MAP-I Joint Doctoral Program 2008/09

“Architecture and Design of Large-Scale Software Systems”

Course Proposal

by

Ademar Aguiar, Universidade do Porto
Ricardo J. Machado, Universidade do Minho
Luís Almeida, Universidade de Aveiro

A. Course Description

1. Theme, importance and contextualization

[scientific areas, similar courses in equivalent international teaching programmes]

The architecture of a software system describes the global structure in terms of its components, external properties and its interrelations. As software systems grow in scale and complexity, it becomes increasingly more important to understand them at many abstraction levels other than algorithms, functions, objects or components, and by different kinds of people, such as procurers, acquirers, producers, integrators, trainers, and users.

Defining the right architecture at its different perspectives (logical, physical and technological) is crucial for the successful development, operation, and evolution of medium and large-scale software systems, but it will be even more important in the future with the advent of ultra-large systems.

In addition to the scale factor, the complexity due to the wide variety of computing platforms and software technologies that can be used to create applications also raises the difficulties and importance of designing software systems. This diversity is increasingly introducing new challenges due mainly to the physical integration and spontaneous interoperation, quantity and heterogeneity of computing devices, services and applications that may be part at any moment of a large-scale software system. The complexity lies not in the individual components but in the integration of those components to create a functioning system that ensures certain non-functional requirements, not only at the moment of its initial design but also during all the expected lifetime of the system, whose control is typically no more centralized. Examples of related application domains include ambient intelligence, e-Applications, and mobile services. Global markets and world-wide actuation also demands the deployment of large-scale enterprise (software) architectures as technological solutions to support business information systems.

Architectural models for such large-scale systems must be tailored to allow the dynamic construction and allocation of customized applications to heterogeneous computing devices, with different computational or interface capabilities. Therefore, for highly complex and large-scale software systems, many design and architectural challenges must be addressed. Many of these are software engineering challenges that must take into consideration aspects not only related with individual computing devices, but also with the entire system obtained from the cooperation of diverse, dispersed, integrated or mobile computing devices that in conjunction contribute to the achievement of the overall system objectives.

General purpose of the proposed course

This course aims to provide students with an appreciation of current research issues on architecture and design of software systems and give them hands-on design experience. The course features a group project that reinforces the readings and lectures on seminal work and current research topics. The project also teaches students about the design process in general, including refining a specification, partitioning functionality, creating interfaces between subsystems, working in teams, and planning.

Considering the methodological emphasis of the course, it is intended that it will be classified as an UCPC curricular unit within the PhD MAP-I curriculum, since it promotes a methodological insight into aspects of software engineering directed at developing large, complex applications in a manner that reduces development costs, increases the potential for commonality among different members of a closely related product family, and facilitates evolution, possibly at system runtime, instead of a particular approach to a well defined computing technology or set of tangible tools.

In addition, it is important to mention that this course proposal is basically the same as last year's course on the same name, certified by CMU, with only a few refinements, namely to address better the main theme of MAP-i 2008/09's edition.

Scientific areas

This course is compliant with the following categories of the ACM Computing Classification System:

- D.2.11 [Software Engineering]: Software Architectures
- D.2.13 [Software Engineering]: Reusable Software --- Domain engineering
- D.2.13 [Software Engineering]: Reusable Software --- Reuse models
- K.6.3 [Computing Milieux]: Software Management --- Software development
- K.6.3 [Computing Milieux]: Software Management --- Software process
- D.2.2 [Software Engineering]: Design Tools and Techniques.
- General Terms: Design, Languages, Standardization.

References to similar courses

Other courses also addressing the issues of software architecture and design with a focus on large-scale/complex software systems are listed next. These courses are, generally, part of both Master and Doctoral programs thus having some concern with open research topics. Despite the specific differences, as discussed later, they address essentially the same core topic covered by the proposed course, corroborating its importance and timeliness.

CMU "Architectures for Software Systems" (CMU/ASS)[1] deserves a particular reference in this proposal since it was used as a starting point for the development of this course and another one with the same name **"Arquitectura de Sistemas de Software" (FEUP/ASS**[2]) being offered at the Master of Sciences in Informatics Engineering, FEUP, University of Porto, since 2003. This course now being proposed aims to leverage the experience gained with FEUP/ASS by adding new and more advanced parts, both to CMU/ASS and FEUP/ASS courses, namely the large variety of adopted topics and the case study selected, with the goal of addressing new issues related with the design and architecture of software systems, and therefore to motivate and promote advanced research work and doctoral theses on the field.

Other courses also analysed by the authors of this proposal are next identified:

- TCSS 360: Software Development and Quality Assurance Techniques, University of Washington, USA, http://faculty.washington.edu/hanks/Courses/360/a06/.
- SWE 621: Software Modelling and Architectural Design, George Mason University, USA, http://mason.gmu.edu/~hgomaa/swe621/SWE621home.htm.
- SWE 641/SYST 621 Systems Architecture for Large-Scale Systems, George Mason University, USA, http://www.gmu.edu/departments/seor/syllabi/02B/syst621.htm.
- 6.170: Laboratory in Software Engineering, MIT, USA, http://www.mit.edu/~6.170/
- INFOSWA: Software Architecture, Utrecht University, Netherlands, http://www.cs.uu.nl/wiki/Swa/EducationPage.

Most courses involve hands-on experience with some of the technologies and applications of software engineering. There are concerns on how to structure projects to ensure successful learning outcomes for the students. As some critical technologies become more mainstream (meta-CASE tools, frameworks, integrated development environments and automated code generators), it is easier to provide relevant infrastructures for students to build projects.

