A framework based measurements for evaluating an IS quality

Olfa Lamouchi, Amar R.Cherif, Nicole Lévy
PRISM, Université de Versailles St.-Quentin, 45, Avenue des Etats-Unis,
78035 Versailles Cedex, France
olla@prism.uvsq.fr

Abstract

There is an increasing community of researchers trying to draw attention to the importance of evaluation issues. Some approaches can be followed to evaluate the quality of software, but gives no guidance on how to construct a high quality software product and on how to use in practice this model across evaluation process. IS are built upon Software, and the quality of Software effects, if not determine the quality of the IS. There are several difficulties of developing a comprehensive model of IS quality. The concepts, models and measures that work in other fields might be usefully applied to the IS field, but careful analysis and consideration should be given.

This paper describes a practical method that can be used to evaluate the expected quality of IS. Our approach is focused on a direct evaluation of the quality of an IS. Our proposed IS quality model is considered as structured set of properties (such as reliability, maintainability, and so on). These properties are usually presented as a hierarchy of statements. The uppermost ones represent the most general properties, suitable to be understood by the widest variety of people, but very difficult to measure directly. The deeper we sink into this hierarchy, the more technical-audience-oriented and measurable properties are found. These finals properties need to be measured using metrics. For this purpose a metrics measurement-based framework is linked to the defined quality model.

Key words: Quality evaluation, quality measurement, quality model.

1. Introduction

IS evaluation has been the subject of intense interest for the IS research community since the early 1990s. As Smithson and Hirscheim (Smithson 1998) state 'evaluation is difficult'. It is a "wicked problem" (Farbey 1999), (Jones 2001). It has been widely noticed in the literature that information system (IS) evaluation is a very difficult task involving a variety of dimensions (Irani 2002), (Peffers 2002), (Sinclair 1995) and various stakeholders (McAulay 2002).

Ad hoc practices for IS evaluation are frequently reported (Irani 2001) and only simple methods, like payback period, are used in evaluation (Lederer 1993). In addition, Remenyi and Sherwood-Smith (Remenyi 1999) described two practical and very relevant issues in IS evaluation. More action oriented quality ideals and principles for evaluation can be found in Cronholm & Goldkuhl (Cronholm 2002) and in Ågerfalk et al (Ågerfalk 2002).
A common factor in all evaluations is the use of measurements (Venkatraman 1986). Strassman ((Strassmann 1985), p. 100) stresses that: "You cannot measure what is not defined. You also cannot tell whether you have improved something if you have not measured its performance", i.e. the need of an operational definition. This is, however, difficult as stated by Hoebeke (Hoebeke 1990). Hoebeke made this comment in relation to a discussion regarding the use of measurements based on calculations, e.g. financial accounting. Measurements are invariably used in complex sense-making processes where both translations and interpretations take place.

Different types of organisations could have a need for different evaluation criteria. Is the organisation producing goods or services (or both)? Is it a private or public? Different parts of the organisations could have different internal logics and cultures that could affect the evaluation approaches. A quality model links together and defines the various software metrics and measurement techniques that an organisation uses which when measured, the approach taken must be sufficiently general for hybrid hardware and software systems. Some approaches are focused on a direct evaluation of the quality of a software product, and can be implemented using GQM method (Goal-Question-Metric), described for the first time in (Basili 1984) and developed since that time by NASA. The set of goals or quality characteristics can be the same or similar to the one defined in ISO/IEC 9126 (ISO/IEC 2001).
Some form of quality model, explicit or not, always is adopted when an evaluation is performed. Most often the model is implicitly assumed, or is ad hoc defined for the particular evaluation. Within the Information Systems literature, there are many models proposed to assess the quality of Information Systems. One of the more complete and better known is that of DeLone and McLean's model of information systems success (DeLone 2003). This model has been used as a basis for empirical research, and has been refined and extended by many researchers Ozkan and Bilgen (Ozkan 2003) and Andersson and Von Hellens (Andersson 1997). There are several difficulties of developing a

comprehensive model of IS quality. It should be obvious that a definition of IS Quality needs to be holistic, one which encompasses all relevant contexts- both technological and organizational-so that the framework to be constructed takes into account the different work contexts and specific organizational needs that should be considered when evaluating an IS quality. Information systems were not considered as entities that need to be assessed in order their level of quality to be depicted. The major problem with the actually approaches is the lack of comprehensive guidelines to produce a consensus view of quality attributes and the inability to provide common quality criteria. It is very difficult to a customer these approaches. These methods give no guidance on how to construct a high quality IS.

The aim of this study is to present a new approach for direct evaluation of the quality of an IS. Our proposed IS quality model is considered as structured set of properties (such as reliability, maintainability, and so on). These properties are usually presented as a hierarchy of statements. The finals properties need to be measured using metrics. For this purpose a metrics measurement-based framework is linked to the defined quality model. In this evaluation approach conflicts can be removed and an indication of overall quality can be determined using a Fuzzy engine. This paper is organized as follow, the following section addresses the IS evaluation approach. Following this is the evaluation model which is composed of three parts: the quality model, the metric model and the metric-quality model. After this, we give the evaluation scenario which is composed of three parts: the description of the scenario, the interpreter engine and the results interpretation. The paper ends with a conclusion and perspectives.

