Reprint

CASE Tools Evaluation: An Automatic Process Based on Fuzzy Sets Theory

T. Antonakopoulos, K. Agavanakis and V. Makios

The 6th IEEE International Workshop on Rapid System Prototyping– RSP'95

CHAPEL HILL, NC, JUNE 1995

CASE Tools Evaluation : An Automatic Process Based on Fuzzy Sets Theory

T. Antonakopoulos, K. Agavanakis and V. Makios
Laboratory of Electromagnetics, Telecommunications and Technology of Information Division,
Electrical Engineering Department, University of Patras, 26500 Patras, Greece
Tel : +30(61)997286, Fax : +30(61)997342, e-mail : antonako@ee.upatras.gr

Abstract. *This paper describes the architecture of a software tool for CASE products evaluation. The tool is based on the ISO software product quality evaluation process model, which has been extended using fuzzy sets theory for achieving more reliable results. The evaluation tool can determine which CASE product is the most appropriate for a specific application and can generate the initial models of the system for further processing by the selected CASE tool. This is achieved by using an inference process based on fuzzy rules for mapping user's specifications into software metrics. Fuzzy sets are employed to express concepts such as the strength of each criterion involved in the evaluation, its relative importance, etc. The evaluation tool architecture is modular while the fuzzy rules, the definition of the used fuzzy sets and the tool database can be easily modified by the user for accurately describing his requirements.*

I. Introduction

The evolution of software technology has led to the development of systems that can automate complex activities and can cope with existing real-life problems more efficiently, with increased stability and security than their ancestors. However, as software systems capabilities increase, so does their complexity. Most phases of the software life-cycle are complex and time-consuming activities that cannot be easily handled without using computerised design tools. This situation and the need of increased software productivity and quality, leads to the use of CASE (Computer Aided Software Engineering) tools which aim to support phases of the software life-cycle or even to automate some of the involved activities.

A plethora of CASE tools are currently available in order to satisfy the diversity of users needs, since different software life-cycles demand different development processes and tools in order to be sufficiently supported, making more difficult to assess the real capabilities and features of the commercial products, and to understand how they are related to each other functionally and in terms of technology [1]. As a result, the evaluation of computerised tools must be guided by the requirements of the particular software process that the user plans to employ and tailored to the specific requirements of the target system. Besides that, the use of an inappropriate product may be responsible for negative results during the system development. Therefore it becomes apparent that a systematic formal method for evaluating the available CASE products and ensuring a safe selection, is needed.

The effort to qualify the "better" product in relation to the project needs, brings to the surface the problem of determining software quality and selection criteria. ISO/IEC 9126 specifications define software quality characteristics that can be modified and expanded in terms of quality characteristics, sub-characteristics and metrics, in order to be applicable to a specific quality evaluation process. This provides a procedure to objectively define software quality and to determine the criteria for tools usefulness. Although it sounds promising, things get confusing when such a procedure is applied to real problems. The experience shows that it is very easy to define 20 quality sub-characteristics concerning the CASE products with an average of 2-3 metrics for each one of them [2]. This means that 50 relative weights for the metrics and 20 more for the sub-characteristics must be defined. So, it becomes apparent the necessity of a tool that could automate the procedure of criteria selection, relative importance assignment and quality evaluation, particularly when the user needs to refine his evaluation using different specifications. As it is shown in the next section, the ISO proposed evaluation process has to be modified by using a more flexible method, since the way that metrics are recorded into rating levels and that relative importance is declared in the form of relative weights, result to inaccurate decisions at boundary points.

The effort of this work is to describe the architecture of a software tool that automates the formal process of evaluating the quality of CASE products, based on a combination of the ISO/IEC 9126 software quality evaluation model and fuzzy sets theory. This software evaluation tool has been specified to provide a convenient and unified environment for CASE tools selection, by using any information available by the tools characteristics. More specifically it will :

