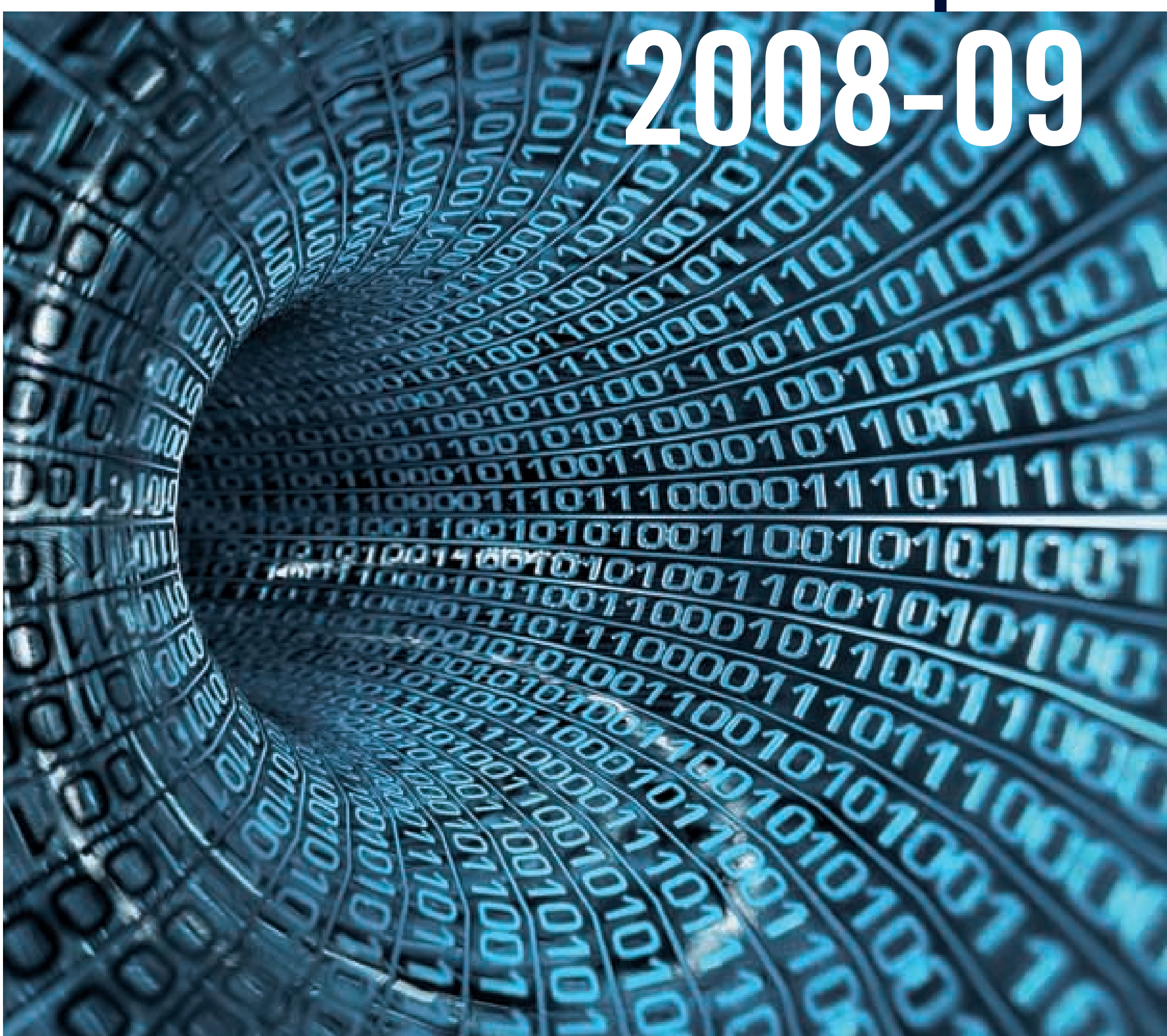
madrid institute
for advanced studies

institute
imdea
software

biennial report
2008-09



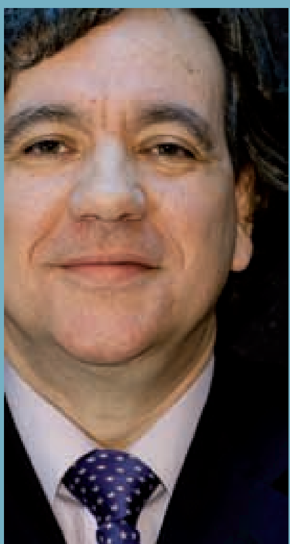
imdea

madrid institute
for advanced studies

institute
imdea
software

biennial report
2008-09

foreword



Manuel Hermenegildo
Director, IMDEA Software Institute

word

Quality research in technology-related areas is arguably the most successful and cost-effective way of generating knowledge, welfare, wealth, and sustainable growth. In early 2006 the Madrid Regional Government gathered a team of officials, scientists from Madrid universities, and representatives of leading Spanish companies which together laid the foundations, objectives, and general thematic area of a research institute in the science and technology of software development –the future IMDEA Software Institute. A group of us were then asked in late 2006 to turn these plans into reality.

Only every now and then is one offered the opportunity to be part of something that can truly make a difference. We accepted the challenge, full of hopes, enthusiasm, and determination. We set out to put in place an organization with the objective of making a key difference in the way research in Computer Science in general, and the science and technology of software development in particular, is done in Spain. A difference capable of attracting to Madrid top talent from all over the world and of increasing the quality, magnitude, and impact of research, in an area with a high potential for raising industrial competitiveness, creating value, and ultimately improving quality of life. The objectives also included closing in part a historical gap, since in contrast to other European countries, there are very few reference research centers in Computer Science in Spain.

With these high objectives and hopes, the Institute was born in 2007, starting operations in temporary premises within the School of Computer Science of the Technical U. of Madrid (UPM), and the first new researchers arrived shortly thereafter. Looking back now, after gathering the material for this 2008-2009 biennial report, the progress towards the goals is encouraging.

Without a doubt, the main acquired strength of IMDEA Software is its people: its researchers and its support staff. The Institute now includes a total of 31 researchers (including the director and the deputy director) from 12 different nationalities (Argentina, Belgium, France, Germany, Mexico, India, Ireland, Macedonia, Spain, Sweden, Ukraine, and USA). They have joined IMDEA Software after having worked at or obtained their Ph.D. degrees from 25 different prestigious universities and research centers in 11 different countries, including Stanford University, Carnegie Mellon U., or Microsoft Research in the USA, INRIA in France, U. of Cambridge in the UK, the Max Planck Software Institute in Germany, or ETH in Zurich, to name just a few. Also, 29 international researchers have visited and given talks at the Institute. In this period IMDEA Software researchers have published 43 articles in top-level refereed conferences and journals (including the top venues in the field, such as PLDI, POPL, TOPLAS, VMCAI, ESOP, CC, IJCAR, CSF, VSTTE, ECOOP, ICLP, etc. and receiving a distinguished paper award and a best presentation award), edited 6 proceedings of major conferences, published 7 invited papers and tutorials, given 13 invited talks and 11 seminars, participated in 37 program committees of international conferences, and been a member of 9 editorial boards of journals and steering committees of conferences.

Also, during this time IMDEA Software researchers have participated in and secured funding from 14 major research projects, 7 of which are funded by the European Union, and collaborated with more than 13 different companies in such projects, including Telefonica, Siemens, Atos, Fredhopper, France Telecom, SAP, Trusted Logic, AbsInt, Airbus, Alcatel, Daimler, EADS, or Deimos. Initial steps towards the creation of a spin-off based on technology developed in the Institute have also been taken.

Major progress was also made in 2008 and 2009 in completing the design and securing the funding for a permanent building on a plot ceded by UPM in the Montegancedo Science and Technology Park, and whose construction is starting at the time of writing these lines.

While much work still lies ahead to achieve the full objectives and dimension of the Institute, it is also clear that it is at this point a vibrant and encouraging reality. My thanks go to all of those –too many to list here– that have believed in and contributed to achieving this ambitious goal.

table of contents

contents

1

general presentation

1.1. Profile

The IMDEA Software Institute (Madrid Institute for Advanced Studies in Software Development Technologies) is a non-profit, independent research institute promoted by the Madrid Regional Government (CM) to perform research of excellence in the methods, languages, and tools that will allow the cost-effective development of software products with sophisticated functionality and high quality, i.e., safe, reliable, and efficient.

The IMDEA Software Institute belongs to the Madrid Institute for Advanced Studies (IMDEA) network, a new institutional framework created to foster social and economic growth in the region of Madrid by promoting research of excellence and technology transfer in a number of strategic areas (water, food, social sciences, energy, materials, nanoscience, networks, and software) with high potential impact.

1.2. Motivation and Goals

The importance of software is continuously increasing. It is the enabling technology in many devices and services which are now an essential part of our lives, and thus software failures can imply high social and economic cost. Developing software of an appropriate level of reliability, security, and performance, and doing so in a cost-effective manner poses today very significant research challenges. At the same time, because of the ubiquity of software, solutions to these challenges can have a significant and pervasive impact on productivity and on the general competitiveness of the economy.

The main mission of the IMDEA Software Institute is to perform research of excellence in methods, languages, and tools that will allow the cost-effective development of software products with sophisticated functionality and high quality, i.e., safe, reliable, and efficient. This research focus includes all phases of the development cycle (analysis, design, implementation, validation, verification, maintenance); its distinguishing feature is the concentration on approaches that are rigorous and at the same time allow building practical tools.

In order to achieve its mission, the IMDEA Software Institute is gathering a critical mass of world-wide, top class researchers, and is at the same time developing synergies between them and the already significant research base and software industry capabilities existing in the region. Although fragmented in different universities and groups, the quality of the research currently performed in the Madrid region in the areas covered by the Institute is competitive at the international level. Also, most of the IT-related companies in Spain (and, specially, their research divisions) are located in the Madrid region, which facilitates collaboration and technology transfer. Thus, the

IMDEA Software Institute brings about the opportunity of having a critical mass of researchers and industrial experts, which can allow for significant improvement in the impact of research.

1.3. Legal Status and Management Structure

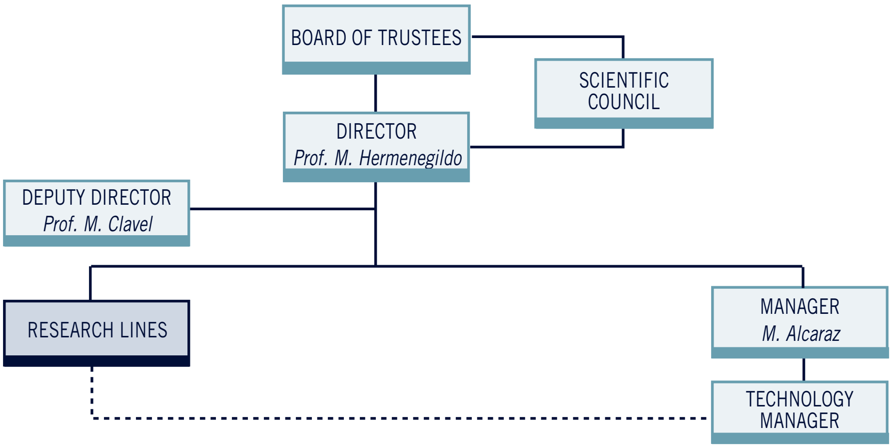


Figure 1.1: Management structure of IMDEA Software

The IMDEA Software Institute is a non-profit independent organization, constituted as a Foundation. This structure brings together the advantages and guarantees offered by the foundation status with the flexible and dynamic management typical of a private body. This combination is deemed necessary to attain the goals of excellence in research, cooperation with industry, and attraction of talented researchers from all over the world. The Institute was created legally on November 23, 2006, following a design that was the result of a collaborative effort between industry and academia, at the initiative of the Madrid Regional Government, and started its activities during 2007.

The main governing body of the Institute is the Board of Trustees. The Board includes representatives of the Madrid Regional Government, universities and research centers of Madrid, scientists with an international reputation in Software Development Technologies, and representatives of companies, together with independent experts. The Board is in charge of guaranteeing the fulfillment of the foundational purpose and the correct administration of the funds and rights that make up the assets of the Institute, maintaining their appropriate returns and utility. The Board appoints the Director, who is the CEO of the Institute, among scientists with a well-established international reputation in Software Development Technologies. The Director fosters and supervises the activities of the Institute, and establishes the distribution and application of the available funds among the goals of the Institute, within the patterns and limits decided by the Board of Trustees. The Director is assisted by the Deputy Director and the General Manager, who takes care of the legal, administrative, and financial activities of the Institute.

The Board of Trustees and the Director are assisted in their functions by the Scientific Advisory Board, a scientific council currently made up of 9 scientists from all over the world with expertise in different areas of research covered by the Institute. The tasks of this scientific council include: to provide advice on and approve the selection of researchers; to provide advice and supervision in the preparation of yearly and longer-term (4-year) strategic plans; to evaluate the performance with respect to those plans; and to give general advice on matters of relevance to the Institute.

1.4. Location

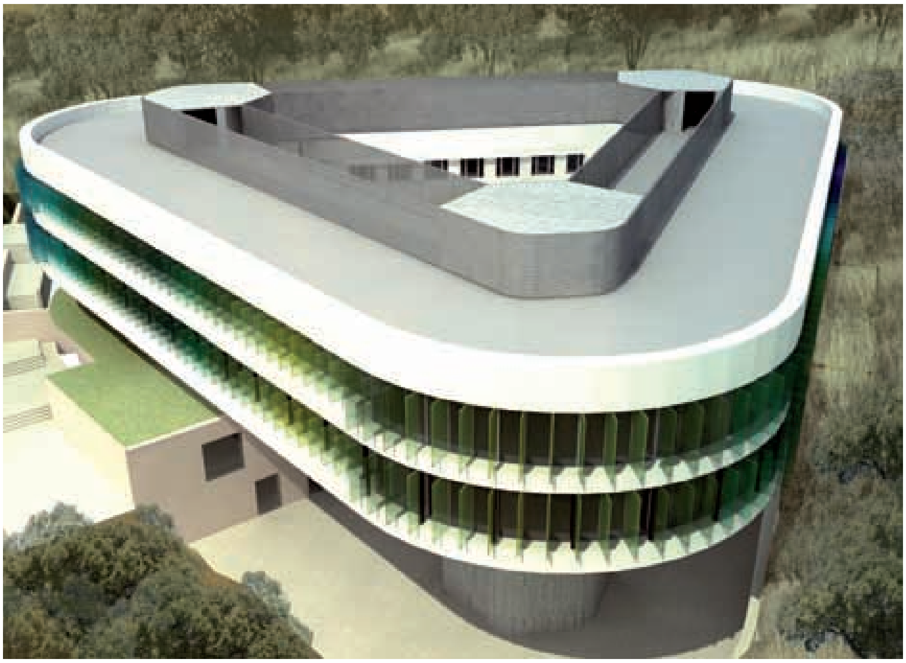


Figure 1.2. IMDEA Software new building (under construction)

The IMDEA Software Institute is temporarily located in a newly remodeled floor of the School of Computer Science of the Technical University of Madrid (UPM), in the Montegancedo Science and Technology Park. A new building, entirely devoted to the IMDEA Software Institute is under construction in a 7,500 m2 plot in the Montegancedo Science and Technology Park, which has been ceded to the Institute by UPM for 50 years.

The new site has excellent access to the Computer Science Department as well as to other new research centers within the Montegancedo Campus Scientific Park. These centers include the Madrid Center for Supercomputing and Visualization (CESVIMA), the Montegancedo Campus UPM company "incubator", the Institute for Home Automation (CEDINT), the Institute for Biotechnology and Plant Genomics (CBGP), the Microgravity Institute, and the Spanish User Support and Operations Centre for ISS payloads (USOC). In particular, the CESVIMA is equipped to study massive storage of information, high performance computing, and advanced interactive visualization and it houses the second largest supercomputer in Spain and one of the largest in Europe. A number of additional research centers are currently under construction in the campus. The new site will also make use of all the convenient new infrastructures that have been completed recently around the campus, such as the recently opened "Montepríncipe" stop of the Madrid Underground and the newly planned UPM Faculty/Student Residence. The campus has been the only campus in Spain to receive a "Campus of Excellence in Research and Technology Transfer" award in the Information and Communications Technologies area from the Ministry of Science and Innovation.

Answering to a call, building designs from six renowned architects were received in early 2009, and a committee including members of the IMDEA Software Institute, the Madrid Regional Government, and external experts selected the winning proposal by Estudio Lamela. The design of the final building has evolved in close consultation with IMDEA Software Institute researchers, which helps guarantee the fulfillment of the Institute scientific needs. Construction of the building is expected to be finished by the end of 2011.

1.5. Members of the Governing Bodies

Board of Trustees

Chairman of the Foundation

Prof. Dr. David S. Warren
State University of New York at Stony Brook, USA.

Vice-chairman of the Foundation

Mr. Jorge Sáinz
Deputy Director for Research, Madrid Regional Government, Spain.

Madrid Regional Government

Excma. Sra. Dña. Alicia Delibes Liniers
Vice-counselor for Education, Madrid Regional Government, Spain.

Ilmo. Sr. José María Rotellar García
General Director for Economics, Statistics, and Technological Innovation. Madrid Regional Government, Spain.

Ilmo. Sr. D. Salvador Victoria Bolívar
Vice-counselor for the Vice-presidency, Spokesperson for the Government, and General Secretary of the Government Council, Madrid Regional Government, Spain.

Mr. Jorge Sáinz
Deputy Director for Research, Madrid Regional Government, Spain. Vice-Chairman of the Foundation.

