Reprint

On Applying Fuzzy Sets in the Evaluation Process of Object-Oriented Supporting CASE Tools

K. Agavanakis, T. Antonakopoulos and V. Makios

The 21st EUROMICRO Conference

COMO, ITALY, SEPTEMBER 1995

On Applying Fuzzy Sets in the Evaluation Process of Object-Oriented Supporting CASE Tools

K. Agavanakis, T. Antonakopoulos 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 quality evaluation procedure for CASE products. The procedure is based on the ISO software product quality evaluation process model, which has been extended using fuzzy sets theory for achieving more reliable results. Fuzzy sets are employed to express concepts such as the strength of each criterion involved in the evaluation, its relative importance, etc. The produced results are tolerant to small changes of the evaluation procedure inputs, which are by nature imprecise. The evaluation procedure architecture is modular while the fuzzy rules and the definition of the used fuzzy sets can be easily modified by the user for accurately describing the requirements of the target system.

I. Introduction

Software engineers need CASE (Computer Aided Software Engineering) tools to support their activities during the entire software life-cycle. The effort of this work is to describe the architecture of a formal process, based on a combination of the ISO/IEC 9126 software quality evaluation model [1] and fuzzy sets theory, that has been used to evaluate the quality of CASE products supporting the Object-Oriented paradigm. It was originally performed within the framework of the RACE II PRISM project [2]. Taking feedback from experience, it has been expanded to provide more accurate results. PRISM's objective is to progress Service Management in Europe and has early been influenced from OO technology. As a result, an appropriate CASE product is needed for the development of each target system that will be based on the suggested Service Management Reference Configuration Framework.

Section II presents the need of a systematic formal method for evaluating the quality of CASE products and presents briefly the ISO/IEC 9126 specifications for software quality evaluation. Section III highlights an automatic quality evaluation process consisting of three main stages, and its influences from fuzzy sets theory. Section IV analyses the process of fuzzy inference, presents the mathematical background and compares it to the classical approach of criteria aggregation. Finally, conclusions from the overall experience of defining the automatic evaluation process are presented in Section V.

II. CASE Technology Evaluation Process

The evaluation of a CASE product must be guided by the requirements of the particular target system, where it will be applied. This is evidenced by the fact that it is not possible to identify an optimal general-purpose software development process; and different software life-cycles demand different tools in order to be sufficiently supported.

In the case of tools that support the OO technology there is the additional problem that there are currently several OO methodologies with many semantic differences among them, which consider different models to represent software systems (e.g. static model, dynamic model, functional model and state transition model). This phenomenon has led the software market to the development of CASE products that support either several OO methodologies or an exclusive one, like the Rational ROSE that supports Booch's OOD methodology [3] and Objectory that supports the Objectory process [4].

The large number of parameters that impact the CASE products evaluation, makes apparent that only a systematic formal method can ensure a safe selection. ISO/IEC 9126 specifications provide a procedure to objectively define software quality and to determine the criteria for evaluating software products. Six main quality evaluation characteristics are defined: *Functionality*, *Reliability*, *Usability*, *Efficiency*, *Maintainability* and *Portability* [1]. When software quality of composite software products has to be evaluated, each quality characteristic is expanded into sub-characteristics that better reflect the product under examination. Although many characteristics have been proposed and used for software quality evaluation, "there is not a widely accepted single set of characteristics, because

0-8186-7127-0/95 $4.00 © 1995 IEEE

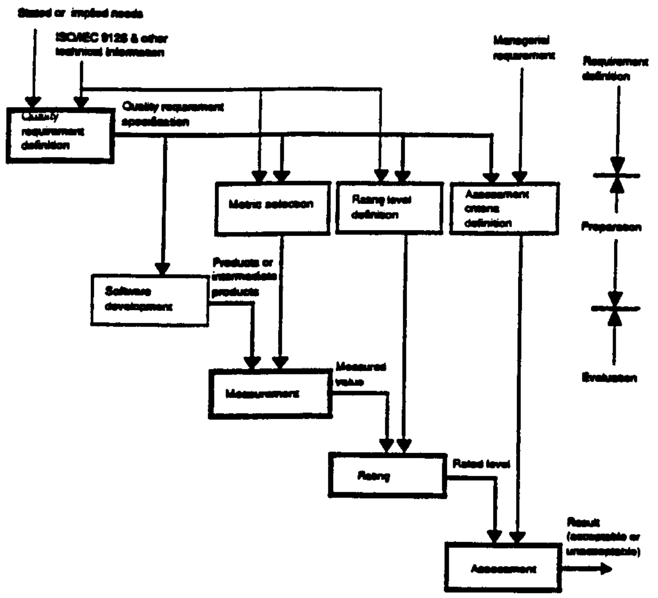

Fig.1 : ISO/IEC 9126 software Product Evaluation Process Model

the set of characteristics that is used in each case, depends on the evaluator opinion“ [5].

