Empir Software Eng (2007) 12:517–549
DOI 10.1007/s10664-007-9038-4

Building measure-based prediction models for UML class diagram maintainability

Marcela Genero · Esperanza Manso · Aaron Visaggio · Gerardo Canfora · Mario Piattini

Published online: 21 March 2007

Editor: Dag Sjoberg

Abstract The usefulness of measures for the analysis and design of object oriented (OO) software is increasingly being recognized in the field of software engineering research. In particular, recognition of the need for early indicators of external quality attributes is increasing. We investigate through experimentation whether a collection of UML class diagram measures could be good predictors of two main subcharacteristics of the maintainability of class diagrams: understandability and modifiability. Results obtained from a controlled experiment and a replica support the idea that useful prediction models for class diagrams understandability and modifiability can be built on the basis of early measures, in particular, measures that capture structural complexity through associations and generalizations. Moreover, these measures seem to be correlated with the subjective perception of the subjects about the complexity of the diagrams. This fact shows, to some extent, that the objective measures capture the same aspects as the subjective ones. However, despite our encouraging findings, further empirical studies, especially using data

M. Genero (✉) · M. Piattini
ALARCOS Research Group, Department of Technologies and Information Systems,
University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
e-mail: Marcela.Genero@uclm.es

M. Piattini
e-mail: Mario.Piattini@uclm.es

E. Manso
GIRO Research Group, Department of Computer Science, University of Valladolid,
Campus Miguel Delibes, E.T.I.C., 47011 Valladolid, Spain
e-mail: manso@infor.uva.es

A. Visaggio · G. Canfora
RCOST—Research Centre on Software Technology, University of Sannio,
Pal. Ex Poste, viale Traiano, 82100 Benevento, Italy

A. Visaggio
e-mail: visaggio@unisannio.it

G. Canfora
e-mail: canfora@unisannio.it

taken from real projects performed in industrial settings, are needed. Such further study will yield a comprehensive body of knowledge and experience about building prediction models for understandability and modifiability.

Keywords Maintainability · Understandability · Modifiability · UML · Class diagrams · Structural complexity · Size · Measures · Empirical validation · Controlled experiments · Prediction model

1 Introduction

Nowadays, the idea that "measuring quality is the key to developing high-quality OO software" is gaining acceptance (Schneidewind 2002). Moreover, in the field of software engineering, it is widely recognized that, in order to develop high-quality software products, the focus should be on measuring the quality of the models that are built at the very beginning of OO software analysis and design (Schneidewind 2002; Briand et al. 1998; Card et al. 2001; Briand et al. 2000; Briand and Wüst 2001, 2002; Bansiya and Davis 2002; Fioravanti and Nesi 2001). This idea received further emphasis with the Model Driven Development (MDD) paradigm (Atkinson and Kühne 2003), in which development effort is focused on the design of models, rather than on coding.

Of course, MDD methods are only as good as the models they help us construct (Selic 2003). In model-driven software engineering, the quality of the models used is extremely important, because it is this that will ultimately determine the quality of the software systems produced.

With respect to the quality of software models, maintainability is of particular interest. As problem domains and software system solutions evolve, so must their models. Models must be sufficiently comprehensible and flexible that modifications that reflect changes in the things they model can be incorporated easily.

Maintainability is an external quality attribute. As such, it can be evaluated directly only when the product is nearly or completely finished. Consequently, in order to make an early evaluation of maintainability, it is necessary to make further indicators available. These indicators should be based on properties of early artifacts (Briand et al. 2000), e.g., the structural properties of class diagrams specified using the Unified Modeling Language (UML) (OMG 2005).[1] UML class diagrams are the key outcome of those early phases and the foundation for all later design and implementation work. Although class diagrams are only one of nine diagram types included in UML, survey research has shown that it is perceived by practitioners as the most important diagram type (Erickson and Siau 2004). Changes in the structure of the system (e.g. because of changing system requirements) need to be reflected by modifications of the class diagram(s); hence the importance of developing diagrams that can easily incorporate changes.

The theoretical basis for developing quantitative models relating to structural properties and external quality attributes has been provided by Briand et al. (1999) (see Fig. 1). This work is the basis for a great amount of empirical research in the area of structural properties of software artifacts (El-Emam 1999, 2001; Poels and Dedene 2000). In the study reported herein, we assumed a similar representation for UML class diagrams. We hypothesized that

[1] Even though we began our research with UML 1.4 (OMG 2001), when UML 2.0 appeared we studied it carefully. However, we did not find differences related to the elements we considered in class diagrams.

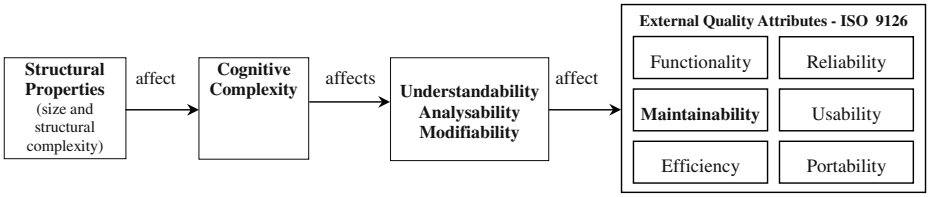


Fig. 1 Relationship between structural properties, cognitive complexity, and external quality attributes, based on Briand et al. (1999) and ISO (2001)

the structural properties (such as structural complexity and size) of a UML class diagram have an effect on its cognitive complexity. Cognitive complexity can be defined as the mental burden placed by the artefact on the people who have to deal with it (e.g. developers, testers, maintainers). High cognitive complexity will result in the production of an artifact that has reduced understandability, which will, in turn, produce undesirable external qualities, such as decreased maintainability.

We proposed a set of eight measures for the structural complexity of UML class diagrams (Genero et al. 2000, 2005; Genero 2002). The proposed measures were based on (1) a theoretical analysis of the ontological structure of UML class diagrams and (2) a review of the literature about the existing measures that can be used to measure the structural complexity and size of UML class diagrams in the initial phases of an OO software development life cycle (Li and Henry 1993; Brito e Abreu and Carapuça 1994; Chidamber and Kemerer 1994; Briand et al. 1997; Marchesi 1998; Bansiya and Davis 2002) The proposed measures are related to the usage of UML relationships, such as associations, dependencies, aggregations and generalizations.[2] In the study reported herein, we also considered traditional OO measures, such as size measures (see Table 1). In what follows, the abbreviations for the measure names will be used.

These measures were validated theoretically according to the DISTANCE framework (Poels and Dedene 2000), in order to guarantee the construct validity of the empirical studies in which these measures were used.

1.1 Objectives of this Research

The proposal of measures is of no value if their practical use is not shown empirically (Basili et al. 1999; Kitchenham et al. 1995; Schneidewind 1992; Cantone and Donzelli 2000). Hence, our main motivation was to investigate, through experimentation, whether the measures we proposed for UML class diagram structural complexity and size could be good predictors of two class diagram characteristics that are related to maintainability: understandability and modifiability. If the predictive power of the proposed measures were to be corroborated by several empirical studies, we really would have identified early indicators of class diagram understandability and modifiability. These indicators would allow OO software designers to make better decisions early in the software development life cycle, thus contributing to the development of better quality OO software. From a

[2] These measures have been defined in a methodological way following a method proposed by Calero et al. (2001), which consists of three main tasks: metric definition, theoretical and empirical validation. However, in this paper, we focus only on empirical validation. Work related to the definition of measures and to theoretical validation can be found in Genero (2002).

Table 1 Measures for UML class diagrams

	Measure name	Measure definition
Size measures	Number of Classes (NC)	The total number of classes in a class diagram. This measure corresponds to Cs (Henderson-Sellers 1996), OA1 (Marchesi 1998) and DSC (Design Size in Classes) (Bansiya and Davis 2002). This is the most common measure of class diagram complexity used in the literature.
	Number of Attributes (NA)	The number of attributes defined across all classes in a class diagram (not including inherited attributes or attributes defined within methods). This includes attributes defined at class and instance level. This is a generalization of the NOA (Number of Attributes) measure (Henderson-Sellers 1996) to class diagram level.
	Number of Methods (NM)	The total number of methods defined across all classes in a class diagram, not including inherited methods (as this would lead to double counting). This includes methods defined at class and instance level. This is a modification of the WMC (Weighted Method per Class) measure (Chidamber and Kemerer 1994), in which (a) the weightings of all methods are 1, and (b) the value is totalled across all classes (as WMC is a class level measure). Weightings are not applicable at the analysis stage because the code for methods is not available at this stage. This also corresponds to the NOM (Number of Methods) measure (Bansiya and Davis 2002; Li and Henry 1993) and the combination of NIM (Number of Instance Methods) and NCM (Number of Class Methods) (Lorenz and Kidd 1994). All of these measures are defined at class level.
Structural complexity measures	Number of Associations (NAssoc)	The total number of association relationships in a class diagram. This is a generalization of the NAS (Number of Associations) measure (Harrison et al. 2000) at class diagram level.
	Number of Aggregations (NAgg)	The total number of aggregation relationships (each "whole-part" pair in an aggregation relationship).
	Number of Dependencies (NDep)	The total number of dependency relationships.
	Number of Generalizations (NGen)	The total number of generalization relationships (each "parent-child" pair in a generalization relationship).
	Number of Generalization Hierarchies (NGenH)	The total number of generalization hierarchies, i.e. it counts the total number of structures with generalization relationships.
	Number on Generalization Hierarchies (NAggH)	The total number of aggregation hierarchies, i.e. it counts the total numbers of "whole-part" structures within a class diagram.
	Maximum DIT (MaxDIT).	The maximum DIT value obtained for each class of the class diagram. The DIT value for a class within a generalization hierarchy is the longest path from the class to the root of the hierarchy (Chidamber and Kemerer 1994).
	Maximum HAgg (MaxHAgg)	The maximum HAgg value obtained for each class of the class diagram. The HAgg value for a class within an aggregation hierarchy is the longest path from the class to the leaves.

practical perspective, the availability of early applicable measures would allow designers or modellers to do the following:

- Make a quantitative comparison of design alternatives, and therefore an objective selection between several class diagram alternatives.
- Assess class diagram understandability and modifiability at an early stage, even during the modelling activity, and, on the basis of this assessment, allocate resources more effectively (e.g. by redesigning high-risk diagrams with respect to understandability or modifiability).

More specifically, our study had the following objectives:

- To find prediction models that relate the measures shown in Table 1 to measures of maintainability. This was undertaken using the data obtained through a controlled experiment carried out by students at the University of Castilla-La Mancha in Spain (henceforth, the Spanish experiment).
- To confirm our findings using the data obtained through a replication of the experiment undertaken with students at the University of Sannio in Italy (henceforth, the Italian experiment).
- To evaluate the "goodness of fit" of the obtained models.
- To compare the findings of the current study with a previous study (Genero et al. 2003a).