Universities and Public Research Bodies

Prof. Dra. Carmen Fernández Chamizo
Vice-president of Information Technologies and Communications, Universidad Complutense de Madrid, Spain.

Prof. Dr. Javier Segovia
Dean of the School of Computer Science, Universidad Politécnica de Madrid, Spain.

Prof. Dr. David Ríos Insúa
Vice-president for International Relations and New Technologies, Universidad Rey Juan Carlos, Spain.

Prof. Dra. Carmen Peláez Martínez
Vice-president for Research, Consejo Superior de Investigaciones Científicas, Spain.

Scientific Trustees

Prof. Dr. David S. Warren
State University of New York at Stony Brook, USA. Chairman of the Foundation.

Prof. Patrick Cousot
École Normale Supériure de Paris (ENS), France.

Prof. Luis Moniz Pereira
Universidade Nova de Lisboa, Portugal.

Prof. José Meseguer
University of Illinois at Urbana Champaign, USA.

Prof. Roberto Di Cosmo
Universitè Paris 7, France.

Expert Trustees

Mr. Roberto Llop
EMC, Madrid, Spain.

Mr. José de la Sota
Managing Director, Fundación para el Conocimiento (MadrI+D), Madrid, Spain.

Industrial Trustees

BBVA
Mr. Eduardo Sicilia Cavanillas.

Board meetings have been attended, as invitees, by representatives of the following companies:

Telefónica I+D
Dr. Richard Benjamins, Director for Technology Strategy at Telefónica R&D.

Deimos Space
Miguel Belló Mora, General Director.

Secretary

Mr. Alejandro Blázquez

Scientific Advisory Board

Chairman of the Board

Prof. David S. Warren
State University of New York at Stony Brook, USA.

Prof. María Alpuente
Universidad Politécnica de Valencia, Spain.

Prof. Roberto Di Cosmo
Universitè Paris 7, France.

Prof. Patrick Cousot
École Normale Supériure de Paris (ENS), France.

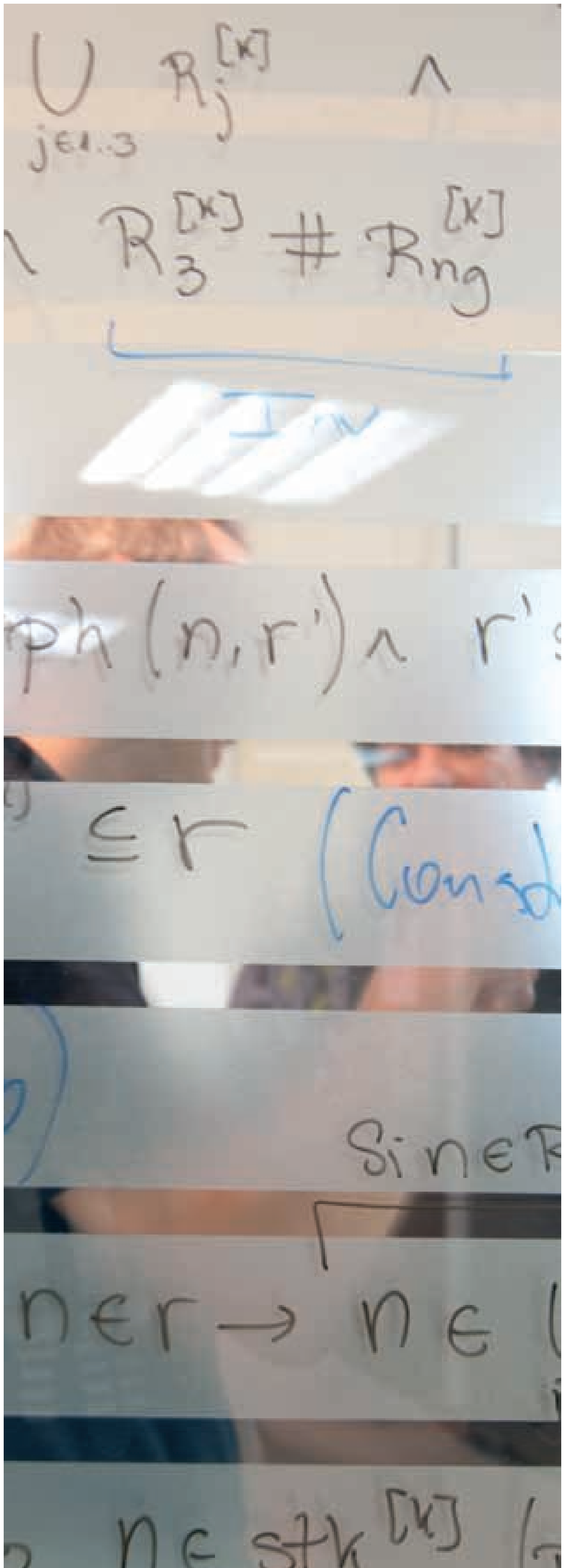
Prof. Veronica Dahl
University Simon Fraser, Vancouver, Canada.

Prof. Herbert Kuchen
Universität Münster, Germany.

Prof. José Meseguer
University of Illinois at Urbana Champaign, USA.

Prof. Luis Moniz Pereira
Universidade Nova de Lisboa, Portugal.

Prof. Martin Wirsing
Ludwig-Maximilians-Universität München, Germany.



2 cooperation framework

2.1. Cooperation with research institutions

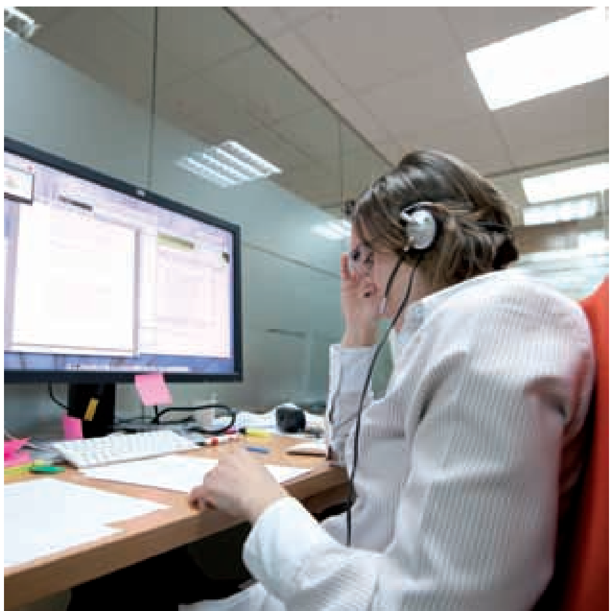
The Institute offers researchers access to and collaboration with universities and other research centers, especially in the Madrid region. The Institute is actively working with these institutions to create a critical mass of researchers capable of producing results which have significant potential impact on industry and society in general. At present the Institute has already signed agreements with the following universities and research centers:

- Universidad Politécnica de Madrid (from November 2007).
- Universidad Complutense de Madrid (from November 2007).
- Universidad Rey Juan Carlos (from January 2008).
- Roskilde University (from June 2008), Denmark.
- Consejo Superior de Investigaciones Científicas (from November 2008).

These agreements establish a framework for the development of collaborations and include the joint use of resources, equipment, and infrastructure, hiring of staff, joint participation in research projects, joint participation in graduate programs, or the association of researchers and research groups with the Institute. To illustrate the scope and importance for the Institute of these agreements, we offer here some highlights. The agreement with the Universidad Politécnica de Madrid includes provisions for the temporary and the permanent locations of the Institute, notably the previously mentioned 7,500 m^2 plot in the Montegancedo Science and Technology Park ceded to the Institute for 50 years. Under the agreement with the Consejo Superior de Investigaciones Científicas, two of its researchers —Cesar Sánchez and Pedro López— are also part of the research staff of the Institute. Finally, under the agreement with Roskilde University, one of its full professors —John Gallagher— is also part-time senior researcher at the Institute.

2.2. Cooperation with industry

IMDEA Software carries out focused collaborations with companies through joint research projects, both with medium and long-term goals, which involve challenges in the research area of the Institute with potential scientific and economic impact. Listed below are some of the companies with which IMDEA Software collaborates (or has collaborated) in joint research projects (the projects are described further in a separate chapter):



Project	Funding Agency	Industrial Partners
MOBIUS	FP6: IP	France Telecom, SAP AG Germany, Trusted Labs
HATS	PF7: IP	Fredhopper
NESSoS	PF7: NoE	Siemens, ATOS
ES_PASS (*)	ITEA2	Airbus France, CS Systèmes d'Information, Thales Avionics, Daimler AG, PSA Peugeot Citroen, Siemens VDO Automotive, Astrium SAS, GTD Barcelona, ALCATEL TSD, IFB Berlin
EzWeb	MITyC	Telefónica I+D, Alimerka, Integrasys, Codesyntax, TreeLogic, Intercom Factory, GESIMDE, Yaco, SoftTelecom
DESAFIOS-10	MICINN	BBVA-GlobalNet, LambdaStream, Deimos Space
PROMETIDOS	Madrid Regional Government	Deimos Space, Atos Origin, BBVA-GlobalNet, Thales, Telefónica I+D

(*) Through associated group at Universidad Politécnica de Madrid.

Other forms of collaboration with industry include the participation of company staff in Institute activities, joint participation in Spanish and European Technological Platforms (for example, the Technology Clusters in the Autonomous Community of Madrid or the INES, NESSI and Internet of the Future platforms), joint scholarships for doctorate or masters work (for example, Deimos Space co-funds one PhD in rigorous development of satellite image processing), transfer of research personnel trained by the Institute to companies (for example, Microsoft Redmond co-funds a four-month visit of an IMDEA postdoctoral researcher to explore technology transfer opportunities), access to the Institute's researchers as consultants, access to the Institute's prospective technology and scientific studies (for example, researchers of the Institute have met with professionals at Telefónica I+D, Canal de Isabel II, Ericsson, Interligare, Lingway, among many others, to present their main research results), and of course, availability for the creation of joint spin-offs for commercial development of technologies created in the Institute. Given the controversial status of software patents (which have recently faced the opposition not only of notorious public figures such Linus Torvalds or Tim Berners-Lee but also of the EU Parliament, and which many currently perceive as a hurdle instead of a help for innovation and profit), IMDEA Software favors the exploitation of its research results through innovative business models, for example those which are based on software products with high technological advantages and which are protected by copyrights that foster their dissemination.

Finally, IMDEA Software associated companies are hi-tech companies interested in a stable and durable relationship with the Institute. Our industrial trustees fall, of course, into this category. Associated companies participate with the Institute in joint research projects and have special and early access to the activities and results produced. These companies contribute to the sustainability of IMDEA Software, according to their capabilities, the common activities, and their specific areas.

3 research

The cost-effective development of complex, safe, reliable, and efficient software is not a simple task, and it cannot be solved by simple “magic bullets” or more enlightened management. The problem affects all stages in the development lifecycle (analysis, design, implementation, verification, maintenance). The IMDEA Software Institute performs research on these aspects along a number of dimensions which include Methodologies (the development and industrial adoption of mathematically rigorous methodologies can improve the software process further), Languages (the basis for expressing software functionality, behavior, and properties), Verification and Validation (semantically well-founded, tool-supported methods to validate code or designs with respect to specifications), and Adequacy/Optimization (the optimal use of resources to achieve a desired goal). To this end:

- The research vision materializes in a number of High-level Research Lines. The current main lines are depicted as rows in Figure 3.1.
- The vision includes also a number of focusing Areas of Application: areas of engineering where the Institute aims and expects to make an impact and which have been identified as priorities in collaboration with industry. The main current areas of application are depicted as columns in Figure 3.1.

These areas of application and high-level research lines are explained further in the rest of the chapter. Finally, two fundamental, cross-cutting issues pervade the vision:

- Tools: well-founded and cost-effective (prototypes of) tools are fundamental study harnesses, demonstrators, and technology transfer vehicles for the techniques for automation of high quality software development.
- Foundations: methods and languages should be built on appropriate mathematical foundations, and at the same time be practical.

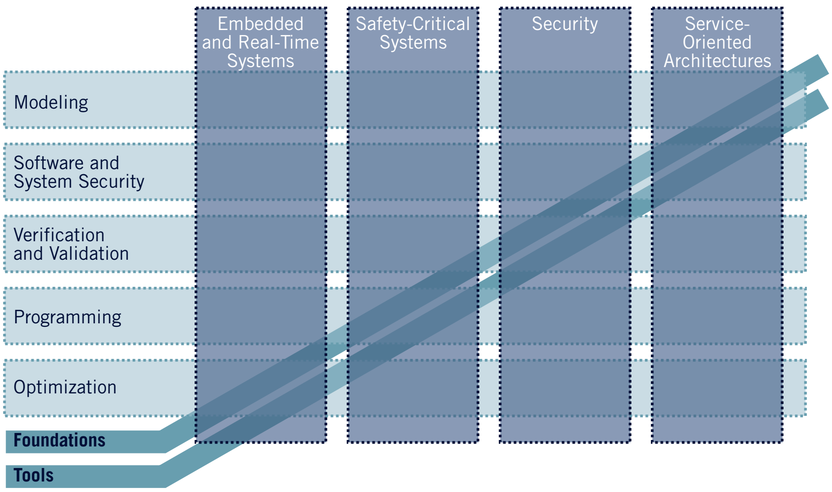


Figure 3.1: Main research lines, application areas, and cross-cutting issues.

3.1. Areas of Application

The following are some areas of application: areas of engineering where the IMDEA Software Institute aims and expects to make an impact.

3.1.1. Embedded and Real-Time Systems

One of the application areas of software where correctness is more critical is embedded systems. An embedded system is a computational artifact that is subject to physical constrains, and whose correct functioning cannot depend on human guidance. In particular, embedded systems are involved in safety-critical applications (such as control systems of automobiles or aircrafts) or systems for highly remote operation (satellite, space, etc.). Embedded systems are also pervasive in areas of high economic impact, like mobile telephony or consumer electronics. Embedded systems must be resource-aware and are often also real-time systems. This means that the computation must be correctly performed within its time constraints, and also with an adequate use of resources. There is a common perception of the potential of rigorous techniques that can improve the quality of embedded software, or the time to market of new devices or families of devices.

Most of the research activities required by embedded and real-time systems and planned at the IMDEA Software Institute are strongly related to the Strategic Research Agenda of the European Technology Platform on Embedded Computing Systems, ARTEMIS.

3.1.2. Safety-Critical Systems

Software is becoming pervasive in areas such as transportation (avionics, automotive), health (diagnosis, therapy), and control (of nuclear plants, of railway signaling systems, of conflict detection systems), where a failure or malfunction may be extremely damaging, even in terms of human lives. The constraints for such safety-critical systems are extremely stringent: the systems must be able to function during extremely long period of times, in presence of human mistakes or hardware or software failures, and provide an acceptable level of services at all times.

Thus, it is urgent to develop methods and tools that help support the development of such dependable software and its (quantitative) evaluation against the aforementioned constraints. To achieve this goal, it is important to build programming languages and software architectures that facilitate the development of fault-tolerant, resilient, and adaptable applications. One particular challenge is to scale existing methods so that they become effective in the context of distributed and networking systems.

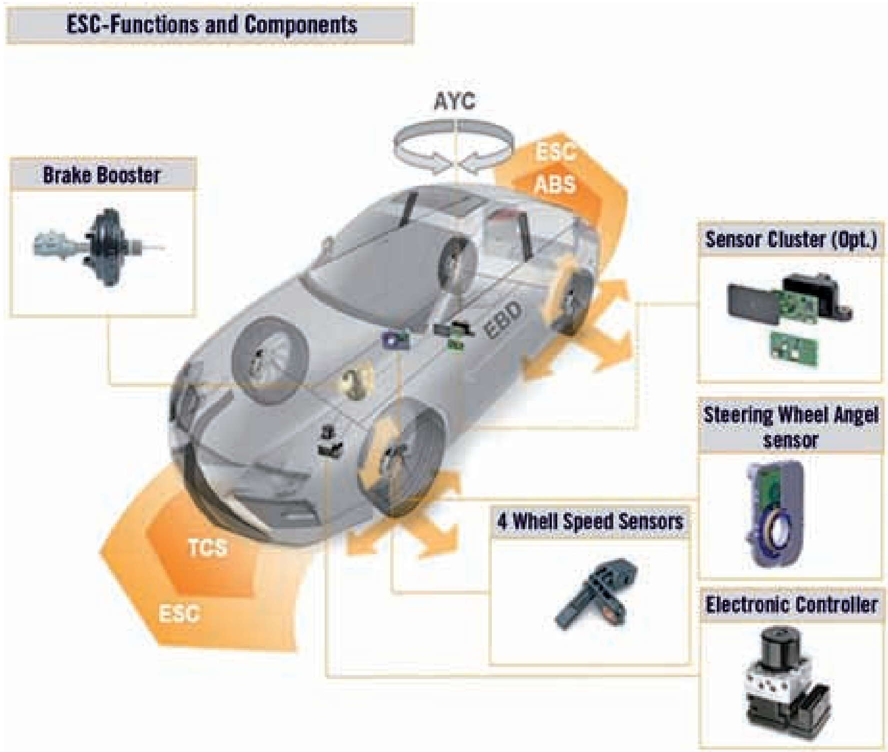


Figure 3.2: Modern cars and trucks contain as many 100 million lines of computer code. This software runs on more than 30 on-board computers and controls vital functions, including the brakes, engine, cruise control, and stability systems. It is under increasing scrutiny in the wake of recent problems with major manufacturers and it is currently impossible to fully test.

3.1.3. Security

As our society increasingly relies on information technology, there is an urgent and unprecedented need to develop new security mechanisms for protecting infrastructures, data, and applications. Several concomitant factors aggravate the problems of information security.