The evaluation process model is 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, as well as the definition of the requirement level and of the relative importance of each quality sub-characteristic. Then during the evaluation preparation, specific quality metrics are selected for each quality sub-characteristic and they are associated with their relative importance, measurement scale, measurement method and rating levels. The last step is to perform the appropriate tests and to use the measured values (in terms of the related rating levels) for the final assessment..

The ISO model is hierarchically structured and can be represented by the general form of Fig.2, where each level in the depicted hierarchy, is referred to as a *criterion level*. 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, or through a “rule of thumb”.

Applying the first step of the ISO/IEC model, three fundamental quality characteristics were defined for the quality evaluation of CASE tools : *Functionality*, *Usability* and *Portability*. The Table.1 presents an indicative list of their sub-characteristics and of metrics that can be used in association with them.

The quality sub-characteristics that have been defined in order to facilitate the assessment of Functionality are : *Suitability, Accuracy, Interoperability, Compliance, Security, Multitasking, Distributed processing, Workgroup support, Version control and Configuration management, Code generation, Document and Report generation, Reverse engineering, Configurability, Integrated Model completeness and consistency checking, Prototype support, On-line help, Testing, Project Management and Development results reuse* [1]. The quality characteristic of Usability has been expanded to the sub-characteristics of *Understandability*, *Learnability* and *Operability* proposed by the ISO/IEC model and finally the Portability has been expanded to *Adaptability*.

Practice showed that this model presents certain drawbacks. 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 desired; but

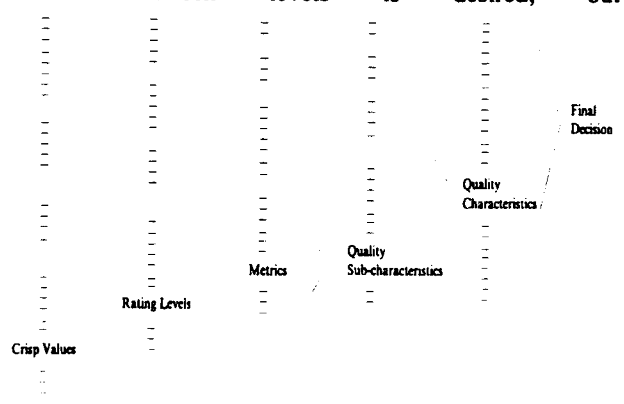

Fig.2 : Evaluation Criteria Hierarchy

Table 1 : CASE products quality evaluation criteria

Quality characteristic	Quality sub-characteristic	Metric
$Q_{.1}$ Functionality	$SQ_{.1\,1}$ Suitability	$MSQ_{.1\,1\,1}$ Methodology realisation
		$MSQ_{.1\,1\,2}$ Product's function realisation
	$SQ_{.1\,3}$ Security	$MSQ_{.1\,3\,1}$ Prevent users from unauthorised access to tools
		$MSQ_{.1\,3\,2}$ Prevent users from unauthorised access to projects
	$SQ_{.1\,4}$ Multitasking	$MSQ_{.1\,4\,1}$ Multitasking facilities
	$SQ_{.1\,5}$ Workgroup support	$MSQ_{.1\,5\,1}$ Workgroup support facilities
	$SQ_{.1\,6}$ Distributed processing	$MSQ_{.1\,6\,1}$ Distributed processing facilities
	$SQ_{.1\,13}$ Prototypes	$MSQ_{.1\,13\,1}$ Timing constraints checking
		$MSQ_{.1\,13\,2}$ Behaviour animation
		$MSQ_{.1\,13\,3}$ Quality evaluation facilities
	$SQ_{.1\,14}$ On-line help	$MSQ_{.1\,14\,1}$ On tools function
		$MSQ_{.1\,14\,2}$ On supporting methodology
	$SQ_{.1\,17}$ Development results reuse	$MSQ_{.1\,17\,1}$ Features that isolate and encapsulate target components
		$MSQ_{.1\,17\,2}$ Browsing/assessment of models by relevant properties
$Q_{.2}$ Usability	$SQ_{.2\,1}$ Understandability	$MSQ_{.2\,1\,1}$ Ease of understanding the concepts of the product
		$MSQ_{.2\,1\,2}$ Ease of understanding the operation of the tools
	$SQ_{.2\,2}$ Learnability	$MSQ_{.2\,2\,1}$ Sufficiency of product's manuals
		$MSQ_{.2\,2\,2}$ Ease of mastering operation of product's tools
	$SQ_{.2\,3}$ Operability	$MSQ_{.2\,3\,1}$ Ease of descriptive operation
		$MSQ_{.2\,3\,2}$ Ease of verifying/translative operation
		$MSQ_{.2\,3\,3}$ Ease of connecting different models and navigating through them
$Q_{.3}$ Portability	$SQ_{.3\,1}$ Adaptability	$MSQ_{.3\,1\,1}$ Ease of adapting existing work to a different environment

with more smooth transition between them. 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. In addition, the restricted number of levels, prevents assessment results of one criterion level to propagate to higher levels, again in an absolute and abrupt way.

