Jornada de Seguimiento de Proyectos en Tecnologías Informáticas, 2003
Programa Nacional de Tecnologías de la Información y las Comunicaciones

ARGO: Improvement of Processes for Decision-Making in Software Engineering Project Management TIC2001-1143-C03

José Javier Dolado *
Universidad del País Vasco - Euskal Herriko Unibertsitatea

José Riquelme †
Universidad de Sevilla

Javier Tuya ‡
Universidad de Oviedo

Abstract

This coordinated project develops, studies and researches into several key activities for **software engineering project management**. The final goal is to provide the project manager with a set of guides and procedures for carrying out a scientific **decision making** in software engineering projects. In this project, we deal with a wide range of management activities from an integrative viewpoint, including both initial phases of the software life cycle (estimation, simulation and planning) and others implemented at the end of the life cycle (software testing, process assessment, post-mortem analysis, etc.). Besides that, and by using **experimental design**, we study several aspects of the UML modelling and the software maintenance phase.

In the area of **cost estimation**, we intend to improve the estimation process by applying **metaheuristics algorithms**, **simulation** and **data mining**. The combination of these techniques tries to overcome the problem of the lack of data at the beginning of the project and to increase the reliability of the predictions. We validate different behaviours in a software development project with project databases (simulated, actual, commercial). By using data mining, we synthesize the **management rules** more useful for software project management. In the software testing phase we assess, by means of metaheuristics, the **testing strategies** that are economically efficient. The main issue addressed in software testing is test case generation, both for imperative programs and non imperative (SQL and databases). In the area of **process assessment**, we look at the effectivity of the last international proposals (ISO-SPICE) in software development firms.

We use *metaheuristics* techniques and information *visualization* techniques as *horizontal techniques*, meaning that they will pervade the three subprojects. Since the process of decision making over all the life cycle requires of an *information system* that allows to collect and to use the data according to the needs, we will develop a system for data

*Email: dolado@si.ehu.es
†Email: riquelme@lsi.us.es
‡Email: tuya@lsi.uniovi.es

collection and for *decision-making support*. This decision support system will be parameterised specifically for our integrated monitoring of software projects.
Keywords: Software engineering, software project management, cost estimation, metrics, data mining, experimentation, project simulation, information visualization, software testing, process assessment, metaheuristics algorithms

1 Project Overview

Our ARGO project is devoted to the study of the different techniques that can be used by humans to cope with the management decision problems in Software Engineering. We think that the management and decision processes can enhance (or may limit) the improvements obtained by the technical instruments available in software technology. In many cases the successful outcome of a project depends more on the manager than in the tools or techniques used in the project. That is why this project studies a wide range of quantitative techniques which are amenable to be used by the software project manager.

Software Engineering will not be a true engineering discipline until the people who really develop the software uses the quantitative techniques for assessing, monitoring, verifying and controlling the products, resources, quality and processes. The joint interest of the three ARGO subgroups is to improve the methods for managing the quantitative information that software project managers use for getting the project done reliably, on time and within budget.

Our endeavors deal with projects, i.e, how to achieve a goal in which people have to carry out tasks with a final purpose, and with changing situations during the process of development. We found very inspiring the greek myth of the Argonauts: a group of mythological heroes who were assembled to rescue the golden fleece, which was guarded by a dragon. The three subprojects were named according to the three demigods who commanded the ARGO vessel: JASON, HERACLES, IDAS. The project began on 28-Dec-2001 and will end on 27-Dec-2004.

1.1 Subproject 1: Jason. *Implementation of Procedures for Estimation, Experimentation, Assessment and Visualization of Software Engineering Projects*

Among the topics addressed by this subproject are:

- *Experimentation in software engineering*. In our field there are constant proposals of new languages and techniques and there are many occasions where the proposals are accepted without further examination. In other fields, such as medicine or civil engineering, new solutions are not adopted until they have been positively evaluated. Although the number of articles dealing with experimentation is growing, there is still the need for further experimentation in software processes and products [7].

- *Software process assessment*. The goal is to have a set of indexes that allow the organization to monitor the status of its software processes. Then, there is the need of having a model and a set of tools that can help to evaluate the quality of the software houses. We may use the Capability Maturity Model (CMM) and the standard SPICE (ISO/IEC TR-15504), Software Process Improvement and Capability dEtermination. The latter is used as the reference model for applying it in our research.