In order to face this challenge, one must provide scalable and rigorous techniques that can be integrated in prevailing software development processes to enforce security of applications. Since many attacks arise at the application level, it is particularly important to achieve security at the level of programming languages, drawing from methods developed in programming language research (design, analysis, and verification), and developing security solutions at a level of abstraction that matches the programming language.

3.1.4. Service Oriented Architectures

Computer infrastructures are evolving towards highly distributed networks able to provide users with a uniform and global access to services. At the same time, selling services has become the biggest growth business in the IT industry. Service Oriented Architectures (SOAs) are an attempt to provide at the level of software the necessary support for effectively programming, deploying, and maintaining services over highly-distributed networks. SOAs draw from many areas of computer science, including software engineering, concurrent and distributed systems, mobile code, and modular and component-based programming. While these areas are well developed in isolation, there remain significant challenges to combine the methodologies that stem from each area in order to deliver cost-effective approaches that support the construction and deployment of electronic services.

Most of the research activities required by SOAs and planned at the IMDEA Software Institute are strongly related to the Strategic Research Agenda of the European Technology Platform on Software and Services (NESSI) and the Future Internet of Services.

3.2. Research lines

3.2.1. Modeling

A model is an abstraction of some aspect of a system (like a blueprint in engineering), which is created to serve particular purposes, for example, to present a human-understandable description of some aspects of the system or to present information in a form that can be mechanically analyzed. The term Model-Driven Engineering (MDE) is used to describe software development approaches in which abstract models of software systems are created and systematically transformed to obtain concrete implementations or skeletons. Model-driven development holds the promise of reducing system development time and improving the quality of the resulting products.

However, in mainstream MDE practice, models are usually informal, with no well-established semantics, and only used for documentation purposes. In fact, modeling has traditionally been a synonym for producing diagrams. Most models consist of a number of "bubbles and arrows" pictures and some accompanying text. The information conveyed by such models has a tendency to be incomplete, informal, imprecise, and sometimes even inconsistent.

In order to address the major challenges current MDE technologies are facing, we believe that the past and present work on formal methods is particularly relevant. Many of the flaws in modeling are caused by the limitations of the diagrams being used. A diagram simply cannot express some of the essential information of a thorough specification. To specify software systems, formal languages offer some clear benefits over the use of diagrams. Formal languages are unambiguous, and cannot be interpreted differently by different people, for example, an analyst and a programmer. Formal languages make a model more precise and detailed, and are subject to manipulation by automated tools to ensure correctness and consistency with other elements of the model. On the other hand, a model completely written in a formal language is often not easily understood. In this sense, we believe that the interaction between the MDE and formal methods communities has a huge potential impact.

At the IMDEA Software Institute we are providing rigorous semantics for current MDE technologies (e.g., OCL, QVT) and we are developing tool-supported methodologies for applying these technologies for building meaningful models: i.e., models that have a clear and rich meaning, and therefore are useful and valuable for developing quality software. At the same time, we are proposing new MDE technologies for specific areas of applications, including software and system security and graphical user interfaces.

3.2.2. Software and system security

The goal of this line is to develop methods and tools that provide an accurate security analysis of systems and software, together with some countermeasures to defeat malicious agents.

While software security traditionally focuses on low-level protection mechanisms such as access control, the popularization of massively distributed systems dramatically increases the number and severity of vulnerabilities at the application level. These vulnerabilities may be exploited by malicious software such as viruses, Trojan horses, etc., but also (unintentionally) by buggy software, with disastrous effects.

Language-based security aims to achieve security at the level of the programming language, with the immediate benefit of countering application-level attacks at the same level at which such attacks arise. Language-based security is attractive to programmers because it allows them to express security policies and enforcement mechanisms within the programming language itself, using well-developed techniques that facilitate a rigorous specification and verification of security policies.

Language-based techniques can guarantee a wide range of policies including confidentiality, integrity, and availability, and their combination. However, their practical adoption has been hindered partly because known enforcement methods are confined to simple policies, such as non-interference for confidentiality. The most pressing challenges are defining unified enforcement mechanisms that support flexible and customizable policies, and developing methods for providing a quantitative assessment of security.

The IMDEA Software Institute is developing rich policy languages that capture precisely common instances of information release. Moreover, these policy languages are directly applicable to powerful abstraction mechanisms that pervade modern programming languages. These policy languages are supported by automated verification procedures, that allow users to detect fraudulent software.

We are also developing accurate methods for a quantitative evaluation of program security. These methods account for covert channels, including timing behavior and resource consumption, and for resistance to common attacks, such as viruses. The ultimate goal is to develop comprehensive adversarial models and effective protection strategies against covert channels.

Language-based methods have been studied primarily for mobile code and very few methods are known to scale to distributed systems. One main challenge is to ensure security of distributed applications, using a combination of cryptographic and language-based methods. Programming language techniques provide an attractive approach to guarantee the security of distributed software, because they allow to reason about programs and their cryptographic libraries in a unified framework. Moreover, programming language techniques are rigorous, and thus are useful to demonstrate beyond reasonable doubt that standard cryptographic systems, some of which have a long history of flawed security proofs and hidden but effective attacks, are secure.

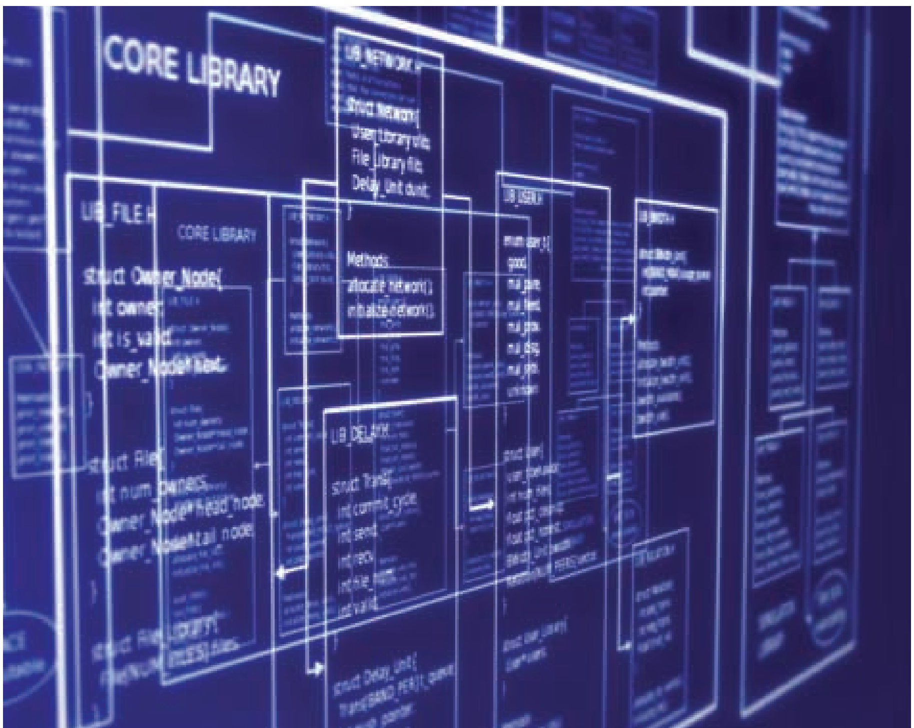
The IMDEA Software Institute is building tools that support the automated analysis of cryptographic systems and provide very strong guarantees of their correctness (cryptographic strength). The tools adopt the game-playing technique, that organizes the construction of cryptographic proofs as sequences of probabilistic games as a natural solution for taming the complexity of performing cryptographic proofs. The tools have been validated experimentally through the verification of widely deployed cryptographic standards.

3.2.3. Verification and Validation

Verification refers to the rigorous demonstration that software is correct; that is, it provides behavioral consistency according to a given specification of its intended behavior. By “intended behavior” we mean the properties that software is expected to satisfy when it is deployed. Software not possessing the properties might be defective: its execution might have unintended consequences. Verified software is software that is free of certain classes of defects because it has been rigorously proven that it satisfies its intended behavior. For these particular classes of defects, the verified software is termed zero-defect software. Such software does not require disclaimers that forgive developer error. Instead such software is guaranteed to be reliable — it behaves as intended.

How do we “rigorously prove” that software is correct? The basic principle is to represent properties as logical formulas so that verification of the properties is akin to proving a theorem using proof techniques from mathematical logic. However, modern software is very complex and typically composed of several components, where each component can be written in a different programming language. For such complex software, proving properties manually is very difficult. The question that arises naturally is this: can the logic-based proof techniques be made to scale so that software can be automatically verified as much as possible so that the manual verification burden is minimized? Apart from managing the complexity of proofs, the benefits of automatic verification are as follows. First, the verification can be repeated whenever necessary and with the same results, thus attesting to the accuracy of verification. Second, proofs can be mechanically checked for correctness. Third, verification results can be reused: once a program has been verified, its specifications can be repeatedly used in verification of a larger piece of software without re-verification.

Researchers at the IMDEA Software Institute are involved in various aspects of automatic software verification. They study expressive languages and logics for specification of properties of software, particularly of software written in modern programming languages such as Java. Once a Java program is decorated with such specifications, off-the-shelf verifiers can be used to generate “verification conditions” which can then be discharged by theorem provers. Researchers are not only studying more efficient verification algorithms and decision procedures for improving theorem proving technology, but also are performing experiments on verifying realistic code such as Java libraries — whose programs are frequently used in building complex software — and design patterns, which provide generic solutions to common software problems. The automated proofs will be made publicly available in a repository linked to the Verified Software Repository of the international Verification Grand Challenge Project.



3.2.4. Advanced Programming and Optimization Tools

The goal of this line is to develop methods and tools that help programmers improve the quality and robustness of the programs they write, allow them to write better programs in a shorter time, and support efficient execution of code through highly optimizing compilers.

Regarding program correctness and robustness the aims are similar to those in verification, but the focus here is on tools that find errors and verify programs during the process of writing such programs, rather than a posteriori. This focus requires efficient and fully automatic program analysis methods.

Abstraction-based techniques provide a unifying framework for this purpose. Their essence is abstract interpretation, a rigorous method which induces a dramatic reduction in the complexity of software analysis. It has been shown powerful enough to, for example, analyze automatically avionics software, a clear example of a large cyber-physical system, consisting of millions of lines of code, and subject to stringent conditions from the DO-178B standards. Researchers at the IMDEA Software Institute are developing tools that show that abstraction techniques can be embedded in development environments for routine use by programmers for on-line debugging, diagnosis, verification, and certificate generation, and that they combine naturally with (and reduce the need for) other techniques such as testing and run-time verification, which currently take more than 90% of overall development cost.

Abstraction-based techniques have also been shown particularly effective for high integrity and embedded software, where the properties of concerns are time and memory consumption, dynamic data sizes, energy consumption, termination, absence of errors or exceptions, etc. Researchers at the IMDEA Software Institute are developing advanced tools for debugging and verification of software with respect to these non-functional properties.

Another important way of improving the programming process, which allows programmers to write better programs in a shorter time, is by improving programming languages. Researchers at the IMDEA Software Institute are working on promising approaches such as extensible and multi-paradigm languages, support for domain-specific languages, support for multi-language applications, and service-oriented architectures.

Regarding the objective of supporting the efficient execution of code, abstraction-based techniques can also be used to ensure that programs are highly optimized before execution, i.e., that they run in the fastest and most resource-efficient way on the platforms and environmental conditions they are deployed on, while maintaining their observable behavior. Typical goals include saving on memory and processing time on sequential processors, adaptive task scheduling in parallel and distributed computers, self-reconfiguration, and automatic adaptation to environmental conditions.

A prominent form of such program optimization is automatic parallelization. As highly parallel processors are becoming an inexpensive and common facility in mainstream computing, there is an opportunity to build much faster, and eventually much better, software. Yet exploiting this enormous potential requires the development of new programming practices that reflect this profound change in the execution paradigm. Two common alternatives are to write parallel programs, using dedicated programming idioms and algorithms that help taming the complexity of parallel programs, or to automatically parallelize existing ones, using compilers for identifying parts of the application that are independent and can thus be run in parallel. Researchers at the IMDEA Software Institute are working on both approaches, developing languages and idioms more suited for parallelism and abstraction-based techniques and tools for allowing detection of common errors in parallel programs and for automatic parallelization of programs.

4 scientists

IMDEA Software strives towards excellence and being competitive with the highest-ranked institutions worldwide. To be successful in this goal, the Institute must attract highly-skilled personnel for the scientific teams and support staff. This is one of the main goals of the Institute, to the point of being a clear measure of its success.

Competition for talent in this area is extremely high at the international level, since essentially all developed countries have identified the tremendous impact that IT has on the economy and the crucial competitive advantage that attracting truly first class researchers entails. To meet this challenge the Institute is creating a world-class working environment that is competitive with similar institutions in Europe and in the US and combines the best aspects of a university department and a research laboratory.

Hiring follows internationally-standard open dissemination procedures with public, international calls for applications launched periodically. These calls are advertised in the appropriate scientific journal(s) and at conferences in the area, as well as in the Institute web page and mailing lists. Appointments at (or promotion to) the Assistant Professor (Junior) and Professor (Senior) levels (i.e., tenure-track hiring and tenure decisions) are approved by the Scientific Advisory Board. Figures 4.1 and 4.2 show the number of applications received during 2008 and 2009 and the location (by continents) of the institutions from which they applied (for senior, junior and postdoctoral positions).

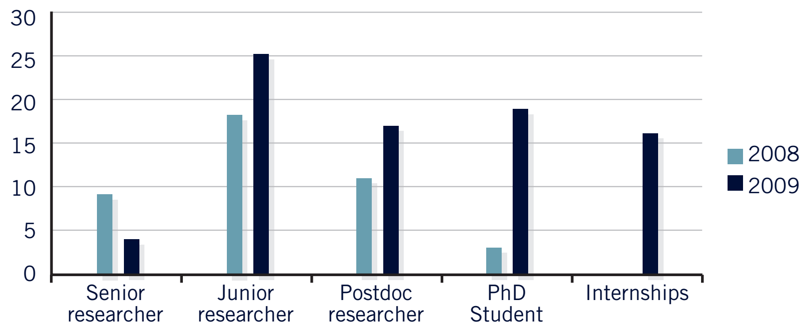


Figure 4.1: Applications received (organized by positions).

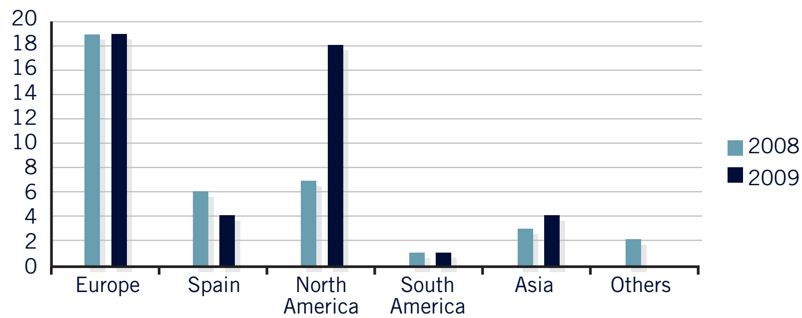
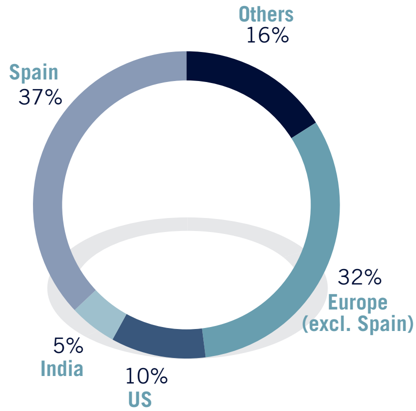


Figure 4.2: Locations of applicants' institutions (only for senior, junior, and postdoc positions), organized by continents.

Nationalities (Senior/Junior/Postdoc)



PhD degree (Senior/Junior/Postdoc)

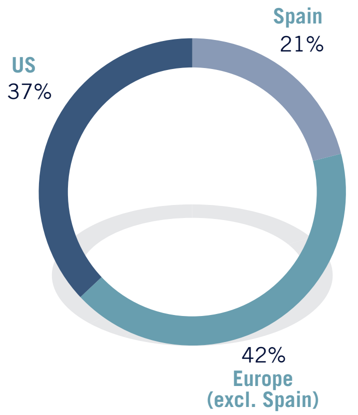
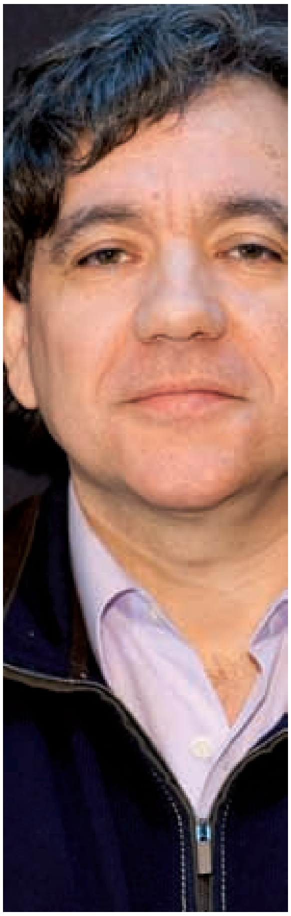


Figure 4.3, Figure 4.4

In addition to staff positions the Institute has its own program of high-quality graduate scholarships, internships, and visiting researchers. In all aspects related to human resources, the Institute follows the recommendations of the European Charter for Researchers and a Code of Conduct for their Recruitment (http://ec.europa.eu/), which it has duly signed.

