Supporting the Evaluation and Selection of CASE Tools

Karin Becker Dilnei Venturini
Instituto de Informática - Pontifícia Universidade Católica do Rio Grande do Sul - Porto Alegre, Brazil
email : {kbecker, dventurini}@inf.pucrs.br

Abstract : *The systematic assessment of CASE tools is a difficult task, which often results on the acquisition of a product that does not solve the software development productivity and quality problems detected by the organization. In this paper, the main features of a Decision Support System (DSS) to aid in the evaluation and selection of CASE tools, are presented This DSS is based on the set of assessment criteria and the underlying decision process proposed by the IEEE Standard 1209-1992. The basic idea is to use the complete hierarchy of assessment criteria proposed in this Standard as a reference decision model, which can then be tailored according to the purposes of the decision process, as well as the characteristics of the user. The process is divided into three phases, namely Preparation, Evaluation and Selection, and specific decision models are associated to each phase in order to guide and support their underlying activities. The phases assume an incremental definition/instantiation of these models, so as to achieve flexibility in the decision process, as well as reuse of previous evaluation/selection results.*
Key-words : *Software Engineering, Decision Support Systems, Case Tools, Multicriteria Decision Problems*

1. Introduction

Software Engineering is an engineering discipline concerned with the practical problems of developing large software systems, involving both technical and non-technical aspects. Like all engineering, it is not just about producing products, but involves producing them in a cost-effective way. The real challenge for software engineers is to develop high-quality software with a finite amount of resources and into a predictable schedule [SOM 92].

Historically, men have always sought for tools as a means to increase productivity and quality, and the term Computer-Aided Software Engineering (CASE) has come into use as a generic term for the automated-support in software engineering. CASE tools, specially when integrated into a common framework or environment, offer a number of functions useful for the various activities of software engineering (e.g. analysis, design, implementation, testing). However, though software practitioners recognize the potential benefits brought by the use of CASE tools [GRU 95][ISO 95], in practice organizations tend to adopt them only in a limited form, or abandon a good percentage of the technology soon after it is implemented [KEM 92][McG 93][MOS 92][FOR 92][FIN 94][BAI 93].

Part of this failure can be explained by the difficulty people have on *assessing* available CASE products, and confronting them with the problems or goals in software development detected by the organization [CHI 92][PRA 93][KAV 96][KIT 96]. As vendors tend to claim about improved quality and productivity resulting from the use of the products they sell, buyers themselves generally have only a vague idea of how to improve productivity with CASE products, and consequently fail to select a product that suits their particular needs [McG 93][BRU 96]. Many organizations choose a CASE product without establishing formal evaluation criteria or thoroughly examining the candidate tools. In fact, software engineers often do not know how to evaluate and select among tools, or when they do, the assessment and acquisition process is in charge of software managers, and not software developers, who will be the ultimate users of the CASE product [CHI 92].

The evaluation and selection of CASE products is clearly a complex decision process, which is fundamental to the successful adoption of CASE tools in organizations. If considered under the framework of Simon's decision making process [SIM 77], it comprises all phases, from intelligence to revision. The *intelligence phase* is concerned with the crisis or opportunity that starts CASE adoption process. The *design phase* involves examining alternative courses of actions, such as looking for and evaluating distinct CASE tools that may solve the "software crisis" identified in the intelligence phase, or even any solution alternative to CASE tool adoption. The *choice phase* is concerned with the selection of one CASE tool, which is typically a selection based on various conflicting criteria (e.g. tradeoff between price and functionality). The *revision phase* involves various post-decisional aspects, such as implementing the solution, assessing whether the goals have been achieved, etc.

A number of works have contributed to a better understanding of this process, but there are no widely accepted, systematic approaches for evaluating a tool's utility. Many works [BEL 94] [McG 93] [MOS 92] [MYO 93] [KIT 96] [KIT 97] [BRU 96] focus on *criteria* for CASE tools assessment, but each one proposes a different set of criteria, often based on the authors' own experience. Moreover, some criteria are difficult to measure, or to compare to each other. An interesting effort towards unification of CASE tools evaluation criteria was performed by the IEEE Software Engineering Standards Subcommittee, resulting in the IEEE Standard 1209 "IEEE Recommended Practice for Evaluation and Selections of CASE Tools" [IEE 92]. This standard is not limited to the proposal of (another) set of *assessment criteria*, but discusses as well the underlying *process*, subdivided into evaluation and selection. Moreover, none of these works discusses the use of a *computer-based system* to support the evaluation and selection of CASE products. Decision Support Systems (DSS) are computer-based system that affect (or have the intent to affect) the way people make decisions [SIL 91]. DSS do not have the goal of replacing decision-makers, but rather to increase their judgmental abilities by providing useful information to base their decisions on.

