Arab Journal of Science
& Research Publishing
Issue (III) - Volume III - June 2017
ISSN: 2518-5780

المجلة العربية للعلوم و نشر الأبحاث
Arab Journal of Sciences & Research Publishing
AJSRP

Design and Analysis of Expert System to Measure Software Quality using EAOMT

Zainab Hussain Yahya

Department of Computer Science - University of Knowledge- Erbil - Iraq

Abstract: In this research a Computer Aided Software Engineering Tool (Case Tool) Has been designed which is called (Expert Analysis For Metrics Tool EAFMT) which is used within design activity , this tool implements one of the famous Artificial Intelligence Techniques which is Expert Systems , which automatically evaluate and enhance system quality. This tool Measures Software Quality within design Activity , Its work depends on the analysis of Class diagrams (A Class Diagram is one of the Unified Modeling Language (UML) Diagrams) , which is done by construction of a parser that extract the necessary information to find design metrics , then each metric is evaluated alone within the Expert System. This tool Aims reducing the resources used and the huge time spent in checking the programs related to design, and Also ensures that the produced system has high quality as the costumer and user expected. This tool performance was tested by applying the hole Automated Teller Machine diagram for mobile devices (which is built using Object Oriented Programming) to it , And it succeeded in analyzing and evaluating diagram Quality based on Expert Systems.

Keywords: Class Diagram, Object Oriented, Chidamber and Kemerer, Cohesion.

1. Introduction:

Artificial intelligence is one of the applications of modern software. Expert systems represent one of the artificial intelligence fields that combine artificial intelligence technology with knowledge in various fields and fields of applications. Therefore, when we study and analyze it, it is extremely important in the representation, storage, programming, and in its field capacity to directly support decisions.

Expert systems occupy the largest part of AI applications compared to other applications [2]. Figure 1 illustrates this:

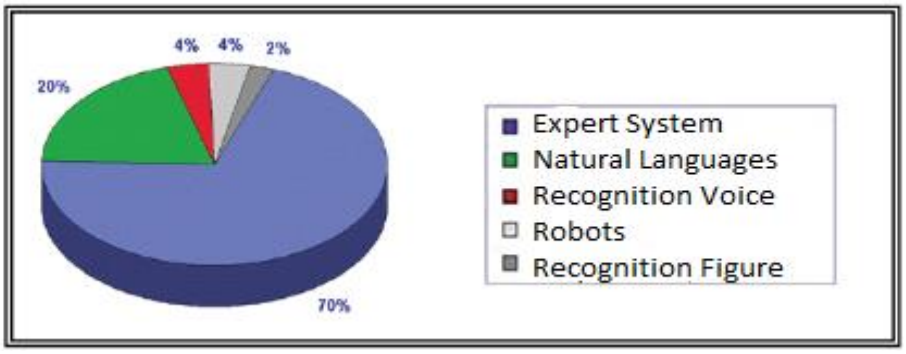

Figure (1) Relative size of expert systems as part of artificial intelligence applications [2]

Expert systems have been applied in the area of software quality standards used during the design phase, if we use the Chidamber and Kemerer standards and the expert systems together to obtain a software tool in which we can measure software quality to reduce the time and effort required to detect and correct errors before we move to stage programming Coding.

Design metrics play an important role in helping developers understand the purpose of software design and thus improve program quality and developer productivity. In this research, OO Object oriented metrics were used, including CK metrics that determine program defects and the results of these measures have an impact It is important to produce programs with high quality designs using kyanic oriented methods [3] OO.

The standards help us to identify the areas of errors and find them for software at the stage where it is developed, which reduces costs and prevents the transmission and accumulation of major errors from the current stage to the subsequent stages, which leads to large errors can result in the failure of the system [4].

2. Previous studies:

• In 2012, Anish Gupte and others discussed a unique set of measures (CK-Chidamber and Kemerer, Metrics for Object Oriented Design (MOOD, Li - Lorenzand Kidd metrics) To the lack of standards covering all aspects of object-oriented systems [4].

• In 2012, Hassan Mohamed Radwan Jameel Al-Quraishi and Waseem Ahmed Al-Quraishi improved the GDEMPER and CUMMER (CK) standards and the purpose of their improvement was to reduce the number of errors during the program design phase and the important effect of these measures in reducing the total number of errors [5].

• In 2012, Aman Kumar Sharma et al. Conducted an empirical analysis of CK and MOOD standards and found that these measures help academic software developers improve software quality during program development using the object-oriented approach [6].

• In 2010, Ayman Al Ahmar developed an initial model of an expert system that helps software project managers and software engineers to choose an appropriate software development methodology. The system was designed successfully using expert system based rules and supported by object oriented models.