- Speed-up the selection of the appropriate CASE tool for

1074-6005/95 $04.00 © 1995 IEEE

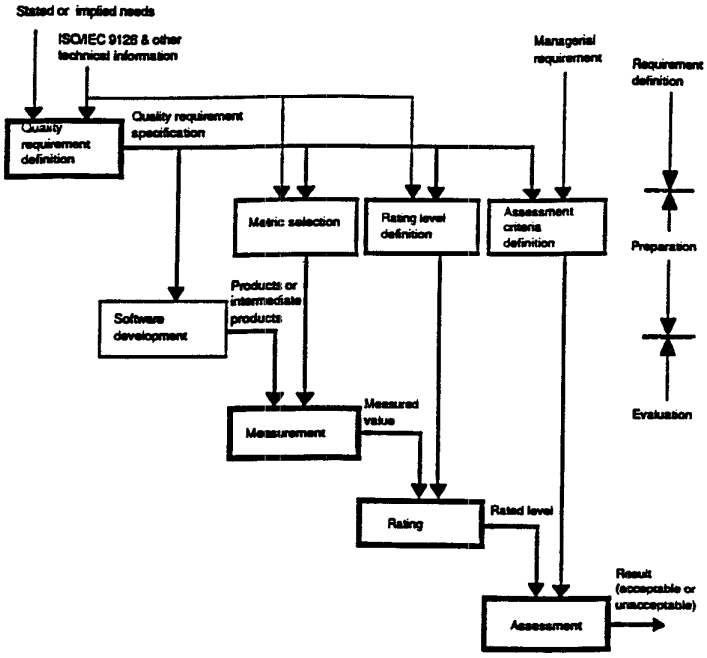

Fig.1 : ISO/IEC Evaluation Process Model

each target system and shorten the beginning of projects development.

- Accelerate the evolution of the system development and contribute to its quality, since the most appropriate CASE product will be chosen, providing the best support for the development life-cycle of the target system.
- Based on the initial system description used in the CASE tools quality evaluation process, it will generate initial models of the system that will be compatible with the selected CASE product.

Section II presents the ISO/IEC 9126 specifications for software quality evaluation and discusses problems that may arise from their application. Section III describes a general overview of the proposed automatic quality evaluation process, and the application of fuzzy sets theory in its decision process. The following three sections provide an analytical presentation of each stage of the proposed model and highlight their advantages. Finally, conclusions from the overall experience of defining the automatic evaluation process based on fuzzy sets theory are presented in Section VII.

II. ISO/IEC 9126 Software Evaluation Process Model

The quality evaluation of a CASE product is quite a complex process, due to the large number of parameters that impact the used criteria. The main process of quality evaluation takes among others the following steps :

- Quality requirement definition.
- Metrics selection and associated measurement scale definition.
- Measurement and rating.
- Assessment.

Although many characteristics have been proposed and used for software quality evaluation, there is not a widely accepted single set of characteristics, because the set of characteristics that is used in each case, depends on the evaluator opinion [3]. The standardised framework of ISO/IEC 9126 quality characteristics for software product evaluation prepared by JTC1/SC7 group, defines six main quality evaluation characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability [4]. When software quality of composite software products has to be evaluated, each quality characteristic has to be refined to sub-characteristics that better reflect the product under examination. According to the ISO specifications, the evaluation process model is the one shown in Fig.1. It consists of three main stages: quality requirement definition, evaluation preparation and evaluation procedure. Quality requirement definition, includes the selection of fundamental quality characteristics and sub-characteristics suitable to the target software. The user also specifies the quality requirement level and the relative importance of each quality sub-characteristic. Then, during the evaluation preparation, specific quality metrics are selected for each quality sub-characteristic and their relative importance is defined. The selection of metrics also includes the definition of the scale of the measurement method and of the rating levels (Fig.2). The last step is to perform the appropriate tests and to record the measured values in terms of the related rating levels; then the final assessment takes place.

