniques and methods for static analysis and validation as well as for testing software systems.

The **model-checking** chapter introduces temporal logics (CTL and LTL) and presents related techniques for model checking [BB et al. 01] [Alur]. The state space explosion problem and practical limits to model checking will be addressed.
The **theorem proving** chapter [RV01] puts emphasis on a theorem prover tool. Additional background knowledge about the logic implemented in the chosen tool will be presented. The theoretical and practical aspects of the tool will be studied.
The **static program analysis** chapter [HNN99] introduces students to classical data-flow analysis [ALSU06] and gives foundations of abstract interpretation theory [CC77,CC79].
The **software-testing** chapter provides an understanding of testing problems, and covers the major test design techniques. The course offers students a view of structural & functional approaches to testing, mutation analysis and probabilistic test approaches. In support of the course, [B90] [AO08] give a solid technical background on test approaches based on the coverage of models, and [CJ02] provides practical insights into process issues & test-related activities.

Textbooks and other References:
[ALSU06] A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman: **Compilers: Principles, Techniques, and Tools**, Addison-Wesley, 2006.
[Alur] Lecture notes on **Computer-Aided Verification** by Rajeev Alur at University of Pennsylvania, Philadelphia, USA, http://www.cis.upenn.edu/cis673/
[AO08] P. Ammann, J. Offutt: **Introduction to Software Testing**, Cambridge University Press, 2008
[B90] B. Beizer: **Software Testing Techniques**, Van Nostrand Reinhold, 1990 (2nd edition)
[BB et al 01] Part I of B. Berard, et al.: **System and Software Verification – Model-Checking Techniques and Tools**, Springer, 2001.

[CC77] 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.
[CC79] P. Cousot, R. Cousot: **Systematic Design of Program Analysis Frameworks**, POPL79, pages 269–282, San Antonio, Texas, 1979.
[CJ02] R. D. Craig, S. P. Jaskiel: **Systematic Software Testing**, Artech House, 2002
[K 99] T. Kropf: **Introduction to Formal Hardware Verification**, Springer, 1999.
[HNN99] C. Hankin, F. Nielson, H. R. Nielson: **Principles of Program Analysis**, Springer, 1999.
[RV01] A. Robinson, A. Voronkov (eds.): **Handbook of Automated Reasoning**, Volume I, North Holland, 2001

Links:

- "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
- "Introduction to software testing " Paul Ammann and Jeff Offutt at George Mason University http://www.cs.gmu.edu/~offutt/softwaretest/powerpoint/ 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

Course *4.12 Usability and User Centred Design for Dependable and Usable Socio-technical Systems*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/usability.pdf
The purpose of this course is to present usability concepts and frameworks that contribute to the notion of usability of interactive systems [Dix et al. 03]. It also introduces techniques for usability evaluation and the notion of development processes that specifically aim to produce usable systems. Command and control systems are the main application area of this course but additional application domains such as web applications [Scapin et al. 2000a] or graphical user interfaces are also considered.

The **Introduction** addresses the basic concepts of usability by providing definitions and examples as well as the principles underlying this concept. It also introduces the notion of affordances in user interfaces [Norman 02] and some examples of usability problems. The pre-requisite for the students for this topic is limited to the course 4.4. on Human Factors: Human and Organisational Behaviour delivered the first semester.

The **ergonomic rules and design guidelines** chapter presents established theories regarding the ergonomics of user interfaces. It presents how such knowledge is gathered, structured and how tools [Vanderdonckt 95] support their access and organisation [Scapin et al. 2000b]. ISO standards on usability guidelines will also be presented and discusses [ISO 9241]. The chapter also introduces the basics of heuristic evaluation and how user testing is central to the evaluation of usability.

The chapter on **work analysis and task analysis** is dedicated to introducing the notion of work and how users' and operators' activities can be analysed and modelled [Diapper & Stanton 04]. It introduces the notions of activity, tasks and scenarios that are critical elements for the understanding of operators at work. Task modelling approaches such as CTT (Concur Task Trees) [Paterno 97], UAN (User Action Notation) [Hix, Siochi & Hartson 90] or GOMS (Goals Operators Methods and Selection rules) [John & Kieras 96] are introduced. Tools, supporting task model edition, simulation and validation are also introduced and their usefulness is shown on various examples and case studies.

The chapter on **usability evaluation** will introduce methods, techniques and tools to support usability evaluation of interactive systems and user interfaces in particular. It will introduce the notion of usability testing and heuristic evaluations [Nielsen 94]. Functioning and setup of usability tests both ecological and within dedicated structures like usability labs will be presented in this chapter.

The chapter **User centred development processes** proposes a broader perspective about usability by presenting the way in which usability-related activities can be related to computer systems. Cost-benefits analysis for usability evaluation [Bias & Mayhew 05] is also presented together with real examples from industrial applications both in the field of safety critical systems and in the field of walk-up and use systems.

The last chapter **Human factor engineering** corresponds to the application of the concepts introduced in course 4.4. **Human Factors: Human and Organisational Behaviour.** It exploits also the elements from task analysis and task modelling presented in the current course. The objective here is the application of previous concepts with a specific emphasis on allocation of functions and organizational errors.

Textbooks:

[Norman 02] D. Norman: **The design of everyday things**, Basic books, 3rd edition, 2002.
[Rasmussen et al. 94] J. Rasmussen, M. A. Pejtersen, L. P. Goldstein, (1994): **Cognitive Systems Engineering**, New York, USA, John Wiley and Sons
[Reason 90] J. Reason: **Human Error**, Cambridge University Press, 1990.
[Dix et al. 03] A. Dix, J. Finlay, G. Abowd, R. Beale: **Human Computer Interaction**, Prentice Hall, 2003 (3rd Edition)
[Rosson & Carroll 02] M. B. Rosson, J. M. Carroll: **Usability Engineering: Scenario-based Development of Human-Computer Interaction**. New York: Morgan Kaufmann Publishers, 2002.