Secondly, although verbal descriptions are used in the case of non-quantitative metrics, it is not always possible to objectively define a rating level. Take for example the metric $MSQ_{.1\,13\,3}$ "Prototypes: Quality evaluation facilities" (Table.1) which has been used with the rating levels :

0: Absent.
1: Few metrics available and predefined evaluation process.
2: Many metrics available and predefined evaluation process.
3: User defined metrics and evaluation process

and $MSQ_{.1\,14\,1}$ "On-line help: On tools function", which has the rating levels :

0: Absent.
1: Help only on the product's most fundamental functions.
2: Help on all product's functions.
3: Help on all product's functions and hypertext facilities

These metrics, have been defined having the ISO/IEC procedure in mind and are not influenced from fuzzy sets theory. Nevertheless, definitions like "few" and "most" have been used, inducing evaluation results that are subjective to the evaluator's personal judgement. This fact, magnifies the problem that the calculations results are not tolerant to small variations of the inputs.

Thirdly, the usual procedure to calculate the score of a criterion level is to use relative weights. For example consider the Q_i quality characteristic that can be expanded to j quality sub-characteristics SQ_{ij}. Each of them depends on k metrics MSQ_{ijk}. The achieved score AS_{ij} of each quality sub-characteristic SQ_{ij} is a function of the measured level ML_{ij} and of the required level RL_{ij}, according to the following formulas:

$$Ml_{ij} = \sum_{k=1}^{n} ML_{ijk} \cdot W(MSQ_{ijk}) \qquad (1)$$

where P_{ijk} is the measured level of the k-metric MSQ_{ijk}, and $W(MSQ_{ijk})$ is the relative weight this metric.

$$RL_{ij} = \sum_{k=1}^{n} RL_{ijk} \cdot W(MSQ_{ijk}). \qquad (2)$$

where RL_{ijk} is the required level for the relevant metric, and

$$AS_{ij} = (ML_{ij} \;/\; RL_{ij}) * 100\% \qquad (3)$$

This approach has the drawback that if all the metrics of one quality sub-characteristic do not have the same number of rating levels, then the portion of the influence that each metric has on the quality sub-characteristic, besides from its relative weight, depends on its number of levels as well. This is due to the fact that larger level values

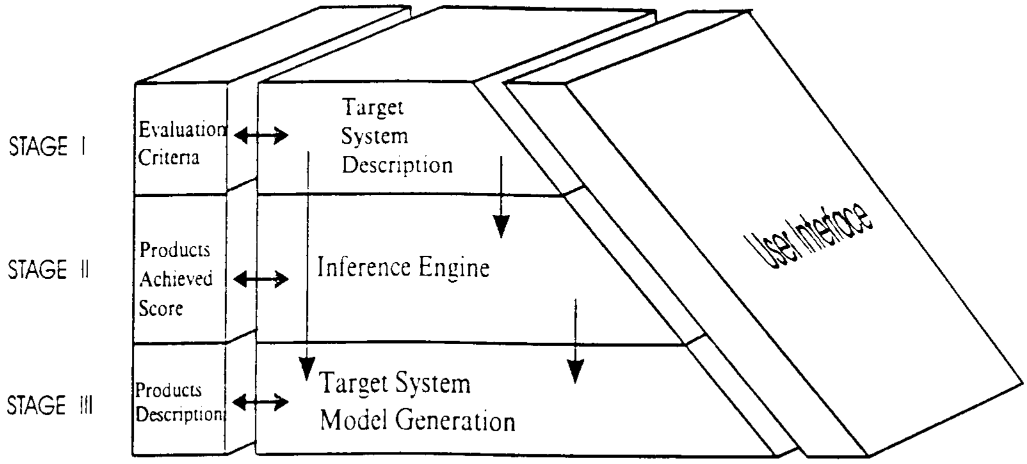

Fig.3 : Block Diagram of the CASE products Quality Evaluation Fuzzy System

give larger sums in Eq.2.

Finally, the described process uses specific crisp values of the criteria relative weights, although, it may not always be possible to accurately define them. The user can be better 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 on the other".

III. Fuzzy-Sets-Based Quality Evaluation Model

Based on the above statements, a CASE products quality evaluation process model has been defined, employing sets theory. From the initial phase of design, it was recognised that this process must be automated by a software system, which is currently under development.

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.