Currently, the scientific staff of the Institute is composed of 5 professors (plus one part-time), 9 assistant professors (3 non-tenure track), 5 postdoctoral researchers, 9 PhD students, and 1 intern. Figures 4.3 and 4.4 summarize, respectively, the nationalities of seniors, juniors, and postdoctoral researchers, and the places from where they obtained their PhD degrees. It can be noted that 63% of IMDEA researchers were born abroad, and 79% received their PhD degree from universities either located in another European country or the US.

scientists



Manuel Hermenegildo

Professor and
Scientific Director

Manuel Hermenegildo received his Ph.D. degree in Computer Science and Engineering from the University of Texas at Austin, USA, in 1986. Since January 1, 2007 he is Full Professor and Scientific Director of the IMDEA Software Institute. He is also a full Prof. of Computer Science at the Tech. U. of Madrid, UPM. Previously to joining the IMDEA Software Institute he held the P. of Asturias Endowed Chair in Information Science and Technology at the U. of New Mexico, USA. He has also been project leader at the MCC research center and Adjunct Assoc. Prof. at the CS Department of the U. of Texas, both in Austin, Texas, USA. He has received the Julio Rey Pastor Spanish National Prize in Mathematics and Information Science and Technology and the Aritmel National prize in Computer Science. He is also one of the most cited Spanish authors in Computer Science. He has published more than 150 refereed scientific papers and monographs and has given numerous keynotes and invited talks in major conferences in these areas. He has also been coordinator and/or principal investigator of many national and international projects, area editor of several journals, and chair and PC member of a large number of conferences. He served as general director for the research funding unit in Spain, as well as member of the European Union's high-level advisory group in information technology (ISTAG), and of the board of directors of the Spanish Scientific Research Council and the Center for Industrial and Technological Development, among other national and international duties.

Research Interests

His main areas of interest include programming language design and implementation; abstract interpretation-based program analysis, verification, debugging and optimization; logic and constraint programming; parallelizing compilers; parallel and distributed processing.

Manuel Clavel

Associate Professor and Deputy Director

Manuel Clavel received his Bachelor's degree in Philosophy from the Universidad de Navarra in 1992, and his Ph.D. from the same university in 1998. Currently, he is Deputy Director and Associate Research Professor at the IMDEA Software Institute, as well as Associate Professor at the Universidad Complutense de Madrid. During his doctoral studies, he was an International Fellow at the Computer Science Laboratory of SRI International (1994 - 1997) and a Visiting Scholar at the Computer Science Department of Stanford University (1995 - 1997). His Ph.D. dissertation was published by the Center for the Study of Language and Information at Stanford University. Since then, he has published over 30 refereed scientific papers. He has also been involved in the supervision of 3 Ph.D. students (1 completed).

Research Interests

His research focuses on rigorous, tool-supported model-driven software development, including: modeling languages, model transformation, model quality assurance, and code-generation. Related interests include specification languages, automated deduction, and theorem proving.

Gilles Barthe

Professor

Gilles Barthe received a Ph.D. in Mathematics from the University of Manchester, UK, in 1993, and an Habilitation à diriger les recherches in Computer Science from the University of Nice, France, in 2004. He joined the IMDEA Software Institute in April 2008. Previously, he was leading the Everest project on formal methods and security at INRIA, France, and a member of the Microsoft Research-INRIA Joint Centre. He also held positions at the University of Minho, Portugal; Chalmers Technical University, Sweden; CWI, Netherlands; University of Nijmegen, Netherlands.
He has published more than 90 refereed scientific papers and has given several invited talks in conferences and research summer schools. He has also been coordinator/principal investigator of over many national and European projects, and served as scientific coordinator of the FP6 FET integrated project «MOBIUS: Mobility, Ubiquity and Security» for enabling proof-carrying code for Java on mobile devices (2005-2009). He has also been chair and PC member of a large number of conferences, and is a member of the editorial board of the Journal of Automated Reasoning.

Research Interests

Gilles' research interests include formal methods, programming languages and program verification, software security, and cryptography, and foundations of mathematics and computer science. His most recent research focuses on the automated certification of cryptographic schemes, and on correctness and security analyses of Java bytecode.

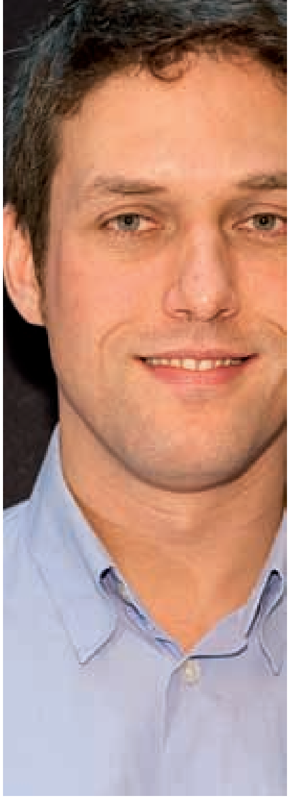
Anindya Banerjee

Professor

Anindya Banerjee received his PhD from Kansas State University, USA, in 1995. After his PhD, Anindya was a postdoctoral researcher, first in the Laboratoire d'Informatique (LIX) of École Polytechnique, Paris and subsequently at the University of Aarhus. He joined the IMDEA Software Institute in February 2009 as Full Professor. Immediately prior to this position, Anindya was Full Professor of Computing and Information Sciences at Kansas State University, USA. He was an Academic Visitor in the Advanced Programming Tools group, IBM T. J. Watson Research Center in 2007 and a Visiting Researcher in the Programming Languages and Methodology group at Microsoft Research in 2007–2008. He was a recipient of the Career Award of the US National Science Foundation in 2001.

Research Interests

Anindya's research interests lie in language-based computer security, program analysis and verification, program logics, concurrency, programming language semantics, abstract interpretation and type systems. His primary research activities over the past couple of years have centered around automatic, modular verification of properties of pointer-based programs and in proving security properties such as confidentiality and integrity properties of programs.



César Sánchez

Assistant Professor

César Sánchez received his Ph.D. degree in Computer Science from Stanford University, USA, in 2007, studying formal methods for distributed algorithms. After a post-doc at the University of California at Santa Cruz, USA, César joined the IMDEA Software Institute in 2008, becoming a Scientific Researcher at the Spanish Council for Scientific Research (CSIC) in 2009. He holds a degree in Ingeniería de Telecomunicación (MSEE) from the Technical University of Madrid (UPM), Spain, in 1998. Funded by a Fellowship from La Caixa, he moved to Stanford University, USA, receiving a M.Sc. in Computer Science in 2001, specializing in Software Theory and Theoretical Computer Science. César is a recipient of the 2006 ACM Frank Anger Memorial Award. He keeps active collaborations with research groups in the USA and Europe.

Research Interests

César's research activities focus on formal methods for reactive systems with emphasis on the development and verification of concurrent, embedded and distributed systems. His foundational research includes the temporal verification of concurrent datatypes, runtime verification, and enhancements of linear temporal logics. In parallel, he is collaborating with industrial partners from the aerospace and embedded sectors to aid in the adoption of formal techniques for software development and validation. Current projects include the interactive formal generation of parallel software for satellite image processing, and the synthesis of advanced online debuggers for testing embedded software.

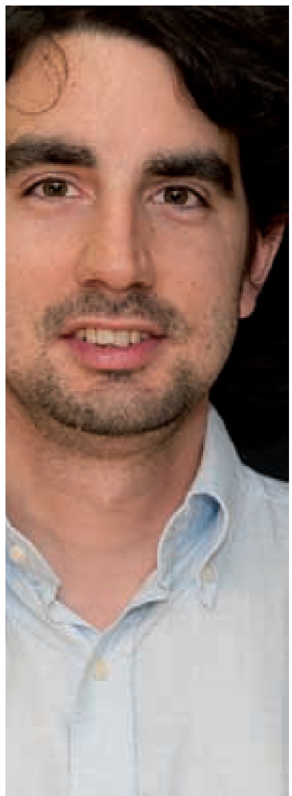
Pierre Ganty

Assistant Professor

Pierre joined the IMDEA Institute in September 2009 after completing a nearly two year postdoc at the University of California, Los Angeles (UCLA). He holds a joint PhD degree in Computer Science from the University of Brussels (ULB), Belgium and from the University of Genova (Unige), Italy that he obtained late 2007. Prior to his PhD, he completed a master and a DEA in computer science from the ULB that he obtained in 2002 and 2004, respectively. During his postdoc, Pierre has been nominated for a campus wide UCLA Chancellor's Award for Postdoctoral Research (15 nominees/1089 postdoctoral scholars).

Research Interests

Pierre's research studies automated analysis techniques for systems with infinitely many states. Many systems are, by nature, infinite and cannot be modeled precisely with finitely many states. Of particular interests are concurrent systems like multithreaded programs or communication protocols or event-based programs. In each of the above classes of systems, there is an unbounded dimension: the number of threads, the number of participants or the number of events; which is best modeled using an infinite state system.
In theory, the analysis of such systems is infeasible unless some precision is lost. In his previous works, he defined over approximation analysis techniques which are useful to prove properties on such systems. His current research has a strong emphasis on complementary under approximation techniques which do not offer complete coverage but are relevant to catch bugs in those systems.



Aleks Nanevski
Assistant Professor

Aleks received his Ph.D. degree in Computer Science from Carnegie Mellon University, USA in 2004. After holding postdoctoral positions at Harvard University (USA), and Microsoft Research, Cambridge (UK), Aleks joined the IMDEA Software Institute in September 2009. Prior to the PhD, Aleks finished his undergraduate studies in Computer Science at the University of Skopje, Macedonia in 1995.

Research Interest
Aleks' research is in the design and implementation of programming languages that facilitate verification of various program properties, ranging from type and memory safety, lack of memory leaks or information leaks, all the way to full functional correctness. His languages and systems unify programming and specification with automated and interactive theorem proving, via a common foundational framework of type theory. He is particularly interested in verifying programs that combine modern higher-order linguistic features such as higher-order functions, polymorphism, abstract types, objects and modules, with imperative ingredients such as pointer arithmetic, pointer aliasing, unstructured control flow, and concurrency.

Alexey Gotsman
Assistant Professor

Alexey Gotsman received his Ph.D. degree in Computer Science from Univeristy of Cambridge, UK in 2009. During his Ph.D. studies, Alexey interned at Microsoft Research Cambridge, UK and Cadence Berkeley Labs, USA. At the moment he is a postdoctoral fellow at Cambridge, and he will be joining IMDEA Software this summer. Prior to his Ph.D., Alexey did his undergraduate and master studies in Applied Mathematics at Dnepropetrovsk National University, Ukraine, interning at the University of Trento, Italy in the process.

Research Interests
Alexey's research interests are in software verification, with particular focus on concurrent systems software. He is interested in developing both logics for reasoning about programs and automatic tools for verifying them. Alexey's research activities include development of such logics and tools for concurrent programs with data structures, liveness properties, and operating systems.

Pedro López-García
Researcher

Pedro López-García received a MS degree and a Ph.D. in Computer Science from the Technical University of Madrid (UPM), Spain in 1994 and 2000, respectively. In May 28, 2008 he got a Scientific Researcher position at the Spanish Council for Scientific Research (CSIC) and joined the IMDEA Software Institute. Immediately prior to this position, he held associate and assistant professor positions at UPM and was deputy director of the Artificial Intelligence unit at the Computer Science Department. He has published about 30 refereed scientific papers (50% of them at conferences and journals of high or very high impact.) He has also been coordinator of the international project ES_PASS and participated as a researcher in many other national and international projects.

Research Interests
His main areas of interest include automatic analysis and verification of global and complex program properties such as resource usage (user defined, execution time, memory, etc.), non-failure and determinism; performance debugging; (automatic) granularity analysis/control for parallel and distributed computing; profiling; unit-testing; type systems; constraint and logic programming.

Mark Marron

Researcher

He received his Ph.D. from the University of New Mexico under the supervision of Deepak Kapur. He joined the IMDEA Software Institute as a postdoctoral researcher in June 2008 and is on temporary leave as a Visiting Researcher at Microsoft Research in Seattle.

Research Interests

His research interests are on developing practical techniques for modeling program behavior and using this information to support error detection and optimization applications. His work to date has focused on the development of static analysis for the program heap which infers region, sharing, footprint and heap based data-dependence information. More recent work has focused on using the information extracted by the analysis to support program parallelization, memory management, error detection, and software engineering applications.

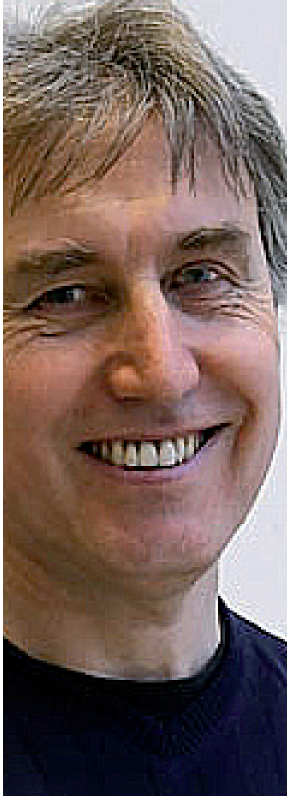
Laurent Mauborgne

Researcher

Laurent Mauborgne received his Ph.D. in Computer Science from Ecole Polytechnique, France, in 1999, and an Habilitation a diriger les recherches from University Paris-Dauphine (France) in 2007. He has been assistant professor at École Normale Supériure, Paris, since 2000, and associate director of computer science studies there since 2006. He was also part-time professor at Ecole polytechnique. He was invited to spend a year at IMDEA Software in August 2009. He published 16 refereed papers in international conferences and 3 papers in reviews. He gave courses in research summer schools and participated in the European projects DAEDALUS and ES_PASS. He was program committee member of the Static Analysis Symposium for 4 years. He is one of the authors of the Astrée analyzer, a tool that proved the absence of run-time errors in critical avionic codes.

Research Interests

The research of Laurent Mauborgne is focused on static analysis of programs and abstract interpretation. The goal is to develop theoretical as well as practical tools to analyze the behaviors of programs. This includes proving safety or temporal properties, optimizing compilation and computing resource usage. Among the recent subjects, he studied the cooperative combination of analyzes in different frameworks.



John Gallagher

Professor (part-time)

John Gallagher received the B.A. (Mathematics with Philosophy) and Ph.D. (Computer Science) degrees from Trinity College Dublin in 1976 and 1983 respectively. He held a research assistantship in Trinity College (1983-4), and post-doc appointments at the Weizmann Institute of Science, Israel (1987-1989) and Katholieke Universiteit Leuven, Belgium (1989). From 1984-1987 he was employed in research and development in a software company in Hamburg, Germany. Between 1990 and 2002 he was a lecturer and later senior lecturer at the University of Bristol, UK. Since 2002 he has been a professor at the University of Roskilde, Denmark, where he is leader of the research group Programming, Logic and Intelligent Systems as well as (part-time) Professor and holds a dual appointment at the IMDEA Software Institute since February 2007. He is a member of the executive committee of the Association of Logic Programming and of the steering committee of the ACM SIGPLAN workshop series on Partial Evaluation and Program Manipulation (PEPM). He is an editorial advisor to the journal Theory and Practice of Logic Programming. He has published approximately 50 peer-reviewed papers which have over 1200 citations.

Research Interests

His research interests focus on program transformation and generation, program analysis, constraint logic programming, rewrite systems, temporal logics, semantics-based emulation of languages and systems, and verification using abstraction and has participated in a number of national and European research projects on these topics.

Juan José Moreno-Navarro

Professor (on leave)

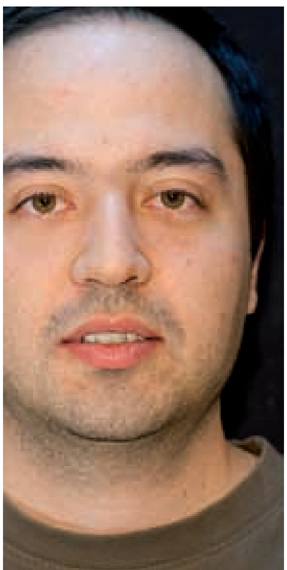
Juan José Moreno-Navarro received his Ph.D. degree in Computer Science from the Technical U. of Madrid (UPM), Spain in 1989. He developed a large part of his Ph.D. at RWTH Aachen (Germany). He has been Full Professor at the UPM Computer Science Department since 1996 and joined the IMDEA Software Institute upon its foundation. He served as Deputy Director up to 2008. He has published more than 100 papers in international conferences, books, and journal publications. He has also participated in several EU-funded and other national and international research projects, founding, leading, and ensuring continuous funding for the BABEL research group for more than 17 years. He has organized, served in program committees, and given invited talks and tutorials in many conferences in the IST field. He is a member of the editorial board of the Electronic Journal of Functional and Logic Programming. He also coordinated the first Erasmus Mundus Master taught in Spain. He has been the founding director of SpaRCIM, the Spanish research consortium in Informatics and Mathematics. He has been responsible for the ICT research program, in charge on International Relations for IST, FP6 IST Committee member, and Spanish National Contact Point for the Spanish Ministry of Education and Science. He has also been the Spanish representative at the ICT COST Committee as well as COST-ICT liaison and observer from ERCIM at the European Science Foundation, vice-chair of the Spanish Society for Software Engineering, and vice-chair of the Spanish Technology Platform on Software and Services INES.

Prof. Moreno-Navarro is currently on leave as Director General for University Policies at the Spanish Ministry of Education.