[Diapper & Stanton 04] D. Diaper, N. A. Stanton (eds.): **The Handbook of Task Analysis for Human-Computer Interaction,** edited by Lawrence Erlbaum Associates, 2004
[Nielsen 94] J. Nielsen: **Usability Engineering**, Morgan Kaufmann, San Francisco, 1994.
[Bias & Mayhew 05] R. G. Bias, D. J. Mayhew (eds.): **Cost-Justifying Usability, Second Edition: An Update for the Internet Age**, Second Edition Morgan Kaufman

Papers:

[Vanderdonckt 95] J. Vanderdonckt, **Accessing Guidelines Information with SIERRA**, in Proc. of 5th IFIP TC 13 Int. Conf. on Human-Computer Interaction INTERACT'95 (Lillehammer, 27-29 June 1995), K. Nordbyn, P.H. Helmersen, D.J. Gilmore & S.A. Arnesen (eds.), Chapman & Hall, Londres, 1995, pp. 311-316.
[ISO 9241] **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
[Scapin et al. 2000a] D. Scapin, J. Vanderdonckt, CH. Farenc, R. Bastide, CH. Bastien, C. Leulier, C. Mariage, PH. Palanque, **Transferring Knowledge of User Interfaces Guidelines to the Web**, in Proc. of Int. Workshop on Tools for Working with Guidelines TFWWG'2000, Springer-Verlag, Londres, 2000, pp. 293-303.
[Scapin et al. 2000b] D. Scapin, C. Leulier, J. Vanderdonckt, C. Mariage, CH. Bastien, CH. Farenc, PH. Palanque, R. Bastide, **A Framework for Organizing Web Usability Guidelines**, in Proc. of 6th Conf. on Human Factors and the Web HFWeb'2000 Austin, Ph. Kortum & E. Kudzinger (eds.), University of Texas, Austin, 2000.
[Paterno 97] F.Paterno', C.Mancini, S.Meniconi. **ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models**, Proceedings Interact'97, Chapman & Hall, July'97, pp. 362-369.
[Hartson, Siochi & Hix 90] Hartson, H. R., Siochi, A. C., and Hix, D. 1990. **The UAN: a user-oriented representation for direct manipulation interface designs**. ACM Transactions on Office Information Systems, 8, 3 (Jul. 1990);
[John & Kieras 96] John, B. E. and Kieras, D. E. 1996. **Using GOMS for user interface design and evaluation: which technique?** ACM Trans. Computer-Human Interaction 3, 4 (Dec. 1996), 287-319

Links:

Links to places where parts of this course are taught:

- University of Lancaster (UK): Human Computer Interaction. See. : http://www.hcibook.com/e3/resources/
- ACL SIGCHI (US) Human Computer Interaction lectures. See list of lectures as well as links to courseware at http://sigchi.org/cdg/index.html
- MIT (US) Pervasive Human Centric Computing See courseware at: http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-883Spring-2006/CourseHome/index.htm
- Human Factors International (US) See certification packages and description. See syllabus at http://www.humanfactors.com/training/usability-training.asp

Course *4.13 Management of Projects*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/project_management.pdf
This course has been conceived to be principally taught by frontal lectures in the class room. Some lab or demo sessions can be considered for the third part of the course to experience or show specific automatic tools (some mentioned in the slides).

Part 1 of the course is introductive and it doesn't need any specific comment on how to teach it.
Part 2 of the course "Project Management as a Process", contains the main concepts and the description of what project management is in practice (it contains the *what*). It should be taught by using an exemplar (real or realistic) project plan document in order to allow the students learn the different phases of a project by understanding the contents and the evolutions of the project plan.
Part 3 describe techniques and methods to perform project management in practice (it contains the *how*). Some specific techniques for performance measurement can be detailed and possibly applied on sample case studies.
Part 4 describes the principal activities the requirements engineering process is composed of. The teaching of these activities is to be made by taking into account the Project Management phases described in Part 2. To do that the specific schemes at the end of each requirements engineering phase description shall be used.

***Textbooks**:*
H. Eisner: **Essentials of Project and System Engineering Management**, Second Edition. John Wiley and Sons, 2002.
J. 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.**
M. B. Chrissis, M. Konrad, S. Shrum: **CMMI Guidelines for Process Integration and Product Improvement**, SEI Series in Software Engineering, 2004.
I. Sommerville, P. Sawyer **Requirements Engineering: A Good Practice Guide**. John Wiley and Sons Ed. 1997.
A. Aurum, C. Wohlin **Engineering and Managing Software Engineering.** Springer Ed. 2005.

Links:
- 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

Course *4.14 Middleware Infrastructures for Application Integration*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/middleware.pdf
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.

The course content follows an increasing difficulty approach. The first part of the course shows how historically integration among software systems has been achieved following a synchronous paradigm. Then the focus goes on a key technology: Web Services. How to achieve reliability and security in web services is then considered. Then component based middleware platforms are introduced and the real cases of J2EE and EJB are investigated. At the end of the first part, web service stack on the top of J2EE will be shown to close the circle.
The second part introduces students to the notion of asynchronous middleware platforms. Publish-subscribe mechanisms to disseminate events in a distributed system are thoroughly investigated. Particular attention will be devoted to different methods of event routing. Then the DDS technology, a specific publish-subscribe technology, is analyzed and the focus will be on how to guarantee QoS properties in event distribution such as persistence, timeliness and availability.

The courseware slides follow the sequence of the contents of the course. The main support texts are the one from Alonso et al. and the one of Tanembaum. Additional material can be retrieved from the web sites of DDS.

***Textbooks**:*
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

Links:
- Università di Roma "La Sapienza", Distributed System Platforms, (in Italian) R. Beraldi http://www.dis.uniroma1.it/~beraldi/PSD0708/
- Università di Bologna, Middleware. F. Panzieri http://courses.web.cs.unibo.it/SistemiMiddleware/MaterialeDiRiferimento

Course *4.15 Software Reliability Engineering*
ReSIST Courseware:
http://resist.isti.cnr.it/files/corsi/courseware_slides/software_reliability_eng.pdf
As software systems are playing an increasing role in real life systems, there is an imperative need for improving software reliability in order to improve system dependability and reduce maintenance cost.
Measurement is an important part of the software reliability engineering activities. It is essential for understanding the underlying phenomena, making correlation between observations, identifying weakness, controlling important issues and evaluating software reliability.
The purpose of this course is to give a global overview of approaches to software reliability analysis, evaluation and improvement.

The course is composed of five chapters.
The first chapter gives the **motivation for software reliability engeneering**, during system development and operational life, including software maintenance.
The second chapter is devoted to the **methods for software reliability engineering**, based essentially on failure and correction data collection, data validation and

analysis, as well as descriptive statistics and reliability trend analysis.
The third chapter addresses **Software dependability evaluation**. In particular, it presents a set of reliability growth models and some models in stable reliability, before giving an overview and an example of dependability benchmarks for Off-the-Shelf software components.
The fourth chapter is concerned with **software reliability improvement** approaches and with the **maturity of the software development process.**
Finally the fifth chapter shows the application of the approaches presented in the course to case studies.