"One of the main features that differentiate fuzzy logic from traditional logical systems, is that it provides a method for representing the meaning of both non-fuzzy and fuzzy predicate-modifiers exemplified by "not", "very", "more or less", "a little" and so on. This, in turn, leads to a system for computing with variables whose values are words or sentences in a natural or synthetic language" [6]. Therefore it becomes clear that if product measurements, user 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.3.

It is considered that CASE products are studied and their most important characteristics and operational features, such as supported models and methodologies, Operating System(OS) and operating platform, are stored in a database for future reference. General quality characteristics, sub-characteristics and metrics are also defined and stored, providing a pool of available criteria for the evaluation process. Measurements for each CASE product related to all predefined criteria are stored in a separate database, in the form of fuzzy variables values.

During the first stage, the user supplies a description of the target system using the provided design toolset that is most appropriate for the addressed problem. The system reacts by providing a list of possible quality criteria, since each available toolset is related with certain predefined criteria. Although a careful survey can end with a vast majority of quality criteria appropriate for CASE products evaluation, it is evident that not all criteria are applicable to all target systems. The motive idea in the evaluation procedure is to use a short list of criteria, depending on the nature of the problem and to ignore unrelated features [7].

The Inference Engine that dominates in the second stage of the evaluation model, is essentially an adapting fuzzy system. It contains the necessary fuzzy rules for the inference realisation and calculates the achieved score of the elements of a criterion level higher than that of the input. The first input is the list of the criteria that will be actually used for the evaluation, generated by the first stage. The second input is user preferences for the selected criteria and consists of a state vector of linguistic variables expressing the relative importance of each criterion

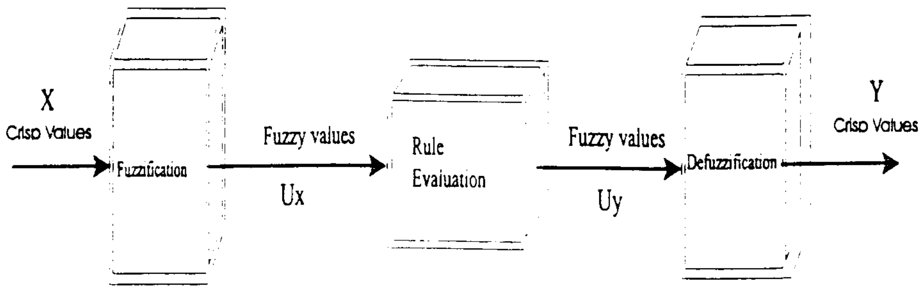

Fig.4 : General fuzzy system

through the linguistic variable *Relative Importance.* The values of this fuzzy variable are {LOW, MEDIUM, HIGH} and the associated membership functions are $\{M^{LOW}, M^{MEDIUM}, M^{HIGH}\}$, which can be configured during the actual operation of the system.

The third input of this stage is the set of measurements for each CASE product and for all the selected criteria, in either crisp or fuzzy form. Measurements are stored in the system database, in the form of values of an other linguistic variable, which refers to the degree that the product satisfies the corresponding criterion. This fuzzy variable is named *Criterion Score* and takes the values {BAD, MODERATE, GOOD}. Their associated membership functions are $\{M^{BAD}, M^{MODERATE}, M^{GOOD}\}$. The output of the Inference Engine is a state vector describing the achievements of each criterion of the higher level, in either crisp or fuzzy form (using the same type of the Criterion Score linguistic variable).

It must be noted that the use of the same three fuzzy variables for all metrics is not restrictive, since membership functions can be freely assigned with any value in the range [0,1], simulating the use of a virtually infinite number of rating levels. As a result the relative importance of quality criteria, is no longer influenced from their range.

Any achieved score during the measurements is propagated to higher levels if it coincides with relative importance HIGH for the relevant criteria and with $M^{HIGH} \rightarrow 1$, 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 influence the defuzzification result, leading to a more accurate and smooth reaction. 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 stage and the results of each criterion level are used for the calculations of the next higher one. Otherwise, the output is transformed to crisp values through a defuzzification process.

In stage III the initial target system description of stage I is processed according to the product that has been selected. The system database contains specifications and descriptions of the capabilities and features of all the available products. 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.

IV. Inference Engine Description and Comparison to Classical Approach

A general fuzzy system is shown in fig. 4. The first step in fuzzy logic processing involves a domain transformation called fuzzification. Crisp inputs are transformed into fuzzy inputs through the use of membership functions, explicitly defined for each input. Then the fuzzy processor uses linguistic rules to determine what output should occur in response to a given set of input values; this procedure is called rule evaluation (also referred to as fuzzy inference). All significant fuzzy outputs, are combined into a specific result for the output variable during the last step, called defuzzification.