First, related work is described and compared to the objectives of the current work. Sections 3 and 4 contain descriptions of the original experiment carried out in Spain and its replica carried out in Italy, respectively. Section 5 presents the analysis and interpretation of the data. Section 6 analyses possible threats to validity. Section 7 compares the current study with a previous one is presented. Section 8 concludes and outlines directions for future research.

2 Related Work

Several reviews of the most up-to-date empirical studies related to OO measures (Basili et al. 1999; Kitchenham et al. 1995; Schneidewind 1992; Cantone and Donzelli 2000; Briand and Wüst 2002) have shown that the dependent variables that are investigated most frequently are fault-proneness (probability of fault detection), the number of faults or changes in a class, the effort of various development activities, and expert opinion about the psychological complexity of a class.

Many papers focus on maintainability as a dependent variable (Li and Henry 1993; Harrison et al. 2000; Fioravanti and Nesi 2001; Briand et al. 2001), with the principal purpose of proposing prediction models for maintenance tasks. In most of these studies, the independent variables were measured on the basis of the source code and not on the basis of UML class diagrams, and for this reason, the predictions were made later in the development. Nevertheless, several experts, such as Briand and Wüst (2001, 2002) and Card et al. (2001), among others, have noted that the earlier the measurement is performed, the better.

Even though a number of studies define measures for UML class diagrams, most of them (Kim and Boldyreff 2002; Si-Said et al. 2002) do not go beyond the definition step. An exception is the research done by Kiewkanya and Muenchaisiri (2004) and Kiewkanya et al. (2004). They carried out a controlled experiment with undergraduate students, with the purpose of building models for the maintenance level (easy, medium and difficult) of

OO software design, represented by UML class diagrams and sequence diagrams. The models were based on measure values calculated from UML class diagrams and sequence diagrams. They used most of the measures we presented in Table 1. Their study is based on an isolated experiment, and to the best of our knowledge, the authors did not pursue further work to validate the models they proposed. As we noted above, the lack of empirically validated measures that can be used as early indicators of UML class diagrams maintainability was the motivation for a project we have been pursuing for the last 3 years. For that reason, all the related work focuses on our own research.

Here, we will briefly summarise our previous work:

- In Genero et al. (2001), we presented an experiment where subjects were given 24 class diagrams and were asked to evaluate subjectively certain subcharacteristics related to maintainability. Despite the subjective nature of the evaluations, the preliminary findings were encouraging. Statistical analysis revealed that the measures we proposed are highly correlated with maintainability subcharacteristics.
- In Genero et al. (2002), we presented an experiment and its replica, in which subjects were given nine class diagrams and asked to modify them to satisfy new requirements. We found that measures related to aggregation and generalization relationships correlate strongly with modifiability, correctness and completeness.
- In Genero et al. (2003b), we described an experiment in which subjects were given nine class diagrams and were asked to modify them according to three new requirements; in addition, they had to write down the time spent on performing the modifications. We found that the time spent on maintenance seemed to be correlated with all measures, except those related to the number of dependencies.
- In Manso et al. (2003), we carried out a Principal Component Analysis (PCA) based on the data collected from three controlled experiments, in which we attempted to discover which of the measures used might not be redundant. After performing the PCA, we observed from the three data samples that the PCs that contained nonredundant information presented well-known characteristics of OO design, concerning the usage of UML relationships, such as associations, dependencies, aggregations and generalizations. The measures relating to size (such as NC and NM) did not seem to be relevant, while NA was relevant.
- In Genero et al. (2003a), we presented a study similar to the current one. We stated prediction models for different measures of class diagram maintainability, such as Understandability Time, Modifiability Time, Modifiability Correctness and Modifiability Completeness. These models were based on the 11 measures we proposed (see Table 1). The data was taken from one experiment carried out by students in Spain and its replication performed by students in Italy. After a multivariate analysis, we concluded that Understandability Time, Modifiability Correctness and Modifiability Completeness are related to the structural complexity measures NAggH and NGenH, and to one of the size measures, NA. We did not obtain models to explain Modifiability, because there was insufficient linear correlation between it and the measures. Furthermore, variable transformations did not improve the results. The predictive accuracy of the models was good, especially with regard to Modifiability Completeness and Modifiability Correctness. As this study has characteristics similar to the current one, we will compare the two in Section 7.

This paper contributes to our knowledge regarding the practical utility of measures for the structural complexity and size of class diagrams. We do so by discussing one

experiment and its replica, focusing on the usefulness of early measures (see Table 1) to predict the maintainability of UML class diagrams. We took the following measures to increase internal and external validity:

- We used material that is more representative of real cases.
- We tried to cover a wide range of measure values.
- We carried out the experiment in a more controlled environment than in the previous experiments. The subjects were allocated to a room and the execution of the experiment was supervised by a professor.
- We improved the experimental tasks by trying to specify modifications that are similar to those required in real projects.

3 Experiment Description (Spanish Experiment)

We now present a description of the experimental process we followed when carrying out the experiment, following some suggestions provided in (Wohlin et al. 2000; Juristo and Moreno 2001; Kitchenham et al. 2002).

3.1 Definition

Using the GQM template (Basili and Rombach 1988; Basili and Weiss 1984; Van Solingen and Berghout 1999), we defined the goal of our experiment as follows:

Analyse	*UML class diagram structural complexity and size measures*
For the purpose of	*evaluating*
With respect to	*the capability to be used as indicators of UML class diagram understandability and modifiability*
From the point of view of	*the researchers*
In the context of	*Undergraduate Computer Science students at the Department of Information Systems and Technologies at the University of Castilla-La Mancha*

3.2 Planning

The planning phase can be divided into the following steps: context selection, selection of subjects, variable selection, hypotheses formulation, and experiment design.

Context Selection The participants were a group of undergraduate students and hence, the experiment was not run within an industrial software development environment. The subjects were 38 students enrolled in the third year of Computer Science at the Department of Information Systems and Technologies at the University of Castilla-La Mancha in Ciudad Real, Spain. The ability to generalize from this specific context is discussed below, in Section 6. The experiment addressed a real problem, i.e. what indicators may or should be used for assessing and estimating the maintainability of class diagrams. Keeping this in mind, it investigated the correlation between (a) class diagram structural complexity and size measures and (b) two subcharacteristics of maintainability: understandability and modifiability.

Table 2 Linguistic labels for complexity

Five linguistic labels				
Very simple	Relatively simple	Average	Slightly complex	Very complex

Selection of Subjects We selected all the students in the available classes. The subjects had approximately 6 months of experience in designing UML class diagrams, but had 2 years experience in the OO paradigm because they began programming with Java in the first year of their degree.

Variable Selection The independent variables were structural complexity and size of UML class diagram, measured by the measures we proposed (see Table 1). The dependent variables were the following two subcharacteristics of maintainability.[3]

- Understandability: the capability of the class diagram to be understood by its users (such as requirement engineers, data analysts, and database designers).
- Modifiability, which includes two of the subcharacteristics of maintainability that were proposed in the ISO 9126 standard for software quality (ISO 2001):
  - Analysability: The capability of the class diagram to be diagnosed for deficiencies or for the identification of the parts to be modified.
  - Changeability: The capability of the class diagram to be changed when modifications are required.

The dependent variables were measured by two main measures collected from the experimental tasks:

- *Understandability Time* and *Modifiability Time*, which reflect the time, in seconds, that the subjects spent on the experimental tasks.
- *SubComp*, which reflects the subjective perception of the subjects about the complexity of each diagram. It was measured via the judgement of the participants about how complex they find the diagrams (an ordinal scale), according to five linguistic labels (see Table 2).

Hypothesis Formulation Our aim was to test the following hypotheses:

Hypothesis 1 the structural complexity measures (NAssoc, NAgg, NDep, NGen, NAggH, NGenH, MaxHAgg, MaxDIT) and the size measures (NC, NA, NM) are good predictors of *Understandability* and *Modifiability Time*.

Multivariate analysis is commonly used in software engineering experimentation, (Briand and Wüst 2002; Mendes et al. 2002), because it not only examines the relationships between independent (Y) and dependent (X_j) variables, but also considers the independent

[3] Even though understandability was not considered as a subcharacteristic of maintainability by ISO 9126 (2001), we included it because there exists a lot of work related to software measurement that considers understandability as a factor that influences maintainability (Briand et al. 2001; Fenton and Pfleeger 1997; Harrison et al. 2000).

variables by combining them as covariates in a multivariate model, in order to better explain the variance of the dependent variables and ultimately obtain accurate predictions. If the model is linear it means that the expected value of Y increases in β_j per each unit that X_j increases. The null hypothesis is that Y and X_j do not have linear dependence, so β_j is zero. That being so, we selected the Multivariate Linear Model to test Hypothesis 1:

$$Y = \mu + \sum_{j=1}^{r} \beta_j X_j + \varepsilon \qquad \text{(Where the variables are explained below)} \qquad (1)$$

As we would obtain repeated measures for the *Understandability* and *Modifiability Time*, due to the fact that the same UML class diagram would be assigned to several subjects, the obtained data would not be independent. Hence, we decided to use Generalized Estimating Equations (GEE) (Hardin and Hilbe 2002; Davis 2002), an extension of General Linear Models (GLM) that accommodates data that are correlated within clusters; in this case, the class diagrams. In fact, the GEE method allows the incorporation of correlations among observations into the estimation process (Vokác et al. 2004).

Summarizing, the GEE model assumes that:

1. There are several dependent variable measurements $Y_{i1}...Y_{\text{in}}$ in the same class diagram (CD_i).
2. There are J independent variables (X_{ij}) that take values in each CD_i, which could explain the dependent variable Y. In our case, we begin considering the 11 measures shown in Table 1.
3. The "N" measurements within a CD are correlated.
4. The observations among different CDs are uncorrelated.

The GEE method has several steps (Davis 2002):

1. Relate a function of the dependent variable marginal mean $g(Y_{ij})$, called link function, with a linear combination of the independent variables, in this case the measures for CD_i, that is the model (1) explained for each $g(Y_{ij})$. The link function for *Understandability* and *Modifiability Time* was $g(\mu_{ij})=\mu_{ij}$, and whenever needed $\text{Log}(\mu_{ij})$.
2. Describe the variance of Y_{ij} as a function of the mean (this variance is predetermined for dependent variables with a Normal distribution (Davis 2002)).
3. Choose the form of the working correlation matrix, i.e. the covariance matrix of repeated measures. In our case (following the recommendation of Davis and Hardin (Davis 2002; Hardin and Hilbe 2002)) for repeated measurements without natural order, the working correlation matrix was described via the exchangeability matrix. Thus, the model assumes that the correlation between measurements from two different subjects is constant.