Textbooks:
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.

Papers:
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 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.

Links:
- OpenSeminar, Software Reliability Engineering, John Musa, Laurie Williams http://openseminar.org/se/courses/41/modules/206/index/screen.do

Application Track on Resilience in Communication Networks

Course *4.16.1 IP Networks and Services Resilience*
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 basic services (e-mail, Web)' resilience (vulnerabilities' identification, countermeasures analysis, …) is the starting point of this course, followed by the description of FTP's threats (banner grabbing and enumeration, brute force password guessing, bounce attacks, …), IP network scanning and VPN security issues. 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. After a presentation of VoIP's protocols and architecture, the threats to this type of communication systems will be detailed. This lecture ends with the description of techniques to validate and secure IP networks and services infrastructures.

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

Links:
the contents of this course can be found scattered over the following URLs.
- 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)

Course *4.17.1 Resilience of Mobile Applications*

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. After the study of secure access to wireless networks, vulnerabilities of protocols such as IEEE 802.11 and Bluetooth will be described. It is followed by a presentation of (i) security requirements for mobile multimedia applications, and (ii) network protocols (SIP, SRTP) for secure multimedia streaming services. 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 this type of computing more vulnerable to various security-related threads: requirements and deployment techniques for pervasive computing form the last topic covered by this course.

***Textbooks**:*

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

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

Links:

the contents of this course can be found scattered over the following URLs.

- 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 548: *Security and Privacy Issues in Telecommunications*
- TCOM 556: *Cryptography and Network Security*
- 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_enginee ring.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)

Links to general telecoms courses: these URLs are provided to help the students following this application track who need a reminder of basic notions in communication systems.

• Hong Kong University of Science & Technology:
http://www.sengpp.ust.hk/~msc/telecom/05-07/courses.htm#top
• Queen Mary U. of London: http://www.elec.qmul.ac.uk/study/msc/msctel.html
• Stanford University: http://www.stanford.edu/class/ee360/
• Swinburne U. of Technology:
http://courses.swinburne.edu.au/Courses/ViewCourse.aspx?mi=100&id=21080
• Technical U. of Denmark:
http://www.fotonik.dtu.dk/English/Education/Int_Msc.aspx
• Telecom Paris Tech: http://www.telecom-paristech.fr/en/msci/mobile_communications/
• University College London:
http://www.ee.ucl.ac.uk/students/postgraduate/masters/msctelecoms/courses

Application Track on Safety Critical Systems

Course *4.16.2 Development Process and Standards for Safety Critical Applications*
Safety critical applications require special attention to the development process of the software and the system and also awareness of different standards, generic ones as well as application specific ones.
Generally, the development process must follow certain rules. Still, different choices are possible. Applicable life cycle models should be introduced and compared.
In most application areas, safety critical systems need to be approved by a licensing authority or similar. This requires a strict compliance with recommendations and rules, covering the complete life cycle and including documentation. All tools and methods presented must be in accordance with the related standards. These are first the generic ones, which give an introduction and state general rules. Depending on the application area for which the safety critical system is developed, the related application specific standards need to be used.
Within the course, some available and accepted development processes will be described. The influence of licensing / certification on this process will be emphasized. Generic standards as the basic safety standards will be introduced. Depending on the anticipated application area, sector specific standards will be included. Examples of safety critical systems will presented.

Textbooks:

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.

Links:
International Electrotechnical Commission (Standardisation body for all electrical, electronic and related technologies): www.iec.ch
International Organisation for Standardisation (General standardisation body for business, government and society): www.iso.ch

Course *4.17.2 Architectural Issues and Examples of Systems*
The design and the architecture of systems influence different resilience attributes. Therefore the architecture must be chosen carefully.
Within this course, different architectures for systems shall be explained and their influence on the dependability shall be presented. This involves hardware as well as software issues and designs. Different architectures for fault tolerance and failure detection, with fail-safe and fail-operational feature, will be evaluated. Within a network, the communication, the protocols and their implementation are also essential factors.
Examples of real systems and their implementation show the importance of architecture decisions.

Textbooks:
J. L. Hennessy and D. A. Patterson: **Computer Architecture: A Quantitative Approac***h*, 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

Application Track on Resilience in e-Business

Course *4.16.3 Enterprise Security*
The purpose of this course is to offer a broad overview of network defense and countermeasures for E-Business and it is intended to give an introduction to the "best practices" associated with the aforementioned technologies and methodologies.

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 defence 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). The course will end describing the process of security in large enterprises to cope with the increasing complexity of regulations including certification, audits, risk management.

The courseware slides follow the sequence of the contents of the course. The main support texts are the one from Newman and the one of Ortmeier. Material related to governance and compliance processes in IT enterprises can be found in web sites of major technology providers such as IBM. Additional material can be retrieved from the web sites of telco operators.

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

Links:

- 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

Course *4.17.3 Computer and Network Forensics*

The knowledge of computer and network forensics has become essential in securing today's network-centric computing environment. This new course will give the students both the fundamental knowledge and hands-on practice on computer and network forensics. The added exposure to forensics will enhance the marketability of our students and serve the students who carry the skills and knowledge forward into their future careers. Competence will be obtained in using established forensic methods in the handling of electronic evidence; rigorous audit/logging and date archival practices; prevention, detection, apprehension, and prosecution of security violators and cyber criminals.

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.

The main support texts are the one from Carrier and the one of Kruse-Heiser. The course will consist of a final project, a presentation of the project itself as well as a written exam.

Textbooks*:*

B. Carrier: **File System Forensic Analysis**, Addison-Wesley, 2005
C. Prosise, K. Mandia: **Incident Response: Investigating Computer Crime,** Berkeley, California: Osborne/McGraw-Hill, 2001
W. Kruse, J. Heiser: **Computer Forensics: Incident Response Essentials,** Addison-Wesley, 2002
Phillips, Nelson, Enfinger, Stuart: **Guide to Computer Forensics and Investigations**, Course Technology, 2006

Links:

- 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. Conclusions

Deliverables D37 and D38 constitute the full shaping of a comprehensive MSc Curriculum in Resilient Computing that tackles one of the most challenging goals of ReSIST. In our knowledge it is the first attempt to provide a quite comprehensive database of material that can be used for starting an MSc track on this topic with the indication of the available support material for the best success in the teaching/learning process. Dissemination of the results of this effort to the largest possible number of European (and not only) Universities and promotion for its use

will continue after the end of ReSIST. The large convergence on the usefulness of gathering the material and the dedication of people in the community of dependable and resilient computing to maintain updated the material is the best guarantee that this effort will have long lasting benefits to the entire European community of industrialists and academics working in the field. 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.