2. IS evaluation approach

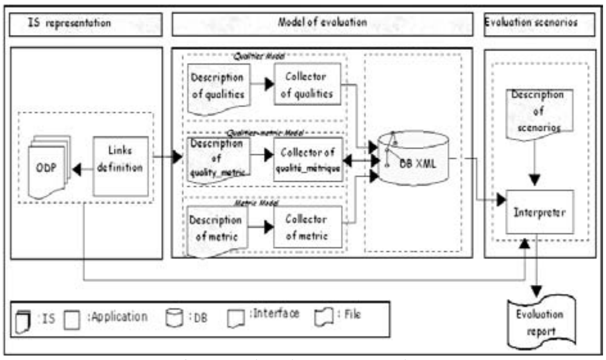

Fig. 1. Structure of IS evaluation approach

To be effective, a framework for IS evaluation must be flexible. Indeed, because of the diversity of the scenarios of evaluation, those cannot be implemented in ad hoc mode. It is necessary to offer to the future users of our framework a flexible device enabling them to describe and modify the scenarios which they wish to implement. Each scenario is based on the quality model, metric model and measurement model. To define a flexible framework, we propose architecture with three components (Fig. 1): IS representation, Model of Evaluation and Scenarios Evaluation.

2.1. IS representation

The first step of our approach is to permit to the users to model their IS using for example RM-ODP. RM-ODP (reference Model-Open Distributed Processing) is a standardized modelling framework to describe a distributed application according to five viewpoints (ISO/IEC 1996), (ISO/IEC 1998), (Putman 2000): Enterprise (objective, business, rules), Information (data), Computational (functional decomposition), Engineering (communication and deployment) and Technology (hardware and software infrastructure).

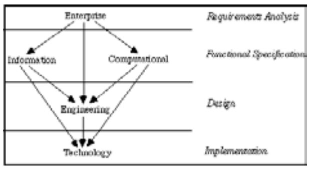

Fig. 2. RM-ODP viewpoints

In a RM-ODP vision, any action on a view may involve the same action or other action in one or more other views but the views are not explicitly linked. To circumvent this problem we chose to use the approach EVOS (Yahiaoui 2005), where the designer can explicitly describe not only the views of his or her system but also the correspondences which exist between these views. Using IS presentation component, the user can:

- Present the IS model according to different viewpoints using RM-ODP,
- Present the linked model that describes the correspondences which exist between the different views using EVOS approach,
- Define a set of metrics that describe particular quality characteristics, measure and collect data from a set of elements of different views of IS.

The evaluation process, witch will be detailed in the following sections, is based on a quality model. This model links together and defines the various software metrics and measurement techniques that an organisation uses which when measured, our approach taken must be sufficiently general for hybrid hardware and software systems. In this work software quality factors that should be taken into account in information systems will be considered. For this purpose, we start by identifying the metrics and measurement approaches that can be used in the IS presentation. These metrics will be associated to a set of variation points located into different views of the IS. The relation between these points expressed by the linked model permit to deduce the consequence of one variation on the global IS architecture.

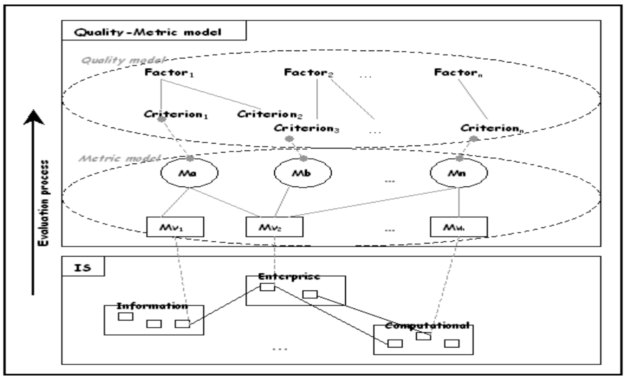

Fig. 3. Evaluation process

2.2. The Evaluation Model

The evaluation model is a predictive model of quality; it makes it possible to identify the forces and the weaknesses of IS according to predetermined criteria. In our approach, the evaluation model is composed of three parts: the quality model, the metric model and the metric-quality model. Before seeing in detail each of these parts we will present the meta-model evaluation (Fig. 4). In this meta-model a factor can be constituted by one or several criteria. Each criterion is measured thanks to metric. A metric can call variable metrics or another metrics. Each metric and variable metric is defined by a measurement function or procedure.

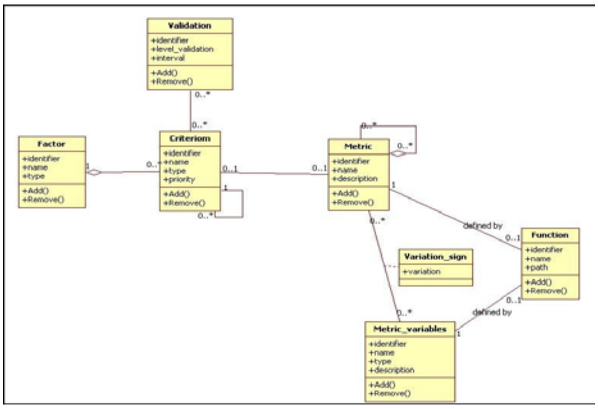

Fig. 4..The meta-model evaluation

Quality comes out as a collection of desired properties, or factors. More precisely, each factor is giving rise to one or more quality goals. Quality factors have been used in literature since the early hierarchical quality models (Boehm 1978). The popularity of theses is reflected in the fact that the International Standard ISO 9126 is based on them. The standard recommends a number of factors such Reliability, usability, maintainability etc. These properties (factors) are usually presented as a hierarchy of statements, with a "expressed-in-terms-of" relation. The uppermost ones represent the most general properties, suitable to be understood by the widest variety of people, but very difficult to measure directly. The deeper we sink into this hierarchy, the more technical-audience-oriented and measurable properties are found.