- *Cost estimation models.* There is no doubt about the importance of estimating and controlling the costs in software development. Basically there are two approaches: parametric models in which the models are driven by data and non-algorithmic methods in which we use heuristics or metaheuristics methods for finding an optimal solution, which is not guaranteed to appear. Neural networks and genetic programming fall in this category.

- *Metaheuristics algorithms applied to software engineering.* The term *metaheuristic algorithm* refers to a set of generic algorithms that can search for optimal solutions in wide multidimensional spaces. We apply them in different situations and purposes (e.g., in cost estimation). Many times, the problems that software engineers face only require a set of good solutions, not an optimal alternative.

1.2 Subproject 2: Heracles. *Definition and Design of a System of Metrics for the Assessment and Estimation of Software Engineering Projects*

- *Databases of software projects.* Given the difficulty of obtaining large amounts of data from actual projects, we simulate dynamic models in order to have databases big enough for post processing. There are real cases in which the decision process is improved by using the data collected [6]. In any of the two situations (actual data or simulated data) the plain question is: Now that we have much data, what can we do with it? The procedures of data mining can deal with large amounts of data.

- *Data Mining.* The concept of Data Mining is defined as the search for *hidden* relationships and global patterns that are embedded within large databases. These relationships, *or rules*, are an invaluable knowledge about the objects related to the data base. However, the main technical problem is that the search for the correct relationships is computationally very expensive. Therefore, we need machine learning and metaheuristics algorithms for adapting the best search strategy.

- *Software Process Improvement.* Current software process models (CMM, SPICE, ...) strongly recommend the application of statistical control and measurement guides to define, implement and evaluate the effects of different process improvement. However, whilst quantitative modelling has been widely used in other fields, it has not been considered enough in the field of software process improvement. During the last decade software process simulation has been used to address a wide diversity of management problems. Some of these problems are related to strategic management, technology adoption, understanding, training and learning, and risk management, among others. We combine traditional estimation models with an intensive utilization of dynamic simulation models of the software process.

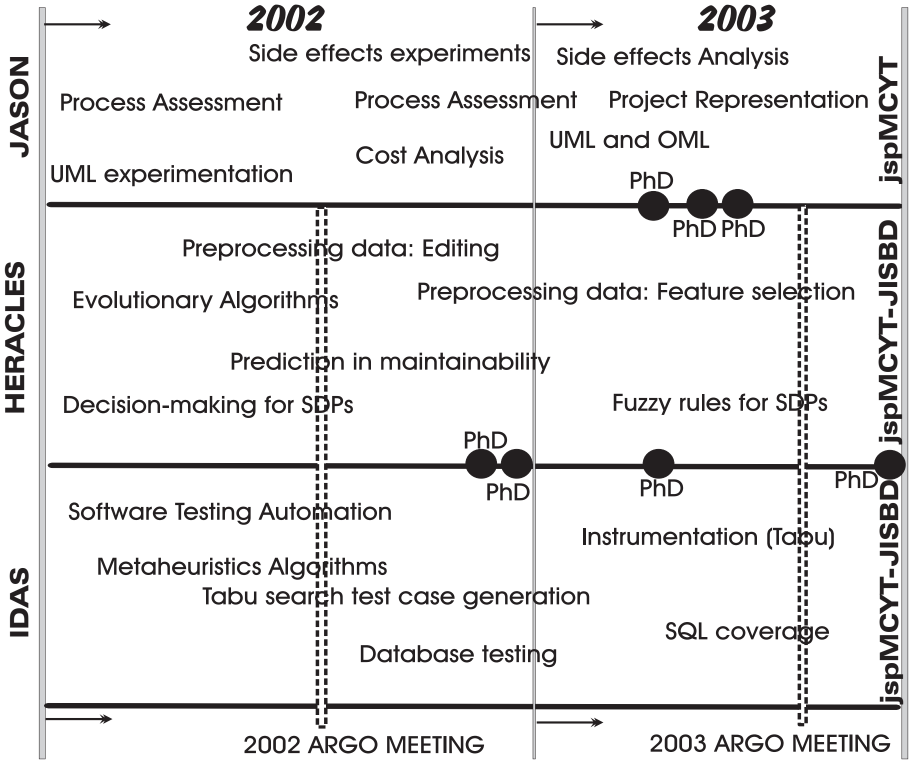

Figure 1: Topics by subgroup.