The extent to which students are left to design as well as implementation varies. Some courses consider the methodological-centered design and prototyping as a critical component of a project,

---

[1] CMU, Master, 17-655: Architectures for Software Systems

[2] FEUP, Master, ASS: Arquitectura de Sistemas de Software

even more important than building a real system. This kind of project encourages creative application of methodologies, even when some technologies are not available. Other courses emphasize the construction with real technologies, to the extent that methodological design is partially ignored. While these projects often may not provide compelling uses of technology, they teach students the importance of integration. We are not judging which is right or wrong. Both approaches have strengths and limitations, and the choice of the kind of project is based on the students' background. However, our perspective is to favor a more methodological approach although with some technological insight.

2. Objectives

Large-scale software systems impose several challenges in terms of design and evolution, namely:

- how to economically align requirements with design architectures and industry structures;
- how to introduce agility (or, flexibility, auto-adaptability) to respond effectively to changes in requirements, system configuration, system environment;
- how to minimize the effort needed to integrate components built independently by different teams, vendors, with different goals, and at different times;
- how to design a system that is easy to be evolved by users in order to provide them unplanned capabilities;
- how to organize development to yield high-quality designs.

In order to help cope with these and other challenges of software systems in terms of design and evolution, especially large-scale ones, this course sets the following objectives:

- to introduce the fundamental concepts underlying the fields of architecture, design, construction and integration of large-software systems;
- to teach how to specify architectural non-functional requirements and quality attributes;
- to teach how to design, understand and evaluate software systems architectures both at the macro-architecture and micro-architecture levels of abstraction to satisfy a set of requirements;
- to present the best practices and best solutions known to design and build high-quality large -scale software systems;
- to familiarize the students with the fundamental concepts of software architecture, the properties and applicability of the different architecture styles, the most popular architectural and design patterns, software components, reusable architectures and the relations of all these concepts with software reuse.

3. Learning results

Pre-requisites for this course include:

- programming ability in an object-oriented programming language such as C++ or Java;
- a basic understanding of software engineering principles;
- and maturity to undertake substantial design projects in a team.

Students are expected to have previously taken undergraduate courses on classical programming paradigms and fundamentals of software design.

Several development challenges should be considered to master the design and architecture of large-scale software systems. After completing this course, students will be able:

- to describe the main architecture styles of software systems;
- to make a precise description of an architecture;

- to idealize different alternative architectures to solve the same problem and evaluate (justifying) which is the best one in terms of design quality;
- to recognize and understand several architectural and design patterns;
- to understand and apply several techniques for reuse of design and architecture knowledge;
- to adopt improved accuracy specifications and notation capabilities to semantically manipulate architectural models towards model execution or model centric development;
- to instantiate patterns and adopt separation of concerns techniques to enhance reuse of architectural models and codifying best practices;
- to configure best practices of software referential design processes to support the development of large-scale software solutions;
- to use development tools to make the previous tasks more efficient;
- to know some technologies to help realize the architectures specified.

4. Course contents

1. INTRODUCTION
    1.1. Software design: fundamental concepts and principles.
    1.2. Software architecture: definitions, concepts, components, connectors, views, quality attributes.
    1.3. Examples of architectures at micro- and macro- levels: design patterns, frameworks, and production lines.
    1.4. Design flow for large-scale software systems.
2. ARCHITECTURE STYLES AND PATTERNS
    2.1. Introduction to architectural styles, reference models and reference architectures.
    2.2. Architectural styles: pipes and filters, data abstraction, object-orientation, event-based systems, layered systems, repositories, interpreters, process-control systems.
    2.3. Software patterns: origins, notion of patterns and pattern languages, kinds of patterns (architectural, design, others).
    2.4. Classic catalogs of patterns: POSA architectural patterns, Gang-of-Four design patterns.
    2.5. Thematic catalogs of patterns: patterns of enterprise application architecture, patterns for enterprise integration.
3. QUALITY ASPECTS AND ATTRIBUTES
    3.1. A quick look at software quality: a corporate perspective.
    3.2. Software quality attributes (performance, dependability, interoperability, maintainability, usability...).
    3.3. Quality attributes and software architectures.
    3.4. Conflicting attributes and design trade-offs.
4. DESIGN AND REFINEMENT OF SOFTWARE ARCHITECTURES
    4.1. UML 2.0 synthesis and meta-modeling capabilities relevant to represent architectural models: superstructure, infrastructure, OCL, MOF.
    4.2. Representation and documentation of software architectures.
    4.3. Variability analysis and software product lines: domain engineering, domain specific languages.
    4.4. Software Process: RUP for Large Enterprises, EUP.
    4.5. Enterprise Software Architectures: EABOK, TOGAF.

5. DEVELOPMENT TECHNOLOGIES?
    5.1. Architectural composition with Eclipse.
    5.2. Middleware for highly distributed (ubiquitous) systems.

5. Teaching methods

The best way to understand and master software design and software architecture is to experience it. In the educational setting, this means:

- learning the fundamental concepts and principles;
- knowing and understanding the solutions and practices proven to be the best, through the exploration of specific examples from the past, so-called case studies;
- and applying the knowledge acquired by imitating and adapting known solutions to a specific problem through hands-on development of a software system, in an individual project.

No textbook adequately covers the course's range of topics, so it will be used a diversity of bibliographic elements (books, journals and conference proceedings).

Readings

All reading assignments come from journals and conference proceedings. Each week, the students must read papers or some few supplemental readings provided. This exposes many students to extensive readings from the research literature for the first time. To help them with their reading, we require them to write a brief summary for each paper, submitted via email at the beginning of the week. We also ask them to submit a list of questions about the readings, which we try to work into the lecture if possible.