The GEE uses the empirical or robust sandwich estimator of the working matrix, which has the advantage of tending asymptotically to the true correlation matrix. Furthermore, the estimates of this GEE model are asymptotically Normal distributed with the specified standard error, and converge to the true values when the sample size becomes large. However, this fact does not preclude the choices of the working correlation matrix and distribution from having an influence on the results. The closer the correlation matrix is to reality, the more accurate the estimates will be. Thus, the null statistical hypothesis to test

model (1) is that the marginal mean $g(Y_{ij})$ is not explained by a linear combination of the independent variables:

$$\mathbf{H}_{0,1} : \beta_j = 0 \forall_j \quad \mathbf{H}_{1,1} : \neg H_{0,1}$$

Hypothesis 2 We also aimed at investigating the degree of correlation between diagram-subjective complexity expressed by *SubComp* and the measures we proposed (see Table 1) through the correlation coefficient (ρ). We could consider *SubComp* as a subjective measure of cognitive complexity. We used the Spearman and the Kendall correlation coefficients (ρ), as the variables scale require. Then, the null hypothesis is that there is no correlation between *SubComp* and each measure we proposed. The null and the alternative hypotheses may be stated as:

$$\mathbf{H}_{0,2} : \rho = 0 \quad \mathbf{H}_{1,2} : \neg H_{0,2}$$

Then this hypothesis links the structural properties box with the cognitive complexity box (see Fig. 1).

Hypothesis 3 We wanted to determine whether each subject's subjective perception of the complexity of each diagram, expressed by *SubComp*, was influenced by the time the subjects spent performing understanding and modifying tasks. Therefore, we formulated the null hypothesis using the Spearman or the Kendall correlation coefficient, that there is no correlation between subjective complexity (*SubComp*) and *Understandability* (ρ_{SU}) or *Modifiability* (ρ_{SM}) *Time*. These hypotheses may be stated as :

$$\mathbf{H}_{0,3\_i} : \rho_{Si} = 0 \quad \mathbf{H}_{1,3\_i} : \neg \mathbf{H}_{0,3\_i} \quad \text{where } i \in \{U, M\}$$

This hypothesis links the cognitive complexity box with the understandability, analysability and modifiability box of Fig. 1.

Experiment Design Given that we had more than one independent variable, it may be thought that a factorial design using a crossed design should be used. However, that would have been impractical, because we would have had to design n^{11} different cases (n being the levels we decided for the independent variables). Moreover, in some cases we might have obtained meaningless class diagrams, which can never appear in real cases. So, we decided to consider nine class diagrams and tried to cover a wide range of measure values (see Table 5).

In addition, when designing the experiment, we had to consider the statistical analysis we wanted to carry out with the empirical data. As we intended to use multivariate regression models, which require data independence to perform a single between-subjects design experiment, each subject was assigned only one test related to one class diagram.

3.3 Operation

The operational phase was divided into three steps: preparation, execution and data validation.

Preparation We decided to have nine UML class diagrams from different application domains, designed to cover a wide range of the measure values (see Table 5). These diagrams were taken from real cases, for example, students' examinations.

An example of the experimental material is shown in Appendix A (Diagram D3). Each diagram had an enclosed test that included a brief description of what the diagram represented, and three tasks:

1. Understandability tasks, where subjects had to complete a questionnaire (four questions) that was designed to determine whether or not they had understood each diagram. They also had to note down how long it took to for them answer the questions. The *Understandability Time*, expressed in seconds, was obtained from that.
2. Modifiability tasks, where subjects had to modify the class diagrams according to four new requirements, and specify the start and end time. The difference between both times is what we called *Modifiability Time* (expressed in seconds). The *Modifiability Time* includes both the time spent analysing what modifications had to be made and the time needed to make them.[4] The activities that the subjects had to carry out are considered as "enhancive maintenance", according to the types of software maintenance proposed by Chapin et al. (2001). The tasks for each class diagram were similar, including adding or replacing attributes, methods, classes, etc.
3. Rating tasks, in which each subject had to rate the complexity of each diagram using a scale that consisted of five linguistic labels (see Table 2). Thus, we obtained the values for the measure *SubComp*. A similar rating task was required in (Harrison et al. 1998), but for rating the complexity of a class.[5]

The subjects were given an intensive training session before the experiment took place. Moreover, we evaluated the experience of the subjects, by giving them one test similar to those which we used in the experiment. Once this test was finished, we collected the data. We added up the *Understandability* and the *Modifiability Time* and according to this time value, we formed two groups: G1 (which included the subjects who spent less time) and G2 (which included the subjects who spent more time). We placed 19 subjects in each group. In order to avoid the bias produced by the experience of the students, we intended to distribute each diagram to the same number of subjects of G1 and G2 (blocked design by experience).

Execution After the training session, the experiment was executed. The subjects were allocated to a room and supervised by a professor. They were randomly assigned one class diagram on paper. As we expressed before, we tried to assign each diagram to the same number of subjects (see Table 3).[6] The professor who monitored the experiment explained to them how to do the tests. They were allowed a total of one hour to complete them.

Data Validation We collected all the data, including the *Understandability Time*, the *Modifiability Time* and the subjects' rating about the complexity of each diagram obtained from the responses of the tests. The measure values were calculated automatically by means of a measuring tool that we designed (García et al. 2003).

[4] We considered the time spent analysing what modifications had to be made and the time needed to perform them together, because it is difficult to separate both times in a reliable way.

[5] For carrying out the data analysis, we assigned numbers to each linguistic level as follows: "Very simple" corresponded to 1 and "Very complex" corresponded to 5.

[6] Most of the diagrams were assigned to four subjects, but diagrams E3 and M3 were assigned to five subjects because there were 38 subjects and nine diagrams. The subjects were our students and we could not exclude two of them solely on the basis of design considerations.

The time measures are meaningless unless a minimum level of quality is delivered. Hence, we considered the following correctness and completeness measures, which were used to exclude those observations that did not fulfil a minimum quality requirement (observations with less than 75% correctness and completeness were not included in the data analysis):

- *Understandability Correctness* = Number of correct answers/Number of answers.
- *Understandability Completeness* = Number of correct answers/Number of questions.
- *Modifiability Correctness* = Number of correct modifications/Number of modifications done.
- *Modifiability Completeness* = Number of correct modifications/Number of modifications required.

Hence, we only considered 38 subjects for understandability tasks and 29 subjects for modifiability tasks.

4 Replication of the Experiment (Italian Experiment)

The experiment we conducted in Italy tested exactly the same hypotheses as the basic experiment (Brooks et al. 1996; Basili et al. 1999). Hence, we see it as a replication. The main characteristics of this replica were as follows:

- The subjects were 23 undergraduate students enrolled in the final year of Computer Science at the University of Sannio, in Benevento, Italy.
- By the time the experiment was carried out, all of the students had taken two courses on Software Engineering, in which they learned in depth how to design UML class diagrams.
- The experimental material upon which the subjects had to work was translated into English (see Appendix A for an example), so the subjects did not work in their native language. This fact might have biased the results, because not all the subjects were sufficiently skilled in the use of English and occasionally needed extra time to ask the professor who was monitoring the experiment about the meaning of some statements.
- As in Spanish experiment, we selected a between-subjects and blocked design by experience (as balanced as possible because of the number of available subjects). Table 4 shows the assignment of subjects to diagrams (tests).

Table 3 Assignment of subjects to diagrams in the Spanish experiment

Diagrams	Subjects
E1	24, 25, 34, 37
E2	26, 27, 28, 29
E3	5, 30, 31, 32, 33
M1	3, 17, 18, 35
M2	7, 8, 19, 20
M3	4, 6, 21, 22, 23
D1	1, 9,10, 36
D2	2, 12, 13, 14
D3	11, 15, 6, 38

Table 4 Assignment of diagrams to subjects in the Italian experiment

Diagrams	Subjects
E1	12, 18
E2	3, 8
E3	5, 10, 15
M1	6, 13, 20
M2	2, 23
M3	4, 9, 17
D1	1, 14, 19
D2	22,16
D3	7, 11, 21

– As in the Spanish experiment, subjects were given an intensive training session before the experiment was executed. We allowed the subjects 1 hour to perform the tasks assigned in the experiment.

After running the replication, we collected all the empirical data. For obtaining the prediction models we excluded two subjects, who had *Understandability Correctness* and *Completeness* scores less than 0.75. These subjects completed only 50% of the tasks, but they solved them correctly. We surmise that the low percentage of tasks completed was due to language problems.

Analysing *Modifiability Correctness* and *Completeness*, we also excluded two subjects to obtain the prediction models, as in the previous case. In all, we only considered 21 subjects for studying the *Understandability* and *Modifiability Time* in the Italian experiment.

5 Data Analysis and Interpretation

The data we want to analyse in the Spanish sample covers 38 and 29 subjects for *Understandability* and *Modifiability Time*, respectively. In the Italian sample it covers 21 subjects for both dependent variables. We considered both data sets (Spanish and Italian) to study the hypotheses of interest described in Section 3.2, because the behaviour could be different across the populations.

We will analyse the data in the following steps[7]:

1. We present the following descriptive studies:

   - To characterize the independent variables (the measures we presented in Table 1) we undertake a descriptive study, showing the range of the measure values for the class diagrams considered in the experiment (Section 5.1.1).
   - To assess the degree of correlation between the measures, we perform a PCA (Section 5.1.2). From the PCA results it is possible to find new uncorrelated dimensions, which information is useful for selecting the measures for testing Hypothesis 1.
   - In order to understand their behaviour and to study similarities or differences between the two populations, we also carry out a descriptive study with the dependent variables (Section 5.1.3).

[7] All the data analysis was carried out using SPSS 12.0 (SPSS 2002) and SAS 8.0 (SAS 1999).