Research Interests

His research interests include all aspects related to declarative and rigorous software development technologies. This includes software development techniques, specially specification languages, automatic generation of code (programming from specifications), and software services description. His interests also include declarative languages (functional and logic programming) and, specially the integration of functional and logic programming, including the expressiveness of such these languages for real world applications. He has led the design and implementation of the language BABEL and now takes part in the activities of the international committee involved in the design of the new language Curry.

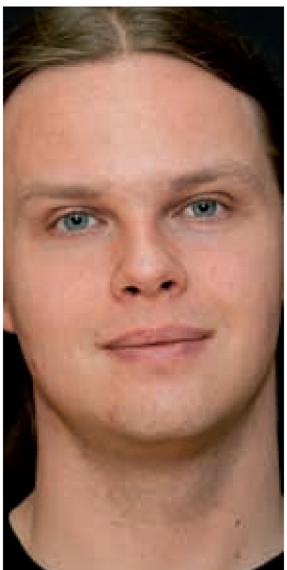
postdoctoral researchers



César Kunz
Postdoctoral Researcher

César Kunz received a Computer Science degree from the National University of Córdoba (UNC), Argentina in 2004. He continued his studies at INRIA, France, funded by the FP6 FET integrated project «MOBIUS: Mobility, Ubiquity and Security», and received a Ph.D. from the École des Mines de Paris (ENSMP), France in February, 2009. He joined the IMDEA Software Institute as a postdoctoral researcher in February 2009.

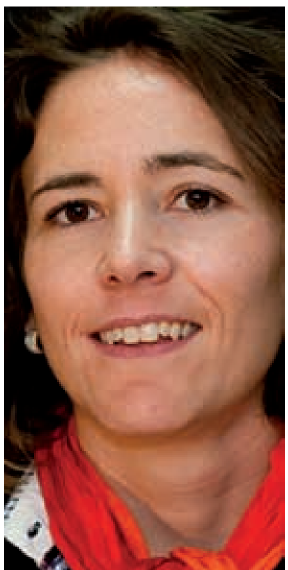
Research Interests
His research interests lie around formal program analysis and verification, abstract interpretation, and program transformation. His primary research activities are centered on the certification of program correctness, the verification of compiler optimizations, and the transformation of verification results in the presence of program transformations.



Daniel Hedin
Postdoctoral Researcher

He received his Ph.D. from Chalmers University of Technology under the supervision of David Sands. He joined the IMDEA Software Institute as a postdoctoral researcher in November 2008.

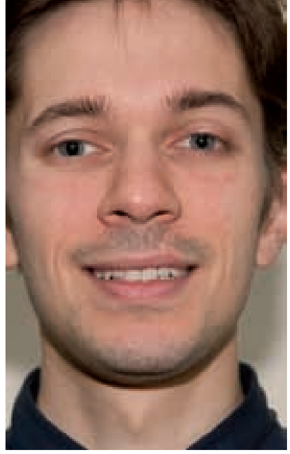
Research Interests
His research interests are on static analysis of programs, and formal verification of program analyses. His earlier work revolved around type based enforcement of noninterference, together with formalizations of their correctness in Coq. Recent work has been focused on formal certification of game based crypto proofs using CertiCrypt, and exploring the possibility of automating such proofs.



Marina Egea
Postdoctoral Researcher

She received her Ph.D. from the Universidad Complutense de Madrid in 2008, under the supervision of Manuel Clavel. She joined the IMDEA Software Institute as a postdoctoral researcher in February 2010. Previously, she held a postdoctoral position at ETH Zurich, in the Information and Security Group, lead by Prof. David Basin.

Research Interests
Her research focuses on model-driven software development, with special emphasis on model-driven software and system security. In this context, she is involved in the formal definition of security-design modeling languages, as well as in the implementation of tools supporting rigorous analysis and transformation of security-design models.



Alexander Malkis
Postdoctoral Researcher

Alexander has obtained his Diploma degree from the University of Saarland, Germany, in 2004-2005, for a work on polyforms (in other terminilogy, bond animals) under the guidance of Prof. Dr. Raimund Seidel; during his studies Alexander was financed by the prominent foundation "Studienstiftung des deutschen Volkes". He continued his studies in Saarbruecken and Freiburg, funded by the Max-Planck society and the DFG (German science foundation), obtaining his PhD thesis in 2010 at the University of Freiburg for a work on verification of multithreaded programs under guidance of Prof. Dr. Andreas Podelski. In April 2010, he joined IMDEA Software.

Research Interests
There is a range of topics in which Alexander is interested in, among them: polynomial verification of large program classes; emptiness of language intersection (complexity and algorithms); thread simulations, liveness, procedure abstractions under concurrency; a working verifier for multithreaded C; verifying multithreaded programs with rich structure and semantics, e.g. with heap, probabilism, recursion, for multicore systems; modeling biological and social systems; and synthesis of multithreaded embedded software.
Currently, Alexander is pushing the bounds of automatic verifiability of tree-based synchronization protocols that resist sequential verification attacks.

4.2. New Faculty & Faculty on leave

Boris Kopf	Assistant Professor	September 1, 2010
Juan Caballero	Assistant Professor	September 1, 2010
Ruy Ley Wild	Postdoctoral Research	November 1, 2010
Juan José Moreno	Professor	On leave

4.3. Visiting Faculty

Visiting Faculty	Institution	Period
Peter Stuckey	Univ. of Melbourne	Sep. 2008–Jan. 2009
María G. de la Banda	Monash Univ.	Sep. 2008–Jan. 2009
Laurent Mauborgne	École Normale Supériure	Sep. 2009–Aug. 2010
Alan Mycroft	Cambridge Univ.	Oct. 2009
Pedro D' Argenio	Univ. Nacional de Córdoba	Oct. 2009

research assistants

ph.d. students

Álvaro García
Research Assistant

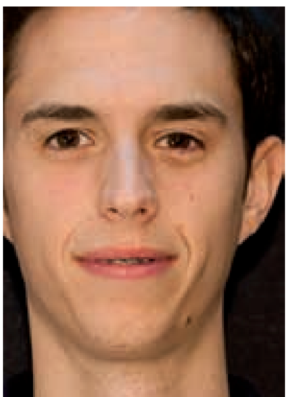
Degree: Technical University of Madrid (UPM), Spain.
Research: Type theory, dependent types and genericity, in particular how to extend dependent types in a modular way, with regards to the expression problem.

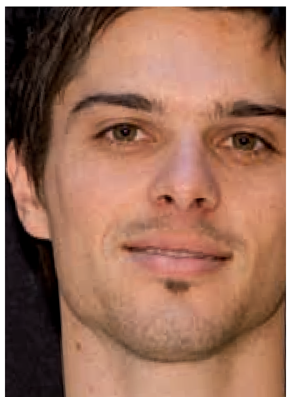
Miguel Ángel García de Dios
Research Assistant

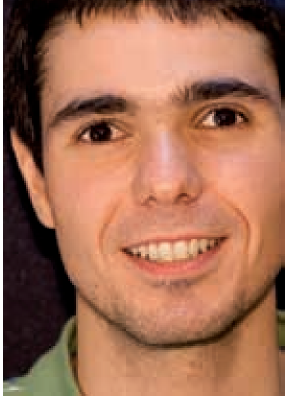
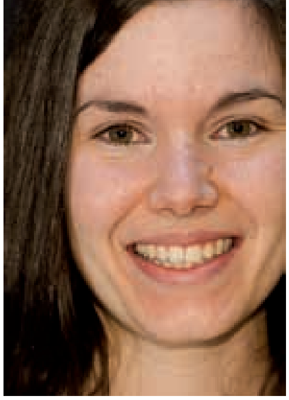
Degree: Universidad Complutense, Spain.
Research: Formal specification and verification, and rigorous tool supported modeling and validation of software systems.

Julián Samborski-Forlese
Research Assistant

Degree: Universidad Nacional de Rosario (UNR), Argentina.
Research: Applications of formal methods and abstract interpretation to program verification; quantum computing; functional programming languages; semantics.





Juan Manuel Crespo
Research Assistant

Degree: Universidad Nacional de Rosario (UNR), Argentina.
Research: Programming language semantics, type theory, functional programming, category theory, logic and software verification.

Federico Olmedo
Research Assistant

Degree: Universidad Nacional de Rosario (UNR), Argentina.
Research: Verification of cryptographic systems and semantics of programming languages.

Teresa Trigo
Research Assistant

Degree: Technical University of Madrid (UPM), Spain.
Research: Software verification techniques based on static analysis and its application to embedded systems. Resource usage analysis and automatic parallelization.

Antonio Artés
Research Assistant

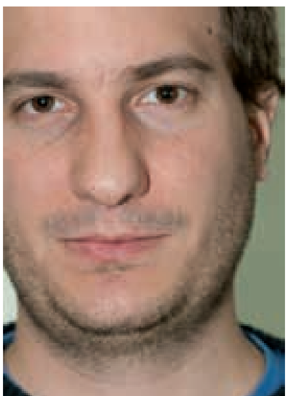
Degree: Technical University of Madrid (UPM), Spain.
Research: Power-aware, temperature-aware and reliability-aware design of low power semiconductor devices.

Santiago Zanella
Research Assistant

Degree: Universidad Nacional de Rosario (UNR), Argentina.
Research: Formal methods in general, and in particular in: verification of cryptographic systems, language-based security, semantics of programming languages, type theory, theorem provers and process calculi.

Alejandro Sánchez
Research Assistant

Degree: Universidad Nacional de Cordoba (UNC), Argentina.
Research: Formal methods, program verification, dynamic memory analysis, concurrent systems, type theory, functional programming.

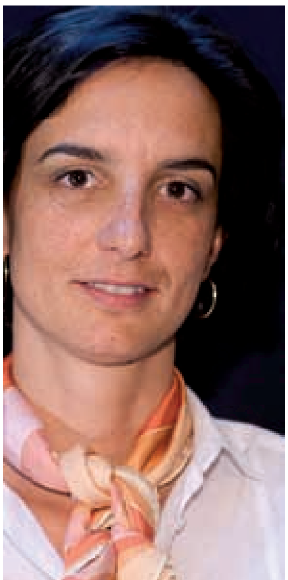

4.5. Interns

Intern	Period	Nationality
Julián Samborski-Forlese	May 2008–Nov. 2008	Argentina
Juan Manuel Crespo	Apr. 2008–Oct. 2008	Argentina
Javier Valdazo	Sept. 2008–March 2009	Argentina
Leonardo Scandolo	Sept. 2008–March 2009	Argentina
Alejandro Sánchez	Dec. 2008–May 2009	Argentina
Santiago Romero	Apr. 2009–Dec. 2009	Argentina
Pablo Buiras	Apr. 2009–Dec. 2009	Argentina
Vincent Laporte	May 2009–Aug. 2009	France
Sylviane Heraud	Jun. 2009–Jul. 2009	France
Carolina Dania	Oct. 2009–Apr. 2010	Argentina
Gerardo Huck	Oct. 2009–Apr. 2010	Argentina

4.6. Administration & IT Support

Researchers at the IMDEA Software Institute are provided with adequate administrative and technical support such that they can concentrate their efforts on scientific activities. Our administrative and technical support staff is currently co-funded by different projects.

María Alcaraz	General Manager	full-time	MBA, MSc. Economics
Paola Huerta	Assistant	full-time	MSc. History
Tania Rodríguez	Assistant	part-time	MSc. Economics
Juan Céspedes	System Administrator	part-time	MSc. Electrical Engineering
Inés Huertas	System Administrator	part-time	Bach. Telematics



5

research projects

5.1. Research Projects

Research activities and technology transfer for industry are normally carried out within the framework of research projects funded by national or international funding agencies or directly through contracts with industry. IMDEA Software is currently participating (and has participated) in a number of research projects which are briefly summarized below:

MOBIUS

Mobility, Ubiquity and Security. *Enabling proof-carrying code for Java on mobile devices*

Funding: European Union, FET Global Computing Proactive Initiative- 6th Framework Program
Duration: 2005-2009
Scientific Coordinator: Prof. Gilles Barthe

Mobius is a European Integrated Project developing novel technologies for trustworthy global computing, using proof-carrying code to give users independent guarantees of the safety and security of Java applications for their mobile phones and PDAs. Prof. Gilles Barthe was the project scientific coordinator and, from 2008 on, the IMDEA Software Institute performed the administrative project management as the project coordinator. Mobius involved 17 partners, including 3 companies (namely, France Telecom, SAP AG Germany, and Trusted Labs), from 10 countries. The budget for the project was approximately 8 M Euros.

Global computing means that applications today may run anywhere, with data and code moving freely between servers, PCs and other devices: this kind of mobility over the ubiquitous internet magnifies the challenge of making sure that such software runs safely and reliably. In this context, the Mobius project focuses on securing applications downloaded to the Java MIDP platform: globally deployed across a host of phones, this is the common runtime environment for a myriad mobile applications.

Techniques of static analysis make it possible to check program behavior by analyzing source code before it ever executes. But mobile code means that this assurance must somehow travel with the application to reach the user. Conventional digital signatures use cryptography to identify who supplied a program; the breakthrough of proof-carrying code is to give mathematical proofs that guarantee the security of the code itself. We can strengthen digital signatures with digital evidence.

Key features of the Mobius security architecture are:

- Innovative trust management, with digital evidence of program behavior that can be independently checked by users or any third party.
- Static enforcement, checking code before it starts; adaptable to manage a range of user security concerns, and configurable to match the real-world mix of mobile platforms.
- Modularity, allowing developers to build up trusted applications from trusted components.

The IMDEA Software Institute developed key innovative technologies for mobile code security, including the design and implementation of efficient and automated methods for the enforcement of resource control and information flow policies, the development of advanced compilation infrastructures that support the automatic generation of digital evidence, and the design and implementation of highly reliable infrastructures to verify digital evidence.

AMAROUT Europe

Funding: European Union, Marie Curie Action (PEOPLE-COFUND) - 7th Framework Program
Duration: 2009-2013
General Coordinator: Prof. Manuel Hermenegildo

AMAROUT Europe is a Marie Curie Action (PEOPLE-COFUND) to foster and consolidate the European Research Area by attracting to Europe and, in particular, to the region of Madrid (Spain) top research talent. AMAROUT contributes with IMDEA to the goal of turning Madrid into one of the top knowledge generation regions in Europe. To accomplish this, the AMAROUT program finances up to 132 researchers to join the IMDEA network of research institutes for one year (renewable up to twice). The total budget for the program is around 11 M Euros of which the European Union cofinances 40%.

Both "experienced" and "very experienced" researchers from any country (worldwide) can apply for AMAROUT fellowships at any of the eight IMDEA Institutes participating in the program (Software, Energy, Food, Materials, Nanoscience, Networks, Water, and Social Sciences). The AMAROUT Selection Committee consists of eight Evaluation Panels, one for each of the participating IMDEA Institutes. Each Evaluation Panel is formed by the Director of the Institute, three members of its Scientific Advisory Board, and two external, independent peer reviewers. The main AMAROUT selection criteria is the candidate's demonstrated ability and commitment to research, as well as the match of experience and interests with the research theme and lines of the IMDEA Institute chosen by the candidate.

The AMAROUT Program is a joint-initiative from eight IMDEA research institutes. The IMDEA Software Institute operates as the beneficiary. As such, IMDEA Software is also in charge of the project management structure: Scientific Committee (SC); Fellowships Management Unit (FMU); Secretary and Local Board of Prospective (BP). The FMU is responsible for the overall program management. IMDEA Software chairs the project team meetings (quarterly). The FMU is supported in its activities by the Secretary (administration, financial, H&M, welcoming) to fulfill the personnel-related, administrative and financial requirements of the Program and the EC. The secretary is commanded by the IMDEA Software. The SC is responsible of the definition of the scientific lines and of the appraisal of the correct implementation of the scientific Program. The IMDEA Software director is the leader of the SC.

HATS

Highly Adaptable and Trustworthy Software using Formal Models

Funding: European Union, FET Focused Call Forever Yours – 7th Framework Program
Duration: 2009-2013
Principal Investigator: Prof. Gilles Barthe

HATS is an Integrated Project funded by the European Union within the 7th Framework Program. The main outcome envisaged by this project is an integrated architectural framework and a methodology for rigorous development of highly adaptable and trustworthy software. The IMDEA Software Institute is one of the research centers in a consortium of 8 academic partners, 2 industrial research centers, and 1 SML, from 7 countries. The budget for the project is approximately 6 M Euros.

Software systems are central for the infrastructure of modern society. To justify the huge investments such systems need to live for decades. This requires software which is highly adaptable. Software systems must support a high degree of spatial variability to accommodate a range of requirements and operating conditions, and temporal evolvability to allow these parameters to change over time.

Current approaches to reusability and maintenance are inadequate to cope with the dynamics and longevity of future software applications and infrastructures, e.g. for e-commerce, e-health and e-government. At the same time, we rely increasingly on systems that provide a high degree of trustworthiness. The major challenge facing software construction in the next decades is high adaptability combined with trustworthiness.

A severe limitation of current development practices is the missing rigor of models and property specifications. Without a formal notation of distributed, component-based systems it is impossible to achieve automation for consistency checking, enforcement of security, generation of trustworthy code, etc. Furthermore, it does not suffice to simply extend current formal approaches. We propose to take an empirically successful, yet informal software development paradigm and put it on a formal basis.

Specifically, in HATS we will turn software product family (SWPF) development into a rigorous approach. The technical core of the project is an Abstract Behavioral Specification language which will allow precise description of SWPF features and components and their instances. The main project outcome is a methodological and tool framework achieving not merely far-reaching automation in maintaining dynamically evolving software, but an unprecedented level of trust while informal processes are replaced with rigorous analyses based on formal semantics.