2.2.1. Quality Model

The Quality model can be defined by a set of views concerning the product. Each view is decomposed into several factors. A factor is decomposed into several criteria. The factors are in general external attributes (but also internal attributes: testability, effectiveness, etc). Each criterion is defined by a set of metric. For example, in the model of McCall (McCall 1977) the reliability factor is broken up into coherence, precision, fault-tolerance and simplicity.

Description of Quality model

In our approach, the quality model is presented in form of tree structure in which each factor can have one or more criteria. Each criterion can have one or more sub-criteria.

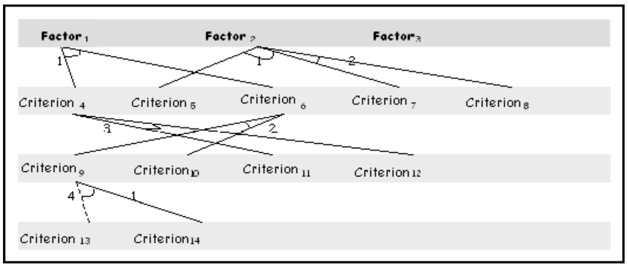

Fig. 5. The tree structure of the Factors and Criteria model

The different feature properties that links factors and criteria and sub-criteria are represented using the following symbols: (1)';' alternative, (2)'∧' mandatory, (3) '∨' or, (4) '?' optional, (5) '∅' empty mode.

Consider the example (e.g.:Fig. 5) we can give the following expressions:

Quality goal$=(Factor_1;Factor_2;..;Factor_5)$,
$Factor_1=(Criterion_4;Criterion_6)$,
$Factor_2=(Criterion_5;Criterion_7\wedge Criterion_8)$,
$Criterion_4=(Criterion_{11}\vee Criterion_{12})$,
$Criterion_5=(\emptyset)$
$Criterion_6=((Criterion_9\wedge Criterion_{10}))$,
$Criterion_7=(\emptyset)$, $Criterion_8=(\emptyset)$,
$Criterion_9=(Criterion_{14};Criterion_{13}?)$,
$Criterion_{10}=(\emptyset)$, $Criterion_{11}=(\emptyset)$,
$Criterion_{12}=(\emptyset)$, $Criterion_{13}=(\emptyset)$,
$Criterion_{14}=(\emptyset)$.

- The quality model is composed of a unit of three factors and ten Criteria. Facteur1 is composed of two criteria, on which we do not have any condition.
- Criterion6 is composed of two criteria. To validate Criterion6 its two criteria should obligatorily be validated
- Criterion9 is composed of two criteria. To evaluate Criterion9, the validation of Criterion13 is optional.
- $Facteur_2$ is composed of three criteria. To evaluate $Facteur_2$, the evaluation $Criterion_8$ is obligatory. The evaluation of the two other criteria depends

on the choice made by the user when defining his/her evaluation scenario.

- Criterion4 is composed by two criteria. To evaluate Criterion4, only one of its criteria must be evaluated.

The user can enter all the necessary information about the quality model by using the interface description component (Fig. 1, "component description of qualities"). This component contains for example the following input forms:

Factor
Name :
Type:
For:
(Viewpoints, IS)
Components :
(Criteria, Priorities)

Criterion
Name :
Type:
Father :
For:
(Viewpoints, IS)
Components :
(Criteria, Priorities)

Table.1. Factor form input & Criterion form input.

Example: The Portability factor model
In this example, one quality factor is defined: Portability. Because of the short lifespan of high performance computing platforms and because many applications are developed and run in a distributed heterogeneous environment, most parallel programmers will work on a number of platforms simultaneously or over time. Programmers are understandably reluctant to learn a new performance tool every time they move to a new platform. Thus, we consider portability to be an important feature.
So the model of quality (Fig. 6), is composed of only one factor "Portability". In the model of quality ISO9126, the "Portability" is made up of four criteria: "Adaptability", "Instability", "Conformity", and "Interchangeability". For our example, we have to keep the criterion "Adaptability" and the criterion "Availability" corresponding to the criterion "Instability" in the model of quality ISO9126.

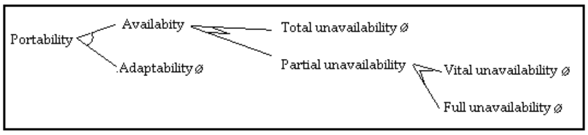

Fig. 6 . The model of the Portability factor

Portability: is characterized by the capacity of a system to function more or less easily in various execution environments. **Availability:** is the capacity of an information processing system to being in operating state for the set of functions which are allotted to him, at one precise moment and during time envisaged. **Total unavailability:** A system is completely unavailability is in general a system stopped totally or saturated. **Partial unavailability:** A system is partially inalienable. **Vital unavailability:** A system is partially inalienable when one or many vital functions are not available. **Full unavailability:** A system is partially inalienable when one or many of its functions are not available. **Adaptability:** Capacity of the information system to receive easily new functionalities.

The model of the criterion "Portability" is presented as follow:
Quality goal =(Portability),
Portability=(Availability;Adaptability),
Availability=(Total_unavailability $\vee$ *Partial_unavailability),*
Partial_unavailability=(Vital_ unavailability $\wedge$ *Full_ unavailability),*
Total_unavailability=(Ø),
Availability=(Ø),
Vital_ unavailability=(Ø),
Full_ unavailability=(Ø).