This work presents the striking features of a DSS aimed at aiding in the evaluation and selection of CASE tools. This DSS is based on the set of assessment criteria and decision process proposed by the IEEE Standard 1209. As pointed out in [KEE 78], the starting point of the development of any DSS is the comprehension of the *process* underlying the types of decisions to be supported, and the process suggested by [IEE 92] was used as a reference to detect a number of useful functions to enhance the judgmental capabilities of decision-makers. The selection of CASE tools is approached as a multi-criteria decision problem [KEY 76], and the set of assessment criteria proposed by the standard is offered as a reference model that can be further refined and adapted to each specific situation, using the functions provided by the DSS. The overall decision process assumed by the DSS is divided into three phases, namely Preparation, Evaluation and Selection, and specific decision models are associated to each phase in order to guide and support their underlying activities. The phases induce an incremental definition/instantiation of these models, so as to achieve flexibility in the decision process, as well as reuse of previous evaluation/selection results.

The rest of this paper is organized as follows. Section 2 discusses the works found in the literature addressing the problem of CASE tools assessment. The striking features of the proposed DSS are presented in Section 3, and the phases of the underlying decision process are discussed in the remaining of the paper. The Preparation Phase is presented in Section 4, the Evaluation Phase is detailed in Section 5, and Section 6 discusses the Selection Phase. Conclusions are drawn in Section 7, where future work is also highlighted.

2. Related Work

As already mentioned, there are no widely accepted, systematic approaches for evaluating CASE tools' utility [KAV 96][CHI 92]. There are consulting firms, trade publications and private analysis services that market CASE products evaluation reports and comparisons. [CHI 92] points out that the main shortcomings of these reports are that they are based on subjective information (at best), the evaluation criteria are not sound and often change with each report, it is not clear whether the results are repeatable, and comparisons published by different firms cannot themselves be compared. [CHI 92] also stresses that good articles that disseminate sound work on CASE tools assessment require a lot of effort, and are rare to find.

Proposals on how to perform systematic CASE tool assessment can be found in works such as [BEL 94] [McG 93] [MOS 92] [MYO 93] [KIT 96] [KIT 97] [BRU 96]. Each proposal includes a distinct set of criteria, which reflects the experience and points of view of each respective author. The criteria also vary in the level of objectivity in which they can be measured. Works such as [McG 93][BEL 94][BRU 96] provide general guidelines to be observed when buying a product. For instance, [McG 93] presents a checklist for buyers of CASE products, where each criterion is stated in the form of a question, such as "on what does the vendor base his claim of improvements in productivity and quality?". The set of criteria proposed in [MOS 92][MYO 93][KIT 97], on the other hand, can be measured objectively, and some works even suggest weights for the criteria.

The IEEE Standard 1209 "IEEE Recommended Practice for Evaluation and Selection of CASE Tools" [IEE 92] represents an effort towards unification of CASE tools assessment criteria. This standard is not limited to the proposal of (another) set of evaluation criteria, but discusses as well the underlying decision process. It has also the sponsoring of the IEEE Technical Committee on Software Engineering, which gathers the most renowned researchers and practitioners of the field. The overall process suggested by the Standard 1209 can be performed to satisfy several purposes, including:

- evaluation of several CASE tools and the selection of one or more;
- evaluation of one or more CASE tools with the resulting data maintained for future reference;
- selection of one or more CASE tools using data from previous evaluations.

The process is thus divided into two aspects, evaluation and selection. The purpose of the *evaluation process* is to determine a CASE tool's functionality and quality for use. The results of this assessment should be recorded in an appropriate manner such that they may be useful in a subsequent selection process. The evaluation process involves the definition of the evaluation task, the identification of assessment criteria, the identification of candidate CASE tools, the evaluation of these tools according to the criteria, and the reporting of the results. The Standard 1209 proposes a set of assessment criteria, organized hierarchically, as illustrated in Figure 1. Each criterion is described in a consistent manner, and possible measurement domains are suggested for some of them. During the *selection process*, the purpose of the selection is defined, and selection criteria are identified and weighted. The selection can either follow the evaluation process, or be performed on the basis of previous evaluation results available. The reader can refer to [IEE 92] for a more complete description of the assessment criteria and recommended evaluation/selection processes.