The IMDEA Software Institute is responsible for the development of a highly adaptable architecture that allows cost-effective verification of the executable programs that will be automatically generated from Abstract Behavioral Specifications. The security architecture will be specifically directed towards security policies expressed using information flow and functional correctness policies.

NESSoS

Network of Excellence on Engineering Secure Future Internet Software Services and Systems

Funding: European Union, Cooperation Program (NoE) – 7th Framework Program
Duration: 2010-2013
Principal Investigator: Prof. Manuel Clavel

The Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS) aims at constituting and integrating a long lasting research community on engineering secure software based services and systems. The NESSoS consortium involves 12 partners, including 2 companies (namely, Siemens and ATOS), from 7 countries. The budget for the project is approximately 3.5 M Euros.

The domain of Engineering Secure Software Services covers a collection of engineering activities that aim for the creation of software services —i.e. ICT services delivered through the deployment of software systems— that are both behaviorally correct (typically guided by software engineering principles) as well as secure (typically guided by security engineering principles). The specific engineering activities range from requirements engineering and analysis, over the creation of architectures, high-level and detailed design into implementation through the reuse and composition of existing artifacts, as well as through the programming of new entities, typically components and services.

The approach of engineering secure software services is based on the principle of addressing security issues from the very beginning in system design and analysis, thus contributing to reduce system and service vulnerabilities, improve the necessary assurance level, thereby considering risk and cost issues during development in order to prioritize investments.

IMDEA Software leads the researcher mobility program within the consortium. This program is a mechanism that supports the integration of activities across the various sites: it brings together researchers working on related topics; it drives knowledge exchange and knowledge generation through union and diversity; and, finally, it increases the capability of joint cooperation among researchers. IMDEA Software also plays a prominent role in three research workpackages: secure service architectures and design; programming environments for secure and composable services; and security assurance for services.

EzWeb

Funding: Spanish Ministry of Industry, Tourism and Trade - Avanza2 Plan
Duration: 2007-2009
Principal Investigator: Prof. Manuel Hermenegildo

EzWeb is a collaborative project funded by MITyC, within the framework of The National Plan for Scientific Research, Development and Technological Innovation 2008-2011. The project is based on the development of key technologies to be employed in building the front end layer of a new-generation, Service-Oriented Architecture (SOA) that supports the following criteria:

- End-users must feel fully empowered. They must be able to self-serve from a wide range of available resources, providing access to content and application services, in order to set up their own personalized operating environment in a highly flexible and dynamic way ("Do it yourself", IKEA philosophy).

- Active participation of users has to be enabled, allowing them to create resources as well as share and exchange both knowledge and resources with others and learn together, thus accelerating the way innovations and improvements in productivity are incorporated.

- Interaction must be adapted and relevant to context, giving the term «context» the widest possible meaning, in a way that comprises both user context (knowledge, profile, preferences, language, information about social networks the user belongs to, etc.) and delivery context (static and dynamic characteristics of the device used for access, geographical and time location, connection bandwidth, etc.). Dynamic context variability and user mobility must also be taken into consideration.

IMDEA Software has contributed to the EzWeb project providing a formal semantics for the main components of the EzWeb platform, thanks to which its software can be in principle checked for correctness and its behavior can be (automatically) reasoned about.

DESAFIOS-10

High-Quality, Reliable, Distributed, and Secure Software Development

Funding: Spanish Ministry of Science and Innovation
Duration: 2010-2013
Principal Investigator: Prof. Gilles Barthe

The overall goal of the DESAFIOS-10 is to contribute both foundations and technologies helpful in the development of software systems with certified quality and reliability, typically based on formal methods and declarative programming. The consortium involves groups from three different Institutions (Universidad Complutense de Madrid, Universidad Politécnica de Madrid, and IMDEA Software).

This project arises as a natural evolution of previous coordinated project DESAFIOS, involving only the research groups from the Universidad Complutense de Madrid and the Universidad Politécnica de Madrid. However, DESAFIOS-10 emphasizes the security and reliability aspects of this research, which is precisely the workpackage lead by IMDEA Software.

PROMETIDOS

Methods for Rigorous Software Development

Funding: Regional Government of Madrid
Duration: 2010-2013
Principal Investigator: Prof. Gilles Barthe

PROMETIDOS-CM research program is focused in four main areas: declarative programming, to develop the next generation of languages for services; specification and validation, to provide a solid foundation for the description and analysis of services; reliability and security, to guarantee robust solutions from start to end; and efficiency, to optimize quality of service with respect to performance. A common goal for all these research lines is the development of tools that will rigorously support their scientific results and that could be eventually transferred to industry.

PROMETIDOS-CM is a consortium involving groups from Universidad Complutense de Madrid, Universidad Politécnica de Madrid, and IMDEA Software.

Comunidad de Madrid
La Suma de Todos

5.2. Projects w/Associated Groups

ES_PASS

Embedded Software Product-based Assurance

Funding: ITEA2 cluster of EUREKA Program; MITyC – PROFIT and AVANZA2
Duration: 2007-2009
Principal Investigator: Profs. Manuel Hermenegildo and Pedro López-García

The research goal of ES_PASS is to improve and integrate state-of-the-art software verification techniques based on static analysis in existing industrial engineering processes in the domain of critical embedded systems.

ES_PASS Embedded SW Boarding Pass
ITEA 2
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, TURISMO Y COMERCIO

Technology and tools for verification of critical properties in software are provided by the Technology Providers to the Industrial Domains (in particular, to the aerospace, automotive and railway transportation domains). With the benefit of the experience in the development of critical systems, industry sectors bring requirements, evaluate the tools, and assess their impacts on engineering processes. Technology providers improve the industrial-strength of their technology and improve dissemination.

The technology providers within the ES_PASS consortium are: AbsInt Angewandte Informatik GmbH, CEA-LIST, École Normale Supériure, EADS CCR, CNRS FèRIA federation, Fraunhofer FIRST, Compiler Design Lab, Technical University of Munich, Tel-Aviv University, Saarland University, and Universidad Politécnica de Madrid (in part through IMDEA Software personnel). The industrial end-users in the ES_PASS project are: Airbus France, CS Systèmes d'Information and Thales Avionics (for the aeronautics domain); Daimler AG, PSA Peugeot Citroen and Siemens VDO Automotive (for the automotive domain); Astrium SAS and GTD Barcelona (for the aerospace domain); and, ALCATEL TSD and IFB Berlin (for the railway transportation domain).

Project	Duration	Description	Funding Agency
S-CUBE	2008-2012	European network of software and services	European Union – NoE
GGCC	2006-2008	Global GNU compiler collection	ITEA; MITyC – PROFIT
DOVES	2009-2013	Development of verifiable and efficient software	MICINN
DESAFIOS	2006-2009	Development of high-quality, reliable, distributed and secure software	MICINN
MERIT	2005-2008	Resource-aware and verifiable mobile computing	MICINN
PROMESAS-CM	2006-2009	Methods for the development of high-quality and secure software	Regional Government of Madrid

5.3. Fellowships

- *Marie Curie AMAROUT Incoming Fellowships,* European Union, 2009. **Aleks Nanevski, Pierre Ganty** and **Laurent Mauborgne.**
- *Incentive for the Incorporation and Intensification of Research Activity (I3) Fellowships,* Spanish Ministry of Science and Innovation, 2009. **Gilles Barthe.**
- *Predoctoral Grants,* Madrid Regional Government, 2009. **Álvaro García** and **Teresa Trigo.**

6

dissemination of results

6.1. Publications

publications

09

6.1.1. Refereed Publications

2009

1. V. Benjamin Livshits, Aditya V. Nori, Sriram K. Rajamani, *Anindya Banerjee*: Merlin: specification inference for explicit information flow problems. PLDI 2009: 75-86.

2. *Anindya Banerjee*: Semantics and Enforcement of Expressive Information Flow Policies. Formal Aspects in Security and Trust 2009: 1-3. **Invited paper**.

3. Avraham Shinnar, Marco Pistoia, *Anindya Banerjee*: A language for information flow: dynamic tracking in multiple interdependent dimensions. PLAS 2009: 125-131.

4. *Gilles Barthe*, Benjamin Grégoire, *Santiago Zanella Béguelin*: Formal certification of code-based cryptographic proofs. POPL 2009: 90-101.

5. *Santiago Zanella Béguelin*, *Gilles Barthe*, Benjamin Grégoire, *Federico Olmedo*: Formally Certifying the Security of Digital Signature Schemes. IEEE Symposium on Security and Privacy 2009: 237-250.

6. *Gilles Barthe*, Benjamin Grégoire, Sylvain Heraud, *César Kunz*, Anne Pacalet: Implementing a Direct Method for Certificate Translation. ICFEM 2009: 541-560.

7. *Gilles Barthe*, Benjamin Grégoire, *César Kunz*, Tamara Rezk: Certificate translation for optimizing compilers. ACM Trans. Program. Lang. Syst. 31(5): (2009).

8. Vlad Rusu, *Manuel Clavel*: Vérification d'invariants pour des systèmes spécifiés en logique de réécriture. JFLA 2009: 317-350.

9. *Manuel Clavel*, Francisco Durán, Steven Eker, Santiago Escobar, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, Carolyn L. Talcott: Unification and Narrowing in Maude 2.4. RTA 2009: 380-390.

10. David A. Basin, *Manuel Clavel*, Jürgen Doser, *Marina Egea*: Automated analysis of security-design models. Information & Software Technology 51(5): 815-831 (2009).

11. Henning Christiansen, *John P. Gallagher*: Non-discriminating Arguments and Their Uses. ICLP 2009: 55-69.

12. Elvira Albert, *John P. Gallagher*, Miguel Gómez-Zamalloa, Germán Puebla: Type-based homeomorphic embedding for online termination. Inf. Process. Lett. 109(15): 879-886 (2009).

13. G. Banda and *J. P. Gallagher*. Constraint-based abstraction of a model checker for infinite state systems. In A. Wolf and U. Geske, editors, Proceedings of the 23rd Workshop on (Constraint) Logic Programming, Potsdam, Germany. Institute of Computer Science, Universitat Potsdam, 2009.

14. *Pierre Ganty*, Nicolas Maquet, Jean-François Raskin: Fixpoint Guided Abstraction Refinement for Alternating Automata. CIAA 2009: 155-164.

15. *Pierre Ganty*, Rupak Majumdar, Andrey Rybalchenko: Verifying liveness for asynchronous programs. POPL 2009: 102-113.

16. *Pierre Ganty*, Rupak Majumdar: Analyzing Real-Time Event-Driven Programs. FORMATS 2009: 164-178.

17. *Mark Marron*, Deepak Kapur, *Manuel V. Hermenegildo*: Identification of logically related heap regions. ISMM 2009: 89-98. **Best Presentation Award**.

18. Edison Mera, *Pedro López-García*, *Manuel V. Hermenegildo*: Integrating Software Testing and Run-Time Checking in an Assertion Verification Framework. ICLP 2009: 281-295.

19. *Leonardo Scandolo*, *César Kunz*, *Manuel V. Hermenegildo*: Program Parallelization Using Synchronized Pipelining. LOPSTR 2009: 173-187.

20. Daniel Cabeza Gras, *Manuel V. Hermenegildo*: Non-strict independence-based program parallelization using sharing and freeness information. Theor. Comput. Sci. 410(46): 4704-4723 (2009).

21. Jorge Navas, Mario Méndez-Lojo, *Manuel V. Hermenegildo*: User-Definable Resource Usage Bounds Analysis for Java Bytecode. Electr. Notes Theor. Comput. Sci. 253(5): 65-82 (2009).

22. Dragan Ivanovic, Manuel Carro and *Manuel Hermenegildo*. An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Proactive Monitoring. Pre-proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), November 2009.

23. Dragan Ivanovic, José Francisco Morales, Manuel Carro and *Manuel Hermenegildo*. Towards Structured State Threading in Prolog. CICLOPS 2009, 15 pages, July 2009.

24. Pablo Chico de Guzmán, Manuel Carro, *Manuel V. Hermenegildo*: Towards a Complete Scheme for Tabled Execution Based on Program Transformation. PADL 2009: 224-238.

25. Pablo Chico de Guzmán, Manuel Carro, *Manuel V. Hermenegildo*: A Tabling Implementation Based on Variables with Multiple Bindings. ICLP 2009: 190-204.

26. Ángel Herranz-Nieva, Julio Mariño, Manuel Carro, *Juan José Moreno-Navarro*: Modeling Concurrent Systems with Shared Resources. FMICS 2009: 102-116.

27. Susana Muñoz-Hernández, *Juan José Moreno-Navarro*: Extending Constructive Logic Negation with Types. Electr. Notes Theor. Comput. Sci. 246: 183-198 (2009).

28. Guillem Marpons-Ucero, Julio Mariño, Manuel Carro, Ángel Herranz-Nieva, Lars-Åke Fredlund, *Juan José Moreno-Navarro*, Álvaro Polo: A Coding Rule Conformance Checker Integrated into GCC. Electr. Notes Theor. Comput. Sci. 248: 149-159 (2009).

29. Toufik Taibi, Ángel Herranz-Nieva, *Juan José Moreno-Navarro*: Stepwise Refinement Validation of Design Patterns Formalized in TLA+ using the TLC Model Checker. Journal of Object Technology 8(2): 137-161 (2009).

30. *Aleksandar Nanevski*, Paul Govereau, Greg Morrisett: Towards type-theoretic semantics for transactional concurrency. TLDI 2009: 79-90.

publications 08

2008

1. *Anindya Banerjee*, David A. Naumann, Stan Rosenberg: Regional Logic for Local Reasoning about Global Invariants. ECOOP 2008: 387-411. **Distinguished Paper Award**.

2. *Anindya Banerjee*, David A. Naumann, Stan Rosenberg: Expressive Declassification Policies and Modular Static Enforcement. IEEE Symposium on Security and Privacy 2008: 339-353.

3. *Anindya Banerjee*, Michael Barnett, David A. Naumann: Boogie Meets Regions: A Verification Experience Report. VSTTE 2008: 177-191.

4. *Gilles Barthe*, Benjamin Grégoire, Colin Riba: Type-Based Termination with Sized Products. CSL 2008: 493-507.

5. *Gilles Barthe*: Certificate Translation. VERIFY 2008. **Invited paper**.

6. *Gilles Barthe*, Benjamin Grégoire, Sylvain Heraud, *Santiago Zanella Béguelin*: Formal Certification of ElGamal Encryption. Formal Aspects in Security and Trust 2008: 1-19. **Invited paper**.

7. *Gilles Barthe*, *César Kunz*: Certificate translation for specification-preserving advices. FOAL 2008: 9-18.

8. *Gilles Barthe*, *César Kunz*: An Introduction to Certificate Translation. FOSAD 2008: 51-95. **Tutorial**.

9. *Gilles Barthe*, *César Kunz*, David Pichardie, Julián Samborski-Forlese: Preservation of Proof Obligations for Hybrid Verification Methods. SEFM 2008: 127-136.

10. *Gilles Barthe*, Frank S. de Boer: Formal Methods for Open Object-Based Distributed Systems, 10th IFIP WG 6.1 International Conference, FMOODS 2008, Oslo, Norway, June 4-6, 2008, Proceedings Springer 2008.

11. *Gilles Barthe*, *César Kunz*, Jorge Luis Sacchini: Certified Reasoning in Memory Hierarchies. APLAS 2008: 75-90.

12. *Gilles Barthe*, Salvador Cavadini, Tamara Rezk: Tractable Enforcement of Declassification Policies. CSF 2008: 83-97.

13. *Gilles Barthe*, *César Kunz*: Certificate Translation in Abstract Interpretation. ESOP 2008: 368-382.

14. *Gilles Barthe*, Benjamin Grégoire, Mariela Pavlova: Preservation of Proof Obligations from Java to the Java Virtual Machine. IJCAR 2008: 83-99.

15. *Manuel Clavel*, Viviane da Silva, Christiano Braga, Marina Egea: Model-Driven Security in Practice: An Industrial Experience. ECMDA-FA 2008: 326-337.

16. *Manuel Clavel*, Marina Egea, Miguel Angel García de Dios: Building an Efficient Component for OCL Evaluation. ECEASST 15: (2008).

refereed publications

17. E. Albert, *J. Gallagher*, M. Gómez-Zamalloa and G. Puebla. Type-based Homeomorphic Embedding and its Applications to Online Partial Evaluation. LOPSTR, LNCS, Num. 4915, pp. 23-42, Springer-Verlag, February 2008.

18. Tobias Gedell, *Daniel Hedin*: Abstract Interpretation Plugins for Type Systems. AMAST 2008: 184-198.

19. Aslan Askarov, *Daniel Hedin*, Andrei Sabelfeld: Cryptographically-masked flows. Theor. Comput. Sci. 402(2-3): 82-101 (2008).

20. Mario Méndez-Lojo, *Manuel V. Hermenegildo*: Precise Set Sharing Analysis for Java-Style Programs. VMCAI 2008: 172-187.

21. Mario Méndez-Lojo, Ondrej Lhoták, *Manuel V. Hermenegildo*: Efficient Set Sharing Using ZBDDs. LCPC 2008: 47-63.

22. J. Navas, M. Méndez-Lojo and M. Hermenegildo. Safe Upper-bounds Inference of Energy Consumption for Java Bytecode Applications. In The Sixth NASA Langley Formal Methods Workshop (LFM 08), April 2008. Extended Abstract.

23. *Manuel V. Hermenegildo*, Francisco Bueno, Manuel Carro, *Pedro López-García*, José F. Morales, Germán Puebla: An Overview of the Ciao Multiparadigm Language and Program Development Environment and Its Design Philosophy. Concurrency, Graphs and Models 2008: 209-237. **Invited paper**.

