

A model for assessment module selection in product certification

F. Qiu,^a R. Foley^b & R. Cole^a

^a*Software Measurement Laboratory*

^b*Department of Computer Studies, Glasgow
Caledonian University, Cowcaddens Road,
Glasgow, G4 0BA, UK*

ABSTRACT

In this paper a model is presented for the selection of software product assessment modules, called 'Assessment bricks'. This is part of a development of an overall architecture for a tool (A Software Assessment Advisor) to act as an automated assistant to aid the assessment procedure.

The model proposed consists of two sub-models: a *suitability* model and an *applicability* model. The suitability model is used to select an initial set of bricks which are suitable for the proposed assessment. The criteria used in this selection include: software characteristics to be assessed; product category; degree of relevance of bricks to identified characteristics and usability of bricks in previous assessments. The selection process involves a fuzzy algorithm and a set of rules. The applicability model provides further refinement of the selection depending upon the availability of product parts (such as documentation), data and tools.

This brick selection model is being incorporated in the Software Assessment Advisor for testing and validation.

INTRODUCTION

As indicated in an earlier paper[1], a Software Assessment Advisor(SAA) is a necessary tool to assist with the task of selecting a set of relevant software attributes and a suitable subset of assessment modules at the first stage of software assessment and certification. The framework of this SAA has been developed by using Expert Systems techniques. The SAA system contains a hybrid knowledge representation, a reasoning strategy and a knowledge learning

4 Software Quality Management

algorithm. The knowledge representation takes a rule-based form for the procedural knowledge which performs the attribute and assessment module (brick) selection and a frame-based form for the declarative knowledge which includes the knowledge of product aspects and the prior assessment experiences from the case studies. The reasoning strategy concerns a rule-based deduction for the basic reasoning in the selection and a frame-based closest example identification to supplement the basic reasoning. The knowledge learning algorithm involves case-based updating on the importance weights used in the rule-based selection. Further to the development of the SAA framework in [1], the first phase of implementation includes the program modules of system scheduler, input/output scheduler, reasoning scheduler, rule-based attribute selection, frame-based case identifying and a user friendly interface in Windows for man-machine communication. The implemented part is written in C and SunView currently running under UNIX on SUN workstations.

A key next step in building the SAA is the development of an appropriate model for assessment module selection given that a set of relevant software attributes has been suggested for assessment and certification. About 100 assessment modules, called 'Assessment bricks', have been produced in the ESPRIT project SCOPE[2] as a modular approach for software product assessment. The approach aims to combine the use of standard assessment packages to suit the needs of any specific software product evaluation. A 'Brick' encapsulates a set of measurement and assessment techniques. The basic idea in this modular approach is to consider a brick as a black-box for its selection and the evaluation of its results. This allows an effective selection and application of measurement and assessment techniques from the large set of those available. The feasibility of applying bricks has been demonstrated by a substantial number of case studies selected from specific examples of industrial software development [3] [4]. One of the main problems which has arisen concerns the selection of an appropriate set of bricks for a specific product assessment. It has been indicated by a majority of the case studies that a 'brick selection assistant' tool is necessary for an appropriate and efficient selection.

A model is proposed in this paper to meet the brick selection requirements; this will complete the SAA system framework. The model is constructed by using knowledge elicited from case study experiences. It consists of two sub-models: a *suitability* model and an *applicability* model. The suitability model is used to select an initial set of bricks which are suitable for the proposed assessment. This selection is based on several criteria, a fuzzy algorithm and a set of selection rules. The applicability model provides further refinement of the selection depending upon the availability of product parts and tools.

The following sections introduce the basic concept of assessment bricks, discuss the factors involved in the brick selection, describe the brick selection model and give an illustrative example of the selection process.



ASSESSMENT BRICKS IN THE PRODUCT CERTIFICATION PROCESS

The need for a modular framework for quality assessment has been investigated in the SCOPE project. It was indicated[5]: "A large majority of software assessment approaches are either oriented towards very specific products or based on the evaluation of the development process. An assessment approach aimed at all types of software product is missing, but the reason is not the lack of applicable techniques. Many quantitative techniques applicable to the various elements of a software product have been extensively experimented. However, their application is not standardised and no global framework exists." The concept of 'Assessment brick' proposed by SCOPE[6] is aimed at the development of such a modular framework for quality assessment that could combine the use of standard assessment packages to suit the needs of any specific software evaluation.