1.3 Subproject 3: Idas. *Process Improvement in the Monitoring, Verification and Validation of Software Engineering Projects*

- *Software verification and validation (V&V).* Among the activities of V&V, software reviews and testing are the main tasks performed. In case of reviews, and according to the [IEEE Std 1028-1997] there are management reviews, technical reviews, inspections, walkthroughs and auditing. Inspections are a widely acknowlegded useful technique that, unfortunately, suffers from different defects [5].

- *Software testing with metaheuristic algorithms.* Because software testing is an expensive process, there is an increasing interest in the use of different techniques and tools to automate this process. In this project we focus on the automation of test case generation using metaheuristic algorithms [1]. There are previous works in the use of different optimization algorithms and approaches such as mutation testing, but metaheuristic techniques are not much used (with the exception of genetic algorithms).

- *Software testing for database systems.* Although there are some tools that are useful for performing regression testing of database intensive programs, the automated support for test case generation and test case reduction, when a test case is composed by a set of database entries and some input parameters, is a topic not well covered by current research. Our goal is to improve the automation for this testing process when programs are mainly written using SQL sentences and run against a database [2].

- *Software verification and validation process.* The goal is that the definition of processes and metrics for supporting the testing process be integrated with the use of automation tools.

2 Managing the ARGO Project

2.1 Coordinating the Jason-Heracles-Idas subprojects

We have had a very fluent coordination among the projects at the subproject level, and also at the general level, with technical meetings which served as catalysts for new ideas. The core of the whole group had participated in previous projects (CICYT TIC 1179-E, CICYT TIC99-0351 and others), and this has made it easier to have a smooth collaboration in the tasks of organization and communication. We have had two annual meetings for the entire project, within a workshop format, in which the works in progress were presented to the whole group and debated. As result of that interaction new ideas have emerged and some studies have been refocused.

The project has a web page with the information that may be relevant for public interest (http://www.sc.ehu.es/jiwdocoj/argo/argo.html). Also, thanks to the effort put in this project we have continued organizing the Workshop *Apoyo a la Decisión en Ingeniería del Software, ADIS (http://www.sc.ehu.es/jiwdocoj/remis/cfpadis2003.htm)–Decision Support in Software Engineering.*

Figure 1 shows the temporal evolution of the topics researched. The large black dots represent the doctoral thesis that were defended by members of the project. Subgroup Idas has planned several Ph.D. Dissertations for 2004. In July of every year we have the [2002-2003-2004] ARGO MEETING. Figure 2 shows the expected research to be carried out until the end of the project.

2.2 Topics developed in the Jason subproject

The results of this subproject have been focused on

- **Experimentation in Software Engineering.**
  Experiment about *side effects*. We applied a crossover design on different tests involving fragments of C code that included increment and decrement operators. Each test had a side effect version and a side-effect free counterpart. The variables measured in the treatments were the number of correct answers and the time spent in answering. The results show that the side effect operators considered significantly reduce performance in comprehension-related tasks, providing empirical justification for the belief that side effects are harmful.
  Experiments about the dynamic modeling in UML. The goal of the first empirical study was to compare the semantic comprehension of three different notations for