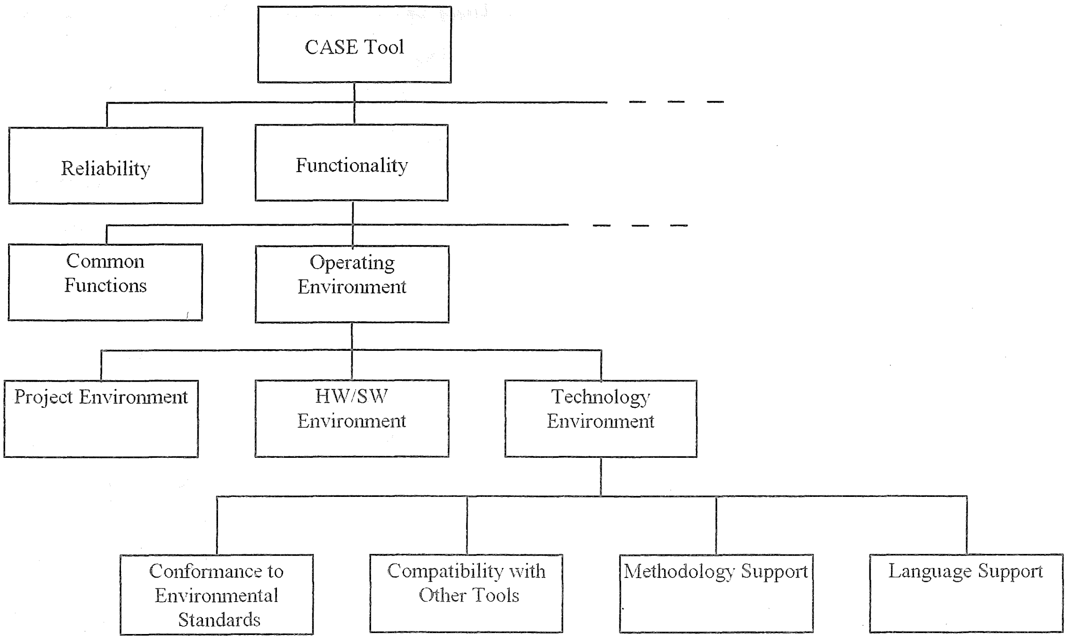

Figure 1 - Partial View of IEEE Standard 1209 Assessment Criteria Hierarchy

As a final remark, most works emphasize the assesment of technical characteristics of CASE tools, but the importance of non-technical aspects (e.g. social, organizational, political, etc) should not be neglected. Mosley points out that "tool evaluation is only the tip of the iceberg! Transitioning the tool into actual use is the rest of the iceberg" [MOS 92]. There are also works that discuss how the adoption of CASE tools is influenced by organizational and political aspects [GRU 95][AAE 93][PRA 93] [KIT 96], developers' maturity [ISO 95][IIV 95], familiarity with the supported software development methodology [KEM 92][JAN 95], among other factors. The IEEE P1348 standard "Recommended Practice for the Adoption of CASE Tools" [IEE 94] provides guidance to enhance the probability of success in the adoption of CASE technology, addressing issues such as definition of CASE needs, development of CASE adoption strategy, etc.

3. A Decision Support System for CASE Evaluation and Selection

The DSS presented in this paper has to goal of supporting the evaluation and/or selection of CASE tools. As already mentioned, DSS do not have the goal of replacing decision-makers. Instead, it offers data, models and a number of useful functions to enhance their judgmental abilities. Key to the successful development of any DSS is the analysis of the underlying *decision process* to be supported [KEE 78]. This analysis has the goal of understanding the data manipulated, the decision models that may be helpful, and the various functions for manipulating them, in order to propose a system that is in accordance with the way people make decisions, but in a more effective way.

Two key ideas guided the design of this DSS. First, it was based on the adaptation of the *decision process* described in the IEEE Standard 1209, which considers CASE tool evaluation and selection as two distinct, but interrelated processes. Second, it regards the evaluation and/or selection of CASE tools as a *multi-criteria decision problem* [KEY 76] [SAG 81], in which decisions are taken considering multiple and conflicting criteria. The basic idea is to use the complete hierarchy of assessment criteria proposed in the Standard 1209 (from now on referred to as the IEEE hierarchy) as a *reference decision model* that can be tailored according to the purposes of the decision process, as well as the characteristics of the user. Other types of decision models are proposed to capture and guide various aspects of this tailoring.

The decision process assumed by this DSS is an adaptation of the one proposed in [IEE 92], which was simplified to enable automated support through a DSS. The adapted process is divided into three phases, referred to as *Preparation*, *Evaluation* and *Selection*. The first two phases correspond to the Standard 1209 evaluation process, and the third one to the selection process. The decision elements comprised in each phase are captured by a set of decision models, which are manipulated according to two aspects: *model definition* and *model instantiation*. The division of the process into phases, together with the corresponding models, aim at providing a flexible process, in which it is possible a) to return to any point in order to try out different scenarios, and b) to reuse previous experiences (models and model instances) resulting from previous CASE tool evaluation/selection processes.

The overall process, with related models and model instances, is depicted in Figure 2. From a model definition point of view, the goal of the Preparation Phase is to define the purposes of the assessment, and to translate those purposes in terms of a set of criteria and filters, which compose the *Tailored Assessment Model* (TAM). The *IEEE Hierarchy* (IEEEH) is the reference model that can be used to help in this definition. From an instantiation point of view, the Preparation Phase involves searching for CASE tools, collecting data about each candidate tool, and filtering interesting candidates for further consideration. The model defined in the Evaluation Phase is the *Evaluation Model* (EM), which completes a given TAM by defining conversion functions, used to transform all data collected for a given candidate tool into a common scale, such that all criteria are comparable. As it will be discussed in detail later, a significant distinction is assumed between gathering data about candidate tools (Preparation Phase) and assigning meaning to this data (Evaluation Phase). Another aspect comprised in EM definition is the specification of weights for all criteria, which are used to calculate the value for composed criteria, i.e. criteria that aggregate other criteria, such as the criterion *Functionality* in Figure 1. At instance level, it is then possible to calculate for every candidate CASE tool the value of all composed criteria, considering the corresponding TAM instances, as well as to associate a meaning to the raw values collected in the previous phase. In the Selection Phase, the best CASE tool(s) is(are) chosen according to the defined selection method (e.g. linear additive model, conjunctive elimination), and two types of selection models are available for taking that decision, namely *Additive Selection Model* (ASM) and *Restrictions Selection Model* (RSM). These models thus refine a given EM by specifying the selection criteria, according to the chosen method. Each phase is further detailed in the remaining of this paper, where the striking characteristics of each decision model type and related operations are described. Due to space limitations, formal definitions of models will not be presented, but the reader can refer to [VEN 97] for a complete description.