Table 5 Measure values for each UML class diagram

DIAGRAM	Measure values										
	NC	NA	NM	NAssoc	NAgg	NDep	NGen	NGenH	NAggH	MaxDIT	MaxHAgg
E1	6	28	52	5	0	0	0	0	0	0	0
E2	7	23	60	3	1	1	2	1	1	1	1
E3	7	19	39	3	3	1	1	1	1	1	3
M1	13	39	96	6	5	1	4	2	2	1	4
M2	9	31	78	6	3	1	0	0	2	0	2
M3	14	27	60	11	0	0	6	1	0	2	0
D1	30	54	128	12	7	3	17	1	3	4	4
D2	52	76	35	15	19	8	21	7	2	4	7
D3	39	65	71	11	6	8	23	2	3	3	2
SUMMARY											
Mean±SE	19.7± 5.6	40.22± 6.7	68.8± 9.7	8.0± 1.4	4.9± 2.0	2.6± 1.1	8.2± 3.1	1.7± 0.7	1.6± 0.4	1.8± 0.5	2.6± 0.8

2. We test Hypothesis 1 to model the relationship between the independent and the dependent variables, using GEE models (Section 5.2). This enables us to obtain models for the *Understandability* and *Modifiability Time*.
3. We test Hypothesis 2 using the Spearman and Kendall correlation coefficients (Section 5.3).
4. We test Hypothesis 3 by using the Spearman and Kendall correlation coefficients (Section 5.4).

5.1 Descriptive Study

5.1.1 Independent Variables Description

The measure values of the selected UML class diagrams are shown in Table 5. The last row shows the Mean and Standard Error (SE). The nine class diagrams have measure values that characterize them and limit the results of this study. These measure values increase from E1 to D3, which is in accord with the results obtained using cluster analysis FOR classifying the nine class diagrams. The degree of complexity, in ascending order, was Ei, Mi and Di.

5.1.2 Correlation Between the Independent Variables

The 11 measures selected as independent variables measure some aspects of the structural complexity and size of the class diagrams and can be considered as dimensions of the complexity. These variables will be used to build models that explain the dependent variables. However, the models will have different behaviour and meaning, depending on whether or not these dimensions are orthogonal. So it is interesting to study if there are correlations between them. The PCA is a good way to study these correlations; furthermore, it shows the extent to which the measures capture different orthogonal dimensions (Johnson

Table 6 Principal rotated components for measures of structural complexity

Measures	PCs			
	FNAG	FNAS	FNAGGH	FNDEP
NASSOC	0.315	0.896	0.102	0.176
NAGG	0.837	0.394	0.233	0.277
NDEP	0.405	0.462	0.368	0.696
NGEN	0.258	0.745	0.391	0.466
NGENH	0.841	0.372	−0.048	0.371
NAGGH	0.206	0.252	0.923	0.181
MAXDIT	0.343	0.843	0.300	0.162
MAXHAGG	0.888	0.209	0.402	0.009

1998; Kleinbaum et al. 1987). The new dimensions obtained through the PCA will be used as explanatory variables to obtain models for the dependent variables.

We performed two PCAs (Johnson 1998; Kleinbaum et al. 1987) with a varimax rotation (for size measures and for complexity measures) because there was insufficient data to perform only one. Examination of the PCA results revealed that the three dimensions related to size measures could be reduced to two, and the eight dimensions due to structural complexity measures could be reduced to four, in the following way (see Tables 6 and 7):

– As Table 6 shows, the information captured by structural complexity measures reduced to four PCs, each capturing different dimensions. As in most cases when carrying out a PCA, it is not easy to explain the meaning of these dimensions. It is obvious that FNDEP is strongly related to the dependencies. FNAG seems to capture information related to aggregations and complexity, due to the number of generalization hierarchies. FNAS seems to capture coupling due to the associations and generalization. FNAGGH captures the complexity due to the number of aggregation hierarchies.
– Size measures capture two aspects (see Table 7): one related to the functionality of a software system, reflected in a class diagram by methods (FNM), and the other to the data to be managed by a software system reflected by classes and their attributes (FNA).

5.1.3 Dependent Variables Description

We also considered descriptive statistics of *Understandability* and *Modifiability Time* by class diagram, in order to summarize the behaviour of these variables within the Spanish data and the Italian data. Figures 2 and 3 show that the box-plot medians of *Modifiability Time* are greater than the *Understandability Time* medians, for both the Spanish and Italian

Table 7 Principal rotated components for size measures

Measures	PCs	
	FNA	FNM
NC	0.995	−0.025
NA	0.990	0.102
NM	0.037	0.999

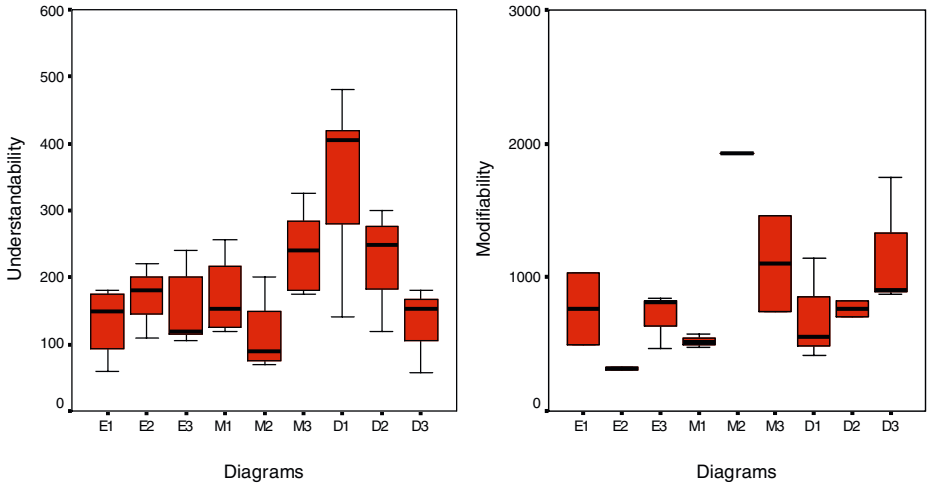


Fig. 2 *Understandability* and *Modifiability Time* box-plots by class diagram (Spanish experiment)

populations. This was expected, because the understandability tasks were easier than modifiability tasks.

Figures 2 and 3 show that the medians of *Understandability Time* for the Italian data are greater than the medians for the Spanish data in all the diagrams. The biggest difference (289 s) was observed for diagram E1, and the smallest difference (25 s) for diagram D1. The medians of *Modifiability Time* for the Italian data are greater than the medians for the Spanish data in all the diagrams, except for E1, E2 and D1. For the diagrams that have bigger values in the Italian data than in the Spanish data, the biggest difference was observed in diagram M2 (1,599 s) and the smallest difference in diagram D2 (98 s). For the diagrams that have bigger values in the Spanish data than in the Italian data, the biggest value was observed in diagram E2 (193 s) and the smallest in D1 (8 s).

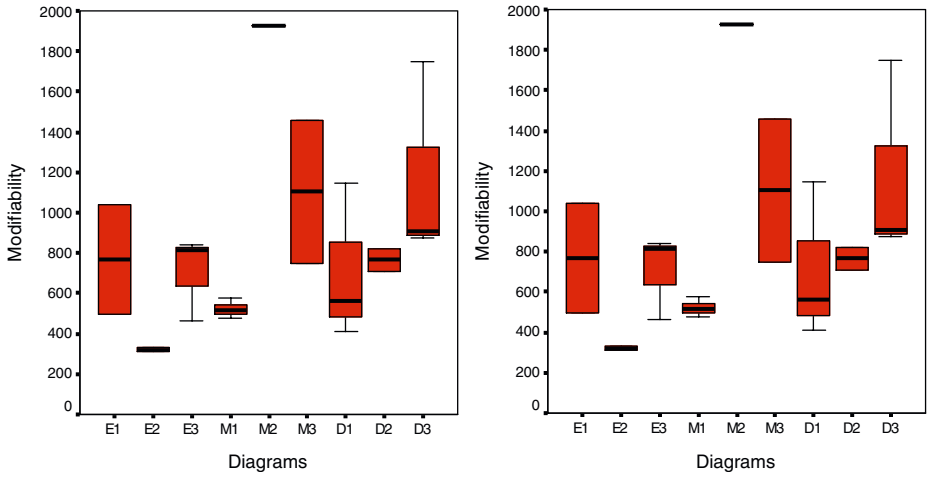


Fig. 3 *Understandability* and *Modifiability Time* box-plots by class diagram (Italian experiment)

Table 8 Descriptive statistics for *Understandability/Modifiability Completeness and Correctness*

Dependent variables	Mean	SE (Standard Error)	Median	IQR (Interquartile Range)
Italian data				
Understandability Correctness	0.902	0.034	1.000	0.250
Understandability Completeness	0.902	0.034	1.000	0.250
Modifiability Correctness	0.848	0.034	0.750	0.250
Modifiability Completeness	0.848	0.034	0.750	0.250
Spanish data				
Understandability Correctness	0.921	0.019	1.000	0.250
Understandability Completeness	0.915	0.020	1.000	0.250
Modifiability Correctness	0.814	0.031	0.750	0.250
Modifiability Completeness	0.757	0.037	0.750	0.312

In general, the time expended by the Italian subjects was greater than that expended by the Spanish ones, excepting for *Modifiability Time* in three diagrams, E1, E2 and D1, and in this last diagram the difference was eight seconds. This also happened in the study reported in (Genero et al. 2003a). The likely reason for this is that subjects from Italy did not use their native language (as stated above, the experiment was in English) and they needed more time to understand the tasks, while the Spanish subjects received all the material in Spanish.

Analysing the quality of the experimental tasks, we observed that *Completeness* and *Correctness* varied little in both populations; they take values of 1.00, 0.75 and 0.50 in most of the cases, but show different behaviour across the two populations, as shown in Table 8. *Completeness* and *Correctness* in the Italian data have identical values for *Modifiability* and *Understandability Times*; furthermore, all the understandability (modifiability) tasks were performed, even though they were not all performed completely correct. In the Spanish data, these variables do not have the same behaviour; looking at *Modifiability Completeness* (0.757) it seems that the modifiability tasks were more difficult for the Spanish subjects than for the Italian ones (0.848). However, the Spanish subjects obtained better results for the understandability tasks than the Italian ones.

The value of IQR (the difference between the third and first quartiles) is similar for all variables, excepting *Modifiability Completeness* in Spanish data. This means that the statistical dispersion was similar in most cases, i.e. for each variable, 50% of the values lay within in an interval that has an amplitude of 0.25.

The distributions of the *Understandability* and *Modifiability Time* by class diagrams were studied using the Shapiro–Wilk test (Shapiro and Wilk 1965). We hypothesized that both variables would be Normal distributed. The results obtained, at a 0.15 [8] significance level were as follows:

- Nonsignificant for Spanish *Modifiability Time* in all the diagrams examined, except for diagram M3. The *Understandability Time* had nonsignificant results for all the diagrams examined, except for M2.
- Nonsignificant for Italian *Understandability Time* in all cases, and *Modifiability Time* had nonsignificant results, except for diagram D3.