2.2.2. Metric model

The Metric model is a set of metric which is used to quantify an aspect of IS. The utility of these metrics is double: on the one hand, they make it possible to anticipate the needs and to envisage the consequence resources; in addition, they can help the designer or the developer to better understanding the architecture of IS.

Description of the Metric Model
One of the requirements of our framework is flexibility. Any moment, it is possible to add or modify the criteria of IS evaluation. These two operations require sometimes the addition of new metrics. The calculation of metric implies the notion of the metric variables (Mv). The Mvs are basic measurement functions extracted from IS or data collected by the designers. The majority of existent metric collection forces the user to write code to define each new metric. In our work, we chose to develop an approach allowing reducing the effort when adding news metrics. The idea consists in fact that an Mv is used in the calculation of one or more metric, and a metric can be used in the calculation of one or more other metrics. As Mvs are pre-implemented, the value of metric is obtained by the evaluation of its expression. It is necessary to write code when the calculation of a metric requires the definition of a new Mv.

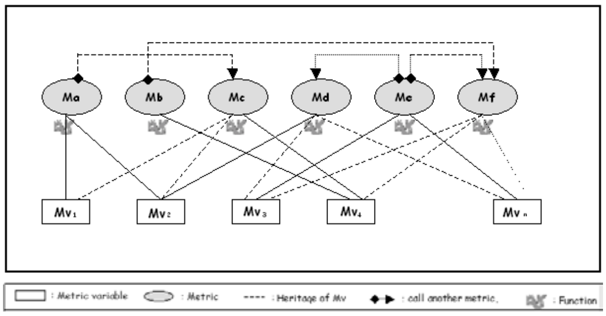

Fig. 7. Metric Model

The user can enter all the necessary information about the metric model by using the interface description component (Fig. 1), "component description of metric"). This component contains for example the following input forms:

Metric	Metric variable	Function
Name :	Name :	Name :
Description:	Description:	Description:
Function :	Function :	Path :
Components :	**Components :**	**Components :**
Metric	Ø	Ø
Vm	**Condition :**	
Condition :		

Table. 2 . Metric form input, metric variable input and measurement function input

The metric model (Fig. 7) is presented as follow:

Ma={Mv1; Mv2}, Mb={ Mv4}, Mc={Ma; Mv4}, Md={Me; Mv2},
Me={(Mv3,; Mvn}, Mf={Me ;Mb},

Example: Metric model of the Portability factor
In this example, the metric correspondents with the criteria extracted from the quality model of Fig. **4**. In order to define this metric we based on the expression of availability of software given by Akoka (Akoka 2002).

Metric of Availability		
Code	**Name**	**formula of computation**
FUA	Full unavailability	FUA=(NYFH/ NYSerH)
VitUA	Vital unavailability	VitUA= (NyVitFH / NYSerH)
TUA	Total unavailability	TUA=(NYTFH / NYSerH)

Table. 3 . Availability metric

NYSerH = A number of operating hours of the software per year, NYFH = A number of hours when at least a function is not available, NYVitFH = A number of hours when at least a vital function is not available, NYTFH= A number of hours of total stop due to a failure, NYFH>= NYVitFH >= NYTFH

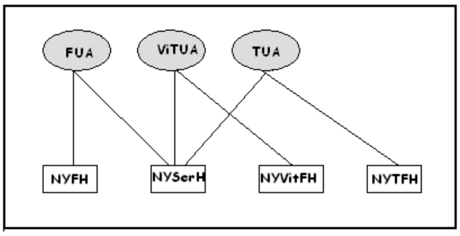

Fig. 8. Availability metric model

The metric model is:
VITUA={NYSerH; NYVitFH},
FUA ={NYSerH; NYFH}, TUA={NYTFH ; NYSerH},

2.2.3. Quality_Metric model

Description of the Quality_Metric model
As we already mentioned, the factors and the criteria are evaluated by the metrics, and each metric is composed by one or many variable metrics 'Mv'. In consequence, the validation of qualities depends on the variation of the Mv. In the Quality_Metric model we seek to present the variations of the Mv that permits the satisfaction of all qualities presented in the Quality model. The conflict removal mechanism is implemented by using the metric_variable_tradeoffs (MvT) algorithm. Given two metric_variables (Mv), Mv A and Mv B, the possible relationships between these two Mvs are as follows: 1) Neutral: An improvement to the Mv A is unlikely to affect the Mv B. 2) Direct: An improvement to the Mv A is likely to cause an improvement to the Mv B. 3) Inverse: An improvement to the Mv A is likely to cause a degradation to the Mv B.