Reference to textbooks

- 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. Freely available at: http://www.di.unito.it/~greatspn/bookdownloadform.html
- 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. Aurum, C. Wohlin **Engineering and Managing Software Engineering.** Springer Ed. 2005.
- 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.
- R. G. Bias, D. J. Mayhew (eds.): **Cost-Justifying Usability, Second Edition: An Update for the Internet Age**, Second Edition Morgan Kaufman
- P. Bishop (ed.): **Dependability of Critical Computer Systems – 3, Techniques Directory**, ISBN 1-85166-544-7.
- A. Burns, A. Wellings: **Real-Time Systems and Programming Languages (Third Edition): Ada 95, Real-Time Java and Real-Time POSIX**, Addison Wesley, March 2001, 611 p., ISBN: 0201729881.
- G. Buttazzo: **Measuring the Performance of Schedulability Tests**, Journal of Real-Time Systems, Springer Netherlands, Volume 30, N° 1-2, May, 2005, pp.129-154.
- 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
- 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.
- 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
- D. Diaper, N. A. Stanton (eds.): **The Handbook of Task Analysis for Human-**

Computer Interaction, edited by Lawrence Erlbaum Associates, 2004
- A. Dix, J. Finlay, G. Abowd, R. Beale: **Human Computer Interaction**, Prentice Hall, 2003 (3rd Edition)S. Even: **Graph Algorithms**, Computer Science Press, 1979.
- H. Eisner: **Essentials of Project and System Engineering Management**, Second Edition. John Wiley and Sons, 2002.
- W. Emmerich **Engineering distributed objects**, John Wiley, 2000
- N. Ferguson and B. Schneider: **Practical Cryptography,** John Wiley &Sons, 2003
- J.L. Gross and J. Yellen (Eds.): **Handbook of graph theory**, CRC Press, 2003.
- Hankin, F. Nielson, H. R. Nielson: **Principles of Program Analysis**, Springer, 1999.
- J. L. Hennessy and D. A. Patterson: **Computer Architecture: A Quantitative Approac*h***, 2nd Edition, Morgan Kaufmann Publishing Co., Menlo Park, CA.
- M. Huth, 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
- R. Guerraoui, L. Rodrigues: **Introduction to Reliable Distributed Programming,** Springer, 2006.
- 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/
- G. Karmakar, 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.
- T. Kropf: **Introduction to Formal Hardware Verification**, Springer, 1999.
- W. Kruse, J. Heiser: **Computer Forensics: Incident Response Essentials,** Addison-Wesley, 2002
- J-C. Laprie et al.: **Guide de la sûreté de fonctionnement**, Cepaduès Editions,1995 (in French)
- 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. http://www.cacr.math.uwaterloo.ca/hac/
- 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.
- P. J. Ortmeier: **Security Management**, Prentice Hall, 2004
- A. Patterson and J. L. Hennessy: **Computer Organization and Design. The Hardware - Software Interface,**. Morgan Kaufmann Publishers, San Francisco, CA
- Phillips, Nelson, Enfinger, Stuart: **Guide to Computer Forensics and Investigations**, Course Technology, 2006
- T. Porter, J. Kanclirz Jr.: **Practical VoIP Security,** Syngress, 2006

- C. Prosise, 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
- S. Ross: **Probability Models for Computer Science**, Academic Press, 2002, San Diego, CA
- M. B. Rosson, J. M. Carroll: **Usability Engineering: Scenario-based Development of Human-Computer Interaction**. New York: Morgan Kaufmann Publishers, 2002.
- 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** (Second Edition), McGraw-Hill, 2007. http://www.cs.bris.ac.uk/~nigel/Crypto_Book/
- I. Sommerville, P. Sawyer: **Requirements Engineering: A Good Practice Guide**. John Wiley and Sons Ed. 1997.
- W. Stallings: **Cryptography and Network Security**, (4th Edition), Prentice Hall
- A. Tanenbaum, M. Van Steen: **Distributed Systems,** (2nd 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
- P. Verissimo and L. Rodrigues: **Distributed Systems for System Architects**, Kluwer, 2001
- A. Wellings: **Concurrent and Real-Time Programming in Java**, John Wiley & Sons Inc., October 2004, 431 p., ISBN-13: 9780470844373.
- D. Wickens and J. G. Hollands: **Engineering Psychology and Human Performance**. 3rd edition, 1999, Prentice Hall.

Appendix. Freely available courseware material

In this Appendix an annotated list of freely available courseware material is presented. It is the outcome of a survey made worldwide on the web.
All material is on-line on the ReSIST web site http://www.resist-noe.org/.
The material can be categorized into 16 groups:

1. Probability and Statistics
2. Cryptology and Cryptography
3. Mathematical Logic
4. Dependable Systems
5. Distributed Systems
6. Software Reliability Engineering
7. Security
8. Privacy
9. HCI and Interactive Systems
10. General Topics
11. Testing, Verification and Validation
12. Real-Time Systems
13. Mathematical Programming and Operations Research
14. Graph Theory
15. Stochastic Network Optimization
16. Pattern Recognition Resources

1. Probability and Statistics:

Slides for ADVANCED PROBABILITY K. Trivedi:
http://www.ee.duke.edu/~kst/
This set of slides is based on the well-known book **Probability and Statistics with Reliability, Queuing, and Computer Science Applications,** John Wiley and Sons, New York, 2001 by Kishor Trivedi of Duke University. This is the second edition of the book of the same author with the same title published by Prentice-Hall.
The book provides a comprehensive introduction to probability, statistics and stochastic processes, and leads the reader to a natural understanding of how these concepts and related methods can be applied in the modelling, analysis and quantitative evaluation of the performance and reliability of systems. The major merit of the book is that its pages contain, in the same readable language, classic material together with recent advancements and achievements distilled from the author's research activity and experience in the area of the modelling and evaluation of stochastic systems.