The concept of 'Assessment bricks'

The concept of a 'brick' is of an assessment module that encapsulates a set of measurement and assessment techniques. The module can be standardised and used to build up a software assessment plan. Essentially a 'brick' is a document that describes an assessment technique, a standard way of applying it and the result it can provide when used during the assessment task. There are two categories of 'brick': basic brick and composite brick. A basic brick combines elementary metrics that can be obtained using the same technique or related techniques. A composite brick combines the results of lower level 'bricks' in order to produce a more abstract result. The composite bricks are organised as a hierarchy.

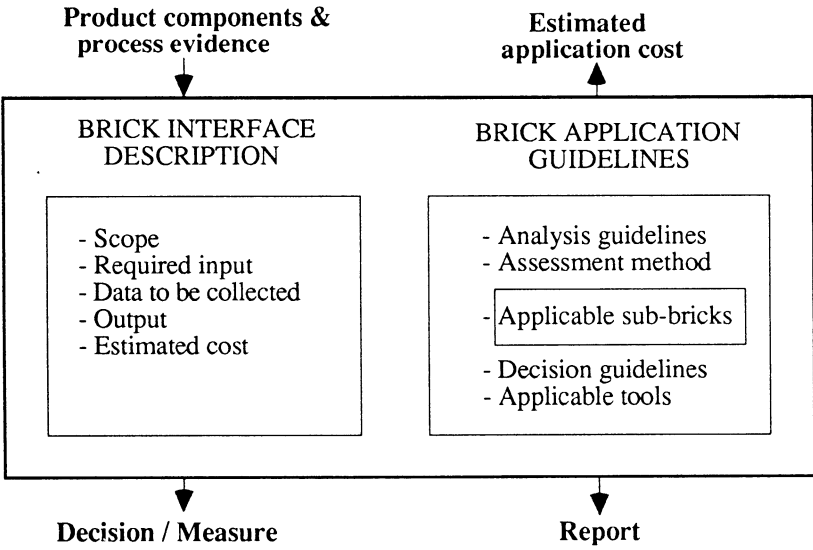


Fig. 1 The brick document framework

6 Software Quality Management

The brick document consists of two parts: the interface description and the application guideline. The interface description provides the necessary information to enable the decision whether to apply the brick, while the application guidelines contain the expertise needed to apply the assessment techniques (Fig.1 [5]). When the 'brick' is composite, the guidelines concentrate on the choice of applicable 'sub-bricks' and the way to combine results of the applied sub-bricks.

The benefits and the problems of using 'Bricks'

The bricks discussed above have been developed in the SCOPE project. Currently there are about 100 bricks in the Brick Library [7]. The feasibility of applying bricks has been demonstrated by a substantial number of case studies selected from specific examples of industrial software development. The benefits of using bricks have also been identified by the case studies. These include the offer of a flexible approach for assessment and certification, the ease of working with bricks, and the provision of sufficient and reliable results. It is felt [8] that the bricks provide a concentrated and guided assessment basis which eliminates, to a certain extent, much of the opinion based conclusions and made by the assessor. The bricks help the assessor to focus on particular aspects of the product quality (e.g. usability, maintainability), and present invaluable information, much of which would undoubtedly be omitted (or forgotten about) if no formal assessment method was being followed.

However one of the main problems which has arisen concerns the selection of an appropriate set of bricks from the complete library for a specific product assessment. User comments have included: "It takes a long time carrying out the initial selection of the applicable bricks" and "the bricks are too many to be individually analysed in detail"[9]. Due to the fact that there is such a large number of assessment bricks, it has been indicated by a majority of the case studies that a brick selection aid is necessary for an appropriate and efficient selection.

FACTORS DETERMINING BRICK SELECTION

Summarised from the case study experience there are several factors which determine whether a brick should be selected for the assessment.

Software characteristics

The bricks were produced and categorised under the six software characteristics derived from the international standard IOS/IEC 9126 [10] namely Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. It follows that where a sub-set of these characteristics are identified for a particular assessment, the range of the bricks that could be applied will be restricted accordingly.

Brick relevance

In ISO/IEC 9126 each software characteristic is described further by some sub-characteristics. For example, sub-characteristics of Functionality include Suitability, Accurateness, Interoperability, Compliance and Security. Basically the relevance degrees are summarised from case study experience according to these sub-characteristics and product categories. The degrees are categorised at four levels:

- very relevant,
- relevant,
- not very relevant and
- not relevant.

