Empirical Software Engineering, 10, 437–466, 2005.
© 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.

A Characterization Schema for Software Testing Techniques

SIRA VEGAS svegas@fi.upm.es
Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain

VICTOR BASILI basili@cs.umd.edu
Department of Computer Science, University of Maryland, Maryland, USA

Editors: Forrest Shull and Natalia Juristo

Abstract. One of the major problems within the software testing area is how to get a suitable set of cases to test a software system. This set should assure maximum effectiveness with the least possible number of test cases. There are now numerous testing techniques available for generating test cases. However, many are never used, and just a few are used over and over again. Testers have little (if any) information about the available techniques, their usefulness and, generally, how suited they are to the project at hand upon, which to base their decision on which testing techniques to use. This paper presents the results of developing and evaluating an artefact (specifically, a characterization schema) to assist with testing technique selection. When instantiated for a variety of techniques, the schema provides developers with a catalogue containing enough information for them to select the best suited techniques for a given project. This assures that the decisions they make are based on objective knowledge of the techniques rather than perceptions, suppositions and assumptions.

Keywords: Software testing, testing technique selection, characterization schema.

1. The Problem of Selecting Testing Techniques

As Harrold (2000) claims, evaluation is a highly important process, as it is directed at assuring software quality. According to Beizer (1990), testing is considered as one of the most costly development processes, sometimes exceeding fifty per cent of total development costs. By one estimate (RTI, 2002), software consumers and organisations incur approximately US$50B in losses from defective software each year. This estimate suggests industry-wide deficiency in testing. One of the factors that influence the cost of testing is the number of test cases used. The more test cases are generated, the longer it will take to specify, execute and analyse the tests. This is what makes it unworkable to run all possible combinations of input values, that is, rules out exhaustive testing (Myers, 1970). So, the tests are run on a relatively small set of cases, previously chosen from the universe of system inputs. The choice of test cases is of utmost importance, not only because the resulting set should be of minimum size, but also because this set must reflect, on the basis of a small number of inputs, the behaviour of the system for the input universe.

Testing techniques are used to find a suitable set of test cases. There are now some several tens of techniques, which raises the question of what difference there is between these techniques. We can find distinctions in the literature as regards the mechanical (testing books) and technical and context aspects (theoretical research articles, simulations and experiments) of the techniques. However, what the best suited techniques for evaluating a given system aspect are remains an open question (Bertolino, 2004). The truth is that, although there are some papers that compare techniques, there are no studies that analyse the applicability conditions of a technique at length, or assess, for each technique, what the relevant parameters for its applicability conditions are.

Although the question of which are the best-suited techniques for developing the set of test cases for testing a given system apparently poses enormous difficulties, it is, nevertheless, a question testers face every time they have to test a system. And, how is it answered at present? Neither systematically, nor following well-defined guidelines. Indeed, of all the existing testing techniques, some are never considered for use at all, and others are used over again in different projects without even examining, after use, whether or not they were really useful. The decisions made by developers are not so much haphazard as limited insofar as their knowledge of the techniques is. There are two main reasons why developers do not make good choices:

- The information available about the techniques is normally distributed across different sources of information (books, articles and even people). This means that developers do not have an overall idea of what techniques are available and of all the information of interest about each testing technique.

- They have no access to pragmatic information concerning each testing technique unless they have used it before. Developers do not tend to share the knowledge they acquire by using testing techniques with others. This means that they miss out on the chance of learning about the experiences of others.

The problem we address here is how to identify relevant information for selecting testing techniques. The aim of solving this problem is to help testers to choose the best suited testing techniques for each project during the testing technique selection process. Indeed, testers will not need to be acquainted with the technique to the extent of having used it or knowing how it is applied to make the selection.

The proposed solution is called a characterization schema and is not confined to identifying useful selection criteria, but also provides an infrastructure for storing the information identified and specified for each existing technique. Although the primary objective of the characterization schema is to help software developers with selection, a schema of this sort will also benefit testing researchers, by focusing their research on knowledge that is now missing about testing techniques.

The schema describes the properties of all the techniques according to the same pattern. The schema will be instantiated once for each technique represented in the catalogue. Accordingly, it will be possible to build a repository containing all the techniques of interest to a given organisation, as shown in Figure 1.

The article is organised as follows. Section 2 discusses related work. Section 3 sets out the goals of the research. Section 4 describes how the schema was generated. Sections 5–7 show how the schema was evaluated and Section 8 discusses our conclusions.

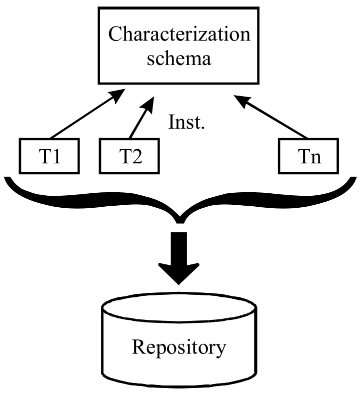

Figure 1. Schema instantiation.

2. Related Work

Although the characterization problem has not been specifically studied in the testing area, there have been attempts at comparing testing techniques. In particular, there are studies aiming to establish differences and similarities between testing techniques. For an analysis of these studies, see Juristo et al. (2004). The goal of these studies is not the same as the work presented here, and they either deal with selected aspects of interest about the techniques and/or are not exhaustive as regards the universe of testing techniques. Therefore, they are not a solution for the problem with which we are concerned here, although the knowledge they input can be used in the characterization schema that we propose.

On the other hand, there are several areas that deal with problems related to information characterization and packaging. Even though these efforts do not fall within the testing area, they have been considered as related research, since they also aim to characterize relevant information.

Attempts have been made to characterize different software artefacts within the area of reuse. When software artefacts are reused, there is usually a reuse repository in which these artefacts are kept. Characterizations should be available for selecting the best-suited artefact from the repository. For example, Prieto-Díaz (1989) or Kontio et al. (1996) have proposed characterization schemas for reusable modules, but the information that is important for selecting reusable models is not relevant for selecting testing techniques. The proposal by Basili and Rombach (1991) for characterizing all software element types (processes, products, techniques, etc.) also falls within the area of reuse. Henninger (1996) employs a similar approach, trying to capture the knowledge of software developers. The schemas proposed by these researchers do not take into account specific testing technique characteristics.

The area of technologies selection covers work on characterizing techniques, methods and tools related to software development for later use. Birk (1997) proposes a generic characterization schema for software technology selection. He also provides a series of guidelines for generating the relevant characteristics of a given technology. This schema is too generic and is not useful for selecting testing techniques. Maiden and Rugg (1996) suggest an approach for selecting requirements elicitation techniques, which means that it cannot be used for testing techniques.

Therefore, there is no research that specifically deals with the problem of characterising testing techniques. The work that there is on characterization schemas for other artefacts cannot be used, because many of the attributes of which they are composed are specific to the artefacts that they characterize and make no sense for testing techniques. On the other hand, the proposed general-purpose characterization schemas (which are supposed to characterize any software development artefact) have been very useful as a starting point for getting an idea of what type of attributes may be important. However, we have not been able to use them as a solution to the problem of selecting testing techniques, because, on the one hand, the attributes of which these schemas are composed need to be instantiated for testing techniques and, on the other hand, testing technique-specific attributes need to be added to the characterization schema.

3. Research Goals

As there are numerous techniques that are not universally applicable, it makes sense to talk of a need for selection. It is generally accepted that the selection process involves comparing given technique characteristics with project needs to select the techniques whose applicability conditions are most resemblant of the conditions of the project at hand. Accordingly, the selection problem can be divided into two subproblems:

- Appropriate identification of the relevant characteristics for selection.