As it can be seen from the above description, the ISO model is hierarchically structured as shown in Fig.3. For the remainder of this paper, each level in the depicted

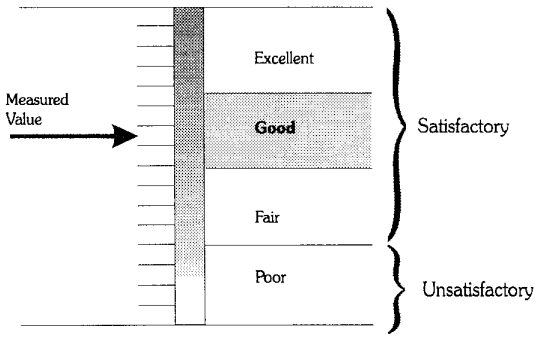

Fig. 2 : Measured value and rated level

hierarchy, will be referred to as a *criterion level*. It is clear that in order to evaluate the score of the elements of any specific criterion level above the metrics, the elements of the immediately lower criterion level must be grouped and their relative importance must be taken into account. If the analysis includes non-quantitative metrics, as it mostly happens in a CASE products survey, then the relevant rating levels have to be defined either through a verbal description of when each level is valid, or through a "rule of thumb". Practice and experience have shown that the above model, presents certain difficulties in its application.

First of all, the fact that each measured value is reflected to a particular rating level, results to low precision; the accuracy of the results depends on the number of the rating levels, defined for each metric. Indeed, if a metric has a quantitative scale, then a small change on the measured value around the limits of two neighbouring rating levels, may cause an abrupt change of the output during the assessment of the current criterion level. This can be partially solved if a large number of levels is used. The trade-off is that as the number of levels increases, the distance between them is decreasing; however the reason for using levels is to group similar values and a large distance between levels is needed. In the case of a non-quantitative metric, which is the most interesting situation for this work, it is very difficult or impossible to verbally define a large number of levels in terms of product's properties, whereas a small number makes the choice very difficult and subjective. The restricted number of levels prevents assessment results of one criterion level to propagate to higher levels.

Second, when the results of one criterion level are summarised in order to get the achieved score for a higher level, e.g. when the metrics of a quality sub-characteristic are summarised, then the usual procedure is to use relative weights. The achieved score of each product is a function of the measured level and the required level, according to the following formulas:

$$\text{Measured level} = \sum_{k=1}^{n} P_k M_{ijk} \qquad (1)$$

where M_{ijk} is the relative weight of the k-metric of the SQ_{ij} quality sub-characteristic of the Q_i quality characteristic and P_k is the achieved rated level of a specific CASE product related to M_{ijk}.

$$\text{Required level} = \sum_{k=1}^{n} R_k M_{ijk}\ , \qquad (2)$$

where R_k is the required level for the relevant metric, and

$$\text{Achieved score} = ((\text{Measured level})/(\text{Required level}))*100\% \qquad (3)$$

This approach has the drawback that if all the metrics of the relevant quality sub-characteristic do not have the same scale then the percentage that each metric influences the quality sub-characteristic apart from the relative weight, depends on its scale as well.

Third, the need of relative weights for the process of assessment, oblige the user to provide specific crisp values, which do not always correctly reflect his preferences. The user can more easily be expressed in a fuzzy way, as he does in real life situations, when he says e.g. "I strongly prefer good quality on this metric more than the other". The mapping of such a preference to the crisp values of relative weights, does not perfectly reflects reality.

Finally, even if a list of possible metrics has been predefined, the potential user is trapped in a maze of quality characteristics, sub-characteristics and metrics that may refer to capabilities of the CASE product that are not related with the kind of system that he is interested in. Therefore, a computerised tool is needed to help the user to select only the most appropriate criteria from the available ones.