The degree of brick relevance to the software characteristics for a certain product category, which is used in the selection process, is based on the relevance degrees according to the sub-characteristics. The calculations are detailed in the next section.

Brick usability

The SCOPE case study evaluations include a measure of usability of the bricks tested. The usable status of the bricks was grouped into the following categories:

- Approved-no modification necessary,
- Usable-minor modifications necessary,
- Usable-with difficulty and
- Unusable-major modifications required.

Brick input

In the application of the bricks the specific product parts (such as requirement specification, source code) which are required as input for an individual brick need to be considered. Any lack of the required inputs will make the brick inapplicable by definition.

Required tools

The bricks can be divided into two groups based on their styles: checklist-based bricks and tool-based bricks. Each checklist-based brick contains a questionnaire, the answers to which are used in the calculation for the assessment result. The tool-based bricks require certain tool support in their application. For example, the Source Code Complexity Brick is used to assess some aspects of the software attribute Maintainability through the computation of several complexity metrics. The metrics are derived automatically from the source code using a suitable static analyser such as Logiscope[11]. Therefore these tools required in the brick application are necessary factors for the judgement of brick applicability.

8 Software Quality Management

THE BRICK SELECTION MODEL

Based on the analysis of these listed factors, the brick selection model developed consists of two sub-models: a *suitability* model and an *applicability* model.

Suitability model

The suitability model is used to select an initial set of bricks which are suitable for the proposed assessment, shown in Fig.2. This selection is based on the following criteria:

- software characteristics to be assessed;
- product category;
- degree of relevance of bricks to identified characteristics;
- usability of bricks in previous assessments.

A Synthetic Evaluation Method[12] based on fuzzy logic, and a set of selection rules, are also involved in the selection.

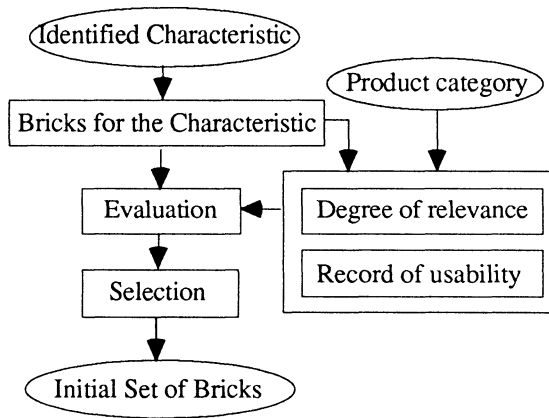


Fig.2 Suitability model in brick selection

The components of the suitability model As shown in Fig. 2 the model consists of the following components:

- 'Identified Characteristic' is the software characteristic which has been identified by the Attribute Selection step as the attribute that should be assessed in the assessment procedure.
- 'Bricks for the Characteristic' are the bricks obtained from the Brick Library, which are categorised under the identified Characteristic.
- 'Product category' involves currently 17 different categories of software systems such as Management Information System, Control System. Details are found in Appendix A of [13].

- 'Degree of relevance' stands for the relevance degree of bricks to the identified software characteristic for a certain product category and calculated from the basic relevance degrees to the corresponding sub-characteristics. Further explanations are given in the sub-section of 'The calculation of the relevance degree'.
- 'Record of usability' is the record of the brick usable status category in previous assessment as mentioned in the section on Factors Determining Brick Selection.
- 'Evaluation' is the process of producing a synthetic judgement of the brick suitability for the assessment. This process is based on a fuzzy algorithm - Synthetic Evaluation Method which is described in the following sub-section.
- 'Selection' is the process of determining an initial set of bricks which are suitable for the assessment. A set of selection rules, discussed in the following sub-section, is used for this process.

The process of brick evaluation and selection The judgement for each brick suitability for the assessment is performed by the Synthetic Evaluation Method[12] based on fuzzy logic. There are three basic elements involved in the method. These are

Factor set $F = \{f_1, f_2\} = \{usability, relevance\};$

Judgement set $J = \{j_1, j_2, j_3, j_4\}$
 $= \{very\ satisfied, satisfied, not\ very\ satisfied, not\ satisfied\};$

Single factor evaluation $\underline{S}(f), f \in F.$

The factor set includes the factors, which can be weighted, considered in the evaluation. The judgement set contains the judgement degrees used in the evaluation. The single factor evaluation identifies the evaluation set for each factor in the factor set.