During the last few weeks of the course, we no longer require reading summaries, to give students more time to focus on the project.

Classes

The class meetings are meant to be conversational, and we encourage students to ask questions and make comments. Consequently, the discussion may follow tangents to the prepared lecture, but they should be fruitful, informative, and thought provoking.

Individual research project

For the individual research project, we base the grade on an oral presentation (for a more methodological project) or a demonstration (for a more technological project), and a final written report. A few weeks into the course, we hand out descriptions of possible projects. The students have a week to look over the project descriptions before chosing one of them. No two students can work on the same project. We make the project descriptions intentionally vague, since it gives them considerable leeway in making design decisions. Having too specific descriptions would force students down a design path that they might not choose on their own.

Once students complete their project, they must demonstrate it, make an oral presentation, and submit a final written report. The report has two major pieces: first, the discussion of the project's major design decisions and trade-offs; second, one section entitled "If I could do it all over again..." describing what they would do differently if they could have a second chance to start from the beginning.

Lectures Plan

WEEK	TOPIC	SUBTOPIC	READINGS	LECTURER
1	Introduction		[Garlan and Shaw, 1994]	AMA, LDA, RJM
2	Architecture Styles and Patterns	Architectural styles: pipes and filters, data abstraction, object-orientation, event-based systems, layered systems, repositories, interpreters, process-control systems	[Shaw and Garlan, 1996]	AMA
3	Architecture Styles and Patterns	Software patterns: origins, essentials, concepts and terminology. Classic examples of patterns: POSA and GoF catalogues.	[Buschmann et al., 1996] [Gamma et al., 1995]	AMA
4	Architecture Styles and Patterns	Thematic examples of patterns: Patterns of Enterprise Application Architecture, Patterns for Enterprise Integration	[Fowler, 2002] [Hohpe et al, 2003]	AMA
5	Quality Aspects and Attributes	Identification of quality attributes, evaluation of software architectures.	[Reagan, 2002]	LDA
6	Quality Aspects and Attributes	Non-functional attributes, conflicting attributes, general taxonomy.	[Barbaci et al., 1995] [Clements et al., 1995]	LDA
7	Design and Refinement of Software Architectures	UML 2.0 synthesis and meta-modelling capabilities. Variability Analysis and Software Product Lines.	[Gomaa, 2005] [Medvidovic et al., 2002]	RJM
8	Design and Refinement of Software Architectures	Software Process	[Leffingwell, 2008] [Ambler et al., 2008]	RJM
9	Design and Refinement of Software Architectures	Representation and Documentation of software architectures. Paradigms for composing highly flexible software systems	[Smith and Williams, 2002] [D'Souza et al., 1999]	AMA
10	Design and Refinement of Software Architectures	Enterprise Software Architectures	[EABOK] [TOGAF]	RJM
11	Development Technologies	Architectural composition with Eclipse.		AMA
12	Development Technologies	Middleware for highly distributed (ubiquitous) systems.	[RUNE, S2005]	LDA

Lecturers: Ademar Aguiar (AMA), Luís Almeida (LDA), Ricardo J. Machado (RJM).

References for suggested readings

[ShawGarlan1996] Mary Shaw, David Garlan, “Software Architecture: Perspectives on an Emerging Discipline”, Prentice Hall, 1996, ISBN 0-131-82957-2.

[BassEtAl2003] Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice”, 2nd edition, Addison-Wesley, 2003, ISBN 0-321-15495-9.

[POSABook1] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, “Pattern-Oriented Software Architecture, Volume 1: A System of Patterns”, John Wiley & Sons, 1996, ISBN 0-471-95869-7.

[GoFBook1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, “Design Patterns - Elements of Reusable Object-Oriented Software”, Addison Wesley, 1995, ISBN 0-201-63361-2.

[Fowler, 2002] Martin Fowler, “Patterns of Enterprise Application Architecture”. Addison Wesley, 2002.

[Hohpe et al, 2003] Gregor Hohpe, Bobby Woolf, “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”. Addison Wesley, 2003.

[BarbEtAl1995] Mario Barbaci, Thomas H. Longstaff, Mark H. Klein, Charles B. Weinstock, “Quality Attributes”, Carnegie Mellon University/Software Engineering Institute, Technical Report CMU/SEI-95-TR-021, December 1995.

[Regan2002] Gerard O'Regan, “A Practical Approach to Software Quality”, Springer-Verlag, 2002, ISBN 0-387-95321-3.

[ClemEtAl95] Clements, P., Bass, L., Kazman, R., Abowd, G. “Predicting Software Quality by Architecture-Level Evaluation,” 485-497. Proceedings of the 5th International Conference on Software Quality, Austin, Texas, October 23-26, 1995. Milwaukee, Wi.: American Society for Quality Control, 1995.

[RUNES2005] FP6 IP "RUNES, Deliverable D5.1 “Survey of Middleware for Networked Embedded Systems”, available on-line at http://www.ist-runes.org/docs/deliverables/D5_01.pdf. January 2005.

[Gomaa, 2005] Hassan Gomaa. “Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures”. Addison Wesley, 2005.

[Douglass, 2003] Bruce Powel Douglass. “Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems”. Addison Wesley, 2003.

[Smith and Williams, 2002] Connie U. Smith, Lloyd G. Williams. “Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software”. Addison Wesley, 2002.

[D'Souza et al., 1999] Desmond Francis D'Souza, Alan Cameron Wills. “Objects, Components, and Frameworks with UML: The Catalysis(SM) Approach”. Addison Wesley, 1999.

[Leffingwell, 2008] Dean Leffingwell. “Scaling Software Agility: Best Practices for Large Enterprises”. Addison Wesley, 2008.