A fuzzy set F in a universe of discourse U is characterised by a membership function M_F which takes values in the interval [0,1], i.e. $M_F: U \rightarrow [0, 1]$. Thus a fuzzy set may be represented by a set of ordered pairs,

$$F=\{(u, M_F(u)) \mid u \in U, M_F: U \rightarrow [0, 1]\} \tag{4}$$

If $M_F(u)>0$, then u is called supporting value. A linguistic variable x in a universe of discourse U is characterised by a term set

$$T(x)=\{T_x^1, T_x^2, \dots T_x^k\} \tag{5}$$

which is a set of names of linguistic values of x with each value T_x^r, r=1..k being a fuzzy number with a membership function M_x^r defined on U by an associated set

$$M(x)=\{M_x^1, M_x^2, \dots, M_x^k\} \tag{6}$$

Therefore M(x) is a semantic rule for associating each value with its meaning [8]. For example, if the score of the metric $MSQ_{.1.2.1}$: "Use of existing text editors" is considered as a fuzzy variable, it could be related with the term set T(x)={BAD, MODERATE, GOOD}.

Following the above definitions, the input vector X of a fuzzy system which includes the input state linguistic variables Xi's and the output vector Y which includes the output state linguistic variables Yi's can be considered as

$$X = \{ (X_r, U_r, \{ T_{Xr}^1, T_{Xr}^2, \ldots, T_{Xr}^{kr} \}, \{ M_{Xr}^1, M_{Xr}^2, \ldots, M_{Xr}^{kr} \}) \mid r=1,\ldots, n \} \quad (7)$$

$$Y = \{ (Y_r, U_r, \{ T_{Yr}^1, T_{Yr}^2, \ldots, T_{Yr}^{kr} \}, \{ M_{Yr}^1, M_{Yr}^2, \ldots, M_{Yr}^{kr} \}) \mid r=1,\ldots, m \} \quad (8)$$

The fuzzy processor contains a set of fuzzy logic rules R. If there is a rule of the form

$$\text{IF} \quad ((X_1 \text{ is } T_{X1})^q \text{ AND } (X_2 \text{ is } T_{X2}))^q$$
$$\text{THEN} \quad (Y=T_Y)^q, \quad 1 \leq q \leq \text{number of rules} \quad (9)$$

then the firing strength a^q of this rule is

$$a^q = M_{X1}(w_1)^q \wedge M_{X2}(w_2)^q, \quad (10)$$

where w_1, w_2 are linguistic values of X_1, X_2 and $\wedge$ denotes fuzzy-AND operation and can be evaluated either as $\min(M_{X1}(w_1)^q, M_{X2}(w_2)^q)$ (intersection) or as $M_{X1}(w_1)^q * M_{X2}(w_2)^q$ (algebraic product).

The membership function $M'_Y(w)^q$ of the output of the corresponding rule, becomes

$$M'_Y(w)^q = (a^q \wedge M_Y(w)^q) * C^q, \quad (11)$$

where C^i is a multiplicative weighting factor named *Contribution Weight* associated with the q-rule. The Contribution Weight has usually the values 1 (meaning that the rule is taken into account) or 0 (meaning that the rule does not affect the inference process); but it can take all the values in the interval [0,1] in order to declare a relative importance between the rules.

Different rules give different results $M'_Y(w)^q$ for the same linguistic value w of the output variable Y and they are combined to give a final value $M'_Y(w)$:

$$M'_Y(w) = M'_Y(w)^{q1} \vee M'_Y(w)^{q2} \vee \ldots \vee M'_Y(w)^{qn} \quad (12)$$

where $\vee$ is the fuzzy-OR operator. $M_Y(w) = M_Y(w)^1 \vee M_Y(w)^2$ is either defined as $\max(M_Y(w)^1, M_Y(w)^2)$ or as $\min(1, M_Y(w)^1 + M_Y(w)^2)$. The result is a membership function curve.

Before feeding the signal to the output, a defuzzification process is performed. Among the commonly used defuzzification strategies, the *Centre Of Area* method yields a superior result [9]. Let w_{Pr} be the support value at which the membership function $M_{Yr}(w_{Pr})^P$ reaches the maximum value $w_r = w_{Pr}$. Then the defuzzified output is

$$y = \frac{\Sigma_{Pr} M'_{Yr}(w_r)^{Pr} w_{Pr}}{\Sigma_{Pr} M_{Yr}(w_r)^{Pr}} \quad (13)$$

The block diagram of the proposed criteria aggregation process, is depicted in Fig.5. Let X_r, r=1...k, be the fuzzy variables expressing the score of a group of k criteria of one criterion level, which correspond to the output fuzzy variable Y, corresponding to a criterion score of the next higher level. If crisp values are available, X_r are produced from the fuzzification process. Fuzzy values for X_r are taken from the term set

$$T(X) : T(X) = \{ T_{Xr}^1 = \text{BAD}, T_{Xr}^2 = \text{MODERATE}, T_{Xr}^3 = \text{GOOD} \}, \quad (14)$$
$$M(X) = \{ M_{Xr}^1 = M_{Xr}^{BAD}, M_{Xr}^2 = M_{Xr}^{MODERATE}, M_{Xr}^3 = M_{Xr}^{GOOD} \} \quad (15)$$