For a specific brick used in previous assessments, suppose that $u_i \in [0,1]$ ($i=1, 2, 3, 4$) is the percentage of the factor 'usability' in the factor set judged at the degrees of

- 'very satisfied' (i.e. Approved-no modification necessary),
- 'satisfied' (i.e. Usable-minor modifications necessary),
- 'not very satisfied' (i.e. Usable-with difficulty) and
- 'not satisfied' (i.e. Unusable-major modifications required)

respectively. Let $r_i \in [0,1]$ ($i=1, 2, 3, 4$) take a similar meaning corresponding to second factor 'relevance' in the factor set. Then a single factor evaluation matrix for this specific brick is obtained as

10 Software Quality Management

$$\tilde{S} = \begin{pmatrix} u_1 & u_2 & u_3 & u_4 \\ r_1 & r_2 & r_3 & r_4 \end{pmatrix}.$$

Suppose that the two factors to be considered (i.e. *usability* and *relevance*) are weighted by w_1 and w_2 ($w_1, w_2 \in [0,1]$ and $w_1+w_2=1$) respectively in the evaluation of this brick, then the weight set is

$$\tilde{W} = (w_1 \ w_2).$$

Thus a synthetic evaluation of each brick on its suitability for the assessment is made by

$$\begin{aligned} \tilde{E} &= \tilde{W} \circ \tilde{S}, \\ &= (w_1 \ w_2) \circ \begin{pmatrix} u_1 & u_2 & u_3 & u_4 \\ r_1 & r_2 & r_3 & r_4 \end{pmatrix}, \\ &= (e_1 \ e_2 \ e_3 \ e_4); \end{aligned}$$

where $e_i \in [0,1]$ and

$$e_i = \max(\min(w_1, u_i), \min(w_2, r_i)) \quad (i=1, 2, 3, 4)$$

represents the rate of the suitability judged considering both factors at the degree of *very satisfied*, *satisfied*, *not very satisfied* and *not satisfied* respectively. This rate is then normalized by $(e_1' \ e_2' \ e_3' \ e_4')$, where

$$e_i' = e_i / (e_1 + e_2 + e_3 + e_4), \quad (i=1, 2, 3, 4), \quad e_1' + e_2' + e_3' + e_4' = 1.$$

This evaluation has the effect of favouring the selection of either the bigger weight or the higher judgement degree.

The selection process in the model for each brick is based on the above evaluation and involves selection rules of the following type:

IF $e_1' \geq 0.33$ or
 $e_1' + e_2' \geq 0.5$ or
 $e_1' + e_2' + e_3' \geq 0.67$;
 THEN the brick is a member of the initial set.

This process ensures that a brick is selected only when a combination of weighting values and judgement degree is high enough.

Calculation of the relevance degree As referred to previously the basic relevance degree of the brick applies to sub-characteristics and the relevance degree to the characteristic is calculated from these basic degrees. This calculation is also based on the Synthetic Evaluation Method which has been used in the evaluation process and described above.

For any software characteristic identified for the assessment, the three basic elements in the Synthetic Evaluation Method are

Factor set $F = \{f_1, f_2, \dots, f_n\} = \{sub_1, sub_2, \dots, sub_n\}$;

Judgement set $J = \{j_1, j_2, j_3, j_4\}$
 $= \{\text{very relevant}, \text{relevant}, \text{not very relevant}, \text{not relevant}\}$;

Single factor evaluation $\underline{S}(f), f \in F$.

where sub_i ($i = 1, 2, \dots, n$) in the factor set is the i th sub-characteristic considered in the evaluation. Suppose a given sub_i then $sub_{ij} \in [0,1]$ ($j = 1, 2, 3, 4$) is the percentage of brick relevance degree to ' sub_i ' judged at '*very relevant*', '*relevant*', '*not very relevant*' and '*not relevant*' level in previous assessments respectively. The single factor evaluation matrix is written as

$$\underline{S} = \begin{pmatrix} sub_{11} & sub_{12} & sub_{13} & sub_{14} \\ sub_{21} & sub_{22} & sub_{23} & sub_{24} \\ \vdots & \vdots & \vdots & \vdots \\ sub_{n1} & sub_{n2} & sub_{n3} & sub_{n4} \end{pmatrix}.$$