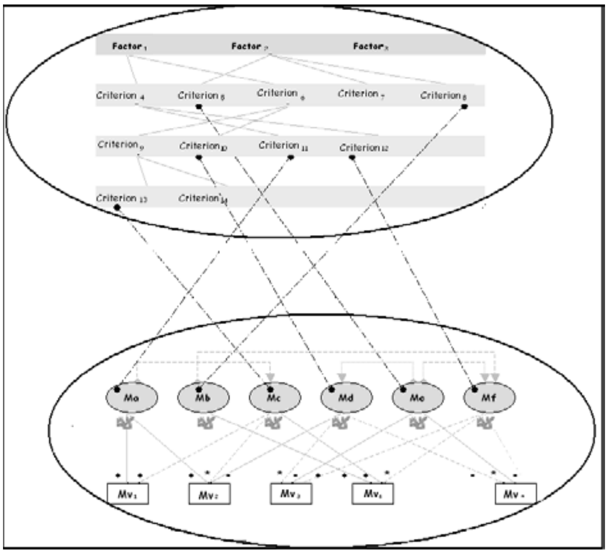

Fig. 9. Quality _Metric model

The Quality_Metric model is presented as follow:
*Criterion5={Me,(Mv3 ,-), (Mv N, *)};*
Criterion 8={Mb,(Mv 4 ,+)};
Criterion 10={Md,(Mv 2 ,-), (Mv 3 ,), (Mv n ,*)};*
Criterion 11={};
Criterion 12={};
Criterion 13={}.

The variations of the metric variables in each metric are represented by the sign of quality:
(+): more the result of the Mv is high more the criterion is satisfied;
(-): more the result of the Mv is low more the criterion is satisfied;
(*): the result of the Mv is neutral; its variation does not impact the evaluation of the criterion.

	Mv_1	Mv_2	Mv_3	Mv_4	Mv_N
Criterion $_5$			-		*
Criterion $_8$				+	
Criterion $_{10}$		-	*		*
Criterion $_{11}$	+	+			
Criterion $_{12}$			+	*	-
Criterion $_{13}$	+	*		+	

Table. 4. Variations of Mv/criterion

We can find certain couples (Criterion, Mv) which have conflicts of interests as in the case of couples (Criterion10, Mv2) and (Criterion11, Mv2) of the

Table. **4**. The result of Mv2 must be increasingly high to validate the Criterion11, but to validate the Criterion10; the result of Mv2 must have an increasingly low result. These conflicts of interests will lead the user to sacrifice certain factors of quality in order to satisfy the most important factors.

Example: Quality_Metric model of the Portability factor
In this example, the Quality_Metric model corresponds to the criteria extracted from the Quality model (Fig. 6) and the Metric model (Fig. 7).
The Quality_Metric model is:
FA={(NYFH ,-), (NYSerH,+)} ;
ViTA={(NYSerH,+), (NyVitFH,-)};
TUA={(NYTFH ,-), (NYSerH,+)}.

2.3. Evaluation Scenario

One of the objectives of our approach is to give to the expert the possibility of creating his/her own evaluation scenarios and of safeguarding them. The evaluation scenario is composed of three parts: the description of the scenario, the fuzzy interpreter and the interpretations. Before seeing in detail each of these parts we will present the meta-model scenario (Fig. 10). In the meta-model scenario, a scenario is composed of a set of qualities (quality_scenario) to be evaluated and of a set of variables (scenario_variables). A comparison scenario can call upon one or more other scenarios. Each set of qualities can contain one or more factors and/or one or more criteria. Each set of variables can contain one or more metrics and/or one or more variable metrics. The result of each scenario is saved in (Scenario_ result).

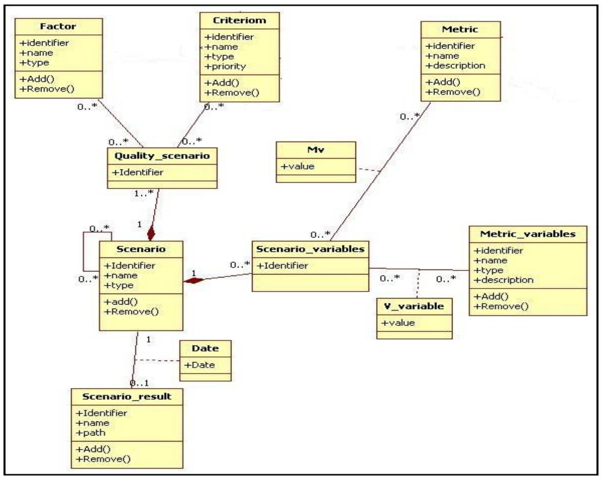

Fig. 10. The meta-model scenario

2.3.1. Description of the scenarios

The expert has the possibility of creating his/her own evaluation scenarios and of safeguarding them. The definition of these scenarios is made in the part "Description of the scenarios" of Fig. 1. An evaluation scenario must have at least one quality to evaluate. In our approach we propose two types of scenarios: simple scenarios and comparison scenarios.

- The simple scenarios allow the evaluation of one or more factors or/and of criteria, without or with conditions (variation of one or many Vm, variation of one or many metric).
- The comparison scenarios allow the comparison between two versions, of the same scenario, realised during two different times in order to seek the differences.

2.3.2. Interpreter

The role of the Interpreter is to execute the scenarios already defined in the component "Interpreter" of Fig. 1. Once the execution is finished, the interpreter passes the result to the component "Evaluation report" of Fig. 1. The Interpreter is composed of three different parts: "measurements", "fuzzy interpretation", and finally the part "interpretations".