3 - Expert system:

3-1 What is the expert system:

Is a branch of artificial intelligence designed to take the expertise of human experts and transfer them to the computer and store them, and through the inference engine draws the conclusions of the specific problem [2]?

3-2 parts of the expert system:

The expert system consists of four parts:

1 - User Interface: Provides a means of communication between the user and parts of the expert system through which it can enter queries about the specific problem and get smart answers to these queries.

2 - Knowledge base: contains the laws of problem solving, procedures, basic data that have to do with the area of the problem such as facts and intuition used by the human expert to solve the problem, usually the knowledge base of its stock in the form of:

Then actions If condition ...

3 - Work Memory: refers to the tasks assigned to certain data, the problem under implementation, which represent data related to the current problem to be solved.

4. Inference engine: A general control mechanism that matches the tasks assigned to the data (in the working memory) on the knowledge base in order to arrive at a specific solution or conclusion that organizes the problem data and the search mechanism in the knowledge base by applying the laws [8].

3-3 system expert:

The expert system provides us with an appropriate environment to overcome all problems by integrating new human capabilities with the efficiency of computers [8] [9] [1]. These have several benefits including:

1. Access to rare expertise, as expert systems provide knowledge and expertise in a particular field and allow others to use it easily.
2. Improving productivity, expert systems work faster and faster than the human element and reduce costs resulting from individual errors.
3. The possibility of working in dangerous situations, expert systems can work in the conditions of the line, such as high or very low temperatures and other conditions that expose people to hazards such as chemical or nuclear errors.
4. Flexibility, expert systems are flexible in solutions provided to users, based on the type of inputs take the rules used to solve problems.
5. Working under unconfirmed information.
6. The possibility of transferring knowledge to places geographically separated.
7. Do not be influenced by personal emotions and user reactions to the results are more objective.

4. Software Quality:

In spite of the urgent need to obtain high quality in the software industry in order to distinguish from others, the cost and time schedule is also one of the main factors affecting the software industry, the program that must be produced at a reasonable cost and reasonable time and be of high quality, these standards The three are essential in determining the software project.

Quality is considered one of the most important factors in the design of all business areas. Quality is defined as "meeting user requirements" [10] [2] and it "achieves excellent levels of efficiency for use [11]. [2] The market is based on quality "based on customer satisfaction. Baldrige, a quality standards organization, uses the same quality concept as" user-leading quality ", which makes user requirements a key consideration. [12] [2] By ISO9001-00 as "a measure of the degree to which the software meets the requirements of a set of echoes Essential Essentials ".

The concept of program quality needs to be further clarified. International standards have established six basic qualities of program quality [13]:

1. Functionality: The ability to provide jobs that meet the implicit needs that programs need when using them.
2. Reliability: Ability to provide services free of any failure.
3. Usability: The ability to understand, learn and use the program easily.
4. Efficiency: The ability to provide adequate performance reflecting the amount of resources used.
5. Maintainability: The ability to modify to make corrections, improvements or adaptations with other environments.
6. Adaptability: The ability to adapt to different environments without applying any procedure or, in other words, using them for other purposes.

That the quality of the software depends primarily on Design Class design, if the design is not good and not detected in the early stages of software development, much effort is required to fix this design defect. In order to solve this problem, the design standards have been used for the following stages, which help us to detect design flaws in the early stages of software development, as well as independent tools for measuring the quality of programs in Java before and after implementation. [14]

In this research, the design standards were combined as one package along with other design principles such as Cohesion, as well as linking them with the expert system. To make full use of these standards, we can improve the quality of the software production. The use of these standards gives a lot of efficiency and performance to the software.

Coherence is one of the most important standard criteria that will be explained below.

1. Cohesion

It is a measure of the strength of the relationship between the parts of the unit and focuses on the responsibilities of one unit and is applied in a program oriented. If the function is tightly linked to the elements of a particular class and is similar in many respects it is said that the class has high coherence. The cohesion is classified into seven levels (ascending order from the least cohesive to the highest cohesion) Coincidental is the assembly of units together regardless For the function of each unit such as classes, logical coherence is the assembly of parts of certain units that are logically categorized to do the same, such as the keyboard and the mouse being placed as inputs to the system. Temporal coherence is part of the units that

are assembled during processing, From the time of implementation of the program, for example, when the implementation shows us R. The user is alerted to the user for an error in execution. Procedural is a compilation of a part of the units that follow a particular sequence in execution. For example, the function of checking the file before it is opened, communicational cohesion is the assembly of parts of units that work on the same data Units that operate on the same data field, sequential assembly of units whose output is an input to other units, and finally functional coherence is the compilation of units that all contribute to a single specific task, for example, string parsing in XML. A less cohesive unit is of a cohesive coherence, carrying out one or more core functions, and the most cohesive unit is functional cohesion that performs only one basic function [4].