Suppose that the factors considered are weighted by $w_i \in [0,1]$ ($i=1, 2, \dots, n$; $w_1+w_2+\dots+w_n=1$) respectively in the evaluation. Then the weight set is written

$$\underline{W} = (w_1 \ w_2 \ \dots \ w_n).$$

Thus a synthetic evaluation to each brick on its relevance degree to the identified software characteristic is made by

$$\underline{E} = \underline{W} \circ \underline{S},$$

12 Software Quality Management

$$= (w_1 \ w_2 \ \dots \ w_n) \circ \begin{pmatrix} sub_{11} & sub_{12} & sub_{13} & sub_{14} \\ sub_{21} & sub_{22} & sub_{23} & sub_{24} \\ \vdots & \vdots & \vdots & \vdots \\ sub_{n1} & sub_{n2} & sub_{n3} & sub_{n4} \end{pmatrix},$$

$$= (e_1 \ e_2 \ e_3 \ e_4);$$

where $e_i \in [0,1]$ ($i=1, 2, 3, 4$) is the brick relevance degree to the characteristic at the levels of 'very relevant', 'relevant', 'not very relevant' and 'not relevant' respectively, which is judged considering all the sub-characteristics currently involved.

Applicability model

The applicability model, shown in Fig.3, provides further refinement of the selection depending upon the availability of product parts (such as documentation), data and tools.

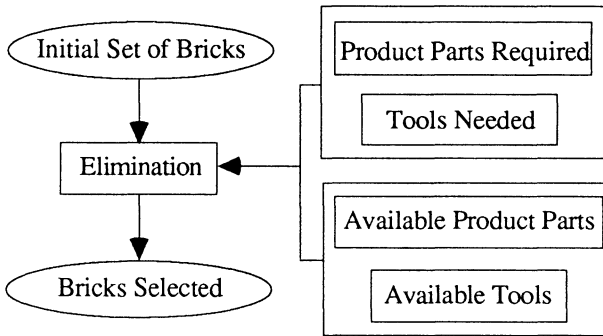


Fig.3 Applicability model in brick selection

In the model shown in Fig.3 'Product Parts Required' represents the product parts required by each brick as the input of the brick, and 'Tools Needed' involves the tools required in the brick application. The 'Elimination' part is a relatively simple procedure, and determines the final set of bricks which are suitable and applicable for the assessment. The elimination of a brick from the initial set is accomplished by, first, comparing the required and available product parts, and then comparing the required and available tools. For each brick if any required product part is not available, or any required tool is not available; then the brick is eliminated and is not a member of the final selected

brick set. Bricks in the final selected brick set are suggested as the suitable and applicable bricks for the assessment.

AN ILLUSTRATIVE EXAMPLE

We take a simple example as an illustration of the brick selection process. Suppose that the software product to be assessed is a network communication system, 'Maintainability' is one of the selected software characteristics to be assessed, and identified sub-characteristics are 'Analyzability', 'Changeability', 'Stability' and 'Testability'. The 25 bricks in the Brick Library, which are categorised for assessing 'Maintainability', need to be evaluated through the selection process. We use 'Source Code-Consistency Brick' as an example. Others follow a similar procedure.

The relevance degree of this chosen brick to attribute 'Maintainability' is calculated from the basic relevance degrees to the identified sub-characteristics. Based the case study results, for our network communication system the relevance degree of this brick to sub-characteristic 'Analyzability' is 0, 1, 0 and 0 at levels of *very relevant*, *relevant*, *not very relevant* and *not relevant* respectively, i.e. (0, 1, 0, 0). In the similar way the relevance degree to 'Changeability' is (0.5, 0.5, 0, 0), to 'Stability' is (0, 0.8, 0.1, 0.1) and to 'Testability' is (0, 0.3, 0.4, 0.2). So the single factor evaluation matrix for the relevance degree of this specific brick is obtained as

$$\underset{\sim}{S} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0.3 & 0.4 & 0.2 \end{bmatrix}.$$

Suppose for this specific product these sub-characteristics are weighted equally in the evaluation. Then

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{W} \circ \underset{\sim}{S}, \\ &= (0.25 \ 0.25 \ 0.25 \ 0.25) \circ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0.3 & 0.4 & 0.2 \end{bmatrix}, \\ &= (0.25 \ 0.25 \ 0.25 \ 0.2) . \end{aligned}$$

14 Software Quality Management

The corresponding normalised form of this solution is $(0.26 \ 0.26 \ 0.26 \ 0.22)$ and is the relevance degree of 'Source Code-Consistency Brick' to software attribute 'Maintainability' in this example at the four different levels.