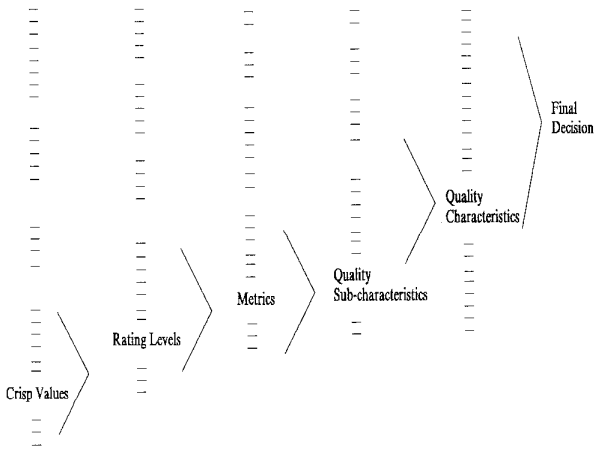

Fig.3 : Evaluation Criteria Hierarchy

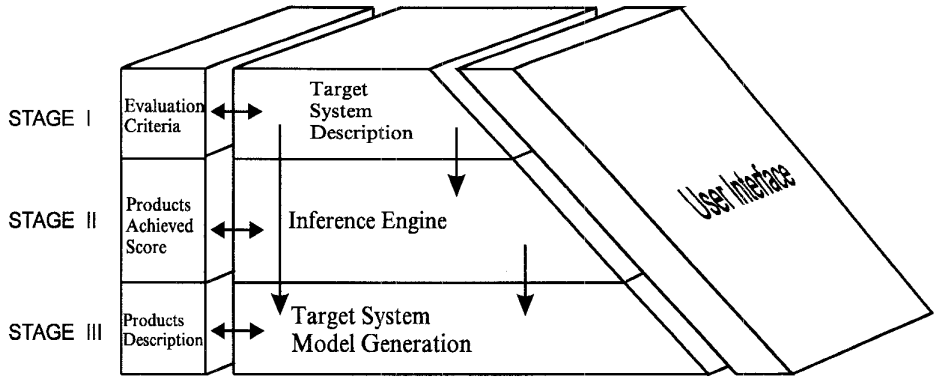

Fig. 4 : General Block Diagram for the CASE product Fuzzy Evaluation System

III. CASE Products Quality Evaluation Model

Based on the above statements, we have been guided to the design of a CASE products quality evaluation process model, that is influenced from ISO/IEC 9126 specifications, but is based on fuzzy sets theory. Although this work was originally concentrated only on quality evaluation, soon it became clear that the proper quality evaluation procedure can shorten the total development time if it provides not only the indication of the best tool for a specific problem but also to generate the initial models for the software development process. That is to integrate the tools selection process in the software life-cycle instead of starting the development after selecting the appropriate tools. Attention has been given to automate the process by a software system, which is currently under development. The selection of fuzzy sets theory has been proven quite natural and promising for these types of problems. Fuzzy models use the high level of abstraction known as approximate reasoning to encode and manage knowledge. A fuzzy set encodes the degree to which objects and events have features associated with the set. The calculus of fuzzy rules provides a systematic and mathematically rigorous way of handling systems that deal with imprecise, ambiguous and vague input-output relationships. When fuzzy sets theory is applied to address decision problems, such as the selection of a CASE product, which is a process subjective to the relative importance of the used criteria, the meaning of a lexically imprecise proposition is represented as an elastic constraint on a fuzzy variable; and the answer of a query is deduced through a propagation of elastic constraints [5].

Therefore it becomes clear that if products measurement, users preferences and decision logic are expressed and governed by fuzzy sets theory concepts, then an appropriate fuzzy inference system will ensure an accurate result from these fuzzy inputs. The general architecture of the proposed model, is depicted in Fig.4, where the evaluation process is divided into three main stages. CASE products are studied and their most important characteristics and operational features, such as supported models and methodologies, operating system and operating platform, are stored for future reference. General quality characteristics, sub-characteristics and metrics are defined, providing a pool of available quality criteria. Measurements for each CASE product related to all predefined criteria are stored in a separate database, in the form of fuzzy variables values. This is considered to be a distinct activity from evaluation, as the same measurements may serve for multiple searches with different user preferences or target systems.

In the first stage, the user supplies a description of the target system using the provided toolset that is most appropriate for the specific problem. The system reacts by providing a list of possible quality criteria, since each available toolset is related with certain predefined criteria. The Inference Engine that dominates in the second stage, is essentially an adapting fuzzy system, which has as inputs measurements, users preferences for specific criteria and their related fuzzy sets membership functions. It also contains the fuzzy rules and calculates at the output the achieved score of the elements of a higher criterion level than that of the input. After the selection of a specific CASE product, the initial target system description is processed according to the specifications imported from the CASE products description. This results to the generation of a model of the target system that is compatible with the selected product and can be used to begin the system development.

Stages one and three are interactive, meaning that the user is actively involved in the data process and can control the results. Stage two takes the user preferences as input but

processes the available information independently. Of course, all three stages can be configured through an adapting mechanism, in order to be customised for the specific type of problem.