The output variable takes values from a similar term set, but it is defined in a different universe of discourse and membership functions are not the same :

$$T(Y) = \{ T_Y^1 = \text{BAD}, T_Y^2 = \text{MODERATE}, T_Y^3 = \text{GOOD} \} \quad (16)$$
$$M(Y) = \{ M_Y^{BAD}, M_Y^{MODERATE}, M_Y^{GOOD} \} \quad (17)$$

In order to express better each metric, each input membership M_{xr}^P function can be defined differently according to its own universe of discourse, but semantically will mean the same thing, e.g. different metrics have different scales and levels of what is good and bad but

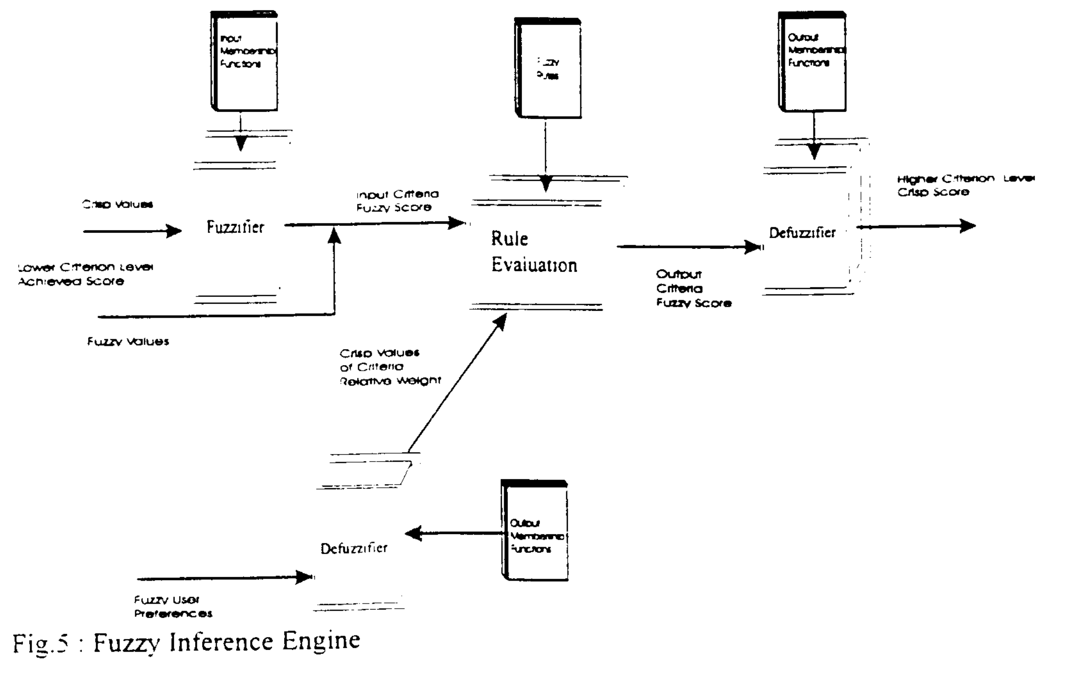

Fig.5 : Fuzzy Inference Engine

when summarising the results, it is the semantic concept, that is to say the linguistic value "GOOD" or "BAD", that is taken into account.

The Inference Engine contains simple fuzzy rules of the form :

$$\text{IF } (X_q \text{ is BAD})^q \quad \text{THEN } (Y=\text{BAD})^q \qquad (18)$$
$$\text{IF } (X_q \text{ is MODERATE})^q \quad \text{THEN } (Y=\text{MODERATE})^q \qquad (19)$$
$$\text{IF } (X_q \text{ is GOOD})^q \quad \text{THEN } (Y=\text{GOOD})^q \qquad (20)$$

where $q = 1..n$

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

$$M_Y^w = \min(1, (C^w)^1 \cdot (M_Y^w)^1 + (C^w)^2 \cdot (M_Y^w)^2 + \ldots + (C^w)^r \cdot (M_Y^w)^r). \qquad (21)$$

The crisp values of contribution weights are calculated during the criteria relative importance defuzzification phase and they are divided by their total sum before they are fed to the rule evaluation block, in order to ensure that their final total sum equals to 1.