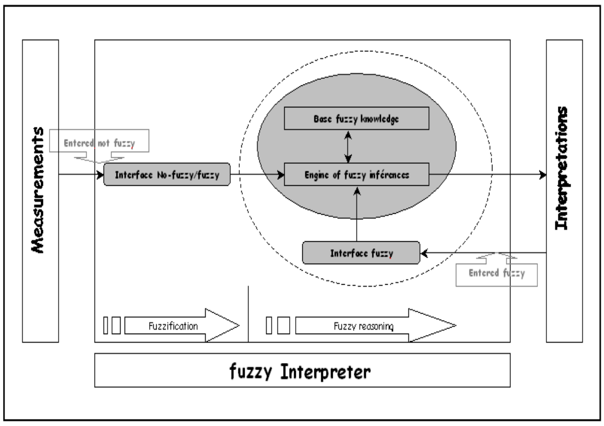

Fig. 11. The architecture of the Interpreter

2.3.2.1 Measurements

The part "Measurements" has the role:

- Research of metrics which correspond to the criteria without sub-criteria.
- Execution of metrics and stocking of their results.
- Routing the results towards the part "fuzzy interpreter".

2.3.2.2 Fuzzy Interpreter

In many quality standards, the evaluation of their levels of validation by metrics, poses the problem of the definition of the thresholds values. From which value, we can consider that criteria is very good, good, medium or weak? The difficulty is more as this value has an impact on the final evaluation of the quality factors. In order to counter this difficulty, we propose the use of a fuzzy threshold. The objective associated with a criterion will be described like a fuzzy set.
For example the criterion C11 is very strongly validated if the result of its metric is equal to or higher than 0.85, whereas having 0.85 tightened it. This fixed cut does not make it possible to consider a good evaluation. A more realistic approach consists in

defining intervals. These intervals make it possible to introduce uncertainty on the thresholds.
We will linguistically express the levels of validation of the quality factors, then to project them on [0,1] (Table. 5). Thus, fuzzy logic makes it possible to express this concept by allotting a degree of truth ranging between 0 and 1 (Fig. 12). Thus, the C11 criterion is very strong with a degree of truth of 0.45 and strong with degrees of truth of 0.50.

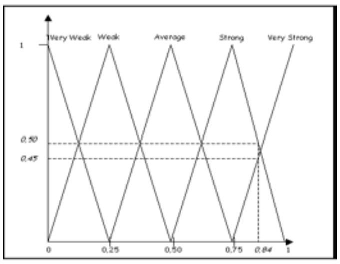

Fig. 12. Memberships functions of the fuzzy variable C11

Linguistic appreciation	**Level of validation**	**Numerical convention in [0,1]**	**Ordinal convention**
Very strong	Completely validated	[0,75..1]	A
Strong	Rather validated	[0,5..1]	B
Average	Fairly validated	[0,25..0,75]	C
Weak	Slightly validated	[0..0,50]	D
Very Weak	Very slightly validated	[0..0,25]	E

Table. 5. Expression conventions of the levels of validity of the Criteria

The aggregation of criteria
Two criteria can be crossed according to the principle of "all or nothing" or then by introducing nuances. The principle of "all or nothing" excludes very compromised between the two criteria and results in two aggregation operations: the conjunction or disjunction. The conjunction is used in the case one wishes the simultaneous satisfaction of the two criteria (logical "and"), i.e. the total evaluation cannot be better than the worst of the partial evaluations. For example in the aggregation of the criterion C9 and C10, the attitude of the decision maker (according to the quality model) implies the simultaneous satisfaction of the two criteria. That wants to say that if C4 is fairly validated (average) and C6 is completely validated (very strong), the result of the aggregation of the two criteria will be most unfavourable of both, i.e. will be fairly validated (average). Disjunction is used in the cases of redundant criteria (logical "or"), i.e. the total evaluation will be equal to the best partial evaluations. For example in aggregation of the criterion C11 and C12, the attitude of the decision maker (according to the quality model) implies satisfaction redundancy of these two criteria. That wants to say that if C11 is fairly validated (average) and C12 is completely validated (very strong), the result of the aggregation of the two criteria will be the most favourable of both, i.e. will be completely validated (very strong). A third attitude of the decision maker leaves the rule 'all or nothing' in order to introduce nuances into aggregation. If the objectives become nuances, the compromise between the two criteria becomes one of the natural attitudes of the decision maker. Thus the total evaluation is at an intermediate level between the partial evaluations. For example in the aggregation of the criterion C4 and C6, if C4 is fairly validated (average) and C6 is completely validated (very strong), the result of the aggregation of the two criteria will be rather validated (strong).
The process of aggregation treats many fuzzy sets. This phase constitutes the essential part of the reasoning in the fuzzy systems: it consists in creating a final fuzzy field which represents the combined interactions of all the rules of a fuzzy model (Cox 1997).

Example 1
In this example we seek to evaluate the criterion of the partial availability of an IS (Fig. 13). According to the model, to validate the criterion "partial availability" it is necessary to validate the two criteria "full unavailability" and "vital unavailability", (full unavailability <= vital unavailability).

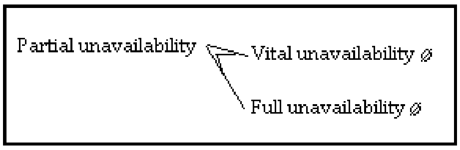

Fig. 13. The model of the Availability criterion

Initially, we must, to evaluate the two criteria "full unavailability = 0,15" and "vital unavailability = 0,16", it is possible to apply the following inference rules:

Rules 1: ***IF*** *Full unavailability is* ***Weak***
AND *Vital unavailability is* ***Weak***
THEN *The partial availability is* ***Average***

Rules 2: ***IF*** *Full unavailability is* ***Weak***
AND *Vital unavailability is* ***Very Weak***
THEN *The partial availability is* ***Strong***

Rules 3: ***IF*** *Full unavailability is* ***Very Weak***
AND *Vital unavailability is* ***Weak***
THEN *The partial availability is* ***Strong***