IV. Target System Description

Although the organisation of quality criteria in characteristics, sub-characteristics and metrics provides the means for the formalisation of the evaluation process, it is usually very difficult for the user to handle the large number of the involved criteria. A careful survey for CASE products evaluation could possibly end with a vast majority of appropriate quality criteria, but it is evident that not all are applicable to every target system. Depending on the nature of the addressed problem, the user may not be interested to some of them, whereas he has reasons to set priority to some others.

In the proposed system, a design tool is provided to the user, which can briefly, but adequately, express all the important aspects of the target system that should be considered when judging on the importance of the criteria. Then it would be much easier for the user to select through an interactive environment only those criteria that are mostly important for his task. Indeed such a system would :

- facilitate the formalisation of the functional specifications of the target system and the distinction of the appropriate criteria.
- semi-automate the selection of the used criteria. The system will contain a database where special system descriptive components or structures will be associated with relevant criteria, providing a first level selection that can be refined by the user, until the final structure of criteria, has been determined.

The above functions are realised in the first stage of the quality evaluation model (Fig.5). A toolset related to the nature of the addressed problem type is included, in order to customise the procedure of inferring the important evaluation criteria from the users description of the target description.

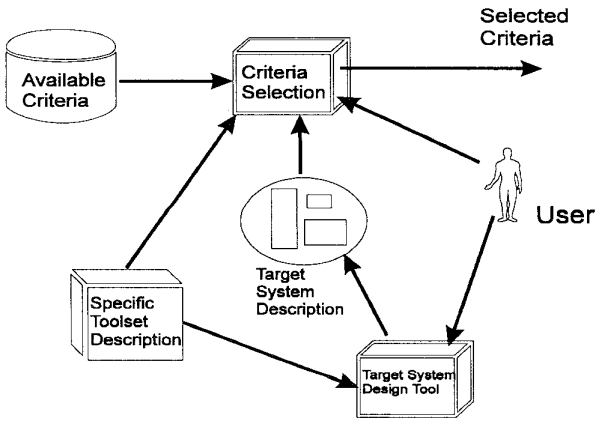

Fig. 5 : Target System Description

V. Inference Engine

The output generated from the first stage of the evaluation process model is the list of the criteria that will be actually used for the evaluation, and is used as an input in the second stage named Inference Engine (Fig.6).

The second input of this stage is the user preferences concerning the selected criteria of a particular level, necessary for the assessment towards the higher level. This consists of a well defined state vector of linguistic variables expressing the relative importance that the user wishes to assign to the elements of the low criterion level. His preference is expressed through a linguistic variable *UserInterest* (UI) assigned to each criterion and referring to his relative interest on the corresponding entity. The related term set is

T (UI) = { VERY WEAK, WEAK, MEDIUM, STRONG, VERY STRONG }

and the fuzzy sets membership functions are

M(UI) = { $M_{VERY\ WEAK}$, M_{WEAK}, M_{MEDIUM}, M_{STRONG} and $M_{VERY\ STRONG}$ }

Membership functions can be configured during the actual operation of the system to better reflect the user's opinion.

The third input of this stage is the set of measurements concerning the achieved score for each CASE product and for all the selected criteria, in either crisp or fuzzy form. These measurements are placed in the system database, in the form of values of another linguistic variable, referring on the degree that the product satisfies the corresponding criterion. This variable is named *CriterionScore* (CS) and has the term set :

T(CS) = {VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH}

and their associated membership functions

M(CS) = {$M_{VERY\ LOW}$, M_{LOW}, M_{MEDIUM}, M_{HIGH} and $M_{VERY\ HIGH}$ }

The output of this level, is a state vector describing the achievements of each higher level criterion using the same type of CriterionScore linguistic variable. Assuming q criteria, the Inference Engine contains simple fuzzy rules of the form :

IF $(CS_q$ is BAD$)^q$ THEN (Output=BAD)^q
IF $(CS_q$ is MODERATE$)^q$ THEN (Output=MODERATE)^q
IF $(CS_q$ is GOOD$)^q$ THEN (Output=GOOD)^q