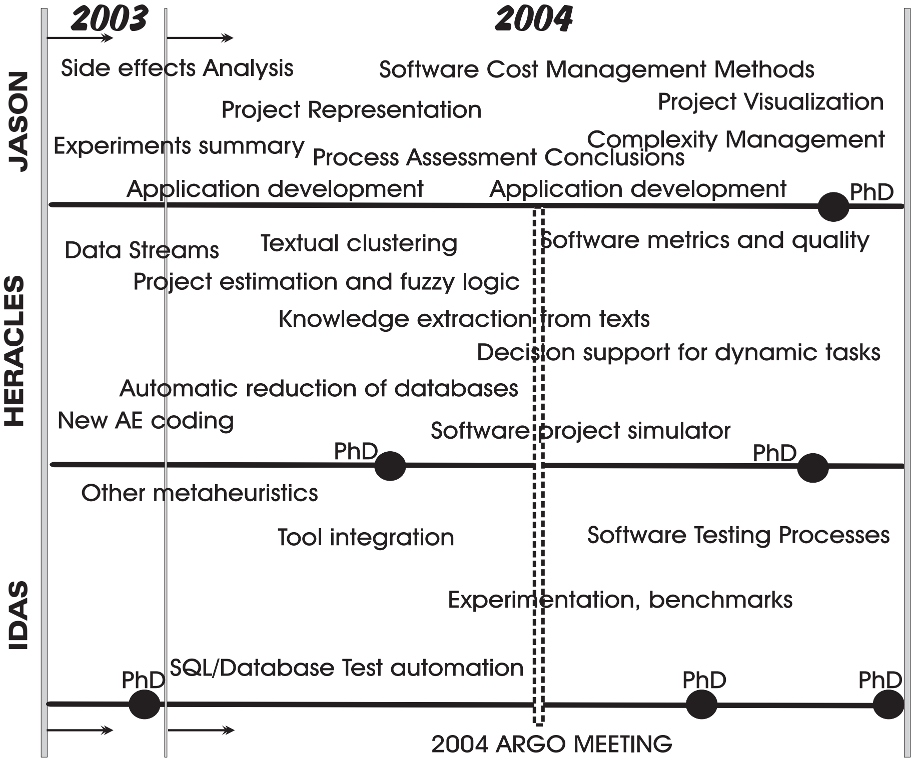

Figure 2: The immediate future of the project.

representing the dynamic behaviour in UML: (a) sequence diagrams, (b) collaboration diagrams, and (c) state diagrams. We performed a 3x3 factorial experimental design with repeated measures. The metrics collected were total time and total score. The main conclusion of this study is that the comprehension of the dynamic modelling in object oriented designs depends on the diagram type and on the complexity of the document. The software project design written in the UML notation is more comprehensible, when the dynamic behaviour is modelled in a sequence diagram. While if it is implemented using a collaboration diagram, the design turns out to be less comprehensible as the application domain, and consequently, the document is more complex.

- **Ontology development**. This work described the initial stages of building an ontology of the activities of software (V&V) as key elements in the management of a software project. There is a need for information in the V&V phases, as many decisions have to be taken with information and data which varies from project to project. The current work has its roots in the issues related to the integration of information in software project management. Among the tasks, data and decisions that the project manager has to deal with, those related to V&V of the software system are critical for a successful outcome. The goal in this work is to model the data, concepts, terms and relations used in the phases, processes and activities of software V&V from the project manager point of view. To do so, we have looked into ontologies in order to model the aforementioned concepts and as a way to integrate the different sources of information.

- **Bayesian networks in software engineering.** Bayesian networks are becoming increasingly popular within the software engineering research community as an effective method of analysing collected data. This work deals with the creation and the use of Bayesian networks and bayesian classifiers in project management. We illustrate this process with examples in the context of software estimation. We highlight some of the difficulties and challenges of using bayesian networks and bayesian classifiers. We discuss how the bayesian approach can be used as a viable technique in Software Engineering in general and for project management in particular, and we also examine the challenges and the open issues.

- **Software process assessment.** The international standard ISO/IEC 15504 provides a guide for software process assessment. Although one of the requirements of this model is its relevancy to all sectors and size of organizations, the limited availability of resources in SMEs (Small and Medium Enterprises) makes its effective application difficult. In this line of research, a method to infer the capability of software lifecycle processes is established. This new method is based on ISO/IEC 15504 and is aimed to define the capability levels of the processes in software SMEs. Ratings are calculated by analysing the answers obtained from a questionnaire which also constitutes an assessment guide for the enterprises to improve their software processes.

2.3 Topics developed in the Heracles subproject

- **Evolutionary algorithms and data mining.** The application of evolutionary algorithms (EA) as search heuristic for data mining techniques is an important topic in the Heracles subproject. EAs have been used to obtain hierarchical set of decision rules with equal accuracy and less number of rules than classical methods. In addition, the EAs have allowed us to discover association rules from numerical attributes and qualitative rules for the decision-making task. We have applied these data mining methods to databases from software development projects (SDP). The aim is to provide decision methods in order to help the project manager to take decisions at any time in the development, by having the adequate number of rules for taking actions.