In order to evaluate the proposed fuzzy-sets-based process, a comparison with the classical approach must be performed. Consider the quality sub-characteristic $SQ_{\cdot 2.2}$: "Learnability". The associated metrics $MSQ_{\cdot 2.2.1}$: "Sufficiency of product's manuals" and $MSQ_{\cdot 2.2.2}$: "Ease of mastering operation of the products tools", have relative weights $C_{\cdot 2.2.1} = 0.3$ and $C_{\cdot 2.2.2} = 0.7$ respectively. In addition, these two metrics are considered to have 3 rating levels. Metric $MSQ_{\cdot 2.2.1}$ is defined as

0. Manuals are inadequately describing the product
1. Manuals describe basic operation
2. Manuals describe the overall process and the logical connection of the supported models

And metric $MSQ_{\cdot 2.2.2}$ is defined as

0. 70 to 100 hours of practice are needed
1. 30 to 70 hours of practice are needed
2. Less than 30 hours of practice are needed

Therefore, the required level of the quality sub-characteristic $SQ_{\cdot 2.2}$, according to Eq.2, is equal to 3*2 ÷ 0.7*2 = 2. Figure 6, depicts the contribution of metric $MSQ_{\cdot 2.2.2}$ to the total value of $SQ_{2.2}$ as a function of the possible values that the metric can take. In addition Fig.7 shows the effect of the same metric to the achieved score of the quality characteristic Q_2 which belongs to one more higher criterion level, assuming that all three quality sub-characteristics below it, have the same relative weight. i.e. $W(SQ_{2.1}) = W(SQ_{2.2}) = W(SQ_{2.3})$ and that three rating levels are defined for each quality sub-characteristic with limiting values at 0.33 and 0.67.

It is obvious that a small change of a metric around the limits of two neighbouring levels, can strongly influence not only the criterion level under estimation, but the next one too. The problem is equally serious in the case of non quantitative metrics, where judgement is subjective to personal estimation and a small difference during measurements may improperly change the evaluation results. Usually, several metrics are combined to give a quality sub-characteristic and several quality sub-characteristics are combined to give a quality characteristic, and it may be possible to select such levels that the evaluation result will be tolerant to most changes of certain metrics without differentiating the results. However, this phenomenon cannot be totally restricted, and there will always be abrupt transitions between assessment levels caused by a very small change of a single metric.

Applying the fuzzy approach in the above example, it has been considered that user preference on the relative importance of criteria is given as shown in Fig.8 and that relative weights produced by the associated defuzzification process are the same as before. Membership functions of the linguistic values are depicted in Fig.9 for the $MSQ_{\cdot 2.2.2}$ metric and in Fig.10 for the output quality sub-characteristic $SQ_{\cdot 2.2}$. Notice that although they have the same form, M^{GOOD} and M^{BAD} have changed place.

Based on these definitions, Fig.11 depicts the value of the quality sub-characteristic $SQ_{\cdot 2.2}$ as a function of the crisp values of the metric $MSQ_{\cdot 2.2.2}$. It is obvious that the transition is smoother than the one depicted in Fig.6 and the evaluation result is more tolerant to imprecise inputs.

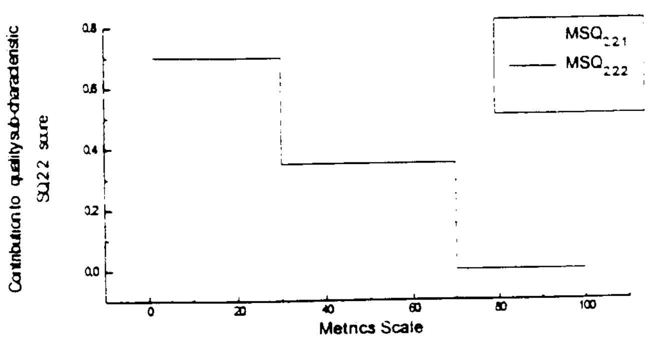

Fig.6 : Classical aggregation of metrics

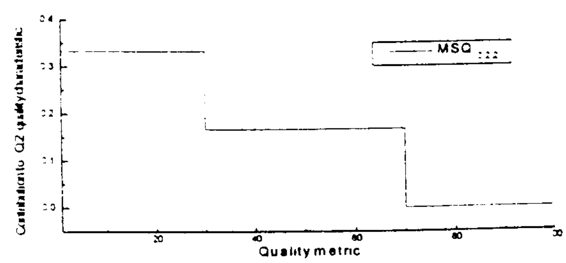

Fig.7 : Influence of a metric to the calculated score of a quality characteristic, using classical approach.