[Ambler et al., 2008] Scott W. Ambler, John Nalbone, Michael J. Vizdos. “The Enterprise Unified Process: Extending the Rational Unified Process”. Addison Wesley, 2008.

[EABOK] “Guide to the (Evolving) Enterprise Architecture Body of Knowledge” URL: http://www.mitre.org/work/tech_papers/tech_papers_04/04_0104/04_0104.pdf

[TOGAF] “The Open Group Architecture Framework Version 8.1.1”, Enterprise Edition URL: https://store.opengroup.org/catalog/product_info.php?products_id=51

6. Evaluation system

The student’s grade for the course is based on an individual research project (65%), an examination (20%), and homework and reading summaries (15%). The individual research report is a survey paper that targets the class as its audience. We encourage students to find a topic related to their own research; for example, some students can use it to find background material for an eventual thesis, if they want to develop one in this field of research.

7. Bibliography

1. [Shaw and Garlan, 1996] Mary Shaw, David Garlan. Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall, 1996.

2. [Bass et al., 2003] Len Bass, Paul Clements, Rick Kazman. “Software Architecture in Practice”, 2nd edition, Addison-Wesley, 2003.
3. [Buschmann et al., 1996] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal. ”Pattern-Oriented Software Architecture, Volume 1: A System of Patterns”. John Wiley & Sons, 1996.
4. [Gamma et al., 1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. “Design Patterns - Elements of Reusable Object-Oriented Software”. Addison Wesley, 1995.
5. [Fowler, 2002] Martin Fowler, “Patterns of Enterprise Application Architecture”. Addison Wesley, 2002.
6. [Hohpe et al, 2003] Gregor Hohpe, Bobby Woolf, “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”. Addison Wesley, 2003.
7. [Barbaci et al., 1995] Mario Barbaci, Thomas H. Longstaff, Mark H. Klein, Charles B. Weinstock. “Quality Attributes”. Carnegie Mellon University/Software Engineering Institute, Technical Report CMU/SEI-95-TR-021, December 1995.
8. [Regan, 2002] Gerard O'Regan. “A Practical Approach to Software Quality”. Springer-Verlag, 2002.
9. [Gomaa, 2005] Hassan Gomaa. “Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures”. Addison Wesley, 2005.
10. [Douglass, 2003] Bruce Powel Douglass. “Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems”. Addison Wesley, 2003.
11. [Smith and Williams, 2002] Connie U. Smith, Lloyd G. Williams. “Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software”. Addison Wesley, 2002.
12. [D'Souza et al., 1999] Desmond Francis D'Souza, Alan Cameron Wills. “Objects, Components, and Frameworks with UML: The Catalysis(SM) Approach”. Addison Wesley, 1999.

B. Lecturers

1. Teaching staff and respective expertise

The course proposed in this document results from the integration of the complementary research interests and expertise of the three authors. It benefits from their individual scientific contributions, making available to the students their research experience in well defined topics within the area of architecture and design of software systems, namely:

- software patterns and frameworks (Ademar Aguiar);
- modelling and refinement of software architectures (Ricardo J. Machado);
- non-functional aspects in software architectures (Luís Almeida).

Such a cooperation in lecturing this course is, to the authors opinion, the most adequate way of coping with the diversity of methods and techniques needed to allow a proper approach to the architecture and design of large-scale software systems. All the three authors present a considerable research achievement at international level, as demonstrated by their short CVs, which is a guarantee of adequate research orientation, a fundamental aspect for a cutting-edge course in a Doctoral program.

2. Short CVs

In the following we include the short CVs of the authors of this course proposal, as of March 2007.

Ademar Aguiar

Education:

- PhD and MSc degrees in Informatics and Computer Engineering (FEUP, 2004 and 1994).
- DEng degree in Electronics and Computer Engineering (FEUP, 1989).

Research interests:

- Software architecture: application frameworks (object-oriented and aspect-oriented); patterns of software; component-based software development.
- Software engineering: agile processes, agile documentation, wiki-based software documentation, object-oriented analysis and design, aspect-oriented software development, software engineering tools.

Current professional situation:

- Assistant Professor at the Faculdade de Engenharia da Universidade do Porto
- Researcher of INESC Porto

Current teaching responsibilities:

- Research Topics in Software Engineering, ProDEI – Doctoral Programme on Informatics Engineering, FEUP
- Architecture of Software Systems, MEI and MIEIC Masters on Informatics Engineering, FEUP
- Agile Methods, MEI and MIEIC Masters on Informatics Engineering, FEUP
- Software Engineering Laboratory, MIEIC Master on Informatics Engineering, FEUP
- Software Engineering, MIEEC Master on Electrical and Computer Engineering, FEUP

Current scientific responsibilities:

- Scientific coordinator of project “Doc-It! “, FEUP/DEEC
- Supervisor of 3 PhD and 9 MSc students (3 MSc students already concluded)

PhD students:

- Artur Jorge da Silva Rocha, Enabling Cooperative Distributed Support for GI-based Decision Making, since 2005
- André Oliveira Restivo, A Disciplined Approach to Composition of Aspects, since 2006
- Nuno Honório Flores, Improving Framework Understanding: a Reverse Engineering Approach, since 2006

Selection of publications:

- Paulo Santos, Ademar Aguiar, "Drag-and-Dock Design Pattern", PLoP’2006, Portland, USA, October de 2006.
- Ademar Aguiar, Gabriel David, “Patterns for Documenting Frameworks: part III", PLoP’2006, Portland, USA, October 2006.
- Ademar Aguiar, Gabriel David, “Patterns for Documenting Frameworks: part II", EuroPLoP’2005, Irsee, Alemanha, July 2006.
- Wiki Based Requirements Documentation of Generic Software Products. Clara Silveira, João Pascoal Faria, Ademar Aguiar, Raul Vidal, AWRE\'05 - The Tenth Australian Workshop on Requirements Engineering, 22 November 2005, Melbourne, Australia, pp. 42-51 (best paper award)
- Ademar Aguiar, Gabriel David, “WikiWiki Weaving Heterogeneous Software Artifacts”, In proceedings of WikiSym 2005 - 2005 International Symposium on Wikis, Oct 17-18, 2005, San Diego, California, U.S.A., co-located with ACM OOPSLA 2005.
- Ademar Aguiar, Gabriel David, “Patterns for Documenting Frameworks: part I", VikingPLoP’2005, Helsinki, Finland, September 2005.
- Nuno Flores, Ademar Aguiar, “JFREEDOM: a Reverse Engineering Tool to Recover Framework Design”, in Serge Demeyer, Kim Mens, Roel Wuyts, Stéphane Ducasse, Proceedings of the 6th ECOOP Workshop on Object-Oriented Reengineering (WOOR’05), Glasgow, July 2005, Springer-Verlag,.
- Nuno Flores, Ademar Aguiar, “Reverse Engineering of Framework Design using a Meta-Patterns-based Approach”, AICCSA’2005 – Workshop on Software Stability: Timeless Architectures and System of Patterns, Cairo, Egipt, January 2005.
- Susanne Jucknath, Jan Wloka, Eric Jul, Sari R. Eldadah, Ademar Aguiar: “The Combined 14th Workshop for PhD Students in Object-Oriented Systems and Doctoral Symposium”, in “Object-Oriented Technology: ECOOP 2004 Workshop Reader”, ECOOP 2004 Workshops, Oslo, Norway, June 14-18, 2004, Final Reports. Lecture Notes in Computer Science 3344 Springer 2005, ISBN 3 540 23988 X.
- Ademar Aguiar, Gabriel David, Greg Badros, "JavaML 2.0: enriching the markup language for Java source code", XATA’2004 – XML, Aplicações e Tecnologias Associadas, Porto, Portugal, February 2004.
- Ademar Aguiar, Gabriel David, Manuel Padilha, "XSDoc: An Extensible Wiki-based Infrastructure for Framework Documentation", VIII Jornadas de Ingeniería del Software y Bases de Datos, Alicante, Spain, November 2003.
- Ademar Aguiar, Gabriel David, "A Minimalist Approach to Framework Documentation", in 13th Workshop for PhD Students in Object-Oriented Systems, Workshop Reader of the

15th European Conference on Object-Oriented Programming, Darmstadt, Germany, July 2003.

- Ademar Aguiar, Gabriel David, "FrameDocMS – um sistema de gestão de conteúdos para documentação de frameworks baseado em XML e WikiWikiWeb", XATA'2003 – XML, Aplicações e Tecnologias Associadas, Braga, Portugal, February 2003.
- Ademar Aguiar, A. Sousa, A. Pinto, "Use-Case Controller", in Proceedings of the Sixth European Conference on Pattern Languages of Programming and Computing, Irsee, Germany, 2001.
- Ademar Aguiar. "A Minimalist Approach to Framework Documentation", in OOPSLA 2000 Companion proceedings of the 15th Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications, Minneapolis, Minnesota, United States of America, October 2000.
- Gabriel David, Maria Cristina Ribeiro, José Torres, Ademar Aguiar, "Metadata for the Preservation and Retrieval of Multimedia Components", in ECAI - Electronic Cultural Atlas Initiative, 2000.
- Ademar Aguiar, "Pseudo-Reference Pattern", in Proceedings of the Fourth European Conference on Pattern Languages of Programming and Computing (EuroPLoP 1999), pages 1–12, Universittsverlag Konstanz.
- Aurélio Pires, Mário Jorge Leitão, Ademar Aguiar, Alexandre Sousa, "SIMAT - Sistema de Informação Municipal: Aplicações Técnicas em Tecnologia SIG" in Ingenium, 2ª Série, Nº 38, pp. 72-75, June 1999.

Ricardo J. Machado

Education:

- PhD and MSc degrees in Informatics and Computer Engineering (UMinho, 2001 and 1996).
- DEng degree in Electronics and Computer Engineering (FEUP, 1994).

Research interests:

- Software Engineering (model-driven development, requirements engineering, software quality)
- Embedded Software (behavioral specification with Petri nets, design automation, simulation & validation)
- Pervasive Information Systems (pervasive workflows, mobility scenarios, technology integration)

Current professional situation:

- Assistant Professor with Tenure at the Information Systems Dept. of University of Minho
- Senior researcher of the Algoritmi Research Centre

Current teaching responsibilities:

- Software Methodologies for Pervasive Computing, MSc degree in Mobile Systems, UMinho
- Quality in Software Process and Product, MSc degree in Information Systems, UMinho
- Programming Methods, DEng degree in Communications Engineering, UMinho

- Development of Software Systems, DEng degree in Information Systems, UMinho

Current scientific responsibilities:

- Scientific coordinator of Software Engineering and Management research Group (SEMAG) at Univ. Minho
- Supervisor of 4 PhD and 12 MSc students (5 MSc students already concluded)
- Scientific coordinator of the following on-going financed research projects: STACOS: Standard based Cooperative Software (FCT/POSI/CHS/48875/2002), uPAIN: Ubiquitous Solutions for Pain Monitoring and Control in Post Surgery Patients (AdI/IDEIA/70/2004/3.1B/00364/007), SOFTAS: Software Development with Aspects (FCT/POSI/EIA/60189/2004; coordinator of T2)
- Founder of the international workshop series MOMPES (International Workshops on Model-Driven Methodologies for Pervasive and Embedded Software); proceedings published by IEEE Computer Society Press; the next edition, the 4th, will be held in cooperation with ETAPS´07
- Currently, PC Chair of QUATIC 2007, PC member of ICSEA 2007, SPAC 2007, ACSD 2007, SIES 2007, WER’07, MOMPES’07, DEST 2007