2. Cryptology and Cryptography:

Slides for CRYPTOLOGY B. Preneel:
http://homes.esat.kuleuven.be/~preneel/classes.html
This set of slides provides an overview of the state of the art in the design of cryptographic algorithms. It reviews the different type of algorithms for encryption and authentication.
The principles are explained of stream ciphers, block ciphers, hash functions, public key encryption algorithms and digital signature schemes. Subsequently the design and evaluation procedures for cryptographic algorithms are discussed. The slides present

also several exercises for the reader. The author requests a notification for using such material.

Book HANDBOOK OF CRYPTOGRAPHY A. Menezes, P. van Oorschot, S. Vanstone:
http://www.cacr.math.uwaterloo.ca/hac/
This site contains the downloadable book for personal use (courtesy of CRC, see copyright notice)

Slides for CRYPTOGRAPHY (in Italian) A. Marchetti Spaccamela:
http://www.dis.uniroma1.it/%7Ealberto/didattica/critto.html
This set of slides is based on the book C Kaufman, R Perlman, M Speciner **Network Security, private communication in a public world**, 2nd edition, Prentice Hall, 2002. The slides present principle of Cryptography and their application to Network Security.

Slides for CRYPTOGRAPHY AND SECURITY (some in German) M. Waldvogel:
http://www.inf.uni-konstanz.de/disy/teaching/ss06/kryptographie/
In this site of Universitat Konstanz several presentations on Cryptography and Security can be found; they cover: Cryptographic checksums, Cryptoanalysis, WEP and WPA, VoIP Security, Why Cryptosystems fail, AES, Xbox Security and Trust and Reputation

Slides for CRYPTOGRAPHY E. Oswald, N. Smart:
http://www.cs.bris.ac.uk/Teaching/Resources/COMS30124/
These slides contain an introduction to cryptography, taught at University of Bristol, UK.

Book CRYPTOGRAPHY, AN INTRODUCTION: SECOND EDITION N. Smart:
http://www.cs.bris.ac.uk/~nigel/Crypto_Book/
This site contains the downloadable book for personal use (see copyright notice)

Slides for CRYPTOGRAPHY M. Backes:
http://www.infsec.cs.uni-sb.de/teaching/SS08/Cryptography/
This site contains the slides by Michael Backes of Saarland University.
This course is an introduction to modern cryptography. It will introduce cryptography from scratch, i.e., no previous knowledge in cryptography or computer security is required. The list of topics comprises:

- Information-theoretic security and the One-time Pad
- Symmetric encryption, stream ciphers, block ciphers, Data Encryption Standard (DES), Advanced Encryption Standard (AES)
- Asymmetric encryption, cryptosystems based on RSA and on the discrete logarithm problem, Cramer-Shoup encryption
- Digital signature schemes
- Cryptographic hash functions
- Selected cryptographic protocols and their security
- Crypto in the "real world"
- Basic concepts of advanced cryptographic primitives and current research topics: bit commitment, zero-knowledge proofs, simulatability, linking formal verification and cryptography

Slides for INFORMATION SECURITY D. Basin :
http://www.infsec.ethz.ch/education/ss08/infsec08
It is a survey of the principles and methods of information security, along with the discussion of selected applications. This includes the following topics: Foundations of Cryptography, Key Management and Trust, Security Protocols, Access Control and Security Policies, Anonymity and Privacy
Access only through username and password

Slides for CRYPTOGRAPHY U. Maurer :
http://www.crypto.ethz.ch/teaching/lectures/Krypto06/
Access only through username and password

Slides for COMPUTER AND NETWORK SECURITY R. L. Rivest, S. Goldwasser :
http://courses.csail.mit.edu/6.857/2008/lecture.html
Access only through username and password

3. Mathematical Logic:

Book MATHEMATICAL LOGIC V. Detlovs, K. Podnieks:
http://www.ltn.lv/~podnieks/mlog/ml.htm
This is a hyper-textual book that contains all material on principle of mathematical logic, progressing from classical propositional and predicate logic to reach constructive propositional and predicate logic and Kripke semantics.

Book LOGIC IN COMPUTER SCIENCE M. Huth, M. Ryan:
http://www.ewidgetsonline.com/cup/widget.aspx?bookid=51/3mLE/ColK5qnmfcLSyg==&buyNowLink=http://sec.ebooks.com/cambridge-add.asp?I=283471&f=3
This site contains the online version of the book for personal use (see copyright notice)

Slides for LOGIC IN COMPUTER SCIENCE M. Huth, M. Ryan:
http://www.cs.bham.ac.uk/research/projects/lics/
It offers several materials, among them an interactive tutor for each chapter.
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

Slides for LOGIC IN COMPUTER SCIENCE T. Coquand:
http://www.cs.chalmers.se/Cs/Grundutb/Kurser/logcs/index.html
The slides present mainly a set of exercises in logic.

Slides for LOGIC IN COMPUTER SCIENCE Julia Lawall, N. Jones :
http://www.diku.dk/undervisning/2005e/213/

Slides for LOGIC IN COMPUTER SCIENCE N. Adersen, J. Lawall:
http://www.diku.dk/undervisning/2004f/202/
This course provides a **sound basis in logic** and an introduction to the **logical frameworks** used in modelling, specifying and verifying computer systems. Propositional and predicate logic are detailed, as well as some specialized logics used for reasoning about the correctness of computer systems. A carefully chosen core of essential terminology is introduced; further technicalities are introduced only where

they are required by the applications.
Numerous examples are given, as well as a full exposition of a fast-growing technique for modelling and verifying computer systems, known as **symbolic model checking**.

4. Dependable Systems:

Slides for PRINCIPLES OF DEPENDABLE SYSTEMS G. Candea:
http://dslab.epfl.ch/courses/pods/winter06-07/index.html
This course offers advanced students a holistic view of the principles that underlie dependable software-centric computing systems, with an emphasis on large-scale distributed systems and Internet services. Lectures cover techniques for high availability, fault tolerance, monitoring, diagnosis; the course looks at how to achieve high availability through fast recovery and graceful service degradation, as well as techniques that leverage redundancy and replication. The utopia of flawless software is discussed, as well as ways to cope with human operator errors, and metrics for evaluating dependability.

Slides for DEPENDABILITY OF COMPUTING SYSTEMS JC. Laprie:
http://www.laas.fr/TSF/courses/N6K-ENAC_Global_2006.pdf
This set of slides of the master course offered at ENAC-ENSICA contains a full digest of the fundamentals of Dependability.