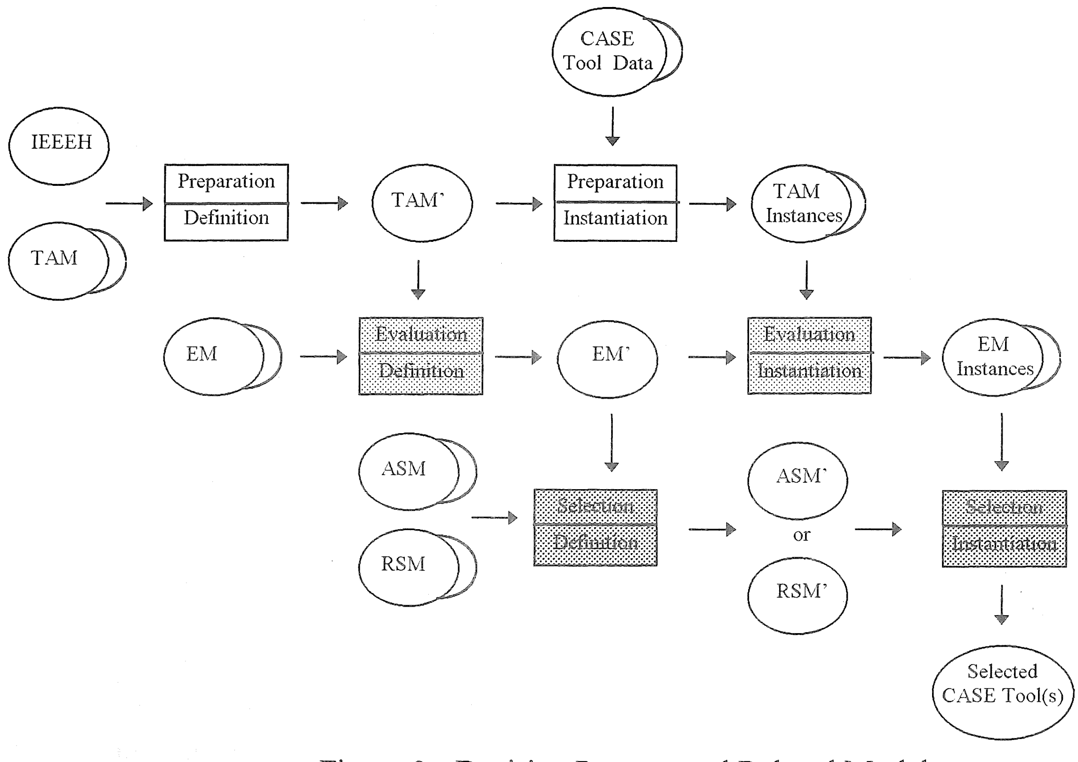

Figure 2 - Decision Process and Related Models

The proposed approach supports the same purposes defined in [IEE 92], namely evaluation of tools for further reference, selection of tools based on available evaluation, and evaluation followed by selection. The results of each phase are captured by the corresponding types of decision models and/or model instances, which are stored in a Model Base, such that they can be reused later in any decision process. Therefore, the phases can be executed sequentially, independently one of another (through the reuse of models and model instances), or any combination of these. The definition and instantiation aspects of each phase can also be executed in combination or independently. This flexibility is depicted in Figure 2, where the different possibilities of combining the activities and models of the suggested generic process are sketched.

The process may also assume a number of different roles, such as evaluator, user, software developer, vendor, etc [VEN 97]. For the purposes of this paper, we will consider only two generic roles: *evaluator* and *user*. The evaluator is the person (or group of people) responsible for the assessment process (evaluation and/or selection), and who is to be supported by the facilities of the DSS. The user is an abstraction representing the organization willing to adopt CASE technology. Eventually, these two roles can be played by the same person (or group of people).

4. Preparation Phase

The Preparation Phase corresponds to the design phase of Simon's decision model [SIM 77], in which different courses of actions are sought for the problem or opportunity identified in the intelligence phase. In the Preparation Phase, this corresponds, on the one hand, to the translation of the purposes of CASE assessment in terms of a set of criteria, and on the other, to the search for candidate CASE tools and the collection of relevant data for the later assessment of these tools.

The purposes of the assessment are defined in a TAM (Tailored Assessment Model). The decision-maker can either *reuse* a TAM available in the Model Base, or *define* his/her own TAM. The definition of a new TAM implies:

a) selection and/or creation of *assessment criteria*;

b) their arrangement in a *hierarchical structure*;

c) assignment of a *measurement domain* for each leaf criterion of this hierarchy;

d) possibly, the definition of *filters* representing mandatory properties that candidate tools must posses.