- Completeness of the set of techniques on which the selection is based.

If we do not take into account all the relevant characteristics for selection, we could overlook an important attribute of the technique that makes it unsuitable for the situation in which it is to be used. If the original set of techniques is not complete, the technique that is best suited for the case at hand could be missing.

Therefore, it is reasonable to expect that the actions to be taken by a tester who intends to make a systematic and balanced selection of testing techniques are:

1. List the techniques from which the selection is to be made.

2. Assess the relevant characteristics for the listed techniques.

3. Assess the characteristics of the project in question.

4. Compare 2 and 3, and decide which is the best-suited technique for the project at hand.

However, the process actually followed, which is much quicker, although less effective, does not usually lead developers to find the best-suited techniques. This process is characterized as follows:

- At best, developers list only the techniques with which they are acquainted and, at worst, which they have already used before.
- The evaluation function used during assessment is subjective and variable, not so often because of the person making the evaluation but because exactly what the relevant characteristics are is not known.

The scenario proposed here to overcome these two problems is as follows: the testers make the selection based not on their knowledge but on the information contained in a repository generated by successive instantiations of the characterization schema proposed here for different techniques. This repository will provide both a complete set of techniques and technique information that is relevant for selection purposes. Accordingly, the evaluation function will have been specified, and the techniques will be selected in full knowledge and not by intuition.

And, how is this repository built? On the one hand, from information provided by testers (hereinafter called testing technique consumers), extracted from their experiences using the techniques and, on the other, from information provided by researchers in the area (hereinafter called testing technique producers), extracted from the results of their research on developing new and studying the applicability conditions of existing techniques. The librarian, who will be the person in charge of maintaining the repository, will supervise all this information.

The idea is based on a hypothesis formulated by Basili and Rombach (1991) that states:

The process of selecting software artefacts is improved using artefact characterization schemas.

So, the objective of this paper is to test an instance of this hypothesis:

The use of a characterization schema improves the process of selecting the testing techniques to be used in a given software project in terms of the quality of and the time it took to get the solutions, compared with using books.

Table 1 shows the aspects to be evaluated and the questions to be answered by the research.

4. Building a Characterization Schema for Testing Techniques

When deciding which information is of interest for selecting testing techniques, we could form an opinion about what information appears to be most relevant. On the basis of a subjective opinion alone, however, we cannot be sure that this information will be of

Table 1. Research questions.

Aspect	Viewpoint	Question
Feasibility	Producer	1. Is it possible to describe at least one testing technique?
	Consumer	2. Is it possible to decide whether or not to use a testing technique in at least one situation?
Flexibility	Producer	3. Can the schema be used to characterize any existing technique?
		4. How formal does a technique have to be for it to be possible to characterize it?
Completeness	Consumer	5. Is the information considered in the schema sufficient for selection purposes?
		6. What information is missing from the schema?
Effectiveness	Consumer	7. Can the schema be used to select the best-suited techniques?
		8. Is there any case in which the schema cannot be used to decide which technique to use?
		9. How can effectiveness be improved?
Efficiency	Consumer	10. How long does selection take using the schema?
		11. How many resources are required to use the schema?
		12. How long does it take to decide whether or not to use a technique?
		13. How could efficiency be improved?
Usability	Consumer	14. How long does it take to learn how to use the schema?
		15. How many explanations are required during schema use?
		16. What sort of explanations are required?
		17. How often is help consulted during schema use and for how long?
		18. Are the names that appear in the schema appropriate?
		19. How can usability be improved?
User satisfaction	Consumer	20. What are the advantages and disadvantages of using the schema?
		21. Would people be prepared to use it?
		22. What improvements could be made?
		23. What do people like and dislike about the schema?
		24. What is a good environment for using the schema?
		25. Is there any superfluous or redundant information?

interest to the people who are going to use the repository in the future or that it is really relevant for the selection process in question. So, the process we followed to generate a schema that would reflect the relevant information for selection as accurately as possible is as follows. Firstly, we built a schema reflecting our view of the selection problem. Then, this schema was complemented with the producers' and consumers' view of the problem. Finally, experts in the testing area inspected the generated schema. At this point, the schema was ready for evaluation. This process is practicable for generating a characterization schema for other SE artefacts or techniques. Vegas et al. (2003) detail the generic process that other researchers can use to generate other characterization schemas. In the following, we discuss the generation of the characterization schema for testing techniques stage by stage.

4.1. First Iteration: Theoretical Schema

As mentioned above, the first step to building the characterization schema is to generate the schema according to our own view. For this purpose, we gathered information from books and analysed testing techniques to study the differences and similarities between them. From this survey, we extracted a number of technique characteristics that need to be specified to be able to make a distinction between testing techniques. Then, we organised the information around the elements involved in this process. More precisely, the information we found relevant for testing technique selection is related to:

- The *technique*. Some testing technique characteristics, such as application cost or how easy they are to understand, can be relevant for deciding on the use of one or another.

- The *results* of applying the technique, or test cases, will also have characteristics of interest for selecting a testing technique. How many test cases the technique generates or the number of repeated test cases are examples of such features.

- Some characteristics of the software or *object* on which the technique is to be applied can determine the use of one technique or another, for example, the programming language used, software size, etc.

- *Agents*. The people who are going to use the technique. It may be more advisable to use one technique than another depending on their characteristics.

The theoretical schema generated in this first step is shown in Table 2.

Table 2. Theoretical schema.

Element	Attribute	Description
Technique	Tools	Available tools that ease the use of the technique
	Comprehensibility	Whether or not the technique is easy to understand
	Cost of application	How much effort is needed to apply the technique
	Sources of information	Where to find information about the technique
	Dependencies	Relationships of one technique to others
	Repeatability	Whether two people generate the same test cases
	Adequacy criterion	Test case generation and stopping rule of the technique
Results	Completeness	Coverage provided by the set of test cases
	Cost of execution	Time the technique takes to execute the test set
	Type of defects	Type of defects the technique helps to discover
	Effectiveness	Capability of the set of test cases to detect defects
	Correctness	Test cases to be deleted from the set
	Adequacy degree	Extent to which the adequacy criterion is achieved
Object	Phase	Stage of development at which the test is to be run
	Element	Elements of the system on which the test acts
	Aspect	Functionality of the system to be tested
	Software architecture	Development paradigm to which the technique is linked
	Software type	Type of software the technique can test
	Programming language	Programming language with which the technique can be used
	Development method	Development method or life cycle to which the technique is linked
Agents	Experience	Experience required to use the technique
	Knowledge	Knowledge required to be able to apply the technique

4.2. Second Iteration: Empirical Schema

Our view of software testing is complemented with the opinion of producers and consumers who are, ultimately, the ones who are going to use the schema. For this purpose, we surveyed a series of producers and consumers, who were asked what information they believed to be relevant for fully describing the properties of (producers) or selecting (consumers) a testing technique.

One of the key tasks for designing the empirical schema was the selection of the respondents. The characteristics of the people involved in the construction of the empirical schema can have a significant influence on the resulting schema. The people involved should be as heterogeneous as possible to assure that the schema does not reflect a unilateral viewpoint. Therefore, we looked for people who played different roles in the testing area. Each of the participant subjects was described for the purpose of examining schema coverage with respect to the type of respondents. The parameters used to describe the respondents are: current position, years of service, company/institution, background and experience in testing. Also, depending on the role played by the respondents (producer or consumer), they were asked respectively: area of interest in software testing or experience in software development.