Slides for DEPENDABILITY AND SECURITY JC. Fabre
http://www.laas.fr/TSF/courses/SEC2007-slides.pdf
This set of slides offers a digest of the concepts of dependability and security and of their interrelationships.

5. Distributed Systems:

Slides for DISTRIBUTED SYSTEMS (in Spanish) R. Jimenez-Peris:
http://lsd.ls.fi.upm.es/lsd/education/sistemas-distribuidos-fundamentos-y-tecnologia-libre-eleccion-ingenieria-distributed-systems-master-level/
This set of slides covers aspects of distributed computing mainly dedicated to interactive consistency, agreement algorithms, transaction processing, replication as well as presenting several case studies.

Slides for DISTRIBUTED SYSTEMS R. Baldoni:
http://www.dis.uniroma1.it/%7Ebaldoni/SD.html
This set of slides covers the following topics: computational models, point-to-point communication channels, failure detectors, time and distributed algorithms, consensus, consistency and broadcast algorithms, software replication techniques, virtual synchrony.

Slides for DISTRIBUTED SYSTEM PLATFORMS R. Beraldi:
http://www.dis.uniroma1.it/~beraldi/PSD0708/
This set of slides covers the following topics: communication in distributed systems, introduction to CORBA , CORBA IDL, introduction to objects, publisher/subscriber systems, Java RMI, ambient computing.

Slides for MIDDLEWARE SYSTEMS F. Panieri:
http://courses.web.cs.unibo.it/SistemiMiddleware/MaterialeDiRiferimento
Access only through username and password

Slides for DISTRIBUTED SYSTEMS N. Suri:
http://www.deeds.informatik.tu-darmstadt.de/teaching/index.html
The site of TU Darmstadt contains many slides related to courses offered during the semesters of different years.

Book DISTRIBUTED SYSTEMS P. Verissimo, L. Rodrigues:
http://www.navigators.di.fc.ul.pt/dssa/
This link points to the book that the authors use for their teaching and contains indirect pointers to interesting sites.

Slides for DISTRIBUTED FAULT-TOLERANT ALGORITHMS R. Guerraoui:
http://www.di.fc.ul.pt/~ler/irdp/teaching.htm
This set of slides is mainly dedicated to extensive analysis of reliable, causal and total order broadcasts, as well as consensus and group membership and shared memory.

Slides for DISTRIBUTED FAULT-TOLERANT AND SECURE ALGORITHMS C. Cachin:
http://www.zurich.ibm.com/~cca/sft06/
These slides contain the lectures given at ETH Zurich on Security and Fault-Tolerance in Distributed Systems.

Slides for DISTRIBUTED SYSTEMS G. Couloris, J. Dollimore, T. Kindberg:
http://www.dcs.qmw.ac.uk/research/distrib/dsbook/
This site is based on the book **Distribute Systems: Concept and Design,** Addison-Wesley, 1994 by G. Couloris, J. Dollimore, T. Kindberg. The interesting thing is that the Instructors Guide, in addition to slides related to the book, includes a line of use of the material (Presentation points) that discuss the objectives of section, the points to emphasize, the possible difficulties and the teaching hints.

6. Software Reliability Engineering:

Slides for SOFTWARE RELIABILITY ENGINEERING J. Musa:
http://openseminar.org/se/courses/41/modules/206/index/screen.do
Reliability is the probability that a system functions correctly, in a given environment over a given amount of time. Some aspects of reliability are availability, correctness, safety, operational usability, etc. Software reliability engineering is the study of operational profiles to determine the best areas to test by estimating the components that will be used the most by the customer and testing those components first.

There are several phases of software reliability engineering. The first is to define the product by looking at the customers, users, external suppliers, and certifying external software. The second step is to implement the operational profiles, which involves looking at system components and determining the probability of their use. Third, the failure intensity object is determined. Fourth, the test cases are created that fit the operation profile and are within the bounds of the failure intensity object. Lastly, reliability data is tracked so that future estimates will be more accurate.
The set of slides and presentations covers all topics listed earlier.

Book HANDBOOK OF SOFTWARE RELIABILITY ENGINEERING M. Lyu:
http://www.cse.cuhk.edu.hk/~lyu/book/reliability/index.html
This site contains the downloadable book for personal use (see copyright notice)

7. Security:

Slides for SECURITY L. Williams, S. Smith:
http://openseminar.org/se/modules/44/index/screen.do
Software security is important, especially in distributed systems, to protect the integrity of the information stored and used by a software system. Questions about security requirements should be asked of the customers during requirements solicitation. The best practice for providing a secure system to customers is to ensure that security concerns are provided for upfront, and during all phases of development. Adding security features to an application, after development of the main functionality is complete, is difficult.
The set of slides and presentations covers all topics listed earlier.

Slides for SECURITY AND FAULT TOLERANCE C. Cachin:
http://www.zurich.ibm.com/~cca/sft06/
http://www.zurich.ibm.com/~cca/sft08/
According to Lamport, a distributed system is one where the crash of a computer that you've never heard of stops you from getting any work done. This course presents methods for building dependable and secure distributed systems. The emphasis is on fault-tolerant and distributed cryptographic protocols. Topics include group communication, failure detectors, reliable broadcast protocols, distributed cryptography, threshold cryptosystems, Byzantine agreement, quorum systems, and replication. Applications to cluster computing, Internet services, and storage systems are also presented.
The course introduces principles and fundamental methods, and shows how they are applied to real-world systems.

Book DEFENCE IN DEPTH: FOUNDATIONS FOR SECURE AND RESILIENT IT ENTERPRISES C.J. May, J. Hammerstein, J. Mattson, K. Rush:
http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06hb003.pdf
This site contains the downloadable book for personal use (see copyright notice)

Book SECURITY ENGINEERING R. Anderson:
http://www.cl.cam.ac.uk/~rja14/book.html
This site contains the downloadable book for personal use (see copyright notice)

Slides for DEPENDABILITY AND SECURITY JC. Fabre
http://www.laas.fr/TSF/courses/SEC2007-slides.pdf
This set of slides offers a digest of the concepts of dependability and security and of their interrelationships.