The IEEE Hierarchy model (IEEEH) can be used as the starting point for that definition. This reference model offers an invaluable framework for selecting the criteria that match the purposes of the assessment, and the characteristics of the user. It fulfills thus an important role in the *learning* that occurs in any decision process as it evolves [COU 96]. For the inexperienced evaluator, it provides a qualified knowledge source to start the learning about CASE tool assessment. For experienced evaluators, it validates or enriches their own knowledge and experience. The decision-maker can also reuse the experience of previous assessments, by adapting an existing TAM. Operations are offered to create new criteria, select existing ones, and to structure criteria hierarchically by including, removing or restructuring the organization of criteria in the hierarchy. Criteria at the bottom of the hierarchy are referred to as elementary or leaf criteria, and the others are referred to as composite or non-leaf criteria. Composite criteria are a means to create higher level abstractions from lower level ones.

An important aspect of assessment is how to *measure* each criterion. Some people feel comfortable by assigning values from a numerical (0..10) or discrete scale (e.g. "very good", "good", ...). Some criteria can be measured objectively (e.g. required memory in Mb), whereas other subjectively (e.g. data integrity - "good"). Moreover, the same criterion can be measured both objectively (e.g. "30Mb of memory)) and subjectively ("fair memory utilization"). We felt that people must *collect data* about candidate tools in the way they are used to or prefer. Also, it is important to give evaluators the possibility of collecting data as objectively as possible, without having necessarily to give an interpretation to those values at this stage of the process (e.g. "support COBOL code generation" vs. "code generation - good"). To accomplish these purposes, *measurement domains* are assigned to the leaf criteria of the hierarchy, which are the criteria actually valued for each candidate tool. Measurement domains are suggested for some of the IEEEH criteria, and as the Model Base is enriched with new TAMs, evaluators can suggest domains for the criteria they created themselves.

Tool assessment can be a very time-consuming activity [MOS 92], and sometimes the best tools are not adequate for solving the detected "software crisis". For instance, consider the situation of a corporation that just did a huge investment on the acquisition of a new hardware platform. For that corporation, the compatibility of the candidate CASE tools with that specific platform may be a fundamental requirement. In order to spare evaluators from the work of collecting data and assessing tools that are clearly inadequate for the identified requirements, filters can be associated to a TAM, i.e. restrictions on the minimum properties that a tool must posses to be further considered as candidate for assessment.

From a model instantiation point of view, the goal is to search for candidate CASE tools, and to gather relevant data about them, using a TAM to guide this process. The result of this task is a set of TAM instances, one for each candidate tool. Various sources can be used for identifying existing tools, such as market surveys, CASE vendors, CASE tool digests and other similar publications. Each TAM instance is related to one CASE tool, of which data is collected and stored for all leaf criteria of the hierarchy, in the domain specified in the TAM. There is no restriction on the number of TAM instances that can be created for a same CASE tool, which is particularly useful for integrating and comparing various points of view. To collect data about a tool, one can examine software and vendor-supplied documentation, interviewing users of the tool, executing demos, etc [IEE 92][IEE 94][MOS 92]. TAM instances can be created any time a new candidate tool is found, and existing TAM instances, created in other assessment processes can be reused, as long as the related TAM is compatible[1].

The filters defined in the TAM can be applied either prior to tool data collection, in order to save time, or later on. Two important aspects about filters must be highlighted. First, the goal of filters is to reduce the number of candidate tools, so as to help focusing on the potential useful ones. No qualitative evaluation of how well a certain tool responds to the criteria specified in a filter is performed at this stage of the process. Second, filter application must be a flexible process. Indeed, as the decision process proceeds, one often has a better insight about the problem being solved. In this sense, evaluators may wish to try out different scenarios with the

[1] For the time being, any two models (e.g. TAM, EM, etc) are compatible only if they are equal. An extended notion of model compatibility is presently under investigation.

tools that have been initially discarded because they did not attend minimum conditions that later on do not seem that important as they were when the process have started. Suppose for example that the filter states maximum price, and that tools costing more than that price were initially discarded. Later on, the evaluator realizes that required functionality is not available for the price the user is willing to pay. So, the evaluator reconsiders some of the tools that were not initially filtered (e.g. to convince the user to spend more).

All these results are stored in the Model Base, such that they can be reused in other processes, or to allow evaluators to return to previous stages in the same process. Figure 3 presents a very simple TAM, with the criteria arranged in a multi-level hierarchy, the domains assigned to the leaf criteria, and the specified filter. Notice that filters may be stated in terms of criteria that are not included in the TAM.