However, if each rule is assigned with a contribution weight $(C_w)^q$, different for each fuzzy value w, then the output is defined as :

$$M^Y{}_w = \min(1, (C_w)^1 * (M_w)^1 + (C_w)^2 * (M_w)^2 + ... + (C_w)^r * (M_w)^r)$$

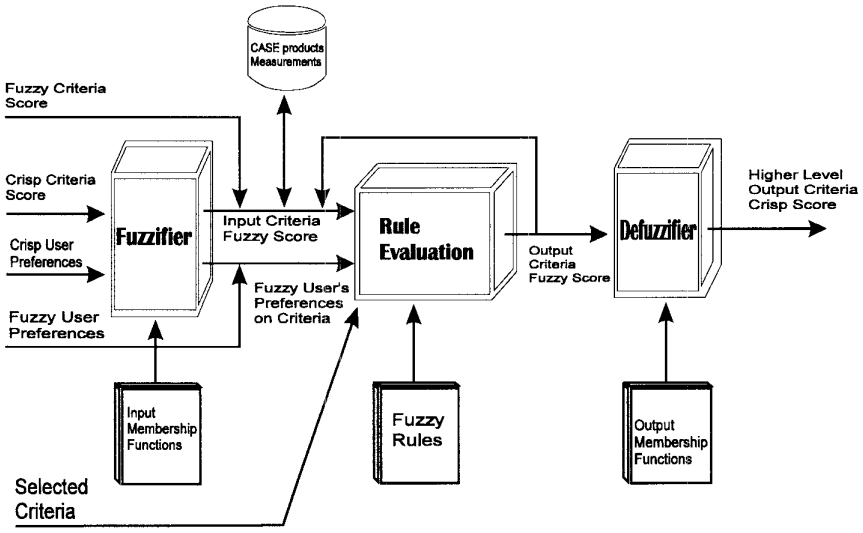

Fig. 6 : Fuzzy Inference Engine

The crisp values of contribution weights are calculated during a separate defuzzification phase concerning the fuzzy values of the criteria relative importance and they are divided by their total sum before they are used to the above equation, in order to ensure that their final total sum equals to 1. It must be stated here, that contribution weights express the degree to which the value of an input criterion affects the value of the output criterion; they cannot change the value of the associated fuzzy variable. Therefore, during defuzzification, a bad score with high relative importance will always attract the crisp value of the output towards the same direction; it will never have the same effect with an input criterion which has good score and low relative importance.

All defuzzification procedures are realised using the *Centre Of Area* method. Let x_{Wi} be the support value at which the membership function $M'^{Yr}_{Wi}(x_{Wi})$ reaches the maximum value $x_i = x_{Wi}$. Then the defuzzified output is

$$y = \frac{\sum_i M'^{Yi}_{Wi}(x_{Wi})\, x_{Wi}}{\sum_i M'^{Yi}_{Wi}(x_{Wi})}$$

The use of the same fuzzy sets for all metrics is not restrictive, since metrics can be freely assigned with any value in the range [0,1] of their relative membership functions, enabling to simulate the use of a virtually infinite number of rating levels. The relative importance among a group of quality criteria, is no longer influenced by the range of each criterion, as they all have the same range effectively formed from the five linguistic variables and their associated membership functions.

Any achieved score during the measurements is propagated to higher levels if it coincides with VERY STRONG user interest for the relevant criteria and with $M_{VERY\ STRONG} \rightarrow 1$, whereas it gradually fades out when it passes through criteria with less interest. Nevertheless, nothing happens abruptly. Even if it fades out under partially positive conditions, a small percentage succeeds to reach the output and to (even partially) influence the defuzzification phase, leading to a more accurate result and smoother transitions. The defuzzification may be postponed until the highest criterion level has been evaluated. Following this procedure, the output of the rule evaluation is retrofitted to the same subsystem and the results of each level are propagated to the higher one. Otherwise, the output is guided to the last block of the subsystem, where defuzzification takes place.

An extension to the defuzzifier in order to support the expression of necessity for the criteria, would also be very useful. There are criteria that are critical for target systems, and their lack or their bad score may determine the whole higher level entity that summarises them in the quality criteria hierarchy. For example, consider a quality sub-characteristic that is related to a group of metrics. Although each metric is associated with a relative importance, one of them may be so important that if it has a bad score, then the whole sub-characteristic should be rejected or marked with this bad score; whereas if it has a satisfying score, then the other metrics and their relative importance should be taken into account as well. Necessity can be expressed, in the defuzzification process setting Evaluation Result = 0 when $M_{VERY\ BAD}$ is close to 1.