Slides for SECURITY in WIRELESS NETWORKS L. Buttyan, J.P. Hubaux
http://secowinet.epfl.ch/index.php?page=slideshow.html
This set of slides is derived by the book of the same authors Security and Cooperation in Wireless Networks, Cambridge University Press, ISBN 9780521873710. Please note copyright notice.

8. Privacy:

Slides for PRIVACY L. Williams, S. Smith:
http://openseminar.org/se/modules/45/index/screen.do

Privacy concerns can arise due to security problems or questions about why certain information is gathered by a website or application. Privacy issues for users of all levels must also be addressed as early in development as the requirements phase, with an emphasis on whom has access to certain information and what will be done with the information. A system should be created that only stores the minimal amount of personal information about a person required to complete a given function.
The set of slides and presentations covers all topics listed earlier.

9. HCI and Interactive Systems:

Slides for HUMAN-MACHINE INTERACTION (in French) P. Palanque:
http://resist.ecs.soton.ac.uk/wiki/images/1/10/designrationale.pdf
http://liihs.irit.fr/palanque/Ps/MasterHM-IntroHCIPalanque.pdf
Learn how to systematically consider various options while analyzing, designing and implementing interactive systems.
Learn the underlying concepts of rationalizing and structuring argumentation during the various phases of the development process of interactive systems
Learn how to organize models and designs in order to provide traceability of the analysis, design and implementation choices.

Slides for INTERACTIVE SYSTEMS (in French) P. Palanque:
http://resist.ecs.soton.ac.uk/wiki/images/b/b9/interactivesystemsengineering.pdf
Learn the specificities of the development of interactive systems. These specificities are explained with respect to the development process, the notation and the implementation.
Basic introduction to Visual Basic 6.0 is given as it proves to be a very efficient tool for high fidelity prototyping of standard interactive systems.

Slides for HUMAN-MACHINE INTERACTION ACM SIG on HCI:
http://sigchi.org/cdg/index.html
These slides gather 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.

Slides for HUMAN INTERFACES and USABILITY U. Texas at San Antonio:
http://vip.cs.utsa.edu/classes/cs6693s2006/lectures/index.html
The slides contain a set of lectures held at UTSA.

Slides for HUMAN-COMPUTER INTERACTION A. Dix, J. Finlay, G. Abowd, R. Beale:
http://www.hcibook.com/e3/resources/
This site contains the slides related to the content of the book Human-Computer Interaction, Prentice-Hall, 2004 by Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale

Slides for PERVASIVE HUMAN CENTRIC COMPUTING MIT OpenCourseware:
http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-883Spring-2006/CourseHome/index.htm
This course is broad, covering a wide range of topics that have to do with the post-pc era of computing. It is a hands-on project course that also includes some foundational subjects. Students will program iPAQ handheld computers, cell phones (series 60 phones), speech processing, vision, Cricket location systems, GPS, and more. Most of

the programming will be using Python®, but Python® can be learned and mastered during the course.
This course features a partial set of lecture notes and a list of readings. In addition, sample quiz questions are available in the exams section.

Book FAILURE IN SAFETY-CRITICAL SYSTEMS: A HANDBOOK OF ACCIDENT AND INCIDENT REPORTING C. Johnson:
http://www.dcs.gla.ac.uk/~johnson/book/
This site contains the downloadable book for personal use (see copyright notice)

10. General Topics:

From the Computer Science Teaching Centre:
http://csta.villanova.edu/CITIDEL/
CITIDEL, the Computing and Information Technology Interactive Digital Educational Library, now accesses over 480,000 entries. CSTC is one of CITIDEL's source collections. CITIDEL is the computing educational portal to the NSDL, the National Science Digital Library. It contains a huge repository of reviewed teaching material on all aspects of Computer Science, including topics related to resilient computing. It may be searched by alphabetical listing, by titles, by authors, by subjects, and by date.

Videos from UC Berkeley:
http://video.google.com/ucberkeley.html
This site contains a set of videos of University of Berkeley. They cover many aspects of Science and Humanities. Of interest for this report what is presented at:
http://video.google.com/videosearch?q=owner%3Aucberkeley+is141&page=1&so=1 on Information Systems.

11. Testing, Verification and Validation:

Slides for COMPUTER-AIDED VERIFICATION R. Alur:
http://www.cis.upenn.edu/cis673/
The course introduces the theory and practice of formal methods for the design and analysis of concurrent and embedded systems. The emphasis is on the underlying logical and automata-theoretic concepts, the algorithmic solutions, and heuristics to cope with the high computational complexity. Topics include:
Models and semantics of reactive systems: states vs. events, nondeterminism vs. concurrency, synchrony vs. asynchrony, safety vs. liveness, refinement preorders, real-time and hybrid systems, open systems.
Verification algorithms: temporal logic model checking, theory of omega automata, games, minimization.
Verification techniques: symbolic model checking, on-the-fly model checking, state space reduction methods, compositional and hierarchical reasoning, abstraction and refinement.

Book COMPUTER-AIDED VERIFICATION R. Alur, T.A. Henzinger:
http://www.cis.upenn.edu/~alur/CIS673/index.html
This site contains the downloadable book for personal use (see copyright notice)

Slides on Topics on VERIFICATION A. Pnueli:
http://www.wisdom.weizmann.ac.il/~amir/
This course concentrates on methods for the verification of reactive systems. The course starts by reviewing the basic models and algorithmic processes for model

checking of reactive systems.
Next, the course extends the models and verification techniques to handle infinite-state systems, using the techniques of deductive verification.

Slides for DEDUCTIVE VERIFICATION OF REACTIVE SYSTEMS A. Pnueli:
http://www.wisdom.weizmann.ac.il/~amir/Course02a/header.html
This course will present methods for the deductive verification of reactive systems, i.e., proving that reactive systems satisfy their specification, given by temporal logic formulas, using deductive (i.e., theorem proving) methods.
Unlike model checking, deductive verification is not restricted to finite-state systems and can handle unrestricted programs with rich data-structures. On the other hand, these techniques are not fully automatic and require user interaction and ingenuity in the design of auxiliary constructs such as invariants and ranking functions and, at a later stage, the effective guidance of a theorem-proving tool.

Slides for TEST AND VERIFICATION E. Fleury et al.:
http://www.cs.auc.dk/~kgl/TOV04/Plan.html
The focus of this course is on techniques and software-tools that can be used to assess the quality and correctness of software systems. The first part of the course considers tools and techniques for formal verification of system designs. The last part of the course considers tools and techniques for testing system implementations.