- **Data preprocessing.** Other research direction consists in improving the data preprocessing techniques. In this sense, we have designed a method of editing (selection of examples or search of representative patterns) with two interesting characteristics: important reduction of the number of instances and lower computational cost. We have also developed a feature selection method (extraction of the most relevant attributes from a database) without distance or statistical calculations.

- **Integrated Framework for Simulation-Based Software Process Improvement.** We have developed an integrated framework for software process improvement according to the CMM. This framework combines traditional estimation static models with an intensive utilization of dynamic simulation models of the software process. The aim of this framework is to support a qualitative and quantitative assessment for software process improvement and decision making to achieve a higher software development process capability according to the CMM. The framework is double-integrated. First, it is based on the systematic integration of dynamic modules to build a dynamic model to model each maturity level proposed in the reference model. As a consequence, a hierarchical set of dynamic models is developed following the same hierarchy of levels suggested in CMM. Second, the dynamic models of the framework are integrated with the use of different static techniques commonly used in planning, control, and process evaluation. The ideas underlying this framework have been materialized in two final products: a hierarchy of dynamic models, and a prototype of a tool to software process improvement design.

- **Decision making in software development projects.** The Simulators of Software Development Projects based on dynamic models have represented a significant advance in front of the traditional techniques of estimating. These simulators enable us to know the evolution of a project before, during and after its execution. But its use in the estimate of the project before beginning the execution has been slowed down by the great number of attributes of the project that it is necessary to know previously. We have presented the improvements that have been added to the simulator developed in our department to facilitate the use of them, and the developed tool, in which we have used data mining and fuzzy logic techniques with the databases generated by the simulator. In this last case, the project manager can know, depending on the decisions that he or she takes, the level of execution of the project objective.

- **Forecasting in software development projects.** The application of certain data mining techniques to numeric databases of SDPs, allows us to obtain qualitative information about the project evolution. Many of these techniques are descriptive, like the clustering, that is why we do not have capacity of forecasting results (project's variables) a priori, from a new data set (project's attributes) of a SDP. To estimate these variables from a new set of values of the attributes, we are proposing some methods (linear regression, k-nearest neighbour, data mining, forecasting), so that the obtained results can be compared between them. The objective is to check if the project variables can be forecasted, from a new series of attributes values, without having to simulate the whole project, and with low error margins.

2.4 Topics developed in the Idas subproject

- **Metaheuristics with Tabu search.** Among the different metaheuristic techniques, we have selected Tabu Search as the first technique to experiment with, because of its capabilities of adaptative memory and responsive exploration. The main effort has been the definition of adequate cost functions, search criteria, memory management (short term and long term tabu lists) and backtracking that would be appropiate to the structure of our problem.
  Tabu search is used to generate test cases for imperative programs using the criteria of branch coverage and has been the subject of an intensive experimentation with programs ranging from few decisions to more than thirty decisions. All results have been compared with random techniques and outperform it (measured in time and test cases generated). Also, Tabu Search reaches the 100% coverage faster than random search. Although it is

difficult to compare to other techniques because there is not a standard set of programs for benchmarking, comparison with other published results that use genetic algoritms gives that tabu search is faster.

- **Tool development.** An instrumentation tool has also been developed. This tool analyzes the program structure and produces the instrumented program, that is plugged into a module with common data structures and the modules including the test case generation algorithm. The resulting executable program is used for test case generation, allowing us to quickly perform the experimentations.

- **Testing SQL programs.** In the area of testing support for SQL programs, the problem of test case generation is more difficult that when we use imperative programs, because of test cases are also composed by database entries. Because of this, the first efforts have been addressed to define some kind of criteria of test completeness. We have defined a coverage metric for database testing that is calculated by taking into account the structure of the SQL query sentence, the relational structure of the data model and the data instances contained in the database. An algorithm has been developed that computes the coverage and supplies advices for:

  - reducing the data instances (rows) in the tables used by the SQL program
  - completing the database with new instances in order to attain a higher coverage
  - detecting possible bugs that could be latent in the database structure.

- **Evaluation of the usefulness of the algorithm and the coverage metric.** It has been performed using an actual database obtained from an industrial maintenance process (the main table is composed by more than a thousand rows), and the algorithm has proven useful for reducing the test database size for detecting additional test cases and for reviewing the data model.