With regard to the companies at which the respondents work, there are two respondents who are faculty members, two respondents who work for centres associated to universities and twelve respondents who belong to different software company sizes (small, medium-sized and large). The group includes two university professors, five software department managers (four from development departments and one from a research department), two project managers, six developers and one scientist. The number of years of service is variable. The experience of the group in software testing also varies and, interestingly, in accordance with the positions held by the respondents. For consumers: the project managers are experienced in planning the testing process within the project, as well as in system testing and acceptance; the software engineers are experienced in unit testing and some in integration testing; the professor is experienced in test planning. The producers are experienced in research. They are all in possession of qualifications ranging from bachelor degrees in computer science, through master degrees in software engineering or computer science, to doctorates in computer science, except two, one of whom is a bachelor in physics (now taking a doctorate in computer science) and the other is a master in electrical engineering. The experience of the consumers in software development varies from three to 22 years, the mean being about 12 years. The areas of interest within software testing for producers are: the study of technique applicability for one and new techniques for the other two.

One of the key questions at this stage is how to decide when to stop gathering information, that is, when the set of information gathered can be considered as representative of the opinion of producers and consumers. Moreover, it is not easy to find people prepared to spend time completing the questionnaire. Therefore, at the same time as information was gathered, we ran a schema stability analysis. Figure 2 shows the results of this analysis, illustrating the accumulated growth speed of the empirical schema.

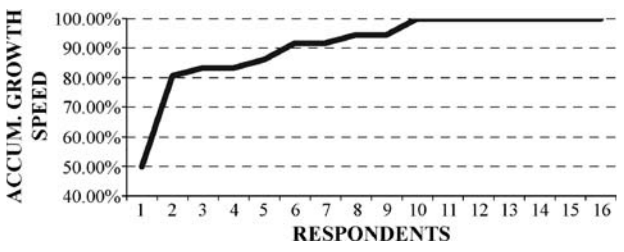

Figure 2. Schema growth speed.

The x-axis of Figure 2 shows the surveyed subjects. The y-axis shows the empirical schema size as a percentage of its final size for this phase at each point. Note that the schema reaches 50% of its final size with the first respondent, rising to 80% for the second. Furthermore, we find that the schema reaches its final size with the tenth respondent. This means that the last six respondents (37.5% of the total) added no new information to the schema. Therefore, the schema could be considered as stable and surveying could stop. Table 3 shows the contents of the empirical schema.

Note that the empirical schema provides some information that did not appear in the theoretical schema (put together from books), since practitioners care about practical issues that theoreticians very often overlook. In particular, we are referring to attributes related to testers' experiences using the technique. To distinguish this information from information appearing in the theoretical schema, we organised the information about testing techniques at two levels:

- *Operational level*. This level is related to the optimal conditions of testing technique operativeness, once given characteristics of the environment in which the technique is to be applied have been determined. This means that it may or may not be appropriate to apply a given technique depending on the knowledge and experience of the personnel and whether or not the available tools are suitable. This level contains all the elements that appeared in the theoretical schema—technique, results, object and agents—, plus a new element, called *tools*, which appeared in the theoretical schema as an attribute and contains information regarding the tools available for a technique.

- *Use level*. This level specifies subjects' earlier experiences of technique use. This level contains two elements:

  – *Project*, which specifies information regarding earlier projects in which the technique was used.

  – *Satisfaction*, which specifies what opinion the technique merits among people who have used it before.

Table 3. Empirical schema.

Level	Element	Attribute	Description
Operational	Technique	Comprehensibility	Whether or not the technique is easy to understand
		Maturity level	How experimental and/or how well validated the technique is
		Cost of application	How much effort is needed to apply the technique
		Inputs	Inputs required to apply the technique
		Adequacy criterion	Test case generation and stopping rule of the technique
		Test data cost	Cost of identifying the test data
		Dependencies	Relationships of one technique to another
		Repeatability	Whether two people generate the same test cases
		Sources of information	Where to find information about the technique
	Results	Coverage	Coverage provided by the set of test cases
		Effectiveness	Capability of the set of cases to detect defects
		Type of defects	Type of defects the technique helps to discover
		Number of generated cases	Number of test cases generated per software size unit
	Object	Phase	Stage of development at which the test is to be run
		Element	Elements of the system on which the test acts
		Aspect	Functionality of the system to be tested
		Software type	Type of software that can be tested using the technique
		Software architecture	Development paradigm to which the technique is linked
		Programming language	Programming language with which the technique can be used
		Development method	Development method or life cycle to which the technique is linked
		Size	What size the software should be to be able to use the technique
	Tools	Identifier	Name of the tool and the manufacturer
		Automation	Part of the technique automated by the tool
		Cost	Cost of tool purchase and maintenance
		Environment	Platform (sw and hw) and programming language with which the tool operates
		Support	Support provided by the tool manufacturer
	Agents	Experience	Experience required to use the technique
		Knowledge	Knowledge required to be able to apply the technique
Use	Project	Reference projects	Earlier projects in which the technique has been used
		Tools used	Tools used with the technique in earlier projects
		Personnel	Personnel who used the technique on earlier projects
	Satisfaction	Opinion	General opinion about the technique after having used it
		Benefits	Benefits of using the technique
		Problems	Problems with using the technique

4.3. Synthesis of Perspectives: Proposal of a Preliminary Schema

The goal of this stage is to synthesise all the viewpoints (our and the producers' and consumers' opinions) to create a single schema, by integrating the theoretical and empirical schemas.

A series of rules were followed to synthesise the two schemas in an orderly fashion:

1. The levels and elements of the synthesised schema will be the union of the levels and elements of the original two schemas.

2. Any attributes that appear in just one of the characterization schemas will appear unchanged in the synthesised schema.

3. Any attributes that appear in both schemas and are equal[1] will appear unchanged in the synthesised schema.

4. Any attributes that appear in the two schemas and are similar[2] will be studied to decide whether they are used to generate one or several attributes.

5. In no case will information be deleted from the characterization schema.

Table 4 shows the results of the synthesis, specifying the source of each schema attribute. The column labelled theoretical indicates the attributes from the theoretical schema and the column labelled empirical designates the attributes from the empirical schema.

It is interesting to note that 14 of the attributes present in the preliminary schema do not appear in the theoretical schema (put together from the testing technique characteristics specified in books). On the other hand, there are only two attributes that are present in the preliminary schema and not in the empirical schema. This means that the empirical and the preliminary schemas are almost exactly the same, except for two attributes. In other words, 55.5% of the attributes of the preliminary schema are common to the original two schemas, whereas the theoretical schema supplies 5.5% and the empirical schema 39% of the other 44.5%. This result is surprising, because the information for the theoretical schema was gathered by examining the differences and similarities between techniques, as well as data found in textbooks regarding the characteristics on which testing technique selection is based, which would appear to be a fairly comprehensive set of sources.

The major omissions of the theoretical schema are pragmatic aspects: the *use* level and the *tools* element. Minor omissions are some attributes of the *technique* element (*maturity level*, *inputs* and *data cost*) and an attribute of the *object* element (*size*). The empirical schema, on the other hand, has theoretical omissions, namely, two attributes of the *results* element (*correctness* and *adequacy degree*).

To get a better understanding of the origin of this difference, let us look at the sources of the empirical schema. Of the 34 attributes of which it is composed, 73.5% are input exclusively by practitioners. On the other hand, researchers input nothing new to the schema, as there is not one attribute input by researchers alone. Finally, 26.5% of the attributes are input by both practitioners and researchers. Taking this into

Table 4. Synthesised schema.