Rules 4: ***IF*** *Full unavailability is* ***Very Weak***
AND *Vital unavailability is* ***Very Weak***
THEN *The partial availability is* ***Very Strong***

The application of these rules gives the following results:

Rules 1 : Average validation (0,55), Rules 2 : Strong validation (0,45), Rules 3 : Strong validation (0,55), Rules 4 : Very Strong validation (0,25).
We have here four rules which can be applied simultaneously to the situation "full unavailability = 0,15" and "vital unavailability = 0,16" and in

consequence, we have four distinct answers corresponding to these rules. The process of aggregation consists in defining the final fuzzy field which represents the interactions between these rules.
We employ the technique of Min/Max aggregation: it carries out the conjunction (by the operator OR) of the consequent fuzzy sets. For each value of the variable "partial availability", we take the largest of the values of membership resulting from the four rules. The equation (Cox 1997)(1) shows how this process works, when it is applied to the whole of the model.

$$U_{sol}(x)=max_{i=[1..4]}(U_{sol}(x),U_{co}(x)_i) \quad \textbf{(1)}$$

Each new value of membership for the solution set Usol(x) (the left member of the equation) is calculated while taking the largest of the values of membership for the consequent fuzzy sets Uco(x)i and the old fuzzy sets solution Usol(x) (the right member of the equation). Rules 3 and 4 give both a response in the fuzzy set "partial availability", there are respectively 0.,45 and 0.55.

Example 2
In this example we seek to evaluate the Availability criterion of an IS. According to the model of Fig. 14, the evaluation of the criterion "Availability" depends on the evaluation of one of the two criteria "Total Unavailability" and "Partial Unavailability". The choice of the criterion is made by the designer during the phase of definition of the scenario.

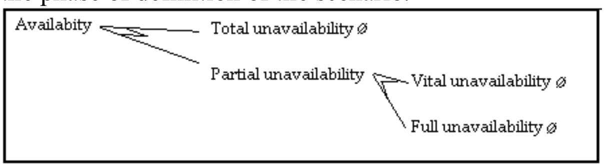

Fig. 14. The model of the criterion "Availability"

We suppose that the designer will select the criterion "Partial Unavailability". The evaluation of this criterion gives a level of validation "Strong", and then it is possible to apply the following inference rule:

Rules 5:	***IF***	*Partial unavailability is* strong
	OR	*Total unavailability is* strong
	THEN	*The availability is* weak

2.3.3. Interpretations

The role of the Interpretations consists in stocking the results of the evaluation of each criterion of the scenario and in generating the final results of the evaluation of the current factor.
The result of the evaluation of a factor **x** is in the following form:
Factor x={(criterionj ,Nv), (criterionj+1 ,Nv),..}

Evaluation Factor x ={(Nv$_i$, ((nbr Nv$_i$/nbr Ct)*100)}

I € [A,E], Nv : level of evaluation, Nbr Ct : the number of criterion pertaining to the factor in the course of evaluation.

2.4. Evaluation report

It remains to evaluate the factor "Portability", after having to evaluate all its criteria and sub-criteria.

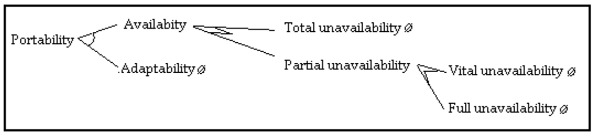

Fig. 15. The model of the criterion "Portability"

The result of the evaluation of the factor "Portability" is in the following form:

Portability={(Availability, Weak), (Portability, Average)}
Portability ={(Weak, 50%),(Average, 50%)}
50% of the criteria of the Portability are slightly validated and 50% of the criteria are fairly validated.

Portability				
Very Weak	Weak	Average	Strong	Very Strong
00%	50%	50%	00%	00%

Table. 6. Table of evaluation of the factor "Portability"

If he/she wants, the user can recover all details of evaluation. For example the user selects the criterion "Availability": *Availability ={Total unavailability, Partial unavailability}.*

Availability (Weak)				
Very Weak	Weak	Average	Strong	Very Strong
00%	100%	00%	00%	00%

Table.7: Table of evaluation of the criterion "Availability"

And if the user select the criterion "Partial unavailability":
Partial unavailability ={Vital unavailability, Full unavailability}.

Partial unavailability (Strong)				
Very Weak	Weak	Average	Strong	Very Strong
00%	00%	25%	50%	25%

Table. 8: Table of evaluation of the criterion "Partial unavailability"

3. Conclusions

Evaluation is an important component of decision-making. The skill with which an organization manages the decision making process will be a major factor in the ultimate value, effectiveness and relevance of its evaluation exercises. The development and expansion of evaluation theory and practice is at the core of several different disciplines. The ultimate goal of software engineering is to find methods for developing high quality software at reasonable cost. Some form of quality model, explicit or not, always is adopted when an evaluation is performed.

In this study we have presented a framework for understanding IS evaluation. The mechanics of the evaluation is based on quality model. This quality model comes out as a collection of desired properties which can be divided into sub properties at various levels. The last level is linked to various software metrics and measurement techniques that an organisation uses. This hierarchical model appears in more deductive way than those presented in literature.

In this evaluation approach conflicts can be removed and an indication of overall quality can be determined. In our work, the objective associated with a criterion will be described like a fuzzy set. The use of a fuzzy threshold permits a more realistic approach consists in defining intervals. These intervals make it possible to introduce uncertainty on the thresholds. This approach gives a good evaluation of IS.