PhD students:

- Alexandre Manuel Tavares Bragança, A Model-Driven Methodology for Software Product Lines Definition Based on UML-F, since 2002
- José Eduardo Moreira Fernandes, Software Development for Pervasive Information Systems: Orientation to Models, since 2003
- Francisco José Monteiro Duarte, Automated Information Systems Generation for Process Oriented Organizations, since 2006
- Paula Alexandra Fernandes Monteiro, Rigorous Model-Driven Transformations for Data-Oriented Software Architectures, since 2006

Responsibilities in technical & scientific institutions:

- R8 Coordinator (Europe, Middle East and Africa) of IEEE Computer Society, by nomination of IEEE CS Board of Governors
- President of the Portuguese representation in the IFIP Technical Committee in Computer Systems Technology (TC10)
- President of CT128, the Portuguese technical committee responsible for analysing the documents produced by JTC1/SC7 from ISO/IEC and by TC311 from CEN/CENELEC in the software and system engineering domain
- Member and responsible for the international affairs of CS03, the Portuguese Information Technologies and Telecommunications Quality Commission of the National Council for Quality (CNQ)
- Member of the Project Control Committee of CCG/UM (within the INI-GraphicsNet)
- Member of the Steering Committee of the Informatics Engineering Society (North Region) of the Portuguese Engineering Association
- Scientific reviewer of Agência de Inovação (within the Portuguese Science Foundation)
- Founder of the following committees: IFIP WG10.5 Special Interest Group on Embedded Systems (SIG-ES), IEEE-IES Technical Committee on Education in Engineering and Industrial Technologies (TCEDU), IFIP WG10.2 Working Group on Embedded Systems

- Member of the following scientific societies: IEEE Computer Society e EASST (European Association of Software Science and Technology)
- Regular scientific reviewer of the following journals: IEEE Transactions on Software Engineering, IEEE Transactions on CAD, IEEE Computer, IEEE Software, IEEE Micro, Nordic Journal of Computing, Elsevier Journal on Computer Standards & Interfaces

Volume edition:

- J.M.Fernandes, R.J.Machado, R.Khedri, S.Clark (Eds.). Proceedings of the 4th International Workshop on Model-Based Methodologies for Pervasive and Embedded Software - MOMPES'07, Braga, Portugal, March, 2007, IEEE Computer Society Press, Los Alamitos, California, U.S.A., (in print).
- B.Kleinjohann, R.J.Machado, C.E.Pereira, P.S.Thiagarajan (Eds.). Proceedings of the 5th IFIP Working Conference on Distributed and Parallel Embedded Systems - DIPES 2006, Braga, Portugal, October, 2006, IFIP Series, Springer-Verlag, New York, U.S.A., [ISBN-0-387-39361-7].
- R.J.Machado, J.M.Fernandes, M.Riebich, B.Schätz (Eds.). Proceedings of the Joint Meeting of the 4th Workshop on Model-Based Engineering of Computer Based Systems and the 3nd International Workshop on Model-Based Methodologies for Pervasive and Embedded Software - MBD/MOMPES'06, Potsdam, Germany, March, 2006, IEEE Computer Society Press, Los Alamitos, California, U.S.A., [ISBN-0-7695-2538-5].
- J.Lilius, R.J.Machado, D.Truscan, J.M.Fernandes, I.Porres (Eds.). Special Edition on Model-Based Methodologies for Pervasive and Embedded Software, Nordic Journal of Computing, vol. 12, no. 3, Fall, 2005, [ISSN-1236-6064].
- J.Lilius, R.J.Machado, D.Truscan, J.M.Fernandes (Eds.). Proceedings of the 2nd International Workshop on Model-Based Methodologies for Pervasive and Embedded Software - MOMPES'05, Rennes, France, June, 2005, TUCS General Publication no. 39, Turku, Finland, [ISBN-952-12-1556-9], [ISSN-1239-1905].
- J.M.Fernandes, J.Lilius, R.J.Machado, I.Porres (Eds.). Proceedings of the 1st International Workshop on Model-Based Methodologies for Pervasive and Embedded Software - MOMPES'04, Hamilton, Ontario, Canada, June, 2004, TUCS General Publication no. 29, Turku, Finland, [ISBN-952-12-1359-0], [ISSN-1239-1905].
- J.Lilius, F.Balarin, R.J.Machado (Eds.). Proceedings of the 3rd IEEE/IFIP/ACM/FME International Conference on Application of Concurrency to System Design - ACSD'03, Guimarães, Portugal, June, 2003, IEEE Computer Society Press, Los Alamitos, California, U.S.A., [ISBN-0-7695-1887-1].