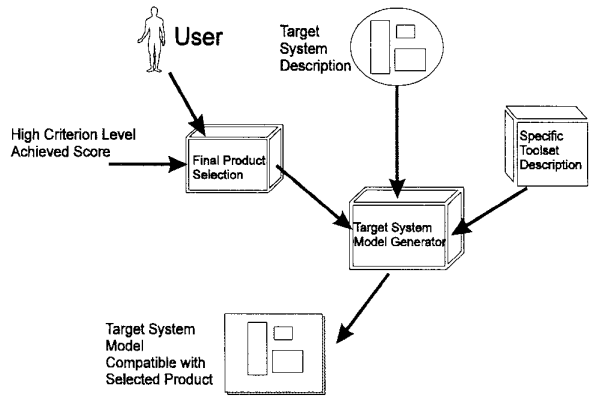

Fig. 7 : Final Selection and Target System Model Generation

VI. Target System Model Generation

The input of this stage, depicted in Fig.7, is a state vector describing the results of each element of the highest criterion level that the user has chosen to evaluate, in the form of the linguistic variable CriterionScore. After the defuzzification of each criterion in the last calculated level, achieved scores are presented to the user through an interactive environment. Possible alternatives for his reaction include :

- to confirm the selection made by the Inference Engine.
- to repeat calculations changing the initial requirements or the specifications of the target system and observe their influence on the output of the evaluation system.
- to take a decision after having consulted the results of the fuzzy quality evaluation.

Finally, according to the selected toolset and the user supplied target system description from the first stage, it is possible to automatically generate initial models of the system under development, which will be compatible with the selected product, e.g. in the case of an Object-Oriented target system, a Class model can be generated if the selected CASE product supports Booch's Object-Oriented design methodology [6] or an Object Model if it supports OMT methodology defined by Rumbaugh et al. [7]. For this purpose the database of the software tool contains a description of the models that each CASE product supports, written in a specific purpose language. The third stage combines this description with the initial description of the target system and generates the appropriate system models.

VII. Conclusions

Software engineers need CASE tools to support their activities during the entire systems life-cycle. The selection of the appropriate product can be very confusing and difficult without the application of a systematic evaluation process and a supporting computerised tool.

Based on the ISO/IEC specifications for software quality, an evaluation process model has been developed and the specifications of a software tool for implementing this model have been determined. The automated tool contains a pool of CASE products quality criteria and employs fuzzy sets theory in order to produce more reliable results from imprecise inputs, like the measurements of the available products in terms of the selected criteria and the user's preference on the criteria. In addition, the user may repetitively apply the process and refine the requirements, in order to find the product that best supports the target system. The software tool will also provide the appropriate system models for the finally selected CASE tool, thus shortening the system development life-cycle.

Acknowledgement

This work was partially supported by the "Research for Advanced Communications in Europe" (RACE II) program, R2041 PRISM project of the European Union.

References

[1]. Al.Fuggeta, "A Classification of CASE Technology", *IEEE Computer*, Dec 1993.
[2]. "Object-Oriented CASE products Survey and Evaluation Process", RACE II Project 2041 PRISM, Univ. Of Patras, Lab. Of Electromagnetics, Dec 1994.
[3]. Takeshige Miyoshi, Motoei Azuma, "An Empirical Study of Evaluating Software Development Environment Quality", *IEEE Transactions on Software Engineering*, May 1993.
[4]. "ISO/IEC 9126 Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use", ISO, 1991.
[5]. L. A. Zadeh , "Fuzzy Logic", *IEEE Computer*, April 1988.
[6]. Grady Booch, "Object-Oriented Analysis and Design with Applications", Benjamin/Cummings (Second Edition 1994).
[7]. J. Rumbaugh, M.Blaha, W.Premerlani, F. Eddy, W. Lorrensen, "Object-Oriented Modelling and Design", Prentice Hall, 1991.