24. Elvira Albert, Germán Puebla, *Manuel V. Hermenegildo*: Abstraction-Carrying Code: a Model for Mobile Code Safety. New Generation Comput. 26(2): 171-204 (2008).

25. E. Trias, J. Navas, E. S. Ackley, S. Forrest and *M. Hermenegildo*. Two Efficient Representations for Set-Sharing Analysis in Logic Programs. In 17th International Workshop on Functional and (Constraint) Logic Programming, WFLP'08, 15 pages, July 2008.

26. Eric Trias, Jorge Navas, Elena S. Ackley, Stephanie Forrest, *Manuel V. Hermenegildo*: Negative Ternary Set-Sharing. ICLP 2008: 301-316.

27. *Mark Marron*, Darko Stefanovic, Deepak Kapur, *Manuel V. Hermenegildo*: Identification of Heap-Carried Data Dependence Via Explicit Store Heap Models. LCPC 2008: 94-108.

28. *Mark Marron*, Mario Méndez-Lojo, *Manuel V. Hermenegildo*, Darko Stefanovic, Deepak Kapur: Sharing analysis of arrays, collections, and recursive structures. PASTE 2008: 43-49.

29. *Mark Marron*, *Manuel V. Hermenegildo*, Deepak Kapur, Darko Stefanovic: Efficient Context-Sensitive Shape Analysis with Graph Based Heap Models. CC 2008: 245-259.

30. Edison Mera, Pedro López-García, Manuel Carro, *Manuel V. Hermenegildo*: Towards execution time estimation in abstract machine-based languages. PPDP 2008: 174-184.

31. José F. Morales, Manuel Carro, *Manuel V. Hermenegildo*: Comparing tag scheme variations using an abstract machine generator. PPDP 2008: 32-43.

32. Amadeo Casas, Manuel Carro, *Manuel V. Hermenegildo*: A High-Level Implementation of Non-deterministic, Unrestricted, Independent And-Parallelism. ICLP 2008: 651-666.

33. Amadeo Casas, Manuel Carro, *Manuel V. Hermenegildo*: Towards a High-Level Implementation of Execution Primitives for Unrestricted, Independent And-Parallelism. PADL 2008: 230-247.

34. Pablo Chico de Guzmán, Manuel Carro, *Manuel V. Hermenegildo*: A Sketch of a Complete Scheme for Tabled Execution Based on Program Transformation. ICLP 2008: 795-800.

35. Pablo Chico de Guzmán, Manuel Carro, *Manuel V. Hermenegildo*, Cláudio Silva, Ricardo Rocha: An Improved Continuation Call-Based Implementation of Tabling. PADL 2008: 197-213.

36. P. Chico de Guzmán, M. Carro and *M. Hermenegildo*. A Program Transformation for Continuation Call-Based Tabled Execution. In Colloquium on Implementation of Constraint and LOgic Programming Systems, 15 pages, University of Udine, December 2008.

37. Pawel Pietrzak, Jesús Correas, Germán Puebla, *Manuel V. Hermenegildo*: A practical type analysis for verification of modular Prolog programs. PEPM 2008: 61-70.

38. Krister M. Swenson, *Mark Marron*, Joel V. Earnest-DeYoung, Bernard M. E. Moret: Approximating the true evolutionary distance between two genomes. ACM Journal of Experimental Algorithmics 12: (2008).

39. Pablo Nogueira, *Juan José Moreno-Navarro*: Bialgebra views: a way for polytypic programming to cohabit with data abstraction. ICFP-WGP 2008: 61-73.

40. Julio Mariño, *Juan José Moreno-Navarro*, Susana Muñoz-Hernández: Implementing Constructive Intensional Negation. New Generation Comput. 27(1): 25-56 (2008).

41. Guillem Marpons-Ucero, Julio Mariño-Carballo, Manuel Carro, Ángel Herranz-Nieva, *Juan José Moreno-Navarro*, Lars-Åke Fredlund: Automatic Coding Rule Conformance Checking Using Logic Programming. PADL 2008: 18-34.

42. *César Sánchez*, Matteo Slanina, Henny B. Sipma, Zohar Manna: The Reaction Algebra: A Formal Language for Event Correlation. Pillars of Computer Science 2008: 596-619. **Invited paper**.

6.1.2. Edited Volumes

2009

1. *Anindya Banerjee*: Formal Techniques for Java-like Programs. ACM Press, 2009.

2. Alessandro Aldini, *Gilles Barthe*, Roberto Gorrieri: Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009 Tutorial Lectures Springer 2009.

2008

1. Elvira Albert, *Anindya Banerjee*, Sophia Drossopoulou, Marieke Huisman, Atsushi Igarashi, Gary T. Leavens, Peter Müller, Tobias Wrigstad: Formal Techniques for Java-Like Programs. ECOOP Workshops 2008: 70-76.

2. Jacques Garrigue, *Manuel V. Hermenegildo*: Functional and Logic Programming, 9th International Symposium, FLOPS 2008, Ise, Japan, April 14-16, 2008. Proceedings Springer 2008.

3. *Gilles Barthe*, Cédric Fournet: Trustworthy Global Computing, Third Symposium, TGC 2007, Sophia-Antipolis, France, November 5-6, 2007, Revised Selected Papers Springer 2008.

4. *M. Hermenegildo*: Workshop on Declarative Aspects of Multicore Programming, DAMP 2008, Informal Proceedings, 2008.

6.1.3. Ph.D. Theses

2009

1. Cesar Kunz: Certificate Translation alongside Program Transformation. Ecole des Mines de Paris, February 2009. Advisor: *Gilles Barthe*.

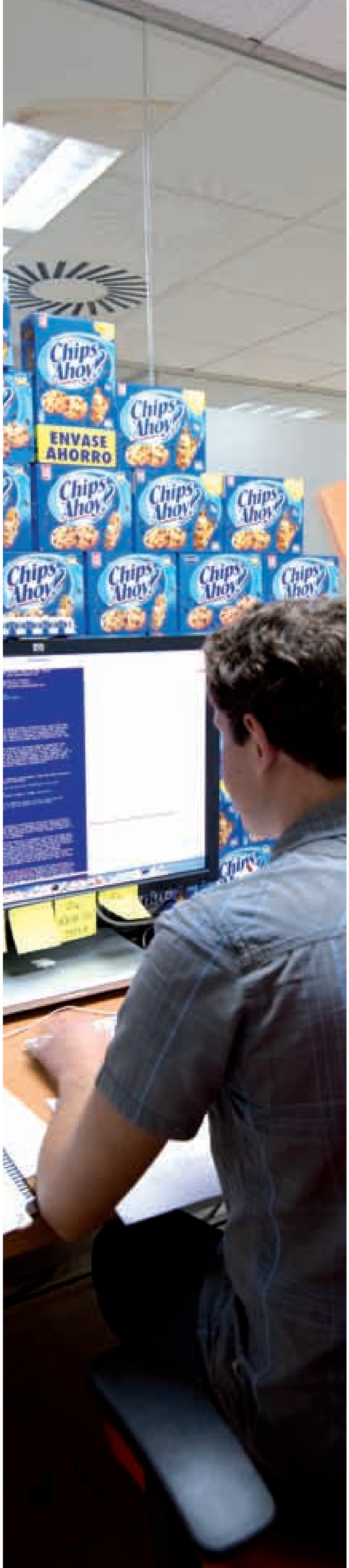
2008

1. Marina Egea: An Executable Formal Semantics for OCL with Applications to Formal Analysis and Validation. Department of Computer Science, U. Complutense de Madrid, November 2008. Advisor: *Manuel Clavel*.

2. Jorge Navas Laserna: Analysis and Inference of Resource Usage Information. Department of Computer Science, U. of New Mexico, USA, August 2008. Advisor: *Manuel Hermenegildo*.

3. Mario Méndez Lojo: A Generic Approach to Bytecode Analysis. Department of Computer Sciencem U. of New Mexico, USA, September 2008. Advisor: *Manuel Hermenegildo*.

4. Amadeo Casas: Automatic Unrestricted Independent And-Parallelism in Declarative Multiparadigm Languages. Department of Electrical and Computer Engineering, U. of New Mexico, USA, 2008. Advisor: *Manuel Hermenegildo*.



talks
invited and plenary

6.2. Invited and Plenary Talks

1. *Anindya Banerjee:* invited talk at FAST workshop (part of Formal Methods 2009).

2. *Gilles Barthe:* invited talks at PLPV'10, CosyProofs'10 (invited tutorial), CIBSI'09, JTRES'09, FLACOS'09, Discotec'09 (invited tutorial), and FAST'08. Invited courses at Asian-Pacific School on Formal Methods and Applications (2009) and Verification Technology, Systems & Applications (2008).

3. *Manuel Hermenegildo:* invited talks at 30 years of Abstract interpretation special session of POPL 2008, San Francisco, CA, USA; European Computer Science Summit'08 (ECSS'08), ETH, Zurich, Switzerland; and Interlink Workshop, July 28, 2008, Urbana-Champaign, Illinois, USA.

4. *Pedro López:* Invited talk at the ES_PASS "Frontiers of Abstract Interpretation 2009" Workshop in Toulouse, France, 2009.

5. *Mark Marron:* Invited talk at the ES_PASS external researchers and dissemination workshop.

invited
seminars &
lectures

6.3. Invited Seminars and Lectures by IMDEA Scientists

1. *Anindya Banerjee:* Microsoft Research, Bangalore, India; Indian Institute of Technology, New Delhi; Indian Institute of Technology, Kanpur; Dagstuhl Seminar on Typing, Analysis and Verification of Heap-Manipulating Programs.

2. *Manuel Clavel:* IRISA (Rennes); ETH Zurich; CITI New University of Lisbon.

3. *Gilles Barthe:* ETH Zurich; Institute of Software of Chinese Academy of Science.

invited speaker series

6.4. Invited Speaker Series

The following external researchers have given invited talks at the IMDEA Software Institute in this period.

2009

1. Sriram Rajamani (Microsoft, India): Verification, Testing and Statistics.

2. Francesco Logozzo (Microsoft, USA): Precise and scalable static analysis of bytecode with Clousot & Code Contracts.

3. Roberto Bagnara (Universita di Parma, Italia): Ranking Functions for Automatic Termination Analysis.

4. Alan Mycroft (University of Cambridge, UK): Strictness Meets Data Flow.

5. Murdoch Gabbay (Heriot-Watt University, Edinburgh): Permissive Nominal Terms.

6. Uri Juhasz (Tel Aviv University, Israel): Modular Analysis of Shared Abstractions.

7. Alvaro Arenas, Researcher (E-Science Centre, STFC Rutherford Appleton Laboratory, UK): Usage Control and Reputation in Grid Systems.

8. Martin Leucker (Institut für Informatik, Technische Universität München, Germany): Don't Know for Multi-Valued Systems.

9. Diego Garbervetsky (Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad De Buenos Aires, Argentina): Invariant-based analysis of dynamic memory requirements.

10. Rob Hierons (School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex, UK): Search Based Testing from State Machines.

11. Luis Moniz Pereira (Centro de Inteligencia Artificial, Universidade Nova de Lisboa, Portugal): On Preferring and Inspecting Abductive Models.

12. Pierre Ganty (University of California at Los Angeles, USA): What's decidable for Asynchronous Programs?

13. Julia Lawall (DIKU University of Copenhagen, Denmark): A Foundation for Flow-Based Program Matching Using Temporal Logic and Model Checking.

14. Amal Ahmed (Toyota Technical Institute, Chicago, USA): Logical Relations: A Step Towards More Secure and Reliable Software.

15. Zaynah Dargaye (INRIA Rocquencourt, France): Mechanized Verification of Functional Language Compiler.

16. Laurent Mauborgne (École Normale Supérieure, France): Disjunctions that Scale Up.

17. Deepak Garg (Computer Science Departament, Carnegie Mellon University, USA): Applying logic to Secure Computer Systems.

18. Aleks Nanevski (Microsoft Research, Cambridge, UK): Programming with Hoare Type Theory.

19. Ralf Lämmel (University of Koblenz-Landau, Germany): The Expression Lemma.

20. Alexey Gotsman (University of Cambridge, UK): Modular verification of concurrent programs with heap.

2008

1. Polyvios Pratikakis (Verimag): Sound, precise and efficient static race detection for multithreaded programs.

2. Pedro R. D'Argenio (Universidad Nacional de Cordoba, Argentina): Quantitative Model Checking Revisited: neither Decidable nor Approximable.

3. Francesco Zappa Nardelli (INRIA Paris-Rocquencourt, France): Relaxed memory models for multiprocessors (or, all what you did not want to know about your multiprocessor).

4. Gogul Balakrishnan (NCE Labs, Princeton, NJ, USA): WYSINWYX: What You See Is Not What You eXecute.

5. Chris Gill (Washington University in St. Louis, USA): Time and Event Based System Software Design and Verification.

6. Gerardo Schneider (University of Oslo): Specification and analysis of electronic contracts.

7. Xavier Leroy (INRIA, Paris-Rocquencourt, France): Formal Verification of a realistic compiler.

8. Jose Meseguer (University of Illinois at Urbana-Champaign, USA): The Temporal Logic of Rewriting.

9. Anindya Banerjee (Kansas State University, USA): Semantics and Enforcement of Information Flow Policies.

Theory Lunch Series

The Institute also holds an internal seminar series to foster communication and collaboration. A total of 27 seminars were given in 2009 and 24 in 2008 within this series.

scientific service & other activities

6.5. Scientific Service & Other Activities

6.5.1. Program Committees

1. *Cesar Sánchez:* ICTAC'09 and PDMC'09.

2. *Manuel Clavel:* OCL'09 and WRLS'08.

3. *Anindya Banerjee:* POPL'09, FTfJP'08, and FTfJP'09 (chair).

4. *Gilles Barthe:* ICALP'09, FM'09, TPHOLs'09, FAST'09, NFM'09, NSS'09, FCC'09, PCC'09, FAST'08, VERIFY'08, ESORICS'08, AMAST'08, PCC'08, PLAS'08, CSF'08, FMOODS'08 (co-chair), and FM'08.

5. *John Gallagher:* WLPE'09 (chair), PPDP'09, LOPSTR'09, M4M'09, PEPM'09, PADL'09, ICLP'08, and SAS'08.

6. *Manuel Hermenegildo:* DAMP'08 (chair), FLOPS'08 (chair), ICLP'08, SAS'08, and ServiceWave'08.

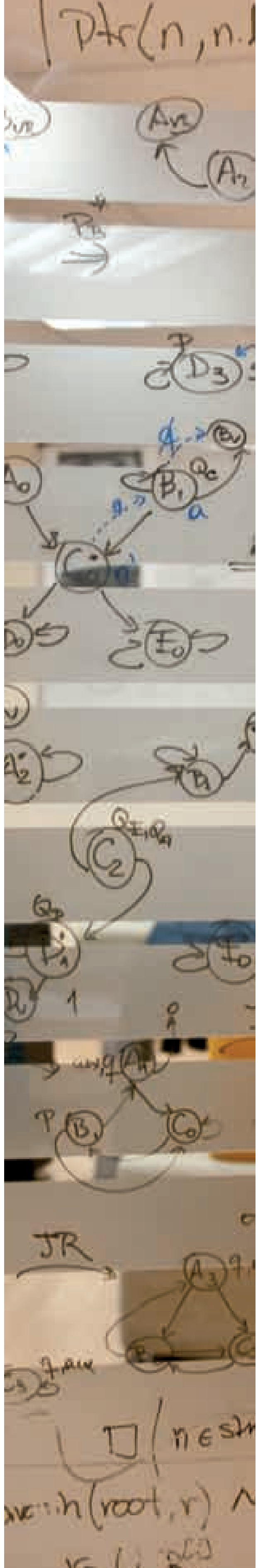
6.5.2. Editorial Boards and Steering Committees

1. *Anindya Banerjee:* Editorial Board of the Journal of Higher Order and Symbolic Computation.

2. *Gilles Barthe:* Editorial Board of the Journal of Automated Reasoning.

3. *John Gallagher:* Editorial Advisor of Theory and Practice of Logic Programming.

4. *Manuel Hermenegildo:* Editorial Advisor of Theory and Practice of Logic Programming; Area Editor of the Journal of Applied Logic; Associate Editor of the Journal of New Generation Computing; Member of the ACM POPL Steering Committee; Member of the ACM PPDP Steering Committee; Member of the FLoC Steering Committee.



7 scientific highlights

high integri

when software has to fly

High Integrity Software: When Software Has To Fly

Some software cannot fail, in some important real world applications. This software is called high integrity software and must be trusted to work dependably in some critical function. Failure in these programs may have catastrophic results in terms of lives or high economic cost. For example, failure in a program used by air traffic controllers could lead to fatal accidents; a failure in a medical system providing treatment could lead to irreversible damage; failures in parts of automobile systems such as brake controllers, apart from being potentially dangerous, could lead to massive and costly recalls. In fact, all of these scenarios have occurred already.

Software engineering practices balance between the cost (and time) to complete a project, and the quality of the outcome. The rapidly growing demand for new, larger software projects with more complex functionality has increased industry demand for software developers. In turn, this need has motivated a trend towards reducing the training necessary for software engineers and developers to enter the job market. However, at the same time, the quality of the software produced has become more and more difficult to guarantee. The issue with software quality is witnessed by the fact that the dominant factor of the overall cost of current industrial software projects is testing, and not building the product itself. Even in non-critical projects testing dominates more than 90% of the total cost.