[8] In this kind of test, is usual to select a significance level greater than 0.10 to reduce the Beta error (i.e. the, the probability of accepting the variable as having a Normal distribution when the distribution is actually not Normal) (Snedecor and Cochran 1989).

Table 9 Spanish *Understandability Time* (SUT) and *Modifiability Time* (SMT) models

Selected models	
Understandability Time	Modifiability Time
Models with PCs $SUT - 1 = 186.426 + 52.954\ \ FNAS$ N=33 QIC=63,475.947	Models with PCs $SMT - 1 = 465.546 + 118.428\ \ FNAS$ N=23 QIC=219,728.227
Models with measures $SUT - 2 = 89.923 + 61.937\ MaxDIT + 1.373\ NA - 24.722\ \ Ndep$ N=35 QIC=101,623.812	Models with measures $SMT - 2 = 449.713 + 76.553 MaxDIT + 1.589\ NM$ N=23 QIC=202,825.245 $SMT - 3 = 344.261 + 72.866\ \ MaxDIT$ N=23 QIC=241,860.09597

Even though the number of observations was limited, we believe that the results support the assumption that the *Understandability* and *Modifiability Time* are not far from having a Normal distribution.

5.2 Testing Hypothesis 1

We used two types of independent variable in the studied model (Model (1)):

- The PCs obtained in Section 5.1.2.
- The 11 measures presented in Table 1. When selecting the measures to be included in a model, we took into account the information provided by the PCs, in order to avoid correlation as much as possible. Because the measures included in the same PC were correlated strongly, the measures included in each model pertain to different PCs.

Table 10 Italian *Understandability Time* (IUT) and *Modifiability Time* (IMT) models

Selected models	
Understandability Time	Modifiability Time
Models with PCs $IUT - 1 = 299.6882 + 71.2318\ \ FNAS$ N=18 QIC=212,294.73663	
Models with measures $IUT - 2 = 170.2822 + 29.8211\ NAssoc - 32.5864\ \ NDep$ N=19 QIC=166,756.48304	Models with measures $IMT - 1 = 565.261 + 12.711\ NC - 72.847\ \ NGenH$ N=17 QIC=658,305.733

5.2.1 Selected Models

The models shown in Tables 9 and 10 were selected from the models obtained from the two collections of independent variables, considering the following criteria:

- The significance level selected to testing $H_{0,1}$: $\beta_i=0\ \forall_I$ was 0.05. If the *p*-value of a coefficient β_I was greater than 0.05, the independent variable X_i was removed from the model.
- There are several ways of evaluating GEE constructed models. The quasilikelihood criterion (QIC) is a well-established goodness-of-fit statistic for model selection, and is adequate for determining the best subset of independent variables for a model (Hardin and Hilbe 2002). It is defined as a function of the squared residuals and the number of the model parameters, capturing the model simplicity and performance. The model with the smallest QIC value is always preferred.
- If we have models with similar conditions with respect to the previous criteria, the fewer variables the better.

The initially selected models were improved by excluding the observations that have extreme residual values, which were detected by looking at residual box-plots. For that reason, the value of *N* changed in each model (see Tables 9 and 10).

In addition, we examined the residuals in order to detect model misspecification, i.e. if the models do not fulfil some of the assumptions for GEE models presented in Section 3.2. (Davis 2002; Hardin and Hilbe 2002). The QIC values of the new models have been clearly improved. Furthermore, if some of the new coefficients were nonsignificant, their independent variables were excluded, which exclusion led to simpler models.

The results obtained from the Spanish data, summarized in Table 9, reveal that:

- The first-selected model SUT-1 has only one independent variable; the PC that includes associations and generalizations (FNAS). It is the best model with smaller QIC.
  The other selected model SUT-2 (based on the measures), has a QIC much different from that for the SUT-1 model. Moreover, it has a problem in interpreting the negative coefficient of Ndep. Despite having selected each measure from a different PC, there is no guarantee that they will be independent. In fact, Ndep is correlated with MaxDIT and NA, so it is necessary to adjust the effect of these correlations with a negative weight in the model.
- The best *Modifiability Time* model built with PCs (SMT-1) has FNAS as an explanatory variable, as does the *Understandability Time* model. The selected model based on the measures, SMT-2, has MaxDIT and NM as independent variables, the latter of which has a measure with negative coefficient, due to its correlation with MaxDIT. We use the model with only MaxDIT as an explanatory variable (SMT-3), because it does not perform badly and it is simpler than SMT-2.

The Italian *Understandability and Modifiability Time* models are summarized in Table 10. We may conclude the following on their basis:

- When PCs are used as independent variables, the *Understandability Time* is explained with FNAS (IUT-1), as in the Spanish model. When the independent variables are the eleven measures, the associations (NAssoc) and dependencies (NDep) explain the *Understandability Time* (IUT-2). This model has a problem inherent to the negative coefficients, due to the correlation between the measures (for the same reason as for the Spanish models).

Table 11 Correlations between measures and *SubComp* (Spanish experiment)

Spanish Data (N=33)	NC	NA	NM	NAssoc	NAgg	NDep	NGen	NGenH	NAggH	Max DIT	Max HAgg
$\tau_{Kendall}$ Coefficient	0.332*	0.231	−0.048	0.327*	0.193	0.206	0.362*	0.352*	0.126	0.378*	0.156
$\rho_{Spearman}$ Coefficient	0.389*	0.285	−0.059	0.380*	0.233	0.245	0.433*	0.393*	0.149	0.433*	0.192

* Significant results at level 0.05
** Significant results at level 0.01

– The PCs did not explain the *Modifiability Time* well. Moreover, when we considered the measures as explanatory variables, we found a set of models, and concluded on the basis of the QIC value that IMT-1 was the best. This model includes two measures, NC and NGenH, the latter having a negative coefficient to compensate for the correlation between them.

5.2.2 Conclusions Regarding Hypothesis 1

We now summarize the findings obtained after testing Hypothesis 1:

– The Italian QIC values are higher than the Spanish ones, and the Italian *Modifiability Time* model has the highest QIC; in fact, the Italian *Understandability* and *Modifiability* Time models did not perform adequately. We think that this is because the Italian experiment was conducted in English and the students had different levels of proficiency in that language. This factor could have biased the results in two ways: (1) it could have increased the times required to perform the tasks, particularly for modifiability tasks, which seem more difficult to complete (see Tables 8 and 9), and (2) it introduced an unknown variable; namely, the proficiency with respect to English of Italian subjects. The language factor could be why good modifiability models were not found in the Italian experiment.
– If we do not consider the Italian *Modifiability Time* model on the grounds just stated, FNAS is a factor that appears in all the models that use PCs as independent variables. That means the complexity due to associations and generalizations (see Table 6) are relevant for explaining *Understandability* and *Modifiability Time* variables. This idea is consistent with the explanatory variables found in the models that use measures as independent variables: MaxDIT appears in all the Spanish models and NAssoc in all the Italian ones.

Table 12 Correlations between measures and *SubComp* (Italian experiment)

Italian Data (N=23)	NC	NA	NM	NAssoc	NAgg	NDep	NGen	N GenH	N AggH	Max DIT	Max HAgg
$\tau_{Kendall}$ Coefficient	0.425*	0.440*	0.145	0.515**	0.384*	0.385*	0.374*	0.434*	0.183	0.528**	0.209
$\rho_{Spearman}$ Coefficient	0.548**	0.56**	0.201	0.627**	0.509*	0.454*	0.463*	0.548**	0.220	0.622**	0.277

* Significant results at level 0.05
** Significant results at level 0.01

5.3 Testing Hypothesis 2

We tested Hypothesis 2, which concerns the correlation between the 11 measures presented in Table 1 and *SubComp*, by means of Spearman and Kendall correlation coefficients (see Tables 11 and 12). The variable MaxDIT, which explains *Understandability* and *Modifiability Time* in the Spanish data, is also correlated with *SubComp* (see Table 11). NM, NA and NDep, which also explain the Spanish *Understandability* and *Modifiability Time* models, seem not to be correlated with *SubComp*. The *N* values shown in Tables 11 and 12 represent the total number of subjects that participated in the experiments, excluding those that had not rated the subjective complexity (five subjects in the Spanish experiment and zero subjects in the Italian experiment).

As we can see from Table 12, the Italian data have eight measures correlated with *SubComp*. Moreover, the measures NAssoc and NDep, which explained *Understandability Time* in the Italian experiment (see Table 10), are also correlated with *SubComp*. NC, which explained *Modifiability Time*, is also correlated with *SubComp*. However, NGenH it is not correlated with SubComp; but we have commented above on the poor behaviour of the Italian *Modifiability Time* models.

In conclusion, there are a number of measures that explain *Understandability* or *Modifiability Time*, such as NA, NC, NAssoc, NDep and MaxDIT, which are also correlated with *SubComp*. Moreover, the measures NAssoc and MaxDIT appear in both populations, connecting *SubComp* and the structural complexity of the class diagrams.

5.4 Testing Hypothesis 3

We tested this hypothesis by means of Spearman and Kendall correlation coefficients.

The *N* value (number of subjects) observed in Table 13 was obtained by considering only those Spanish subjects selected for *Understandability Time* (38) and *Modifiability Time* (29) who had rated *SubComp*. Hence, we excluded five subjects related to understandability tasks and four subjects related to modifiability tasks. We obtained the *N* value for the Italian data in a similar way.

In the Italian data, *SubComp* is correlated neither with *Understandability* nor with *Modifiability Time*. As noted above, this could be due to the fact that the subjects did not use their native language.

The *Understandability Time* in the Spanish data is correlated with *SubComp*. Perhaps the effort needed to perform the understandability tasks has influenced the subjective perception about complexity, because it was the first task the subjects were asked to

Table 13 Correlation between *SubComp*, *Understandability* and *Modifiability Time*

	Spanish data		Italian data	
	r_{Spearman}	τ_{Kendall}	ρ_{Spearman}	τ_{Kendall}
Understandability Time (UT)	0.343*	0.347*	0.410	0.331
p-value	0.049	0.049	0.065	0.061
N	33	33	21	21
Modifiability Time (MT)	0.337	0.277	0.156	0.110
p-value	0.099	0.087	0.500	0.532
N	25	25	21	21

*Significant results at 0.05 level

perform. So, by the time they undertook the modifiability tasks they had learned from the previous tasks. For further work, it would be interesting to study the correlation if the modifiability tasks were performed first.

6 Threats to Validity

We now discuss, in turn, threats to the conclusion, construct, and internal and external validity (Cook and Campbell 1979). Some of these threats were addressed through the experiment design and others after the experiment execution. Our goal here is twofold: (1) to help readers to qualify the results that are presented in this paper, and (2) to propose future research by highlighting some of the issues associated with our study.