Selection of publications:

- R.J.Machado, K.B.Lassen, S.Oliveira, M.Couto, P.Pinto. Requirements Validation: Execution of UML Models with CPN Tools. International Journal on Software Tools for Technology Transfer (STTT), Springer-Verlag, [ISSN-1433-2779], (in print).
- A.Bragança, R.J.Machado. Adopting Computational Independent Models for Derivation of Architectural Requirements of Software Product Lines. J.M.Fernandes, R.J.Machado, R.Khedri, S.Clark (Eds.). Proceedings of the 4th International Workshop on Model-Based Methodologies for Pervasive and Embedded Software - MOMPES'07 (within the 7th European Joint Conferences on Theory and Practice of Software - ETAPS 2007), Braga, Portugal, March, 2007, IEEE Computer Society Press, Los Alamitos, California, U.S.A., (in print).
- J.M.Fernandes, R.J.Machado, P.Monteiro, H.Rodrigues. A Demonstration Case on the Transformation of Software Architectures for Service Specification. B.Kleinjohann, L.Kleinjohann, R.J.Machado, C.E.Pereira, P.S.Thiagarajan (Eds.). From Model-Driven

Design to Resource Management for Distributed Embedded Systems, pp. 235-244, , IFIP Series vol. 225, Springer-Verlag, New York, U.S.A., October, 2006, [ISBN-0387-39361-7].

- F.Duarte, J.M.Fernandes, R.J.Machado. Business Modeling in Process-Oriented Organizations for RUP-based Software Development. P.Loos, P.Fettke (Eds.), Reference Modeling for Business Systems Analysis, chap. V, pp. 98-117, Idea Group Publishing, Hershey, U.S.A., October, 2006, [ISBN-1-59904-054-9].
- A.Bragança, R.J.Machado. Extending the UML2 Metamodel for Complementary Usages of the «extend» Relationship in Use Case Variability Specification. Proceedings of the 10th International Software Product Line Conference - SPLC 2006, Baltimore, Maryland, U.S.A., August, 2006, pp. 123-130, IEEE Computer Society Press, Los Alamitos, California, U.S.A., [ISBN-0-7695-2599-7].
- R.J.Machado, J.M.Fernandes, P.Monteiro, H.Rodrigues. Refinement of Software Architectures by Recursive Model Transformations. J.Münch, M.Vierimaa (Eds.), Proceedings of the 7th International Conference on Product Focused Software Process Improvement - PROFES'06, Amsterdam, Netherland, June, 2006, pp. 422-428, LNCS Series vol. 4034, Springer-Verlag, Berlin Heidelberg, Germany, [ISBN-3-540-34682-1].
- R.J.Machado, J.M.Fernandes. Integration of Embedded Software with Corporate Information Systems. A.Rettberg, M.C.Zanella, F.J.Rammig (Eds.), From Specification to Embedded Systems Application, pp. 169-171, IFIP Series vol. 184, Springer-Verlag, New York, U.S.A., August, 2005, [ISBN-0-387-27557-6].
- R.J.Machado, I.Ramos, J.M.Fernandes. Specification of Requirements Models. A.Aurum, C.Wohlim (Eds.), Engineering and Managing Software Requirements, chap. 3, pp. 47-68, Springer-Verlag, Berlin Heidelberg, Germany, July, 2005, [ISBN-3-540-25043-3].
- R.J.Machado, J.M.Fernandes, P.Monteiro, H.Rodrigues. Transformation of UML Models for Service-Oriented Software Architectures. Proceedings of the 12th IEEE International Conference on the Engineering of Computer Based Systems - ECBS 2005, Greenbelt, Maryland, U.S.A., April, 2005, pp. 173-182, IEEE Computer Society Press, Los Alamitos, California, U.S.A., [ISBN-0-7695-2308-0].
- R.J.Machado, J.M.Fernandes. A Multi-level Design Pattern for Embedded Software. B.Kleinjohann, G.R.Gao, H.Kopetz, L.Kleinjohann, A.Rettberg (Eds.), Design Methods and Applications for Distributed Embedded Systems, pp. 247-256, IFIP Series vol. 150, Kluwer Academic Publishers, Boston, U.S.A., August, 2004, [ISBN-1-4020-8148-0].

Luís Almeida

Education:

- PhD degree in Electrical Engineering (UAveiro, 1999).
- LicEng degree in Electronics and Telecommunications Engineering (UAveiro, 1988).

Research interests:

- Distributed Real-Time Systems (networks, real-time operating systems, traffic/task scheduling, fault tolerance, resource constraints, wireless ad-hoc communication, dynamic reconfigurability)
- Autonomous Mobile Robotics (cooperation, architecture, navigation, localization)

Current professional situation:

- Assistant Professor with Tenure at the Electronics, Telecommunications and Informatics Dept. of University of Aveiro
- Senior researcher at the Research Unit IEETA/UA

Current teaching responsibilities:

- Real-Time Systems, :LicEng degree in Electronics and Telecom. Eng. and Computers and Telematics Eng., UAveiro
- Electronics IV (embedded systems lab), LicEng degree in Electronics and Telecom. Eng., UAveiro

Current scientific responsibilities:

- Coordinator of the Electronics Systems Lab at the Research Unit IEETA/UA
- Member of the Scientific Committee of the science center Fábrica da Ciência Viva, Aveiro.
- Supervisor of 5 PhD students (2 in co-supervision). Supervisor of 3 concluded PhDs (1 in co-supervision) and 1 MSc:
    - Paulo Pedreiras, "Supporting Flexible Real-Time Communication on Distributed Systems", Univ. Aveiro, PhD concluded in June 2003.
    - Joaquim Ferreira, "Fault Tolerance in Flexible Real-Time Communication Systems", Univ. Aveiro, PhD concluded in May 2005.
    - Tullio Facchinetti, "Wireless Real-Time Communication Framework for Mobile Robots", Univ. of Pavia, Italy, PhD (co-supervision) concluded in July 2005.
    - Sergio Pinheiro, "Impacto da Rede de Comunicação em Sistemas de Controlo Distribuídos", Univ. Aveiro, MSc concluded in September 2005.
    - Ricardo Marau, "Supporting Dynamic Quality-of-Service Management with Real-Time Communication Protocols", Univ. Aveiro, PhD on-going, started in January 2005.
    - Rui Santos, "Enhanced Ethernet Switching Technology for Adaptive Hard Real-Time Applications", Univ. Aveiro, PhD on-going, started in October 2006.
    - Frederico Santos, "Architecture for Real-Time Coordination of Multiple Autonomous Mobile Units" Univ. PhD on-going, started in October 2006.
    - Manuel Barranco, "Utilizacion de Topologias en Estrella en la Tecnologia CAN para Mejorar la Dependabilidad", Univ. de les Illes Baleares, Spain, PhD (co-supervision) started in 2003.
    - Iria Estevez, "Tecnicas de Soporte a la Flexibilidad Funcional en Sistemas Embarcados Distribuidos de Tiempo Real", Univ. Carlos III de Madrid, Spain, PhD (co-supervision) started in 2003.
- Scientific coordinator of the following on-going financed research projects: DCCS-QoS: Quality of Service approaches in Distributed Computer Control Systems (GRICES:bilateral cooperation with China), ARTIST2 (EC, 6th FP NoE, IST-004537; local coordinator).
- Currently co-Chair of the real-time embedded systems track at ETFA 2007, member of the Organizing Commitee of SIES 2007 and member of the Program Committee of RoboCup Symposium 2007.

Participation in technical & scientific institutions:

- Member of the following scientific societies: IEEE Computer Society, IFIP WG10.5 Special Interest Group on Embedded Systems, Referees Association of MIUR (Italian Ministry of Science and Education).
- Regular scientific reviewer of the following journals: IEEE Transactions on Industrial Informatics and on Industrial Electronics, ACM Transactions on Embedded Computing

Systems, Kluwer Journak of Real-Time Systems, Elsevier Journals of System Integration, Systems Architecture, Embedded Computing and Microprocessor and Microsystems

Volume edition:

- L. Almeida (coordinated section 24: Networks, within Part III: Adaptive Real-Time Systems for QoS Management). in *Embedded Systems Design*, *The ARTIST Roadmap for Research and Development,* B. Bouyssounouse and J. Sifakis (eds.) Lecture Notes in Computer Science, Vol. 3436, ISBN: 3-540-25107-3, Springer 2005.
- L. Almeida (ed). *Proceedings of the 3rd International Workshop on Real-Time Networks RTN 2004 (satellite of the 16th Euromicro Conference on Real-Time Systems – ECRTS'04 (Catania, Italy).* Universidade de Aveiro, ISBN: 972-789-136-5, Setembro de 2004.
- L. Almeida, P. Pedreiras (eds.). Secção Especial: Sistemas de Tempo-Real, uma Abordagem Prática. *Electrónica e Telecomunicações* 4(2), ISSN: 1645-0493, Janeiro 2004.
- L. Almeida, S. Boverie (eds). *Proceedings of the 5th IFAC Symposium on Intelligent Components and Instruments for Control Applications.* IFAC Publications, ISBN: 0-08-044010-X, Dec 2003.
- L. Almeida (ed). *Proceedings of the Work-in-Progress Session of the 15th Euromicro Conference on Real-Time Systems – ECRTS'03 (Porto, Portugal).* Eições Politema, ISBN: 972-8688-11-3, July 2003.
- L. Almeida, T. Sauter (eds). *Proceedings of the Work-in-Progress Session of the 3rd IEEE Workshop on Factory Communication Systems – WFCS'00 (Porto, Portugal).* Electrónica e Telecomunicações 3(1), Revista do DETUA, ISSN: 1645-0493, September 2000.

Selection of publications:

- Ferreira, J.; Almeida, L.; Fonseca, J.A.; Pedreiras, P.; Martins, E.; Rodriguez-Navas, G.; Rigo, J.; Proenza, J.. Combining Operational Flexibility and Dependability in FTT-CAN. IEEE Transactions on Industrial Informatics, 2(2):95-102, May 2006.
- Barranco, M.; Proenza, J.; Rodriguez-Navas, G.; Almeida, L.. An Active Star Topology for Improving Fault Confinement in CAN Networks. IEEE Transactions on Industrial Informatics, 2(2):78-85, May 2006.
- T Facchinetti, G. Buttazzo, L. Almeida. Dynamic Resource Reservation and Connectivity Tracking to Support Real-Time Communication among Mobile Units, EURASIP Journal on Wireless Communications and Networking, 2005(5): 712-730, December 2005.
- P. Pedreiras, P. Gai, L. Almeida, G. Buttazzo. FTT-Ethernet: A Flexible Real-Time Communication Protocol that Supports Dynamic QoS Management on Ethernet-based Systems. IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Informatics, 1(3), August 2005.
- P. Caspi, A. Sangiovanni-Vincentelli, L. Almeida, A. Benveniste, B. Bouyssounouse, G. Buttazzo, I. Crnkovic, W. Damm, J. Engblom, G. Fohler, M. Garcia Valls, H. Kopetz, Y. Lakhnech, F. Laroussinie, L. Lavagno, G. Lipari, F. Maraninchi, Ph. Peti, J. de la Puente, N. Scaife, J. Sifakis, R. de Simone, M. Torngren, P. Veríssimo, A. Wellings, R. Wilhelm, T. Willemse, W. Yi. Guidelines for a Graduate Curriculum on Embedded Software and Systems. ACM Transactions on Embedded Computing Systems, 4(3):587-611, August 2005.
- U. Nunes, J.A. Fonseca, L. Almeida, R. Araújo, R. Maia. Fieldbuses in Real-Time Control of Autonomous Vehicles. ROBOTICA, special issue on Low Cost / Cost Effective Automation, 21(3), June 2003.