Level	Element	Attribute	Theoretical	Empirical
Operational	Technique	Comprehensibility	X	X
		Maturity level	–	X
		Cost of application	X	X
		Inputs	–	X
		Adequacy criterion	X	X
		Test data cost	–	X
		Dependencies	X	X
		Repeatability	X	X
		Sources of information	X	X
	Results	Completeness	X	X
		Correctness	X	–
		Effectiveness	X	X
		Type of defects	X	X
		# of generated cases	X	X
		Adequacy degree	X	–
	Object	Phase	X	X
		Element	X	X
		Aspect	X	X
		Software type	X	X
		Software architecture	X	X
		Programming language	X	X
		Development method	X	X
		Size	–	X
	Tools	Identifier	X	X
		Automation	–	X
		Cost	–	X
		Environment	–	X
		Support	–	X
	Agents	Experience	X	X
		Knowledge	X	X
Use	Project	Reference projects	–	X
		Tools used	–	X
		Personnel	–	X
	Satisfaction	Opinion	–	X
		Benefits	–	X
		Problems	–	X

account, it can be said that most of the information that is not in the testing books, and therefore does not appear in the theoretical schema, really comes from practitioners. Therefore, the two schemas provide a fully complementary view: theory and practice.

4.4. Schema Improvement: Expert Peer Review

As mentioned above, after synthesising the schemas, we thought it was advisable to send the resulting schema, along with a questionnaire, to a number of experts in the testing area. An expert is defined as highly reputed person in the area of testing, with lengthy experience in both the theory and practice of the testing process and their knowledge of the people involved in this process. The experts were asked to give their opinions of the schema on form and substance. The opinions on form included issuing judgements about the suitability of schema organisation or the names that appear in the schema. The opinions on substance included issuing judgements about the existence of possible redundancies, missing information, etc., in the schema.

Additionally, the expert responses were analysed by checking whether there were contradictory or coincident before finally making a decision on whether or not to accept the suggestion, and if so, how to add the suggestion to the schema. Indeed, a series of rules are followed to decide whether or not to accept the expert suggestions:

1. If the experts disagree, the majority view will be respected.

2. If more than one expert recommends a given change, the recommendation will be taken into account.

3. If only one expert recommends a change, this change will be accepted provided the proposed change is not due to a misinterpretation of the schema, its logic or its contents. In other cases, however, a change is not always as evident as when it is recommended by one rather than several experts. Then, it is the expert's versus our opinion. It is sometimes impossible to reconcile the two viewpoints, and it was decided that our opinion should take precedence, as it did not seem to make sense to make modifications in which we did not believe or about which we were unsure. This happened because experts have a bigger bias towards producers than consumers. Their advice sometimes interfered with consumers' opinions. To safeguard the views of consumers (who are ultimately the beneficiaries of the schema), we opted not to systematically add expert opinions to the schema.

The questionnaire was originally sent to twelve testing experts, of which only four responded. Therefore, the schema was reviewed by four experts, whose particulars are:

- *Expert 1.* Professor of ICMC/USP; expert in testing.

- *Expert 2.* NASA systems analyst at GSFC/Unisys. Her experience in testing includes ten years as a developer and eighteen years as a researcher within the area.

- *Expert 3.* Senior researcher at IEI/CNR (Italy); ten years' experience as a researcher in the software testing area.

- *Expert 4.* Full professor at Portland State University; thirty years' experience as a researcher and professor within the software testing area.

It is clear that all the respondents have lengthy experience mainly as researchers, although some are also experienced as developers within the testing area. Additionally, it is worth mentioning that they are people of international repute within the testing area.

Table 5 shows the results of the expert peer review, leading to the final schema. From this table, we find that the changes made owing to the experts' suggestions are:

- Four attributes have been deleted from the *operational* level: *phase, maturity level*, *adequacy degree* and *software architecture*.

- The *correctness* attribute of the *operational* level was replaced by another named *precision.*

- The *results* element was renamed as *test cases*.

- The *use* level was renamed as *historical* level.

Table 6 shows what values the attributes can take.

5. Feasibility and Flexibility Evaluation from the Producer Viewpoint

The objective of this evaluation is to check schema feasibility from the producer viewpoint, as well as to evaluate schema flexibility. In other words, the primary goal of this evaluation is to find out whether it is possible to locate the information the schema contains, how this information can be used during selection and whether it is possible to instantiate the schema for any existing testing technique. Each of these points is discussed below.

A testing technique was instantiated to check feasibility from the producer viewpoint. The chosen technique was *decision coverage*, from the control flow technique family. The values found for this technique are shown in Table 7. They have been obtained from Beizer (1990); Frankl and Iakounenko (1998); Frankl and Weiss (1993); Hutchins et al. (1994); Myers (1970); Pfleeger (1999); Sommerville (1998); and Wood et al. (1997). When a value is not known the '—' symbol has been used.

The main finding from this instantiation is that the schema is feasible from the producer viewpoint. Other additional conclusions were:

- There is information that is difficult to find, especially information related to reference projects. This is because companies do not like to see their private data published, if they actually keep a record of these data. A cultural change has to take place at companies for it to be possible to get this information.

- There were two schema attributes (*precision* and *completeness*) whose value was not found anywhere. This casts doubts upon the advisability of these two attributes appearing in the schema. However, they are found in both the theoretical and empirical schemas and the experts did not consider them unsuitable. This appears to be relevant

Table 5. Final schema.

Level	Element	Attribute	Description
Operational	Agents	Knowledge	Knowledge required to be able to apply the technique
		Experience	Experience required to be able to apply the technique
	Tools	Identifier	Name of the tool and the manufacturer
		Automation	Part of the technique automated by the tool
		Cost	Cost of tool purchase and maintenance
		Environment	Platform (sw and hw) and programming language with which the tool operates
		Support	Support provided by the tool manufacturer
	Technique	Comprehensibility	Whether or not the technique is easy to understand
		Cost of application	How much effort it takes to apply the technique
		Inputs	Inputs required to apply the technique
		Adequacy criterion	Test case generation and stopping rule
		Test data cost	Cost of identifying the test data
		Dependencies	Relationships of one technique with another
		Repeatability	Whether two people generate the same test cases
		Sources of information	Where to find information about the technique
	Test cases	Completeness	Coverage provided by the set of cases
		Precision	How many repeated test cases the technique generates
		Effectiveness	What capability the set of cases should have to detect defects
		Defect type	Defect types detected in the system
		Number of generated cases	Number of cases generated per software size unit
	Object	Element	Elements of the system on which the test acts
		Aspect	Functionality of the system to be tested
		Software type	Type of software that can be tested using the technique
		Programming language	Programming language with which it can be used
		Development method	Development method or life cycle to which it is linked
		Size	Size that the software should have to be able to use the technique
Historical	Project	Reference projects	Earlier projects in which the technique has been used
		Tools used	Tools used in earlier projects
		Personnel	Personnel who worked on earlier projects
	Satisfaction	Opinion	General opinion about the technique after having used it
		Benefits	Benefits of using the technique
		Problems	Problems with using the technique

Table 6. Attribute values of the schema.