References

Ågerfalk, P. Sjöström, J. Eliasson, E. Cronholm, S. Goldkuhl, G. (2002) "Setting the Scene for Actability Evaluation: Understanding Information Systems in Context", *Proceeding of ECITE 2002*, Paris , France.

Akoka, J. Wattiau, I. (2002) "La Qualité du logiciel".

Andersson, T. Von Hellens, L. (1997) "Information systems work quality", *Information and Software Technology 39*, pp. 837-844.

Basili, VR. Weiss, DM. (1984) "A Methodology for Collecting Valid Software Engineering Data", *IEEE Transactions on Software Engineering.*

Bjørn-Andersen, N. Davis, GB. (1988) "Information Systems Assessment: Issues and Challenges", North-Holland, Amsterdam.

Boehm, B. Brown, J. Kaspar, J. (1978) "Characteristics of Software Quality", *TRWSeries of Software Technology.*

Cox. (1997) "La logique floue pour les affaires et l'industrie", *International Thomson Publishing France, Paris, France.*

Cronholm, S. Goldkuhl, G. (2002) "Actable Information Systems - Quality Ideals Put Into Practice", *the Eleventh Conference On Information Systems*, pp 12-14.

DeLone, WH. McLean, ER. (2003) " The DeLone and McLean Model of Information systems success: A Ten-Year Update", *Journal of Information Systems, Spring*, Vol 19, pp.9-30.

Deutsch, MS. Willis,R. (1988) "Software Quality Engineering", Randall W. Jensen.

Farbey, BL. Targett, D. (1999) "Moving IS evaluation forward: learning themes and research issues", *The Jour-nal of Strategic Information Systems,* Vol 8 (2), pp 189-207.

Hoebeke, L. (1990) "Measuring in Organisations", *Journal of Applied Systems Analysis*, Vol 17, pp 115-122.

Irani, Z. (2002) "Information systems evaluation: navigating through the problem domain" *Information & Management*, Vol 40, No.1, pp 11-24.

Irani, Z. Love, PED. (2001) "The Propagation of Technology Management Taxonomies for Evaluating Investments in Information Systems", *Journal of Management Information Systems*, Vol 17, No.3, pp 161-177.

ISO/IEC 9126. (1988, 1991) "Information technology - Software product evaluation - Quality characteristics and guidelines for their use".

ISO/IEC 9126-1. (2001) "Quality management systems– Requirements. ISO", *Software engineering – Product quality*. ISO/IEC (2001), ISO 9001.

ISO/IEC.10746-1. (1998) "Information technology-Open Distributed Processing–Reference Model: Overview".

ISO/IEC.10746-2. (1996) "Information technology-Open Distributed Processing–Reference Model: Foundations".

ISO/IEC.10746-3. (1996) "Information technology-Open Distributed Processing–Reference Model: Architecture".

Jones, S. Hughes, J. (2001) "Understanding IS Evaluation as a Complex Social Process: A Case Study of a UK Local Authority" *European Journal of Information Systems*, Vol. 10, No.1.

Kitchenham, B. Plfleeger, SL. (1996) "Software Quality", *The Elusive Target, IEEE Software*, pp12-21.

Lederer, AL. Mendelow, AL. (1993) "Information systems planning and the challenge of shifting priorities", *Information & Management*, Vol 24, No.6, pp 319-328.

McAulay, L. Doherty, N. Keval, N. (2002) "The stakeholder dimension in information systems evaluation", *Journal of Information Technology*, pp 241-255.

McCall, J. Walters, G. (1977) "Factors in software quality", *Technical report*, US, Rome Air Development Center Reports.

Ozkan, S. Bilgen, S. (2003) "Notes Towards IS Assessment: A Comparison of two Models within the Context of the Internet", *In proceedings of the IADIS International Conference WWW/Internet*, Algarve, Portugal, Vol 2, pp.1215-1219.

Peffers, K. Saarinen, T. (2002) "Measuring the Business Value of IT Investments: Inferences from A Study of Senior Bank Executives", *Journal of Organizational Computing and Electronic Commerce*, Vol 12, No.1, pp 17-38.

Putman, J-R. (2000) "Architecting with RM-ODP", *Prentice Hall PTR, Published October*.

Remenyi, D. Sherwood-Smith, M. (1999) "Maximise Information Systems Value by Continuous Participative Evaluation", *Logistics Information Management*, Vol 12, No.1, pp 14-31.

Sinclair, D. Zairi, M. (1995) "Performance Measurement as an Obstacle to TQM", *The TQM Magazine*, Vol 7, No. 2, pp 42-45.

Smithson, S. Hirschheim, R. (1998) "Analyzing information systems evaluation: another look at an old problem", European *Journal of Information Systems*, Vol 7, No.3, pp 158-174.

Strassmann, PA. (1985) "Information Payoff: The Transformation of Work in the Electronic Age", *Free Press*, New York, NY.

Stufflebeam, DL. (2001) "Evaluation models", *New Directions for Evaluation*, 89 *(Spring)*, pp7-98.

Venkatraman, N. Ramanujam, V. (1986) " Measurment of Business Performance in Strategy Reserach: A Comparision of Appraoches", *Academy of Management Review*, Vol 11, No.4, pp 801-814.

Yahiaoui, N. Traverson, B. Levy, N. (2005) "A new viewpoint for change management in RM-ODP systems".