Conclusion Validity Conclusion validity is the extent to which conclusions about the existence of a statistical relationship between treatments and outcomes are warranted. In both experiments we considered "convenience samples" (i.e. we selected all the students in the available classes), which threaten the validity of statistical conclusions and external validity.

A limited number of data values were collected during the execution of the experiments, due to the limited amount of time and the number of subjects. We may consider the quality of data collection to be poor, because we used pencil and paper. There is also a potential problem regarding the writing down of times. Although the experiment was supervised, the subjects assumed the responsibility of writing down the correct times and the supervisor did not check the times given by the students.

The quantity and quality of the data collected and the data analysis results (Section 5.2) are favourable to our principal work hypothesis, Hypothesis 1, concerning the use of the measures investigated as explanatory variables for *Understandability* and *Modifiability Time*. Furthermore, the statistical methods used to study Hypotheses 1 to 3 work with robust tests (GEE estimators, Kendall, Spearman and Shapiro-Wilk). It is only the size of the samples that could be a threat to the strength of the conclusions (Wohlin et al. 2000).

Construct Validity Construct validity is the extent to which independent and dependent variables provide accurate measurements of the concepts they purport to measure. The dependent variables, understandability and modifiability of class diagrams, were measured by using different times (*Understandability* and *Modifiability Time*), which are objective measures that reflect the time that the subjects spent on solving the experimental tasks. For this reason, we consider that they provide objective measurements of what they purport to measure. We also took into account another variable, *SubComp*, which reflects the subjective perception of the subjects related to the complexity of the diagrams. Although it is a subjective measure, it does capture what it purports to measure, so it is constructively valid.

To investigate the construct validity of the measures used in this study, the reader may refer to Genero (2002), where the theoretical validation of the measures was presented.

Internal Validity Internal validity is the extent to which conclusions drawn from the causal effect of the independent variables are warranted. The analysis performed here is correlative in nature. We demonstrated that some of the investigated measures have a statistically and practically significant relationship with *Understandability Time* and *Modifiability Time*.

Such statistical relationships do not intrinsically demonstrate a causal relationship. The following issues were addressed:

- Differences among subjects. As Briand et al. (2001) remarked when dealing with small samples in software engineering experiments, variation in participants' skills is a major concern that is difficult to address fully by randomization or blocking. In the study reported herein, there was no great difference among the subjects' experience in modelling with UML, due to the fact that all of them were students and they were taking their first course of Software Engineering. For that reason, error variance due to differences among subjects is reduced.
- Knowledge of the universe of discourse among class diagrams. The class diagrams were from different domains, but were general enough to be easily understood by each of the subjects. This means that knowledge of the domain did not affect internal validity.
- Learning effects. We selected a between-subjects design and the subjects were only given one UML class diagram, so there were no learning effects.
- Fatigue effects. On average, the experiment lasted for less than an hour (this fact was corroborated by summing the total time for each subject), which is much less than typical laboratory sessions in both universities. Hence, fatigue was not relevant in these experiments.
- Persistence effects. As the experiment was run with subjects who had never done a similar experiment, persistence effects were avoided.
- Subject motivation. We motivated students to participate in the experiment, explaining to them that tasks similar to those of the experiment could be performed in examinations or practice.
- Other factors. Plagiarism and influence among students were controlled by the professor who supervised the experiment.
- In the Italian experiment we used English, which is a foreign language for the subjects. The use of English could have been a threat to internal validity in Italy, and it seemed to disturb *Modifiability Time* in particular. This fact could have biased the results in two ways: it increased the times to do the tasks, especially for modifiability tasks, and it introduced an unknown variable, is the proficiency of Italian subjects in the use of English.

External Validity External validity is the extent to which the results of the research can be generalised to the population under study and other research settings. The greater the external validity, the more the results of an empirical study can be generalized with regard to actual software engineering practice. Two threats to validity were identified which limit such generalization:

- Materials and tasks used. In the experiment, we tried to use class diagrams and tasks representative of real cases, but further empirical studies should be conducted that use real cases from software companies.
- Subjects. Because of the difficulty of obtaining professional subjects, we used students from a software engineering course. Now, it is, of course, well-recognized that if results are to be generalized, experiments must be carried out using professionals as subjects. However, the justification for this position is that results can only be generalized if the participants in an experiment have the same level of experience as the populations to which the results are to be generalized. In the case of the experiments reported here, the

Table 14 PCs with measures of the structural complexity used in Genero et al. (2003a)

Measures	PCs		
	FNGEN	FNAGG	FNAS
NAssoc	0.418	0.339	0.811
NAgg	0.398	0.839	0.346
NDep	0.668	0.229	0.675
NGen	0.916	0.134	0.355
NAggH	−0.195	0.963	0.124
NGenH	0.940	−0.032	0.268
MaxHAgg	0.629	0.469	0.558
MaxDIT	0.906	0.010	0.306

tasks assigned did not require a high level of industrial experience were well within the capacities of students. Hence, in this case, we think that the results are generalizable.

7 Comparison with Previous Experimental Work

We now compare the results observed in the current study with those obtained in a previous, similar, study presented in Genero et al. (2003a), for the purpose of deriving a general conclusion about Hypothesis 1. In the previous study, we also collected empirical data from one experiment carried out in Spain and a replica in Italy. We have, therefore, the results of four experiments to synthesize; two of them carried out in Spain and two in Italy. These experiments have a common framework, defined by similar experimental design, measures and hypotheses, so they can be compared. The major difference between them is that in the previous study, we considered all the collected data, whereas in the current one we included only the data with *Undertsandability/Modifiability Correctness* and *Completeness* higher than or equal to 0.75. Moreover, in the previous study we used other statistical techniques to analyze the obtained data. Therefore, in order to integrate these four studies, we analyzed the previous data again, using the GEE model and selecting only those subjects that had *Correctness* and *Completeness* scores higher than or equal to 0.75.

7.1 Results of the Previous Experiments

We first carry out a PCA with the values of the measures obtained from the class diagrams used in the previous experiments (Genero et al. 2003a). Then we perform a descriptive study of *Understandability* and *Modifiability Time*. Then we test Hypothesis 1, and finally compare the obtained results.

Table 15 PCs with the size measures used in Genero et al. (2003a)

Measures	PCs	
	FNC	FNM
NC	0.979	0.088
NA	0.969	0.163
NM	0.126	0.992

7.1.1 Correlation Between Measures

We studied the correlation between measures, as we did in Section 5.1.2. The PCs obtained (see Tables 14 and 15) capture the data (FNC) and functionality (FNM) of the class diagrams. Furthermore, FNGEN captures the complexity due to generalizations, FNAGG the aggregations complexity and FNAS the complexity due to associations and dependencies.

7.1.2 Understandability and Modifiability Time Description

We had 24 subjects in the Spanish experiment and 26 in the Italian one. When we selected the subjects who had *Correctness* and *Completeness* scores higher or equal than 0.75, we excluded six Spanish subjects and ten Italian subjects for *Modifiability Time* prediction models.

We observed differences between the Italian and Spanish dependent variables, as in the current study. The reason could be the same; the use of English in Italy.

7.1.3 Analysis of Hypothesis 1

To study this hypothesis about the *Understandability* and *Modifiability Time* using model (1) (Section 3.2), GEE models were built and selected using the same criteria as in the current study (Section 5.2.1). There were two types of independent variable: the PCs shown in Tables 14 and 15 and the eleven measures considered (see Table 1). When selecting the measures to be included in a model, we took into account the information of the PCs, to avoid correlation as much as possible. Hence, each measure of the model pertained to a different PC.

Selected Models A summary of the measures that explain the selected models in the previous experiments is shown in Table 16. They are ordered by the obtained QIC value. Each group of models is referred in Table 16 as GUS, GMS, GUI and GMI; two groups are related to the Spanish data (*Understandability* and *Modifiability Time*) and the other two are related to the Italian data (*Understandability* and *Modifiability Time*). We selected two models of each type because both models fulfil all the criteria and have very similar QIC values.

The results presented in Table 16 show that:

– When we consider the models that have PCs as independent variables, the functionality (FNM) appears in all of them. This means that FNM is a good explanatory variable,

Table 16 Summary of the variables used in models selected in previous experiments (Genero et al. 2003a)

Spanish data		Italian data	
Understandability Time (GUS)	Modifiability Time (GMS)	Understandability Time (GUI)	Modifiability Time (GMI)
FNM FNAGG	FNM FNAS	FNM FNAGG	FNM FNAGG
FNC FNM	FNC FNGEN	FNC FNAS	FNM FNAS
NA NM	NC MaxHAgg	NA NDep	NAssoc NAgg
NAggH NGenH	NA MaxHAgg	NA NAssoc	NAggH MaxDIT

not only for some models of *Understandability Time* in the Italian and Spanish data, but also for *Modifiability Time* models. However, the PCs that capture size and associations (FNC, FNAS) appear in the majority of the models (except for the *Modifiability Time* models of Italy for FNC and the *Understandability Time* models of Spain for FNAS).
- When we consider the models that have the measures as independent variables, the size (NA) appears in the majority of the models, and also aggregations (NAgg, NAggH or MaxHAgg). Associations (NAssoc) and generalizations (NGenH, MaxDIT) appear in two of the four groups of models.
- In the Italian data, FNM, FNAS and NAssoc are in the two model groups.
- In the Spanish data, the two model groups have FNM, FNC and NA in common.

7.2 Integrating the Results of Hypothesis 1

To summarize the results of the four studies, we excluded the *Modifiability Time* Italian models due to the problem noted in Section 5.2.2. Table 17 shows a summary of the selected models in the current study.

From the tables, we may conclude that:

- The structural complexity due to associations, measured by NAssoc or FNAS, appears in all the model groups, except in the *Understandability Time* Spanish models of the previous study.
- The structural complexity due to generalizations, measured by MaxDIT, NGenH, the PCs FNGEN of the previous study and FNAS of the current study, seems to be relevant for explaining *Understandability* and *Modifiability Time,* except for the understandability Italian models of the previous study.

In conclusion, even the models selected regarding *Understandability Time* (or *Modifiability Time*) did not include the same independent variables and the same coefficients. The measures of structural complexity due to associations and generalizations seem to be the most relevant explanatory variables in the four experiments. The principal reasons for such differences between the models could be the sample size and certain issues related to threats to validity (discussed in Section 6).