Level	Element	Attribute	Values
Operational	Technique	Comprehensibility	(high, medium, low)
		Cost of application	(high, medium, low)
		Inputs	(requirements, code, design, etc.)
		Adequacy criterion	family (data flow, flow control, etc.) and technique (sentence coverage, etc.)
		Test data cost	(high, medium, low)
		Dependencies	Two values: [technique] and dependency type (should be applied before, after, should never be used with, etc.)
		Repeatability	(yes, no)
		Sources of information	(a person, a book, an article, an experiment, etc.)
	Test cases	Completeness	Percentage
		Precision	Percentage
		Effectiveness	Percentage
		Defect type	(control, assignation, initialisation, etc.)
		Number of generated cases	Formula
	Object	Element	(function, procedure, system, subsystem, etc.)
		Aspect	(communications, database, GUI's, etc.)
		Software type	(real time, batch, interactive, expert system, etc.)
		Programming language	(structured, functional, logical, real time, concurrent, etc.)
		Development method	(prototyping, reuse, waterfall, knowledge-based system, etc.)
		Size	(number in KLOC)
	Tools	Identifier	Two values: [tool name] and [company name]
		Automation	(flow chart, mutant generation, test case generation, etc.)
		Cost	Two values: [purchase cost] and [maintenance]
		Environment	Three values: [SW requirements], [HW requirements] and [programming language]
		Support	(24-hour hotline, technical assistance, etc.)
	Agents	Experience	(tool understanding, etc.)
		Knowledge	(cyclomatic complexity, flow charts, etc.)
Historical	Project	Reference projects	[project name]
		Tools used	[tool name]
		Personnel	[people's names]
	Satisfaction	Opinion	[sentence or paragraph explaining the opinion]
		Benefits	[sentence or paragraph explaining the benefits of the technique]
		Problems	[sentence or paragraph explaining the drawbacks of the technique]

information that is not available in the literature on testing techniques. So, it is an omission of the testing literature, not a fault of the schema, as this information is considered relevant from all viewpoints (note that there are not many attributes in the schema of which this can be said). This omission is due to the fact that the knowledge on testing techniques is not yet mature enough.

Table 7. Decision coverage technique.

Level	Elem.	Attribute	Value
Operational	Technique	Comprehensibility	High
		Cost of application	Low
		Inputs	Source code
		Adequacy criterion	Control Flow
		Test data cost	Medium
		Dependencies	Supplemented with techniques that find processing errors
		Repeatability	No
		Sources of information	Sommerville
	Test cases	Completeness	—
		Precision	—
		Defect type	Control
		Effectiveness	48% over the total faults found in the code
		# of generated cases	Exponential # decisions
	Object	Element	Units
		Aspect	Any
		Software type	Any
		Programming language	Any
		Development method	Any
		Size	Medium
	Tools	Identifier	LOGISCOPE
		Automation	Obtain paths
		Cost	€3,000–6,000
		Environment	Windows; C/C++
		Support	24 Hot-line
	Agents	Knowledge	Flow graphs
		Experience	None
Historical	Project	Reference projects	—
		Tools used	—
		Personnel	—
	Satisfaction	Opinion	OK, but should be complemented with others
		Benefits	It is easy to apply
		Problems	No dynamic analyser should be used with real-time and concurrent systems due to code instrumentation

- Information about the testing techniques is sometimes contradictory. This is precisely the case of schema attributes for which the knowledge on testing techniques is not yet mature enough, such as effectiveness (see for example Beizer (1990) and Wood et al. (1997)) or the number of generated cases (see for example what is stated in Weyuker (1990)). Therefore, we sometimes find that different information sources bring to light different, and sometimes contradictory, information about the same attribute. In the case of the number of generated cases, for example, we find that books (mechanical knowledge) differ from research papers (theoretical and empirical knowledge) on the attribute value they give for some techniques. This way, assertions are sometimes made that limit the applicability of the claim in question.

- The metrics used to fill in some attributes are not easy to interpret, such as the metric associated with effectiveness, which is *probability of detecting one fault (over the total set of faults)*. Can this metric be considered valid for specifying how many faults a technique can detect? Would not the *percentage of faults that the technique can detect* or the *probability of detecting a given fault* be a better choice? This problem could be solved if producers used more pragmatic metrics, for example, the ones suggested in the schema.

We decided to select a number of technique families, which covers the variety of techniques between families, and a number of techniques within each family, which covers the variety of techniques within each family, to check **schema flexibility**. Additionally, we opted for well-known techniques, as this gives a better understanding of how the schema is instantiated. The chosen techniques were:

- *Functional techniques*. Boundary value analysis and random testing.
- *Control-flow techniques*. Sentence coverage, decision coverage, path coverage and threads coverage.
- *Data-flow techniques*. All-c-uses, all-p-uses, all-uses, all-du-paths and all-possible-rendezvous.
- *Mutation*. Standard mutation and selective mutation.

We were able to instantiate all the chosen techniques (see Vegas (2002)). Of course, this does not mean that the schema is totally flexible, as it would be necessary to instantiate the schema for *all* existing testing techniques for this purpose. However, the fact that we were able to instantiate a number of techniques that are representative of existing techniques without any problem indicates that the schema is flexible enough to be able to instantiate the huge majority of, if not all, testing techniques.

6. Schema Use: Feasibility Evaluation from the Consumer Viewpoint

Before checking schema feasibility from the consumer viewpoint, we need to prescribe the procedures associated with the use and evolution of the repository. As mentioned in Section 3, the repository will be used directly by producers and consumers and indirectly by the librarian. Producers will be able to provide new information for the repository. Consumers, likewise, will be able to select testing techniques for the projects on which they are working. They will also be able to provide feedback on the contents and the actual structure of the schema from the results of repository use and the selected techniques. Finally, the librarian will update the repository on the basis of the information supplied by producers and consumers, taking care to maintain the coherence of the information it contains. There are, then, five procedures, each associated with repository use, which will also permit its evolution. Figure 3 illustrates these uses.

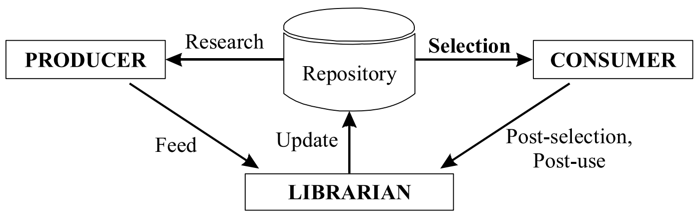

Figure 3. Schema use.

Below, we elaborate on what we consider to be the primary schema use: selection. Two concepts need to be introduced to be able to explain this process:

- *Bounded variables*. These are variables whose value is imposed by the project and cannot be changed during selection. For example, the project development method, the type of software under development, etc.

- *Non-bounded variables*. These are variables whose value can be changed depending on the current selection needs and/or preferences. For example, the characteristics of the people who are to apply the testing techniques or the tools to be used are not necessarily pre-established by the project in question.

The steps for schema use by consumers for selection purposes are established on the basis of these definitions as:

1. Identify the desired technique values for the attributes that belong to the object element of the schema, as well as the *effectiveness* and *defect type* attributes of the test cases element.

2. Identify whether there are other attributes that could impose any sort of constraint on the technique to be chosen (for example, whether given personnel should be used).

3. Compare the desired values of the attributes with the values of each technique the repository contains. Preselect the techniques whose values match the specified values.

4. If the set of preselected techniques is empty, relax one of the constraints and return to step 3.

5. Examine the remaining attribute values of the preselected techniques, paying special attention to the *dependencies* attribute.

We give an example of a selection to check **schema feasibility** from the consumer viewpoint. For this purpose, the repository contained in Vegas (2002) was used. The problem posed was as follows:

A car park management system (concurrent system) is to be built. At this stage of the project, the QA team has identified the key quality attributes of this software system. These were obtained by examining the characteristics of the software under development, as well as its application domain. In this particular case, the essential attributes are: correctness, security and timing.

The project situation is as follows: the system is to be coded in Ada, the development team is fairly experienced in developing similar systems, and almost all the errors they are found to make are typical of concurrent programming. The testing team is also experienced in testing this type of systems.

It was solved as follows.

1. *Determination of bounded variables*, shown in Table 8.