Book CAUSAL SYSTEM ANALYSIS P. Ladkin:
http://www.rvs.uni-bielefeld.de/publications/books/ComputerSafetyBook/index.html
This site contains the downloadable book for personal use (see copyright notice)

Book CONCEPTS, ALGORITHMS, AND TOOLS FOR MODEL CHECKING J-P. Katoen:
http://www.cs.aau.dk/~kgl/VERIFICATION99/katoen2.ps
This site contains the downloadable book for personal use (see copyright notice)

Slides for VALIDATION AND VERIFICATION J. P. Bowen:
http://www.cs.ucl.ac.uk/staff/J.Bowen/GS03/
The course will train students in the principles and techniques of validating and verifying complex software systems. The training will be at an intellectually demanding level and will cover not only the state-of-the practice in validation and verification, but also the most significant trends, problems and results in validation and verification research.
These slides concentrate mainly on testing aspects and on formal methods aspects.

Slides for THEOREM PROVING AND MODEL CHECKING IN PVS E. Clarke, D. Kroening:
http://www.cs.cmu.edu/~emc/15-820A/
This course will cover a number of techniques that have proven to be useful in verifying software and hardware systems including temporal logic, model checking, BDDs, fast SAT procedures, and theorem proving. The focus of the course will be on the PVS theorem prover / model checker developed at SRI (and widely regarded as the most powerful tool in its class). Students will learn how to use PVS for hardware and software verification. Some of the basic decision procedures used in PVS for theorem proving and model checking will be presented. The course will conclude

with a discussion of the limitations of PVS for software and hardware verification and how a tool might be constructed that would be more powerful.

Slides for SYSTEM VALIDATION T. C. Ruys:
http://fmt.cs.utwente.nl/courses/systemvalidation/
The complexity of software and hardware systems is rapidly increasing and so is their vulnerability with respect to errors. This course is focused on validation techniques and methods to assess the correctness and dependability of information- and communication technology (ICT) systems.
The emphasis is on model checking, an automated and very successful validation technique. Close attention will be paid to the model checker SPIN. The central theme of this course is model-checking algorithms for the automated analysis of concurrent software. Besides the basic algorithmic principles, considerable attention will be devoted to optimization techniques that make model checking a verification approach of industrial relevance. Finally, algorithms and software tools will be treated that are aimed at the verification of Java and C++ programs.

Slides for ABSTRACT INTERPRETATION P Cousot, J. C. Hunsaker
http://web.mit.edu/afs/athena.mit.edu/course/16/16.399/www
Abstract Interpretation is a theory of approximation of mathematical structures, in particular those involved in the semantic models of computer systems. Abstract interpretation can be applied to the systematic construction of methods and effective algorithms to approximate undecidable or very complex problems in computer science such that the semantics, the proof, the static analysis, the verification, the safety and the security of software or hardware computer systems. In particular, abstract interpretation-based static analysis, which automatically infers dynamic properties of computer systems, has been very successful these last years to automatically verify complex properties of real-time, safety critical, embedded systems.
The course is an introduction to abstract interpretation with application to static analysis (the automatic, compile-time determination of run-time properties of programs) and software verification (conformance to a specification).

Slides for ABSTRACT INTERPRETATION David Schmidt :
http://santos.cis.ksu.edu/schmidt/Escuela03/home.html
A *static analysis* finitely analyzes a program in advance of the program's execution and extracts information useful for program transformation or program-correctness proofs. *Abstract interpretation* is a foundational framework that can justify the correctness of a static analysis. Many elegant static analyses are synthesized from the abstract interpretations that justify their correctness. This series of lectures introduces abstract interpretation and applies it to static analysis. Standard formats of static analysis are presented, and abstract interpretations are used to synthesize both abstract-operational- and abstract-denotational-semantics definitions and as well as logics and models for temporal-logic model checking.

Slides for SOFTWARE TESTING P. Ammann, J. Offutt:
http://cs.gmu.edu/~offutt/softwaretest/powerpoint/
This course is about concepts and techniques for testing software and assuring its quality. Topics cover software testing at the unit, module, subsystem, and system levels, automatic and manual techniques for generating and validating test data, testing process, static vs. dynamic analysis, functional testing, inspections, and reliability assessment.

The course makes it explicit that all test approaches fundamentally rely on four categories of models to be covered: graphs, logical expressions, input domain characterization and syntactic structures.

12. Real-Time Systems:

Slides for OPERATING SYSTEMS CONCEPTS Avi Silberschatz, Peter Baer Galvin, Greg Gagne :
http://www.os-book.com/
This set of slides covers the full topic of Operating Systems with a large part dedicated to real-time systems.

Slides for REAL-TIME SYSTEMS (in Italian) G. Buttazzo:
http://feanor.sssup.it/~giorgio/srt.html
This set of lectures introduces the topic of real-time systems and the problems related to scheduling algorithms.

Slides for REAL-TIME SYSTEMS (in French) I. Puaut:
http://www.irisa.fr/caps/people/puaut/puaut.html
These slides contain a generic course on real-time systems.

13. Mathematical Programming and Operations Research:

Slides for INTRODUCTION TO MATHEMATICAL PROGRAMMING T. Ralphs:
http://www.lehigh.edu/~tkr2/teaching/ie406/
This courseware from Ted Ralphs of Lehigh University offers a comprehensive material for optimization problems.

Slides for OPERATIONS RESEARCH from tutOR U. Melbourne:
http://www.tutor.ms.unimelb.edu.au/frame.html
This site contains a collection of web-based, online tutorial modules.

14. Graph Theory:

Book GRAPH ALGORITHMS S. Even:
http://www.wisdom.weizmann.ac.il/~oded/even-alg.html
This site contains downloadable excerpts of the book for personal use (see copyright notice)

15. Stochastic Network Optimization M.J. Neely:

http://www-rcf.usc.edu/~mjneely/stochastic/
The page links to papers in the area, and is intended as a resource for researchers and practitioners.

16. Pattern Recognition Resources B. Fisher:

http://homepages.inf.ed.ac.uk/rbf/IAPR/
This web site contains links to resources that can support students, researchers and staff. The most important resources are for students, researchers and educators. These include lists with URLs to:

- Tutorials and surveys
- Explanatory text
- Online demos
- Datasets

- Book lists
- Free code
- Course notes
- Lecture slides
- Course reading lists
- Coursework/homework
- A list of course web pages at many universities