Now the suitability of this brick can be evaluated considering the brick usability record and the relevance degree obtained above. The usability record (from case study data) of this brick is $0.58, 0.17, 0.17$ and 0.08 at levels of *very satisfied*, *satisfied*, *not very satisfied*, and *not satisfied* respectively. The single factor evaluation matrix is

$$\underline{S} = \begin{pmatrix} 0.58 & 0.17 & 0.17 & 0.08 \\ 0.26 & 0.26 & 0.26 & 0.22 \end{pmatrix}.$$

Again if the factors *usability* and *relevance* are equally weighted, then

$$\begin{aligned} \underline{E} &= \underline{W} \circ \underline{S} = (0.5 \ 0.5) \circ \begin{pmatrix} 0.58 & 0.17 & 0.17 & 0.08 \\ 0.26 & 0.26 & 0.26 & 0.22 \end{pmatrix}, \\ &= (0.5 \ 0.26 \ 0.26 \ 0.22). \end{aligned}$$

Its normalised form is given by $e_1' = 0.4$, $e_2' = e_3' = 0.21$ and $e_4' = 0.18$. Since $e_1' = 0.4 > 0.33$ this brick is selected as a member of the initial set.

Through the evaluation of the suitability process this brick has been identified suitable for the assessment and the next process followed is the inspection of its applicability. The 'Source Code-Consistency Brick' is a checklist-based brick. The basic input required by this brick is 'source code'. So where the product part 'source code' is available, this brick *is* applicable for the assessment. Then, as a suitable *and* applicable brick, it *is* a member of the final selected brick set and *is* to be applied for the assessment of the product maintainability. If 'source code' of the product is *not* available, this brick is *inapplicable* and is ignored for the final recommendation.

SUMMARY

The brick selection model discussed here is part of the development work of a Software Assessment Advisor which aims to aid the selection of a set of software attributes and assessment bricks in the building of a plan for individual software product assessment and certification.

The model aims at providing a suitable and efficient selection of assessment bricks from the existing Brick Library for the identified software attributes. Its construction is based on a number of case study experience on using bricks. The main components of this model include:



- a series of selection criteria considering software characteristics to be assessed, product category; degree of relevance of bricks to identified characteristics, usability of bricks in previous assessments, required input of bricks and tools needed for applying bricks;
- a synthetic evaluation of the suitability of bricks for the proposed product assessment and certification, the evaluation is made by using fuzzy logic;
- a rule-based initial selection based on the evaluation result and
- an elimination for a final decision according to availability of product parts and tools.

The brick selection model is being incorporated in the Software Assessment Advisor for testing and validation.



16 Software Quality Management

REFERENCES

1. Qiu, F., Foley, R. and Cole, R. 'Building A Software Assessment Advisor Using Expert System Techniques', in Software Quality Management (Ed. Ross, M., Brebbia, C.A., Staples, G. and Stapleton, J.), pp. 791-805, *Proceedings of the First International Conference on Software Quality Management*, Southampton UK, 1993
2. Software CertificatiOn Programme in Europe , Project No. 2151. *ESPRIT II Projects and Exploratory Actions*, Vol.4 of a series of 8, Sept.1990
3. SCOPE, Task 1.2 & 2.3 Case Studies, Chapter 2, *Technical Annexe, Issue3*, pp. 77-92 & 136-144, July 1990
4. SCOPE, Case Study Reports, 1989-1992
5. Robert, P. and Seigneur, F. 'A Modular Approach for Software Product Assessment: The Brick Concept', SCOPE Technical Report
6. SCOPE, *Certification Model - Final Report*, August 1990
7. McGinley, J. and Hunter, G. *Catalogue of Software Quality Assessment Procedures*, SCOPE Technical Report, Sept.1992
8. Hunter, G. *Case Study GPT3 Evaluation Report*, SCOPE Technical Report, Feb.1993
9. Agostani, L. and Caliman, P. *Monitoring Report for Case Study "ETNI"*, SCOPE Technical Report, March 1992
10. International Organization for Standardization(IOS), International Electronical Commission(IEC), *Information technology-Software product evaluation-Quality characteristics and guidelines for their use*, Draft International Standard ISO/IEC DIS 9126, IOS&IEC, 1990
11. Logiscope Technical Presentation, Verilog LOG/NT.A/C, March 1990
12. Geng, C.Y., Zhao, Y.Q. and Deng, Z.Z. *Fuzzy Sets and Management Decisions* , Electronic Industry Publishing House, Beijing, 1988
13. Qiu, F. *An Approach to Expert System Based Assistance in Software Product Assessment*, SML Technical Report, Oct. 1993