2. *Pre-selection of an initial set of techniques*. The values of the bounded variables identified in the previous step were compared with the technique values contained in the repository. The techniques selected after situation/technique matching are: *boundary value analysis*, *random*, *path coverage*, *all-possible-rendezvous*, *all-c-uses*, *all-p-uses*, *all-uses*, *all-du-paths*, *standard mutation* and *selective mutation*. The *sentence coverage* and *decision coverage* techniques will be rejected because their effectiveness is low, and the technique *threads coverage* will be discarded because it is for object-oriented software.

3. *Identification of the best-suited techniques for selection*. Of the pre-selected techniques, there is one that is specific for Ada-style programming languages. Although there are general-purpose techniques that are more effective, the technique that is specific for concurrent software appears to detect the faults proper to concurrency better than the other techniques. Furthermore, the *path coverage* technique states that when used with concurrent and real-time systems, a dynamic analyser cannot be used as a tool, which is available in this case and it would be good to be able to use. Additionally, the techniques *all-c-uses*, *all-p-uses, all-uses*, *all-du-paths, standard mutation* and *selective mutation* cannot be used without a tool (which is not available in this case). Therefore, the *all-possible-rendezvous* techniques will be selected.

Table 8. Bounded variables.

Level	Element	Attribute	Value
Operational	Test cases	Effectiveness	>50%
	Object	Element	ANY
		Aspect	ANY
		Software type	Real time
		Program. language	Ada
		Size	Medium

However, the dependency attribute states that the technique should be supplemented with a black-box technique. Observing the black-box techniques in the pre-selected set (*boundary value analysis* and *random*), it is found that the *random testing* technique is useful for people with experience in the type of tests to be run and will, therefore, also be selected.

From this, we conclude that the schema is feasible from the consumer viewpoint, as at least one selection has been able to be made.

7. Experimental Evaluation

The objective of this evaluation is to check schema completeness, effectiveness, efficiency, usability and user satisfaction from the viewpoint of the consumers in all cases. The primary aim of this evaluation is to try to understand how schema use affects the testing technique selection process. That is, whether the schema is really an improvement on selection using other resources (basically books) or, contrariwise, it is preferable to carry on using the traditional selection process, because the schema increases the workload; and whether the schema is of assistance to consumers in the sense that it improves the work they do. To assure that the comparison between the schema and books was as balanced as possible, the subjects who participated in the experiment have worked with a catalogue on paper rather than the automated repository in the shape of a tool. For any additional information about the experiment that is not detailed here for reasons of space (forms used, statistical analyses employed, as well as results validity tests, etc.), see Vegas (2002).

The null **hypotheses** of this experiment are:

H_{01}: The efficiency of the selection process is independent of the method used for the purpose and the project in question.

H_{02}: The usability of the method of selection is independent of the method used for the purpose and the project in question.

H_{03}: The completeness of the original set of information for making the selection is independent of the method used for the purpose and the project in question.

H_{04}: The effectiveness of the selection process is independent of the method used for the purpose and the project in question.

The fifth aspect, user satisfaction, will not be considered for establishing a hypothesis that can be refuted by means of statistical evidence. This aspect will be assessed informally, examining the opinions of each subject.

Table 9 reflects the characteristics that either do not influence or are not intended to influence the result of the experiment; that is, the parameters and their assigned values.

Table 9. Parameters for the experiment.

Parameter	Value
Subject experience	Inexperienced
Task to be performed	Testing technique selection
Documentation	Requirements
	Project context
Test	Unit
Attribute	Correctness

Below, we will describe the two **factors** in this experiment (or variables whose value varies), because we want to see how they affect the response variable. These two factors are the selection method and the software project:

- *Method of selection*. This factor has two possible alternatives: books and schema. For selection using books, the subjects will be given three books, which will be the only ones they are allowed to use. The books used are highly reputed (Beizer, 1990; Pfleeger, 1999; and Sommerville, 1998). For selection using the schema, the subjects will be given the repository discussed above and will follow the process defined in Section 6.

- *Software project*. This factor will encompass all the project characteristics, from the software system, through time and financial project constraints, to personnel, etc. The four software projects chosen are: a video club management system (M), a bank loan approval system (B), a car park control system (S) and a system for monitoring pump water level (RT).

The **response variables** of this experiment (outputs that reflect the relationships between the different factor levels) are shown in Table 10. They will be gathered by means of nine forms designed for the purpose.

The experiment was run with 87 subjects, final-year students at the Technical University of Madrid's School of Computing, who had already taken 30 SE credits, including 6 credits on software evaluation. Before carrying out the experiment, a questionnaire was given to the subjects to gather information a priori on their characteristics.

1. *Work experience*. The subjects used for the experiment are 87 final-year students at the School of Computer Science, Technical University of Madrid. 50% of them have no work experience and the other 50% are experienced as follows: 50% have worked as developers, 21% have worked as analysts, 24% have been members of a testing team and 5% have worked as project managers. Of the experienced subjects, 45% have less than six months of work experience, 30% from six months to a year and 25% over a year.

2. *Experience with software testing*. Of the subjects, 75% are acquainted with software testing only through what they have studied at university, whereas the other 25% have run tests as part of their work. As regards experience in testing, 70% routinely run

Table 10. Response variables for the experiment.

Aspect	Response variable
Completeness	Amount of info used during selection
	Sort of the info used during selection
	Amount of info missed during selection
	Sort of the info missed during selection
Usability	# problems found during selection
	Description of the problems found
	# times schema help consulted
	Attributes consulted in help
Effectiveness	# techniques considered during selection
	# techniques selected
	Techniques selected
Efficiency	Time spent studying the techniques
	Selection time
	Time spent consulting doubts about schema
User Satisfaction	Advantages and disadvantages of schema use
	Would you be willing to use it?
	What improvements you would make?
	What did you like or not like?
	Has your view of selection changed?
	What have you learnt?
	Would you do things differently next time?
	Suitability of attribute names and distribution

tests as part of the practical exercises set for their degree course, 6% have only done run tests in small exercises and 24% have been involved in testing as part of real development projects.

3. *Testing technique selection heuristic.* None of the students have a heuristic for selecting testing techniques, either because they have never considered the problem or they do not consider selection to be necessary (they always apply the same techniques).

The subjects were assigned randomly, according to their profile, to the four established groups. The experimental design is discussed below. Table 11 shows how the methods of selection were assigned to groups.

From Table 11, we can see that the subjects of group 1 and 3 will make both selections with the schema and books, respectively. Groups 2 and 4 will make the selection first with books (schema) and then with the schema (books). According to this distribution,

Table 11. Assignation of methods of selection to groups.

	Group 1 (18 pers.)	Group 2 (33 pers.)	Group 3 (18 pers.)	Group 4 (18 pers.)
Selection 1	Schema	Books	Books	Schema
Selection 2	Schema	Schema	Books	Books

groups 2 and 4 can be used to examine the effect of the method of selection (preference was given to group 2 because it represents reality), and the purpose of groups 1 and 3 is to assess the learning effect on these inexperienced subjects. We wanted the size of group 2 to be considerably larger than the others, as this would be the context in which the schema would be introduced into companies, where people already have experience in selection and try out a new method that is supposed to be better.

Continuing with the experimental design, Table 12 shows how projects were assigned to subgroups. Eight subgroups, from A to H, were set up in each group.

The following **internal threats to the validity of the experiment** were identified:

- *Copying*. There is the possibility of students copying each other, as they do the work at home. An attempt was made to solve this problem by telling the students that they would be graded on the basis not of the techniques selected but according to the effort made to do the exercise. Moreover, it should be stressed that the subjects involved in the experiment were all volunteers, which suggests that they were interested from the very start in the sort of work they were doing.