3 Outcomes of the project and main results

We have been very prolific in presenting the works at refereed international conferences, but we have taken care of other activites, such as the global results of Table 1 show. Besides the figures of Table 1, we can say that some of the articles have appeared in the key journals and conferences of our fields. The set of topics researched cover an ample set of quantitative techniques.

3.1 Some published results

Here we list some of the works that represent the topics that we have researched with successful results. The list is not exhaustive (obviously) and only tries to show how the members of the project are progressing. Many publications are indexed in different databases, such as ISI, Compendex, etc. The list of journals of the published works include *Artificial Intelligence Communications, Empirical Software Engineering, IEE Proceedings–Software Engineering, IEEE Trans. on Power Systems, IEEE Trans. on Software Engineering, IEEE Trans. on Systems, Man and Cybernetics (Part B), Information Sciences, Information and Software Technology,*

Type of contribution	Number
International Journals	15
National Journals	1
Ph.D. Dissertations	7
International Conferences	41
National Conferences	29
International Program Committees	15
National Program Committees	6
Events organized	7
PhD Students (grants)	4

Table 1: Current results of the project.

International Journal of Computers, Systems and Signals, Journal of Intelligent and Fuzzy Systems, Journal of Software Maintenance and Evolution, Pattern Recognition, Novática, Revista Colombiana de Computación, Software Quality Journal. For the full information please ask the heads of the project.

1. J.S. Aguilar-Ruiz, J.C. Riquelme and M. Toro, Evolutionary Learning of Hierarchical Decision Rules, *IEEE Systems, Man and Cybernetics Part B*, Vol. 33(2), 324–331, 2003.

2. E. Díaz, J. Tuya and R. Blanco, Automated Software Testing Using a Metaheuristic Technique Based on Tabu Search, *18th IEEE International Conference on Automated Software Engineering,* Montreal, October 2003.

3. J.R. de Diego, J. Tuya, and C. de la Riva, Software Process to specify and verify Reactive Systems. Process and Context generation, *6th World Multiconference on Systemics, Cybernetics and Informatics,* pp. 63–68, Florida, 2002.

4. J.J. Dolado, M. Harman, M.C. Otero and L. Hu, An Empirical Investigation of the Influence of a Type of Side Effects on Program Comprehension, *IEEE Trans. on Software Engineering*, July 2003, Vol. 29, No. 7, pp. 665–670.

5. A. Màs, E. Amengual, Un nuevo método para la aplicación simultánea de ISO/IEC 15504 y ISO 9001:2000 en PYMES de desarrollo de software. *UPGRADE–Novática*, no. 164, julio/agosto, pp. 24–31, 2003.

6. J. Mata, J.L. Alvarez and J.C. Riquelme, Discovering Numeric Association Rules via Evolutionary Algorithm, *Advances in Knowledge Discovery and Data Mining.* Lecture Notes in Artificial Intelligence, Vol. 2336, 40–51, PAKDD 2002.

7. M.C. Otero and J.J. Dolado, An Initial Experimental Assessment of the Dynamic Modelling in UML, *Empirical Software Engineering*, v7, pp. 27–47, 2002.

8. J.C. Riquelme, J.S. Aguilar-Ruiz and M. Toro, Finding representative patterns with ordered projections, *Pattern Recognition*, Vol. 36(4), 1009–1018, 2003.

9. J.C. Riquelme, F. Garijo and M. Toro, Editors. *Special Issue of Journal of Intelligent and Fuzzy Systems*, Vol. 12(3–4) 131–250, 2003.

10. D. Rodríguez, J.J. Dolado and M. Satpathy, Bayesian Networks and Classifiers in Project Management, *2nd Workshop in Series on Empirical Software Engineering*, Roma, September, 2003.

11. M. Ruiz, I. Ramos, M. Toro, A dynamic integrated framework for software process improvement, *Software Quality Journal*, Vol. 10, pp. 181–194, 2002.

12. R. Ruiz, J.C. Riquelme and J.S. Aguilar-Ruiz, Projection-based measure for efficient feature selection *Journal of Intelligent and Fuzzy Systems*, Vol 12(3–4) 175–183, 2002.