8 Conclusions and Future Work

Pursuing the objective of obtaining good predictors of *Understandability* and *Modifiability Time* of UML class diagrams and considering that simple studies rarely provide definitive answers (Miller 2000; Basili et al. 1999), in this work we have presented the following:

Table 17 Summary of the models obtained in the current study

Spanish data		Italian data	
Understandability Time (GUS)	Modifiability Time (GMS)	Understandability Time (GUI)	Modifiability Time (GMS)
FNAS MaxDIT NA NDep	FNAS MaxDIT NM	FNAS NAssoc NDep	*EXCLUDED* *EXCLUDED*

1. A controlled experiment carried out with students enrolled in their third year of Computer Science at University of Castilla-La Mancha in Spain, and its replica with students in the final year of Computer Science at the University of Sannio, in Italy (see Sections 3 and 4).
2. A thorough data analysis, through which we obtained prediction models for the *Understandability* and *Modifiability Time* of the UML class diagrams (Hypothesis 1). These models were built on the basis of the measures we proposed for the size and structural complexity of UML class diagrams (see Section 5).

From testing Hypothesis 1, we conclude that the principal component that captures the structural complexity due to associations (NAssoc) and generalizations (MaxDIT) is a PC that is relevant for explaining *Understandability* and *Modifiability Time*, because it appears in all the models that use PCs as independent variables. This result is consistent with the explanatory variables found in the selected models that use the proposed measures as independent variables: MaxDIT appears in all the Spanish models and NAssoc in all the Italian ones.

The use of English in the Italian experiment could have biased the results in two ways: (1) it could have increased the times the subjects spent performing the required tasks, especially for modifiability tasks, which seem more difficult to complete, and (2) it introduced an unknown variable; namely, the proficiency of Italian subjects in the use of English. This bias may explain why we did not find good *Modifiability Time* models in the Italian experiment. So, we excluded the *Modifiability Time* Italian models. In the future, we hope to improve the design of the experiment, which improvement will include controlling the language used in the experiment.

The results obtained in the current study were confirmed when we compared them with those of a previous study (Genero et al. 2003a). In the previous study, the PC that captures associations (FNAS) was relevant to explaining the majority of the *Understandability* and *Modifiability Time* models. Furthermore, the structural complexity introduced by generalizations, measured by different measures such as MaxDIT or NGenH, was also relevant to explaining the selected models of *Understandability* and *Modifiability Time*.

The results of our test of Hypothesis 2 suggested that the subjective perception of the subjects about the complexity of the diagrams, expressed by *SubComp*, is related to the structural complexity due to associations (NAssoc), generalizations (MaxDIT) and other measures.

From our test of Hypothesis 3, we may conclude that in the Spanish experiment, it seems that subjects were influenced by the effort made (*Understandability Time and Modifiability Time*) when rating the subjective complexity of the diagrams (*SubComp*). In the Italian experiment there was no such correlation, perhaps due to the effect of using English.

In spite of the encouraging findings of the current study, we identified a number of weaknesses. The results of this experiment should therefore be interpreted only as findings, which need to be replicated and corroborated. In addition, because of the nature of the experimental tasks, in which participants made a number of predefined changes to UML class diagrams, this experiment was limited to two characteristics of maintainability: understandability and modifiability. That being so, we propose to conduct further research in the following directions:

- Experimental research: We suggest making a family of experiments to increase the external validity of the results to the extent that the conclusions currently presented can be generalized. Such a family of experiments should also use professionals as subjects

and different experimental models and tasks (class diagrams and maintenance changes taken from actual industrial practice). Furthermore, we will propose new experiment designs that permit corroboration of the influence of associations and generalizations on the maintainability of class diagrams.

- Replications: We will replicate the current experiment in Italy but prepare the material in Italian, rather than English.
- Field studies: We intend to conduct a number of field studies on the effect of size and structural complexity on the maintenance of UML class diagrams in practice. This will involve observations of maintenance practices in OO development environments. Collected data will include:
  - Size and structural complexity of the models, using a wide range of measures defined.
  - Frequency and type of maintenance changes required, which will enable the models to be evaluated for stability.
  - Effort required implementing changes, which will enable the modifiability of the models to be evaluated.

- Even though we focused, in the study reported herein, on measures for the size and structural complexity of UML class diagrams, there are, in the literature, other measures for coupling and cohesion. These measures should be addressed in the future.

Acknowledgements This research is part of the MECENAS project (PBI06-0024) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" and the following projects supported by the "Ministerio de Educación y Ciencia (Spain) and FEDER": TIN2006-15175-C05-05, TIN2004-03145 and TIN2004-06689.

We thank Macario Polo, Félix García and Crescencio Bravo from the University of Castilla-La Mancha for having allowed us to perform the experiment with their students.

The authors are grateful to the anonymous reviewers for insight and feedback to several key issues covered in this research. Thanks to Chris Wright for proofreading the paper.

Appendix A

Here we show, as an example, the UML class diagram D3 and its understandability, modifiability and rating tasks.

Diagram D3

Given the UML class diagram shown below related to the "Management of a Bank" you should perform the following tasks:

Tasks:

1. Answer the following questions:

Write down the start time (indicating hh:mm:ss): _____

1. Is it possible to distinguish between individual and company clients?
2. Is it possible to know when an employee began to work in a branch office?
3. Can an employee belong to more than one branch office?
4. Can the employees be clients of the bank?

Write down the end time (indicating hh:mm:ss): _____

2. Carry out the modifications necessary to satisfy the following requirements:

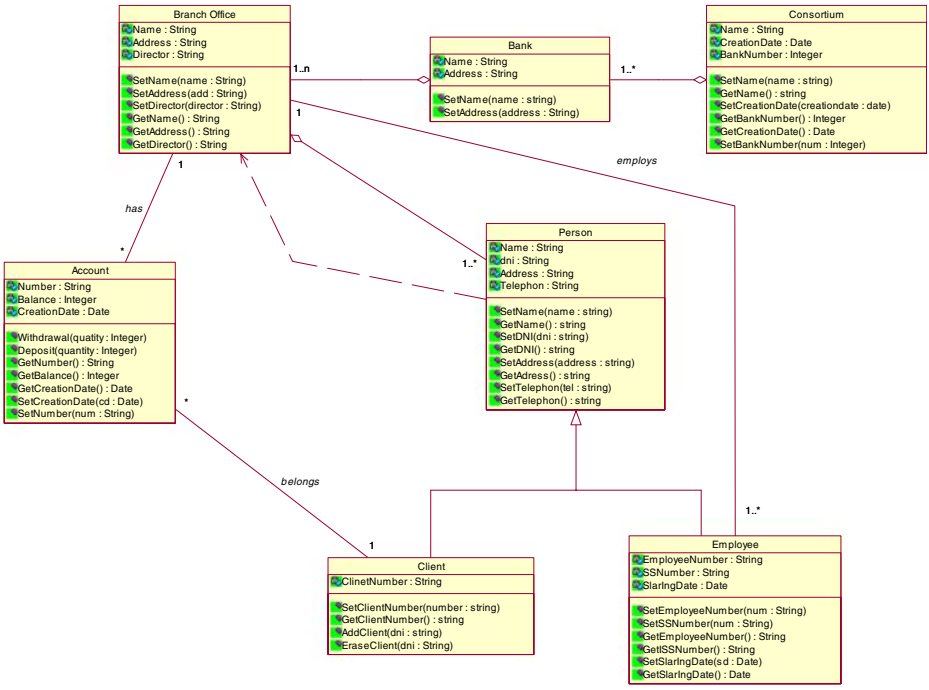
Write down the start time (indicating hh:mm:ss): _____

1. We would like to distinguish between permanent and temporary employees. For temporary employees we would also like to know their start date and the duration of their contracts.
2. We would like to register the transactions of an account, indicating the transaction date, type (deposit withdrawal) and the amount of the transaction. We know that a transaction can only belong to one account and one account can experience many transactions.
3. Can there be employees that are clients of the bank?
4. We would like to identify the manager of every bank and also the employees of each bank.

Write down the end time (indicating hh:mm:ss): _____

3. According to your criterion, evaluate the COMPLEXITY of this class diagram (mark with a cross).

Very simple	Relatively simple	Average	Slightly complex	Very complex



References

Atkinson C, Kühne T (2003) Model-driven development: a metamodeling foundation. IEEE Softw 20(5):36–41

Bansiya J, Davis C (2002) A hierarchical model for object-oriented design quality assessment. IEEE Trans Softw Eng 28(1):4–17

Basili V, Rombach H (1988) The TAME project: towards improvement-oriented software environments. IEEE Trans Softw Eng 14(6):728–738

Basili V, Weiss D (1984) A methodology for collecting valid software engineering data. IEEE Trans Softw Eng 10:728–738

Basili V, Shull F, Lanubile F (1999) Building knowledge through families of experiments. IEEE Trans Softw Eng 25(4):435–437

Briand L, Wüst J (2001) Modeling development effort in object-oriented systems using design properties. IEEE Trans Software Eng 27(11):963–986

Briand L, Wüst J (2002) Empirical studies of quality models in object-oriented systems. In: Zelkowitz (ed) Advances in computers, vol 59. Academic, pp 97–166

Briand L, Devanbu W, Melo W (1997) An investigation into coupling measures for C++. In: 19th International Conference on Software Engineering (ICSE 97), Boston, USA, pp 412–421

Briand L, Wüst J, Lounis H (1998) Investigating quality factors in object-oriented designs: an industrial case study. Technical report ISERN 98-29 (version 2)

Briand L, Wüst J, Lounis H (1999) A comprehensive investigation of quality factors in object-oriented designs: an industrial case study. In: 21st International Conference on Software Engineering, Los Angeles, pp 345–354

Briand L, Arisholm S, Counsell F, Houdek F, Thévenod-Fosse P (2000) Empirical studies of object-oriented artefacts, methods and processes: state of the art and future directions. Emp Softw Eng 4 (4):387–404

Briand L, Bunse C, Daly J (2001) A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. IEEE Trans Softw Eng 27(6):513–530

Brito e Abreu F, Carapuça R (1994) Object-oriented software engineering: measuring and controlling the development process. In: Proceedings of 4th International Conference on Software Quality, Mc Lean, VA, USA, 3–5 October

Brooks A, Daly J, Miller J, Roper M, Wood M (1996) Replication of experimental results in software engineering. Technical report ISERN-96-10. International Software Engineering Research Network

Calero C, Piattini M, Genero M (2001) Method for obtaining correct metrics. In: International Conference on Enterprise and Information Systems (ICEIS'2001), pp 779–784

Cantone G, Donzelli P (2000) Production and maintenance of software measurement models. J Softw Eng, Knowl Eng 5:605–626

Card C, El-Emam K, Scalzo B (2001) Measurement of object-oriented software development projects. In: Software Productivity Consortium NFP