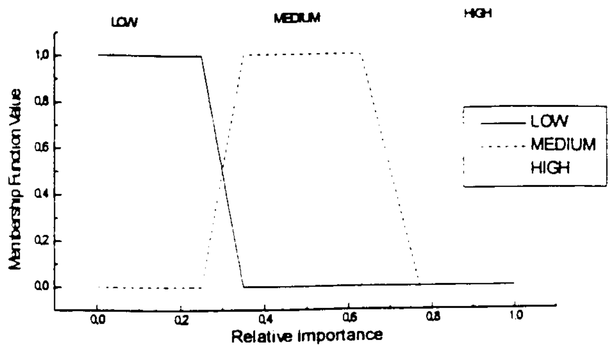

Fig.8 : Fuzzy criteria relative importance

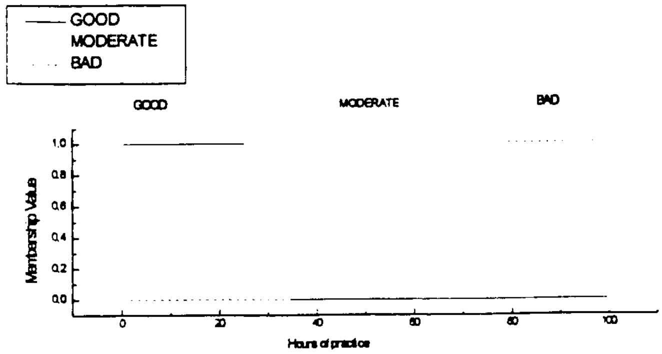

Fig. 9 : Fuzzy values of metric $MSQ._{2.2.2}$

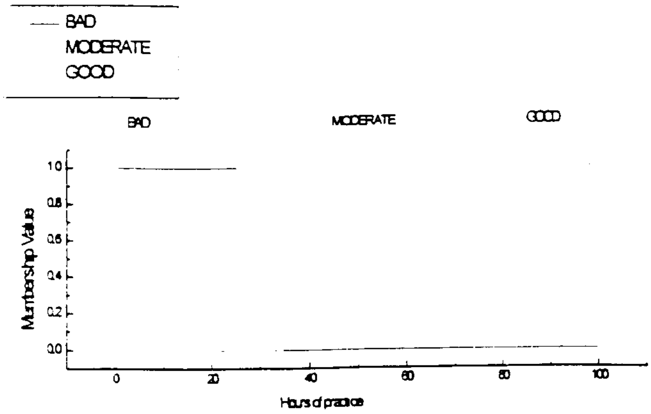

Fig. 10 : Fuzzy values of $SQ._{2.2}$

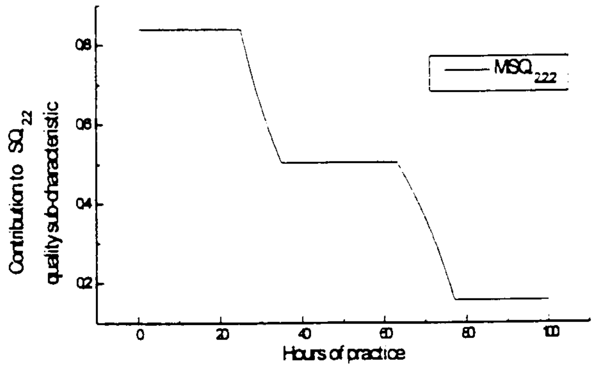

Fig.11 : Quality sub-characteristic $SQ._{2.2}$ as a function of metric $MSQ._{2.2.2}$ using fuzzy sets approach

V. Conclusions

Software development methodologies affect all aspects of the engineering practice, so they should be highly considered when selecting CASE products. Due to the vast variety of target projects needs, 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.

This paper has presented an evaluation process model based on the ISO/IEC specifications for software quality. The proposed model 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. The comparison to the classical approach of criteria aggregation, showed that the result is more tolerant to small changes of the input measurement values and, therefore, more accurate.

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]. "ISO/IEC 9126 Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use", ISO, 1991.

[2]. "Object-Oriented CASE products Survey and Evaluation Process", RACE II Project 2041 PRISM Report, Univ. of Patras, Lab. of Electromagnetics, Dec 1994.

[3]. Grady Booch, "Object-Oriented Analysis and Design with Applications", Benjamin/Cummings (Second Edition 1994).

[4]. Ivar Jacobson, "Object-Oriented Software Engineering - A Use Case Driven Approach", ACM Press/Addison-Wesley, (Fourth edition 1993).

[5]. Takeshige Miyoshi, Motoei Azuma, "An Empirical Study of Evaluating Software Development Environment Quality", *IEEE Transactions on Software Engineering*, May 1993.

[6]. L. A. Zadeh, "Fuzzy Logic", *IEEE Computer*, April 1988.

[7]. Rodney Bell, "Choosing Tools for Analysis and Design", *IEEE Software*, May 1994

[8]. Chin-Teng Lin, George Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System", *IEEE Tranactions on Computers*, vol.40, No.12, Dec 1991

[9]. Kosko Bart, "Neural Networks and Fuzzy Systems", Prentice-Hall International Editions, 1992