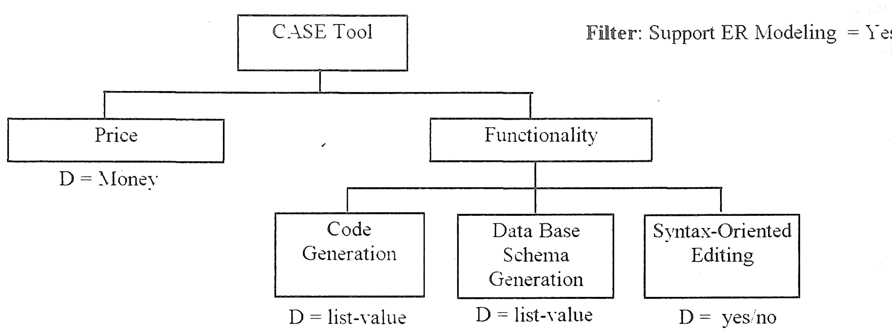

Figure 3 - A TAM Model

5. Evaluation Phase

The Evaluation Phase also corresponds to Simon's design phase. It has the goal of associating meaning to the values collected in the Preparation Phase. As already mentioned, domains are defined for each leaf criterion in the Preparation Phase such that: a) data about each candidate CASE tool can be collected in the domains defined by the evaluators themselves as the most adequate for that purpose, and b) it is possible dissociate *data gathering* from the task of *associating meaning* to those values. For example, one thing is to observe that a given tool generates code for C and COBOL. The other, is to evaluate how useful these languages are, considering the characteristics of the user. For one user, this is excellent, because these languages are the ones required, whereas for others, SQL language, which is not supported, is essential. The flexibility provided by this approach also allows evaluators to try out *different evaluation scenarios*, by associating different meaning to raw values, as well as to *reuse data* collected in previous assessments more easily. Another essential issue in evaluation is that criteria must be comparable, otherwise, evaluation is relative. For example, how good is a tool that requires 15Mb of RAM and supports code generation for SQL and COBOL languages?

The evaluation context is defined in the *Evaluation Model* (EM), which complements a given TAM by:

a) providing *conversion functions* for all leaf criteria of the hierarchy;

b) providing *weights* for to each criteria, which are used to calculate the values for all composite criteria.

The first task has the twofold objective of associating meaning to raw values, according to the characteristics of the evaluation at hand, and well as to bring all values to a common evaluation scale, so that criteria are comparable. The common scale adopted is the numerical interval 0..10, which is the domain assumed for all criteria in an EM. A number of conversion functions are suggested for each domain, and available in the Model Base. When defining an EM, the evaluator can either select among the conversion functions available in the Model Base, or create his/her own. These new functions become then available in the Model Base for future reference. Figure 4 shows an example of an EM derived from the TAM presented in Figure 3. The specification of the conversion function assigned to the criterion *price* (f_price) is shown in the picture.

Weights are required to associate meaning to *composite criteria*, since their values are actually derived from the ones of the criteria they aggregate. Let us consider, for example, the criterion *functionality*, presented in Figure 3. One may evaluate that *code generation* and *schema generation* are essential features, whereas *syntax-oriented editing* is just a nice feature. The value of *functionality* must then reflect this requirement, which is accomplished by assigning different weights for these criteria, as depicted in Figure 4. Notice that the role of weights in our approach differs from the one suggested in the original IEEE process [IEE 92], since here they are related to the evaluation of CASE tools, and not to only the selection process. Indeed, it is our opinion that the evaluation of a tool must be performed according to a certain context, i.e. requirements and constraints of a given user, and therefore, one cannot evaluate a tool unless the importance assigned to the criteria are defined at this stage of the process.

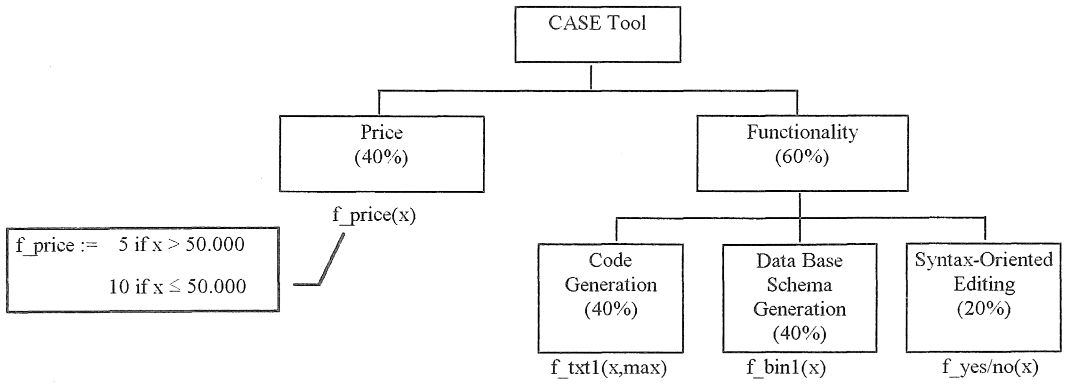

Figure 4 - An EM Model

The definition of distinct conversion functions and weights for a same TAM allows more flexibility in the evaluation process. One can try out different *evaluation scenarios* by defining different combinations of these two types of parameter in distinct EMs. This is particularly useful for analyzing the same tools according to distinct points of views, providing in this way a better insight about the tools and the assessment process, as well as the possibility of integrating and comparing points of views related to different evaluators.

From an instance point of view, an EM instance is related to a TAM instance, from which it a) extracts data for the leaf criteria, b) convert then to a common scale, and c) calculate recursively the values for the non-leaf criteria according to the defined weights. All these results are stored in a Model Base, such that they can be reused later in other processes, or to allow evaluators to return to previous stages in the same process. Notice that at any point of the Evaluation Phase, one can go back to the Preparation Phase, and create/reconsider other TAM instance(s) for evaluation (i.e. consider new candidates).