The quality and reliability requirements of high integrity software justifies the investment in scientific undertakings to create a body of knowledge about how to build more reliable software. These new techniques intend to provide better guarantees of quality, at the price of using more sophisticated methods and tools by properly trained software engineers. Moreover, in the long run these methods could also lead to more productive software processes.

ty software

Researchers from the IMDEA Software Institute have developed – and continue to develop – cutting-edge technologies for high-integrity software following two different approaches. First, a fundamental attempt to create the basic science that can be used to craft the high-integrity software of the future. These techniques are designed to provide the best guarantee of adherence to intended behavior. Completed and ongoing projects include the use of high-order theorem proving to verify programs and libraries, static analysis for functional and non-functional properties of real-time and embedded systems, and temporal verification of reactive systems, in particular concurrent datatypes. At the same time, the IMDEA Software Institute has developed novel lightweight and applicable techniques that can be directly incorporated to improve existing software practices: advanced visualization of heap-manipulating programs, debugging of production system programs, and online monitoring of embedded reactive programs based on runtime verification.

The IMDEA Software Institute is collaborating with the leading aerospace company Deimos, located in the area of Madrid, on the technology transfer of these techniques. This continuing effort started with the rigorous and systematic development of software for satellite image processing. The aim of this project is to develop the tools to interactively synthesize provably correct software, based on a formal approach to software families, applied to image processing. Using these tools software engineers can develop very efficient parallel software that can be verified with independent tools. Moreover, different projects can experience dramatic cost gains by the increase in the level of reuse by the use of software families.

language-ba

Language-based Security: Building Trustworthy Software for the Interconnected World

Current computing environments and infrastructures are increasingly heterogeneous and dynamically changing. Executable mobile programs are everywhere: web pages, email, plug-and-play extensions, JavaScript, on-line games, Word and PowerPoint documents and attachments, electronic banking... Software is constantly being updated and downloaded over the Internet, sometimes without our knowledge or consent. Yet, today's security architectures provide poor protection from faulty software, and even less from malicious software. These security architectures were developed at the time when software was managed and updated infrequently by an experienced administrator, we trusted the (few) programs we ran, physical access to the systems was required to cause any damage to the data, and crashes and outages did not cost billions. As none of these conditions is valid anymore, our information systems have become increasingly susceptible to attacks with potentially devastating consequences.

To accommodate for the new trends in software use and deployment, the IMDEA Software Institute is working on developing new security architectures that are well suited for networked computing systems built from diverse and extensible components. We leverage techniques from programming language and logic design, to address the following issues.

- As mobile programs move on the network, it is important for them to gain trust of the new host by presenting verifiable evidence that they conform with the host's security policy. One of our research concerns is developing languages and logics in which such verifiable evidence can be constructed in the form of a rigorous mathematical proof. Another concern is helping the code producers to construct such mathematical proofs as automatically as possible, without requiring a prohibitive investment of time and resources.

- Various hosts may have various security policies. For example, mobile phones may allow downloading games from certain web-sites, but not from certain others. On the other hand, a hospital information system will probably never concern itself with downloading games, but will focus on ensuring that confidential patient data is not leaked to unauthorized personnel, or to the general public. A research concern here is, again, in designing languages and logics in which a wide variety of security policies can be specified.

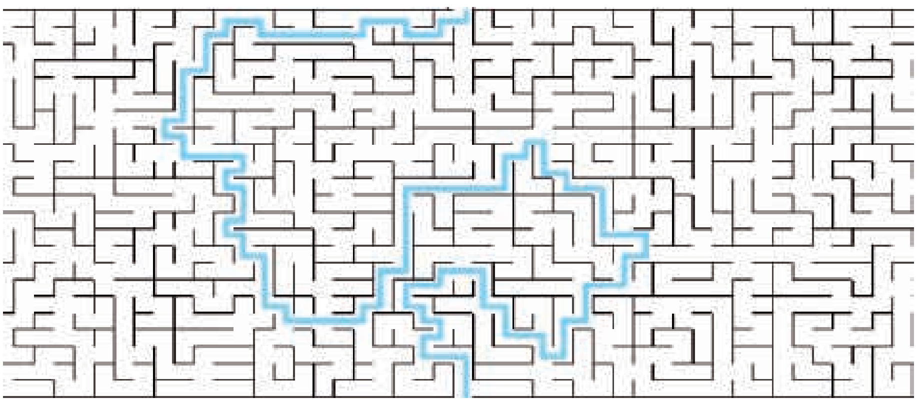
building trustworthy software
for the interconnected world

sed security

One of the general directions that the IMDEA Software Institute is pursuing in our general research in programming languages, and towards the goal of mobile software design in particular, is the development of expressive type systems that integrate programming, specification of security policies, and proving that programs respect the policies, into one and the same language. Such an integration is highly desirable for several reasons. First, programs written in a combined language are equipped with the (condensed) proofs of their security. The code consumer can convince themselves of the code security by inspecting this proof – usually a rather simple operation. Second, combining programs and specification leads to better maintainability and reuse of programs and proofs. For example, if a program is shown secure with respect to the specification of some library, it can readily be linked against any version of that library, without requiring potentially expensive refactoring. Third, programming facilities can be brought to use in the production of specifications and proofs. For example, one can implement decision procedures within the system itself, and use them to automate the parts or the whole of the proof development. This makes the proofs themselves relatively short and manageable, instead of being a serious burden on the programmer or the verifier that they are in today's state-of-the-art verification systems.

We have also shown that such mathematics-based security infrastructure can be put to use in practice. For example, in the Mobius project, jointly with France Telecom and INRIA, we have developed automated enforcement mechanisms for flexible security policies for the Mobile Information Device Profile (MIDP, version 2), the de-facto standard for Java-enabled small devices such as mobile phones, set-top boxes, smart-card terminal equipment, etc. The major achievement of the project is showing the feasibility of on-device checking of mathematical proofs, using dedicated checkers developed and extracted from rigorous mathematical formalizations in the proof assistant Coq.

Finding a path through a maze is difficult, but verifying that the path is valid is easy. Same with proofs.



towards "gree

Towards "Greener" Software: Verifying & Controlling Computing Resource Consumption

verifying & controlling computing resource consumption

The conventional understanding of software correctness is absence of errors or bugs with respect to a functional or behavioral specification, i.e., with respect to *what* the program is supposed to compute or do. However, in an increasing number of computer applications the world outside the computer plays an essential role. For example, embedded systems must control and react to the environment, which in turn establishes constraints about the system's behavior like resource usage and reaction times. This makes it necessary to extend the criteria of correctness with new kinds of aspects including upper and lower bounds on execution time, usage of memory, energy consumed, or user defined resources.

The challenge is to extend debugging and verification techniques and tools to deal with resource usage properties, allowing automated performance debugging and certification of programs. This requires developing novel analysis techniques for resources and also improving more conventional analyses for data shapes, data sizes, metrics, aliasing and sharing, etc. Another novel aspect of resource verification is that static checking must generate answers that go beyond the classical outcomes (true/false/unknown). To be useful, these answers must often include conditions under which these classical outcomes are obtained, including input data size or value ranges. For example, it may be possible to say that the outcome is true if the input data size is in a given range.

Resource usage optimization has also become a leading design constraint in current computing devices. As simple examples, in office environments computers and monitors account for the highest energy consumption after lighting, and many mobile devices are limited by battery capacity.

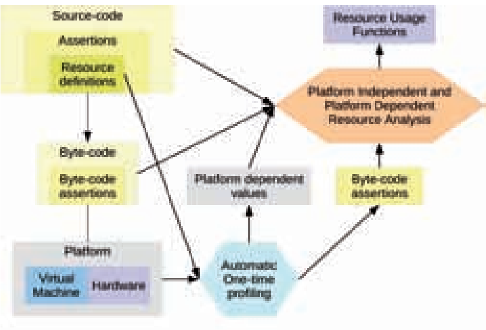
The objective is to develop tools that facilitate the development of "greener" devices, i.e., devices that make a certifiably more efficient use of their available resources. Resource-aware and resource usage-certified programs will allow improving existing devices and applications (like mobile phones and on-board satellite software), enables new uses (like portable medical devices), and reduces the environmental impact of the devices they control.

Researchers from the IMDEA Software Institute are developing state-of-the-art techniques and tools to craft and verify resource-aware software. The pioneering CiaoPP system provides a general framework for computing with high precision the resources consumed by a given piece of software and for debugging/certifying such consumption with respect to specifica-

ner” software

The design of software for mobile devices must consider resource limitations such as battery power. Applications need to optimize their use of available parallelism for power consumption while offering high performance.

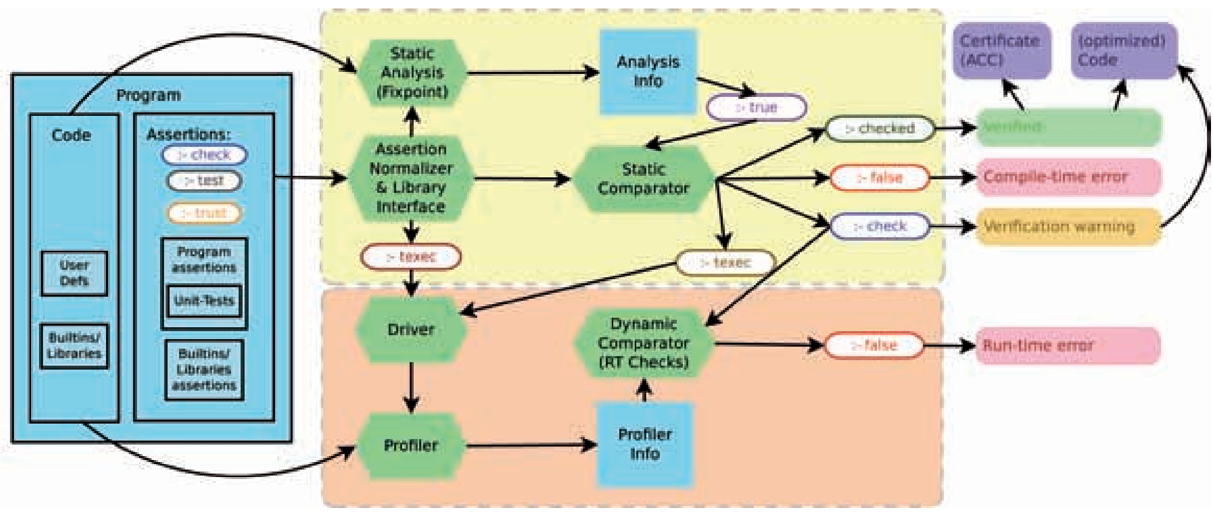
tions. This includes classical concerns like execution time, memory, or disk space as well as other user-defined or platform-dependent resources like energy, network accesses, or opened files. This system has also pioneered analyses whose results are parametric on the values and sizes of inputs, and not only for closed programs. Consequently, it can be used for compositional resource analysis, including libraries and open systems. The platform is highly adaptable to different languages, hardware, and resources because it is built around a customizable static analyzer with a versatile and clear assertion language. Furthermore, within the ES_PASS project IMDEA scientists have shown the applicability of these techniques to automatically determining the resource usage of aerospace software and other embedded industrial applications. This has included demonstrations to the project’s industrial partners and tests on concrete code examples extracted from their application codes.



General framework for resource usage analysis.

Researchers from the IMDEA Software Institute are also pioneering the development of combined hardware/software techniques for increasing the reliability and reducing the power needs of portable medical devices. These applications demand high reliability and a stringent use of resources, which cannot be met with state-of-the-art hardware design and/or conventional software techniques. In this project, IMDEA scientists are using software resource control and parallelization techniques together with specialized hardware which are together aimed at meeting reaction times while decreasing hardware clock frequency in order to meet the power, temperature, and reliability demands. This will enable the design of smaller, more portable, more durable, and more reliable medical devices.

Analysis/verification/testing architecture.



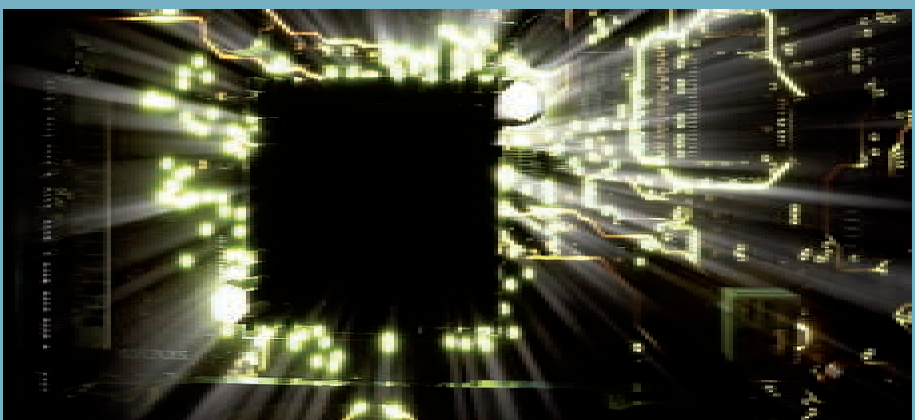
parallelism fo

towards cost-effective exploitation of ubiquitous parallelism

Parallelism for the Masses: Towards Cost-effective Exploitation of Ubiquitous Parallelism

A radical change in computer system architecture is currently taking place. Mainstream microprocessor manufacturers have switched from a monolithic model with a single CPU to a decentralized model with multiple CPU cores in the same chip. *Multicore* processing units with more than 100 computing units are already in production. The reason for this paradigm shift is that intrinsic physical limitations prevent increasing CPU clock frequencies much further. Since new applications always pose increasing performance requirements, parallelism is now being widely used to meet these demands. This paradigm shift is affecting a wide spectrum of computing devices, including laptops, medical devices, smart phones, future automotive and avionics technology, or home entertainment.

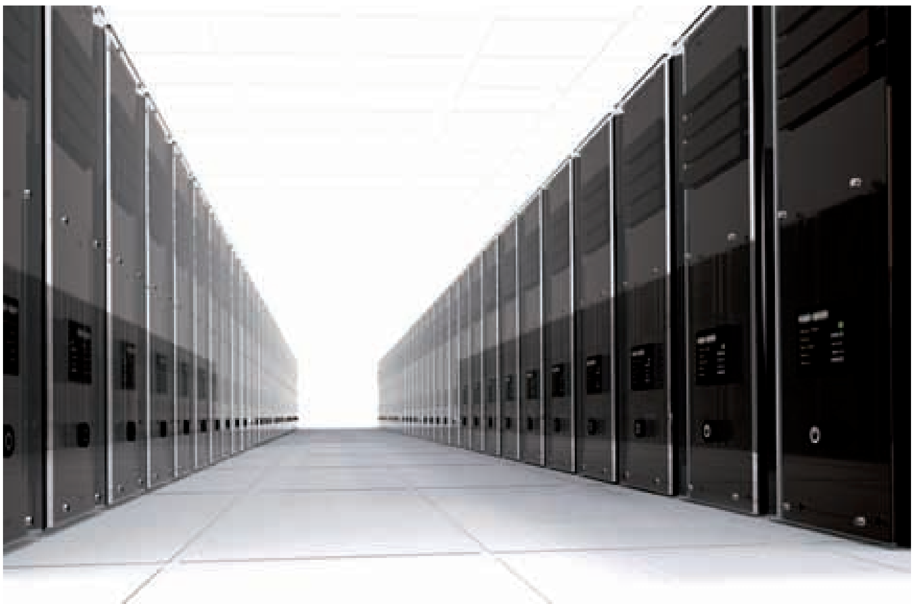
However, current approaches to developing software for these new platforms make poor use of the performance offered by their parallelism. The software industry is in dire need for new approaches for building and verifying multicore software. The problem is that programming highly parallel systems is a very challenging task. One approach consists in taking programs written in traditional, imperative languages and parallelizing them using automated techniques. Although this approach has shown some promises for modest amounts of parallelism the benefits quickly degrade when more processors are added, due to the excessively serial nature of these programming paradigms. The scientific community is thus seeking novel programming paradigms that allow a much higher degree of exploitation of parallelism in a cost-effective way. Another approach consists in writing parallel programs manually but the primitives currently available are in general too low-level and the process too error prone.



r the masses

Scientists at the IMDEA Software Institute are developing two of the more promising approaches to meet these challenges. The first one consists in designing and using more declarative programming languages and developing tools for their automated and user-assisted parallelization. These tools tackle the identification of independent tasks as well as the complex problem of controlling that the tasks generated are of sufficient size, through a combination of static and dynamic resource and data size analyses.

The second is to use programming based on events and tasks which are respectively triggered and completed in no particular order. This approach allows a program to exploit massive parallelism. The challenges stem from the fact that there is no guarantee on the ordering in which the instructions will be executed. Even worse, the underlying architecture provides no guarantee regarding on which CPU core the instructions are going to be executed. For those reasons, parallelism greatly complicates the assessment of correctness of these software systems. In this setting, testing, which is the most used software engineering technique for validating software reliability, provides very little confidence. The problem stems from the limited coverage of testing with respect to the very large spectrum of system's behaviors. By defining and then querying a mathematical model of the system's behavior one can acquire a level of confidence in the system which is not possible with testing. The definition and querying (fully automatically or human-assisted) of those models is another grand challenge that is being addressed at the IMDEA Software Institute.



editor
imdea software institute

graphic design
base 12 diseño y comunicación

D.L.
M-??.???-2010

www.software.imdea.org

La Suma de Todos
Comunidad de Madrid
www.madrid.org

EUROPEAN UNION
STRUCTURAL FUNDS

madrid institute
for advanced studies

institute
iMdea
software

www.software.imdea.org

Contact
software@imdea.org
tel. +34 91 336 37 34
fax +34 91 336 50 18

Facultad de Informática (UPM) · office 3301
Campus Montegancedo
28660 · Boadilla del Monte · Madrid
SPAIN