- *Capability*. It is true that not all the subjects will have the same problem-solving ability. Randomisation should minimise this problem.

- *Method learning*. If subjects apply the same method of selection twice, they will learn from the mistakes they made the first time and will do a better job the second time round (irrespective of how good the method of selection is). This threat is explicitly taken into account in groups 1 and 3. However, we have groups 2 and 4, in which the subjects apply a different method of selection each time, to counteract its effect.

- *Object learning*. If subjects use the same project for the two selections they are to make, they will also learn from the mistakes they made the first time and will do a better job the second time round (irrespective of how good the method of selection is). This has been remedied by having each subject make the two selections for different projects.

Table 12. Assignation of projects to subgroups.

	Selection 1				Selection 2			
REP.	M	B	S	RT	M	B	S	RT
A	X	–	–	–	–	–	X	–
B	X	–	–	–	–	–	–	X
C	–	X	–	–	–	–	X	–
D	–	X	–	–	–	–	–	X
E	–	–	X	–	X	–	–	–
F	–	–	X	–	–	X	–	–
G	–	–	–	X	X	–	–	–
H	–	–	–	X	–	X	–	–

- *Boredom*. Subjects may find the experiment boring and, therefore, their performance may be below normal. It is assumed that the grading of the exercise will motivate the students. Also, the subjects who have performed the experiment are volunteers, which means that they should have at least some interest in the subject.

- *Enthusiasm*. On other occasions, subjects are excited about the prospect of trying out a new method or technique, which means that they work harder on this technique. In this case, the subjects are inexperienced, which suggests that they will work equally as hard on selection with or without the schema.

- *Unconscious formalisation*. As one method is more formal than the other (schema as opposed to an ad hoc process using books), the group that uses the schema first and the books afterwards may make use of the more formal model to the benefit of books. This could mean that books come off better than they really should. In case this happens (group 4), the other three groups will control the possible effect of this group.

- *Procedure*. Something that can, and actually did, occur is that the subjects do not follow the process they were told to for selection using the schema. This can lead to deviations in the results obtained using the schema, whether for better or worse.

And the following **external threats to the validity of the experiment** were identified:

- *Language.* The schema is instantiated in the subjects' native tongue, whereas the books are in English (a language with which many subjects are not well enough acquainted). Although it was not foreseen that the subjects would encounter this difficulty (it was thought that they would have no problem with reading technical texts in English), they did. As the subjects have to read technical texts in English in this country, this problem will mean that the results obtained here cannot be generalised to all countries (at least not to English-speaking countries), as it has benefited the schema.

- *Experience.* The subjects are not experienced, which can mean that it was harder for them to understand testing books (there is the possibility that their vocabulary on the subject is not good enough). This may have been more advantageous for the schema and will mean that the results cannot be generalised to all subject types.

- *Projects.* Four projects were used and an attempt was made for them to be representative of reality. However, experiments with more projects should be run, as not all the possible situations of a software project have been accounted for.

- *Techniques used.* A set of techniques that was as varied and complete as possible was chosen to instantiate the schema. This means that at least two members of all the families of testing techniques are represented. Also some techniques that are representative of specific software (like techniques for object-oriented or real-time software) were included. However, only unit testing techniques were accounted for. This issue would have to be addressed in more detail by running experiments with integration, system and regression testing techniques. Also, more techniques for specific software should be included.

To **analyse the quantitative data** collected during the experiment (all but user satisfaction response variables), we will use two different statistical methods: analysis of variance (ANOVA), and analysis of simple correspondences. We use the ANOVA to study the relationships between the response variables measured on a ratio scale, and the experiment factors (which are measured on a nominal scale). We want to determine whether the differences between the means of the response variable in the groups established by the combinations of factor levels are statistically significant. ANOVA was run after checking that the sample met normality and homocedasticity criteria. For the results of these tests, see (Vegas, 2002). We use the analysis of simple correspondences to describe the relationships between the response variables measured on a nominal scale, and the experiment factors (also on a nominal scale). Our aim is to describe the relationships between the categories of each variable, where similar categories appear close to each other.

Table 13 lists the results of the experiment for the stated hypotheses. The significance level of the results is also given for the quantitative variables.

The results for **schema effectiveness** appear in Table 13. It was found that the original number of techniques used to make the selection is lower for books than for the schema and varies from subject to subject; the number of selected techniques is lower for the schema than for books; and the subjects using books select families of techniques, opting

Table 13. Results for the experiment response variables.

Aspect	Response variable	Results	SIG.
Completeness	Amount of info used during selection	Books (7.79) < Schema (12.6)	0.000
	Sort of the info used during selection	Depends on project and selection method	N/A
	Amount of info missed during selection	Books (2.12) < Schema (3.72)	0.000
	Sort of the info missed during selection	Schema values not instantiated for schema	N/A
Usability	# problems found during selection	Schema (1.51) < Books (2.67)	0.000
	Description of the problems found	Missing information in both cases. With books also redundant information and lack of organisation	N/A
	# times schema help consulted	Low (9% of attributes)	0.000
	Attributes consulted in help	Concepts not intuitive for subjects	N/A
Effectiveness	# techniques considered during selection	Books (8.33) < Schema (13)	0.000
	# techniques selected	Schema (2.26) < Books (3.21)	0.000
	Techniques selected	Better tuned for schema	N/A
Efficiency	Time spent studying the techniques	Schema (84.07) < Books (279.4)	0.000
	Selection time	Schema (145.91) < Books (247.82)	0.000
	Time spent consulting doubts about schema	Negligible (3% of the time)	0.000

for a technique from the family if they are very well acquainted with it. This means that the subjects are unable to distinguish a technique from a family. This could be explained by saying that books are confusing as regards the information they provide. This could be the reason why the subjects tend to select more techniques and techniques with which they are very familiar. Finally, it should be stressed that the schema leads to more precise selections.

Schema efficiency has been examined using the results shown in Table 13. The experiment demonstrated that the schema helps to reduce both the learning and the selection time as compared with books and that the time spent consulting the schema can be considered negligible with respect to the other two. Moreover, the total time required to solve the selection problem is the sum of the learning time, plus the selection and consultation time (which is zero if books were used for selection). Accordingly, it can be concluded that the characterization schema makes selection more efficient.

The results for **schema completeness** are shown in Table 13 and examine both the information used by the subjects during selection and missing information. As regards the information used by subjects during selection, we find that subjects working with the schema used more information for selection purposes than subjects working with books. This is clear advantage of the schema over books, as selection will be better because more information is used. On the other hand, as regards the information that subjects missed during selection, it is interesting to note that subjects working with the schema missed more information than subjects working with books. This, which might, in principle, look like a drawback of the schema, has a logical interpretation, namely, that the subjects were referring to attributes not instantiated for the technique. This implies that it is important for the characterization schema to be completely instantiated for it to be useful to consumers. Another interesting point observed is that subjects are not always able to ascertain information that does not appear in, but can be easily deduced from the schema, such as the time it will take to apply the technique. We can get an idea of this value, if we know the complexity of the technique, the people, the tool and software size.

Schema usability has been examined on the basis of the results shown in Table 13. These variables can be used to compare the schema and books and to assess the schema. From the comparison, it was possible to deduce that the subjects have fewer problems using the schema than books and the frequency of appearance of each problem is lower with the schema. The main problem encountered by the subjects using the schema is the presence of uninstantiated attributes. The main problem of the subjects using books is the poor organisation of the available information, as well as missing information of interest and the existence of information that is unnecessary for selection purposes. From the schema assessment, it was possible to deduce that consultations of the meaning of attributes are generally low, and the attributes that represent concepts that are not intuitive or are difficult for the subjects to interpret are the ones most often consulted.