6. Selection Phase

Given a set of CASE tool evaluation results, evaluators can in the Selection Phase choose the one that most suits the requirements and constraints of the user at hand. For that purpose, various selection methods are available in the DSS, such that, again, the decision can be taken according to the various scenarios analyzed. For the time being, three selection methods were considered for this DSS, namely *linear additive*, *conjunctive elimination* and *disjunctive elimination* [KEE 76][SAG 81].

The linear additive method selects the tool X_j with the highest score, considering that all tools are analyzed according to the same criteria. The score for each tool is calculated according to Formula 1, where c_i is the value assigned to a criterion, and w_i is the weight assigned to that criterion. The linear additive method is classified as a *holistic method* [SAG 81], in the sense that the score for each tool is calculated independently of other tools. It is also regarded as a *compensatory method*, since a high value for a criterion can compensate a low value for another one(s).

$$X_j = \sum_{i=1}^{n} c_i \,.\, w_i \qquad (1)$$

The conjunctive and disjunctive elimination methods are examples of *heuristic methods* [SAG 81], which select alternatives based on the comparison of one alternative with another, or against a standard. In the conjunctive and disjunctive elimination methods, each alternative is compared against some *standard*, which states minimally acceptable value standards for each relevant criterion. In the conjunctive elimination, alternatives that pass the critical standard on *one or more criteria* are retained. It is thus a compensatory method. The disjunctive elimination method is non-compensatory, because an alternative is selected only if it passes *all critical standards*.

To use the first method, the evaluator has to select or define an *Additive Selection Model* (ASM). An ASM is related to an EM, and it states the criterion of the EM hierarchy upon which the selection will be made. Normally, it is the criterion representing the *root* of the hierarchy that will be used, i.e. the tool with the highest overall score, but a given sub-hierarchy can also be defined (i.e. the tool presenting the best *functionality*). At instance level, the evaluator has to select a set of EM instances, and the result will be the tool (or set of tools, in case of a tie) that performs best in the selection criterion (i.e. have the highest score). It is presently under study the extension of the ASM to comprise the selection based on more than one criteria (i.e. one or more disjoint subtrees of the EM hierarchy).

To use the conjunctive or disjunctive elimination methods, the evaluator has to define a *Restrictions Selection Model* (RSM), also related to an EM, and in which it is defined the *criteria* for which critical values will be given, the *critical value(s)* for each criterion, together with the corresponding *operator* (e.g. "functionality, 7, <="). It is also defined the method according with the standard will be compared (i.e. conjunctive or disjunctive elimination).

The flexibility of the Selection Phase comes from the fact of being able to try out different forms of selections based on a same evaluation, by varying any of the following aspects: a) the selection method, b) in the case of linear additive selection, the selection criterion (ASM model) and/or set of EM instances, or c) in the case of conjunctive or disjunctive elimination, by trying out different standards (criteria and/or critical values). One can also try out to select by varying the related EM model. All these results are stored in a Model Base, such that they can be reused in other processes, or to allow the evaluators to return to previous stages in the same process.

7. Conclusions

In this paper we presented the main ideas of a DSS for supporting the evaluation and/or selection of CASE tools. As any DSS, it does not have the goal of replacing the decision-maker, but rather to enhance his/her judgmental abilities by providing useful information, and the possibility of examining a wider number of scenarios before reaching a decision.

The DSS assumes a decision process that is an adaptation of the one proposed in [IEE 92]. Both the decision process and assessment criteria of this standard were considered, because they can be regarded as the highest level of agreement on the subject, not to mention the sponsoring a renowned society as IEEE. The DSS assumes a process divided into 3 phases, each one of them related to a type of decision model that guides the activities underlying the phase. Each phase is further divided into two aspects, namely model definition, that allows the creation of an evaluation/selection framework, and model instantiation, in which actual tools are contrasted to the assessment framework. The flexibility provided by this approach was stressed throughout the paper. Basically, it allows evaluators to start the process from any point, to return to any previous point, if desired, and to extensively reuse the results of previous evaluations/selections.

A prototype DSS is under development, using MS ACCESS for Windows platform, which allows the definition and instantiation of all decision models. The empirical validation of the DSS, with the suggested phases and models, is the next step of this work. We are presently analyzing a case study, based on a reported experience on CASE tools evaluation and selection performed by a Data Processing Center, in cooperation with the evaluator responsible for that acquisition.

Future research topics include, among others, the development of functions supporting easy performance of sensitivity analysis (e.g. graphs, comparisons between models and model instances); the study and development of different forms of compatibility between the various types of models, so as to increase the reuse of previous results; extension of the definition of ASM to comprise selection based on one or more criteria: provision of additional selection methods; extension of the support to group-related activities in decision-making (Group DSS); etc.

References

[AAE93] AAEN, I. CASE user satisfaction - impact evaluation in user organizations. Proceedings of the Sixth International Workshop on Computer-Aided Software Engineering , Singapure, 1993. p. 66-74.