13. M.J. Suárez-Cabal and J. Tuya, Improvement of test data by measuring SQL statement coverage, *Workshop on System Testing and Validation,* Amsterdam, September 2003.

3.2 Conferences

The conferences, workshops and seminars in which the results have been presented, so far, are: [2002] ADIS 2002, COMPSAC 2002, DEXA 2002, EASE 2002, ECAI 2002, FSKD 2002, GECCO 2002, IBERAMIA 2002, VI ICPE 2002, ICSM 2002, ICSSEA-SV 2002, IDEAL 2002, IECON 2002, JISBD 2002, KES 2002, MAEB 2002, METRICS 2002, PAKDD 2002, PROFES 2002, PSCC 2002, SAC 2002, SQM 2002, [2003] ADIS 2003, ASE 2003, CAEPIA 2003, DEXA 2003, EPIA 2003, ESMC 2003, GECCO 2003, ICCS 2003, IEA/AIE 2003, ICEIS 2003, JICS 2003, JISBD 2003, MAEB 2003, PROSIM 2003, SERP 2003, SPICE 2003, SAC 2003, STEP 2003, SV 2003, WSESE 2003.

3.3 Events organized

The members of the project have participated and/or are participating in the organization of the following events: IBERAMIA 2002, Workshop AE in MAEB 2003 and 2004, Special Track about Data Streams in ACM SAC 2004, ADIS 2002, ADIS 2003, JICS 2002 and JICS 2003.

3.4 Ph.D. Dissertations

The following researchers, members of ARGO, have defended their corresponding Ph.D.: José Luis Álvarez Macías, Javier Aroba Páez, Óscar Marbán Gallego, Jacinto Mata Vázquez, Mari Carmen Otero Vidal, Daniel Rodríguez García and Mercedes Ruiz Carreira.

4 Conclusion

Throughout this report we have perceived that the people of the ARGO project are delivering substantial results in the fields related to the analysis of quantitative data for project management. We are addressing all the issues that initially motivated our activities. In the near future, the subproject JASON will focus its activities on project representation and visualization, cost analysis of the software projects and its accounting consequences, software metrics and experimentation, process assessment, software complexity, etc. The main goal will be to provide intelligence in decision-making for software cost management. Subproject HERACLES will mainly continue working on two topics: data mining and decision support for software development. Regarding the first one, we are implementing tools for data streams and clustering algorithms for textual information (for example, e-mails). In the second topic, we will analyse software quality metrics and decision-making for dynamic tasks of a SDP. Subproject IDAS will continue the research on test automation techniques (both for imperative and non imperative programs) by means of exploring new techniques and improving the capabilities of the ones already developed. We will experiment with more complex programs and will provide improved automated support to attain a greater easy of use.

We look forward to the new results, which will increase the knowledge about the use of quantitative techniques in software project management. Also, we are envisaging further cooperations, as a group, for the coming years.

References

[1] J. Clarke, J.J. Dolado, M. Harman, R. Hierons, B. Jones, M. Lumkin, B. Mitchell, S. Mancoridis, K. Rees, M. Roper and M. Shepperd, Reformulating Software Engineering as a Search Problem, *IEE Proceedings–Software Engineering* Vol. 150, Issue 03, June 2003, pp. 161–175.

[2] R. A. Davies, R. J. Beynon and B. F. Jones, Automating the testing of databases, *1st Intl Workshop on Automated Program Analysis, Testing and Verification,* pp. 15-20, 2000.

[3] K. El Emam, *Elements of Software Process Assessment and Improvement*, IEEE Computer Society, 1999.

[4] U.M. Fayyad and R. Uthurusamy, Data Mining and KDD, *Comm. of the ACM*, Vol. 39, No. 11, 1996.

[5] F. Macdonald and J. Miller, ASSIST-a tool to support software inspection, *Information and Software Technology*, 41, pp. 1045–1057, 1999.

[6] D.M Raffo and M.I Kellner, Empirical Analysis in Software Process Simulation Modeling, *Journal of Systems and Software*, vol. 47, N^{o} 9, 2000.

[7] M. V. Zelkowitz and D. Wallace, Experimental validation in software engineering, *Information and Software Technology*, vol. 39, n^{o} 11, oct. 1997, pp. 735–743.