Chapin N, Hale J, Khan K, Ramil J, Tan W (2001) Types of software evolution and software maintenance. J Softw Maint Evol: Res Prac 13:3–30

Chidamber S, Kemerer C (1994) A metrics suite for object oriented design. IEEE Trans Softw Eng 20 (6):476–493

Cook T, Campbell D (1979) Quasi-experimentation: design and analysis issues for field settings. Boston, Houghton Mifflin

Davis C (2002) Statistical methods for the analysis of repeated measurements, Springer

Erickson J, Siau K (2004) Theoretical and practical complexity of UML. In: 10th Americas Conference on Information Systems, New York, USA, pp 1669–1674

Hardin JW, Hilbe JM (2002) Generalized estimating equations. Chapman & Hall, London

El-Emam K (1999) The prediction of faulty classes using object-oriented design metrics, NRC/ERB1064. National Research Council Canada

El-Emam K (2001) Object-oriented metrics: a review on theory and practice, NRC/ERB 1085. National Research Council Canada

Fenton N, Pfleeger S (1997) Software metrics: a rigorous approach, 2nd edn. Chapman & Hall, London

Fioravanti F, Nesi P (2001) Estimation and prediction metrics for adaptive maintenance effort of object-oriented systems. IEEE Trans Software Eng 27(12):1062–1083

García F, Ruiz F, Cruz JA, Piattini M (2003) Integrated measurement for the evaluation and improvement of software processes. In: 9th European Workshop on Software Process Technology (EWSPT'9), Helsinki, Finland. Lecture notes in computer science, vol 2786. pp 129–145

Genero M (2002) Defining and validating metrics for conceptual models, Ph.D. thesis, University of Castilla-La Mancha, Spain

Genero M, Jiménez L, Piattini M (2002) A controlled experiment for validating class diagram structural complexity metrics. In: Bellahsene Z, Patel D, Rolland C (eds) The 8th International Conference on Object-oriented Information Systems (OOIS'2002), Lecture notes in computer science, vol 2425. Springer, Berlin Heidelberg New York, pp 372–383

Genero M, Piattini M, Calero C (2000) Early measures for UML class diagrams. L'Objet. 6(4), Hermes Science Publications, pp 489–515

Genero M, Olivas J, Piattini M, Romero F (2001) Using metrics to predict OO information systems maintainability. In: CAISE 2001, Lecture notes in computer science, vol 2068. Interlaken, Switzerland, pp 388–401

Genero M, Manso M[a]E, Piattini M, Cantone G (2003a) Building UML class diagram maintainability prediction models based on early metrics. In: 9th International Symposium on Software Metrics (METRICS 2003), IEEE Computer Society, pp 263–275

Genero M, Olivas J, Piattini M, Romero F (2003b) Assessing object oriented conceptual models maintainability. In: Poels G et al (eds) International Workshop on Conceptual Modeling Quality (IWCMQ'02), Tampere, Finland. Lecture notes in computer science, vol 2784. Springer, Berlin Heidelberg New York, pp 288–299

Genero, M., Piattini M., and Calero, C. 2005. Metrics for high-level design UML class diagrams: an exploratory analysis. Journal of Object Technology, 4(9). Available at http://www.jot.fm

Harrison R, Counsell S, Nithi R (1998) An investigation into the applicability and validity of object-oriented design metrics. Emp Softw Eng 3:255–273

Harrison R, Counsell S, Nithi R (2000) Experimental assessment of the effect of inheritance on the maintainability of object-oriented systems. J Syst Softw 52:173–179

Henderson-Sellers B (1996) Object-oriented metrics—measures of complexity. Prentice-Hall, Upper Saddle River, NJ, pp 489–515

ISO/IEC 9126-1.2 (2001) Information technology—software product quality—part 1: quality model

Johnson DE (1998) Applied multivariate methods for data analyst. Duxbury Press, Brooks/Cole Publishing Company

Juristo N, Moreno AM[a] (2001) Basics of software engineering experimentation. Kluwer

Kiewkanya M, Muenchaisiri P (2004) Predicting modifiability of UML class and sequence diagrams. In: The Second Workshop on Software Quality (26th International Conference on Software Engineering (ICSE 2004)), pp 53–57

Kiewkanya M, Jindasawat N, Muenchaisiri P (2004) A methodology for constructing maintainability model of object-oriented design. In: Fourth International Conference on Quality Software (QSIC' 04), pp 206–213

Kim H, Boldyreff C (2002) Developing software metrics applicable to UML Models. In: Proceedings of the 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, (QAOOSE 2002)

Kitchenham B, Pfleeger S, Fenton N (1995) Towards a framework for software measurement validation. IEEE Trans Softw Eng 21(12):929–943

Kitchenham B, Pfleeger S, Pickard L, Jones P, Hoaglin D, El Emam K, Rosenberg J (2002) Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng 28(8):721–734

Kleinbaum D, Kupper L, Muller K (1987) Applied regression analysis and other multivariate methods, 2nd edn. Duxbury

Li W, Henry S (1993) Object-oriented metrics that predict maintainability. J Syst Softw 23(2)

Lorenz M, Kidd J (1994) Object-oriented software metrics: a practical guide. Englewood

Manso, M[a]E, Genero M, Piattini M (2003) No-redundant metrics for UML class diagrams structural complexity. In: Eder JY, Missikoff M (eds) CAISE 2003, Lecture notes in computer science, vol 2681. Springer, Berlin Heidelberg New York, pp 127–142

Marchesi M (1998) OOA Metrics for the unified modeling language. In Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering, Florence, Italy, pp 67–73

Mendes E, Watson I, Mosley N, Counsell S (2002) A comparison of development effort estimation techniques for web hypermedia applications. In: 8th IEEE Symposium on Software Metrics (METRICS'02), pp 21–30

Miller J (2000) Applying meta-analytical procedures to software engineering experiments. J Syst Softw 54:29–39

OMG (2001) Unified Modeling Language (UML) Specification, Version 1.4. Object Management Group (OMG)

OMG (2005) Object Management Group. UML 2.0, OMG Document. Available at http://www.omg.org

Poels G, Dedene G (2000) Distance-based software measurement: necessary and sufficient properties for software measures. Inf Softw Technol 42(1):5–46

SAS Institute (1999) SAS/STAT Users Guide, Version 8. SAS Institute, Cary, NC

Schneidewind N (1992) Methodology for validating software metrics. IEEE Trans Softw Eng 18(5):410–422

Schneidewind N (2002) Body of knowledge for software quality measurement. IEEE Computer 35(2):77–83
Selic B (2003) The pragmatics of model-driven development. IEEE Software 20(5):19–25
Shapiro SS, Wilk MB (1965) An analysis of variance test for normality. Biometrika 52:591–611
Si-Said C, Akoka J, Comyn-Wattiau I (2002) Conceptual modelling quality—from EER to UML Schemas Evaluation. In: Spaccapietra S, March, S, Kambayashi Y (eds) 21st International Conference on Conceptual Modeling (ER 2002), Tampere, Finland. LNCS, vol 2503.d pp 499–512
Snedecor GW, Cochran WG (1989) Statistical methods, 8th edn. Iowa State University Press
SPSS (2002) SPSS 11.5. Syntax Reference Guide. Chicago. SPSS Inc
Van Solingen R, Berghout E (1999) The goal/question/metric method: a practical guide for quality improvement of software development. McGraw-Hill, New York
Vokác M, Tichy W, Sjoberg DI, Arisholm E, Aldrin M (2004) A controlled experiment comparing the maintainability of programs designed with and without design patterns—a replication in a real programming environment. Emp Softw Eng 9:149–195
Wohlin C, Runeson P, Höst M, Ohlson M, Regnell B, Wesslén A (2000) Experimentation in software engineering: an introduction. Kluwer

Marcela Genero is an Associate Professor in the Department of Information Systems and Technologies at the University of Castilla-La Mancha, Ciudad Real, Spain. She received her MSc degree in Computer Science from the University of South, Argentine in 1989, and her PhD at the University of Castilla-La Mancha, Ciudad Real, Spain in 2002. Her research interests include empirical software engineering, software metrics, conceptual data models quality, database quality, quality in product lines, quality in MDD, etc. She has published in prestigious journals (Journal of Software Maintenance and Evolution: Research and Practice, L'Objet, Data and Knowledge Engineering, Journal of Object Technology, Journal of Research and Practice in Information Technology), and conferences (CAISE, E/R, MODELS/UML, ISESE, OOIS, SEKE, etc). She edited the books of Mario Piattini and Coral Calero titled "Data and Information Quality" (Kluwer, 2001), and "Metrics for Software Conceptual Models" (Imperial College, 2005). She is a member of ISERN.

M. Esperanza Manso is an Associate Professor in the Department of Computer Language and Systems at the University of Valladolid, Valladolid, Spain. She received her MSc degree in Mathematics from the University of Valladolid. Currently, she is working towards her PhD. Her main research interests are software maintenance, reengineering and reuse experimentation. She is an author of several papers in conferences (OOIS, CAISE, METRICS, ISESE, etc.) and book chapters.

Corrado Aaron Visaggio is an Assistant Professor of Database and Software Testing at the University of Sannio, Italy. He obtained his PhD in Software Engineering at the University of Sannio. He works as a researcher at the Research Centre on Software Technology, at Benvento, Italy. His research interests include empirical software engineering, software security, software process models. He serves on the Editorial Board on the e-Informatica Journal.

Gerardo Canfora is a Full Professor of Computer Science at the Faculty of Engineering and the Director of the Research Centre on Software Technology (RCOST) at the University of Sannio in Benevento, Italy. He serves on the program committees of a number of international conferences. He was a program co-chair of the 1997 International Workshop on Program Comprehension; the 2001 International Conference on Software Maintenance; the 2003 European Conference on Software Maintenance and Reengineering; the 2005 International Workshop on Principles of Software Evolution: He was the General chair of the 2003 European Conference on Software Maintenance and Reengineering and 2006 Working Conference on Reverse Engineering. Currently, he is a program co-chair of the 2007 International Conference on Software Maintenance. His research interests include software maintenance and reverse engineering, service oriented software engineering, and experimental software engineering. He was an associate editor of IEEE Transactions on Software Engineering and he currently serves on the Editorial Board of the Journal of Software Maintenance and Evolution. He is a member of the IEEE Computer Society.

Mario Piattini is MSc and PhD in Computer Science by the Technical University of Madrid. Certified Information System Auditor by ISACA (Information System Audit and Control Association). Full Professor in the Department of Information Systems and Technologies at the University of Castilla-La Mancha, in Ciudad Real, Spain. Author of several books and papers on databases, software engineering and information systems. He leads the ALARCOS research group at the University of Castilla-La Mancha.