[BAI93] BAILER, B.; BAUCKNECHT, K.; SCHATZMANN, C. *Ways to CASE: an analysis of selected CASE experiences.* Proceedings of the Sixth International Workshop on Computer-Aided Software Engineering, Singapure, 1993. p. 36-43.

[BEL94] BELL, R. *Choosing tools for analysis and design.* IEEE Software, 11(3): 121-25, May 1994.

[BRU96] BRUCKHAUS, T.; MADHAVJI, N. H.; JANSSEN, I.; HENSHAW, J. *The impact of tools on software productivity.* IEEE Software, 13(5): 29-37, Sept. 1996.

[CHI92] CHIKOFSKY, E. J.: MARTIN, D. E.: CHANG, H. *Assessing the sate of tools assesment.* IEEE Software, 9(3): 18-21, May 1992.

[COU96] COURBON, J.C. *User-centered decision support systems: design and implementation.* Proceedings of the IFIP WG8.3 Conference on Implementation Aspects for Supporting Management Decisions, London, 1996.

[FIN94] FINLAY, P. M.; MITCHELL, A. C. *Benefits from the introduction of information engineering.* Proceedings of the First European Conference on IT Investment Evaluation, Henley-on-Thames, 1994.

[FOR92] FORTE, G. *Tools fair: out of the lab, onto the shelf.* IEEE Software, 9(3): 70-9, May 1992.

[GRU95] GRUPE, F. H. *An evaluation of the analyst/programmer productivity changes attributable to the use of CASE tools.* Journal of Computer Information Systems, 35(2): 85-90, Winter 1994-1995.

[IEE92] *IEEE Recommended Practice for the Evaluation and Selection of CASE Tools.* IEEE Computer Society, Dez 1992.

[IEE94] *IEEE P1348 Recommended Practice for the Adoption of CASE Tools.* Draft 6.0, IEEE Computer Society, Sept. 1994.

[IIV95] IIVARI, J. *Factors affecting perceptions of CASE effectiveness.* European Journal of Information Systems, 4(3): 143-58, Aug. 1995.

[ISO95] ISODA, S.; YAMAMOTO, S.; KUROKI, H. ; OKA, A. *Evaluation and introduction of the structured methodology and a CASE tool.* Journal of Systems and Softwares, 28(1): 49-58, Jan. 1995.

[JAN95] JANKOWSKI, D. J. *CASE tool selection using methodology to support to chooe the rigth tools for the right job.* Journal of Systems Management, 46(4): 20-6, Jul.-Aug. 1995.

[KAV96] KAVI, K.: NAHOURAII, E. *Software assesment tools.* IEEE Software, 13(5): 23-5, Sep. 1996.

[KEE78] KEEN, P. G. W.; MORTON, M. S. S. *Decision support systems : an organization perspective.* Addison-Wwsley Publishing Co., 1978.

[KEY76] KEENEY, R. L.; RAIFFA, H. *Decision with multiple objectives: preferences and values tradeoffs.* John Wyley & Sons, 1976.

[KEM92] KEMERER, C. F. *How the learning curve affects CASE tools adoption.* IEEE Software, 9(3): 23-8, May 1992.

[KIT96] KITCHENHAM, B. A. *Evaluating software engineering methods and tools - Part 1: the evaluation context and evaluation methods.* ACM Software Engineering Notes, 21(1):11-5, Jan. 1996.

[KIT97] KITCHENHAM, B. A. *Evaluating software engineering methods and tools - Part 6: identifying and scoring features.* ACM Software Engineering Notes, 22(2):16-8, March 1997.

[MCG93] MCGRATH, FRANK *Checklist for buyers of ICASE products.* IEEE Software, 10(9): 108-10, Nov. 1993.

[MIY93] MIYOSHI, T. ; AZUMA, M. *An empirical sutdy of evaluating software development environment quality.* IEEE Transactions on Software Engineering, 19(5): 425-35, May 1993.

[MOS92] MOSLEY, V. *How to assess tools efficiently and quantitatively.* IEEE Software, 9(3): 29-32, May 1992.

[PRA93] PRATHER, B. *Critical failure points of CASE tools evaluation and selection.* Proceedings of the Sixth International Workshop on Computer-Aided Software Engineering, Singapore, 1993.

[SAG81] SAGE, A. P. *Beavioral and organization considerations in the design of information systems for planning and decision support.* IEEE Transactions on Systems, Man and Cybernetics, 9: 640-80, Sep. 1981.

[SIL91] SILVER, Mark S. *Systems that support decision makers: description and analysis.* John Wiley & Sons Ltd.,1991.

[SIM77] SIMON, H. A. *The new science of management decision.* Harper & Row, 1977.

[SOM92] SOMMERVILLE, I. *Software engineering.* 4th ed. John Wiley & Sons, 1992.

[VEN97] VENTURINI, D. *A system to support the evaluation and selection of CASE.* M.Sc. Thesis. Porto Alegre, Instituto de Informática - PUCRS, 1997. (In Portuguese: *Um sistema de apoio à avaliação e seleção de ferramentas de desenvolvimento de software CASE*)