From this, it can be deduced that characterization schema usability is acceptable, although it could be improved by building a tool to assure that all the information is entered.

With respect to **user satisfaction (qualitative response variables)**, the subjects can be said to like the schema. They would be prepared to use (but not instantiate) it if given the opportunity, although they consider it contains too much information. They view the fact that there are uninstantiated attributes or that they have to handle so much information as a drawback.

As mentioned above, we are going to study whether there is **learning** in groups 1 and 3 and the **influence of the method** in groups 2 and 4.

- *Groups 1 and 3*. As regards learning, we find that it affects schema efficiency in these two groups. This means that as subjects become better acquainted with the schema, their selections are faster. We also found that group 3 was delighted with the prospect of using the schema, whereas group 1 failed to appreciate the schema, as they had not had to undertake a selection without it.

- *Groups 2 and 4*. Using one method before another, affects usability. This means that usability was better for books in the group that used first the schema and then books. In relation to what we discussed above, we also found that group 2 were delighted with the schema and group 4 made a more guided selection, as they used the schema for the first selection and books for the second.

8. Conclusions

The main problem met by software developers when choosing the best suited testing techniques for a software project is information. The information on testing techniques is, at best, distributed across many sources of information and, at worst, non-existent. The approach we have taken in this paper to the problem of gathering relevant information about software testing techniques is called a characterization schema. Using this schema, we can build a repository that contains the description of each technique of interest and describes all techniques according to the same pattern so that a decision can be made on whether or not to use a technique without having procedural knowledge of the technique.

An empirical and iterative process has been followed to search for the information that such a characterization schema should contain. This process is empirical because it takes into account the opinions of a variety of people involved in software testing and iterative because it is gradually refined as new opinions are added to the schema.

Finally, the generated schema has been evaluated in two ways. First, it has been instantiated for several testing techniques, by means of which we were able to test the schema for several techniques and check that it is possible to find the required information. Secondly, an experiment was run to check its behaviour against the use of other methods of selection, such as books. This experiment found that schema use is more efficient and complete than the use of books for selection purposes and that selection is less problematic. However, we were not able to demonstrate schema effectiveness, as the current state of testing technique selection is less stable than it was thought to be. We were, however, able to deduce that selections made using the schema are finer tuned than when using books.

Acknowledgment

We would like to thank all the people who have cooperated in this piece of research (producers, consumers, experts and subjects involved in the experiment) for their support.

Notes

1. Two attributes are considered equal if they bear the same name and belong to the same element and level.
2. Two attributes are considered similar if they do not bear the same name or do not belong to the same element or same level, although they represent the same or similar concepts.

References

Basili, V. R., and Rombach, H. D. 1991. Support for comprehensive reuse. *Software Engineering Journal* 6(5): September, 303–316.

Beizer, B. 1990. *Software Testing Techniques*. 2nd edition. Boston, MA: International Thomson Computer Press.

Bertolino, A. 2004. *Guide to the Knowledge Area of Software Testing*. Software Engineering Body of Knowledge. IEEE Computer Society, February. http://www.swebok.org.

Birk, A. 1997. Modelling the application domains of software engineering technologies. *Proceedings of the Twelfth International Conference on Automated Software Engineering (ASE)*. Lake Tahoe, CA, November.

Frankl, P., and Iakounenko, O. 1998. Further empirical tudies of test effectiveness. In *Proceedings of the ACM SIGSOFT International Symposium on Foundations on Software Engineering*. Lake Buena Vista, FL, USA, ACM, November, pp. 153–162.

Frankl, P. G., and Weiss, S. N. 1993. An experimental comparison of the effectiveness of branch testing and data flow testing. *IEEE Transactions on Software Engineering* 19(8): August. 774–787.

Harrold, M. J. 2000. Testing: A roadmap. *Proceedings of the 22nd International Conference on the Future of Software Engineering*. Limerick, Ireland, May, pp. 63–72.

Henninger, S. 1996. Accelerating the successful reuse of problem solving knowledge through the domain lifecycle. *Proceedings of the Fourth International Conference on Software Reuse*. Orlando, FL, April, pp. 124–133.

Hutchins, M., Foster, H., Goradia, T., and Ostrand, T. 1994. Experiments on the effectiveness of dataflow—and controlflow—based test adequacy criteria. In *Proceedings of the 16th International Conference on Software Engineering*. Sorrento, Italy, IEEE, May, pp. 191–200.

Juristo, N., Moreno, A.M., and Vegas, S. 2004. Reviewing 25 years of testing technique experiments. *Empirical Software Engineering Journal* 9(1): 7–44.

Kontio, J., Caldiera, G., and Basili, V. R. 1996. Defining factors, goals and criteria for reusable component evaluation. *Proceedings of the CASCON'96 Conference*. Toronto, Canada, November, pp. 12–14.

Maiden, N. A. M., and Rugg, G. 1996. ACRE: Selecting methods for requirements acquisition. *Software Engineering Journal* 11(3): 183–192.

Myers, G. J. 1970. *The Art of Software Testing*. New York, USA: Wiley-Interscience.

Pfleeger, S. L. 1999. *Software Engineering: Theory and Practice*. New Jersey, USA: Mc-Graw Hill.

Prieto-Díaz, R. 1989. *Software Reusability, Vol 1, Chapter 4. Classification of Reusable Modules*, Addison-Wesley, pp. 99–123.

RTI, 2002. The Economic Impact of Inadequate Infrastructure for Software Testing. Planning Report 02–3, National Institute of Standards and Technology. May.

Sommerville, I. 1998. *Software Engineering*. 5th edition. Harlow, England: Pearson Education.

Vegas, S. 2002. *A Characterization Schema for Selecting Software Testing Techniques*. Ph.D. Thesis, Facultad de Informática, Universidad Politécnica de Madrid. February.

Vegas, S., Juristo, N., and Basili, V. R. 2003. A Process for identifying relevant information for a repository: a case study for testing techniques. *Managing Software Engineering knowledge*. Chapter 10, Springer–Verlag, Berlin, Germany, pp. 199–230.

Weyuker, E. J. 1990. The cost of data flow testing: An empirical study. *IEEE Transactions on Software Engineering* 16(2): February, 121–128.

Wood, M., Roper, M., Brooks, A., and Miller, J. 1997. Comparing and combining software defect detection techniques: A replicated empirical study. *Proceedings of the 6th European Software Engineering Conference*. Zurich, Switzerland, September.

Dr. Sira Vegas is assistant professor with the Computer Science School at the Universidad Politécnica de Madrid in Spain. She was Summer Student at the European Centre for Nuclear Reseach (Geneva) in 1995. From 1998 to 2000 she has been a regular visiting scholar at the University of Maryland. In summer 2002 she has been research visitor at the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern (Germany). Sira has a B.S. and Ph.D. in computer science from the Technical University of Madrid. She is a member of IEEE Computer Society and ACM.

Dr. Victor R. Basili is a Professor of Computer Science at the University of Maryland. He was founding director of the Fraunhofer Center for Experimental Software Engineering, Maryland, and one of the founders of the Software Engineering Laboratory (SEL) at NASA/GSFC. He received a B.S. from Fordham College, an M.S. from Syracuse University, and a Ph.D. in Computer Science from the University of Texas at Austin. He has been working on measuring, evaluating, and improving the software development process and product for over 30 years. Methods for improving software quality include the Goal Question Metric Approach, the Quality Improvement Paradigm, and the Experience Factory organization.