The CK is one of the most common OO design standards and therefore does not need to be compared with other standards. It can be built manually or automatically using a ready tool that performs the same purpose but not as efficient as manual construction.

We will explain the concept of each measurement and measurement mechanism that is carried out as follows:

1- (LCOM) Lack of Cohesion in Method

Is a measure of the difference of functions in the class in terms of variables or attributes [18] [17] [16] The degree of performance of each element of one logical function was measured and the greater the degree that the participation of the element in the performance of different functions in different functions, Reuse or keep the item on one function [19]?

There are two types of LCOM [20] and each of them is calculated as follows:

- LCOM 1

The value of P is the number of pairs of functions that do not have common variables either Q represents the number of pairs of functions that both reach the same variables within the category if the arrival of different groups of variables to the functions P value increases by one and if they share at least one variable Q value is one.

LCOM1 = P-Q

If P> Q

LCOM1 = 0

Otherwise

LCOM1> 0

This measure indicates a lack of cohesion and needs to be restructured and divided into one or more categories. The items in which LCOM is high indicate that there is an error in it and that the function is required.

• LCOM 2 [20]

(0-2), which is the ratio of the number of functions that do not reach specific characteristics to the number of total attributes within the category

LCOM2 = 1-sum (MA) / (m * a) (1)

As:

M = number of functions within the category

A = Number of variables within the category

MA = number of functions that reach variables.

The lower the value of LCOM2, the higher the cohesion and the better design of the class. It indicates that the system classifies the items well and thus increases the possibility of reuse and simplicity as this leads to a cohesive class and the cohesive type increases the degree of packaging.

The high value of LCOM2 increases complexity, reduces packaging and increases the likelihood of errors.

• LCOM3 20] [

The value of which is between 0-1 and its high value indicates greater complexity and probability of errors. If its value (0) indicates high coherence, all functions are related to all variables.

LCOM3 = (m-sum (MA / a) / (m-1) (2)

2- Depth of Inheritance Tree (DIT)

Represents the maximum length of the node to the root of the tree. This scale is represented as a tree. If we have multiple inheritance the depth is classified based on the maximum length of the path started from the node representing the root of the tree. The default depth of the root node is zero. DIT is a measure of the number of previous items that may have an impact on the current class. It is worth mentioning that the deeper tree is the most complex design because of the large number of categories and methods in them. And also indicates the depth of the tree to the large level of internal reuse in the system and this needs to balance between reuse and reduce the resulting complexity,

This measure evaluates the degree of efficiency and reuse that are relevant to ease of comprehension and testability [20] [21] [22].

3- **Number of Children (NOC)**

Is the number of subcategories that are directly related to the main category in the hierarchy. This is a measure of the number of sub-categories that will inherit the functions of the main category, as follows:

1. Large numbers of sub-species increase the likelihood of reuse (genetics is a form of reuse).

2. The greater the number of sub-items, the greater the probability of incorrect abbreviation of the main category.

3. The number of sub-categories gives an idea of the effect of that class on the design.

The DIT scale indicates the extent to which the class is affected by the characteristics of the previous varieties and the NOC scale

Illustrates the potential impact on subtypes [20] [24] [23].

3- Tight Class Cohesion (TCC)

Is a measure of class cohesion based on the direct contact of a pair of functions to be NP (C) is the total number of abstract functions in class C if n of the functions in class C,

NP (C) = n (n-1) / 2 (3)

The functions are connected directly if they are connected to common variables. For NDC (C) the number of direct connections in class C, the TCC is the relative number of directly connected functions that are expressed as follows:

TCC (C) = NDC (C) / NP (C) (4)

The TCC value ranges from [0-1] The greater the TCC value of a given category, the greater the cohesion of that class and the design will be good.

If the value of TCC is low, we will separate the product into one or more varieties to increase reuse, flexibility and inclusiveness as the variety is easier to understand and specific to its structure and function [25].

4- Loose Class Cohesion (LCC)

Is a measure of the possibility of communication whether direct or indirect connection between functions and its purpose is to detect the components of coincidence within the category?

The value of LCC = 1 means that all functions are connected to each other, whether the communication is direct or indirect and that the class will be coherent. If LCC = 0 means that there is no connection between the functions and this case is not Cohesion.

The NIC is the number of functions connected directly or indirectly

LCC = NIC / NP (5)

NP is the total number of abstract functions within class [25].

5. Analysis and Design of EAFM

The design phase is one of the basic stages of software lifecycle development. Moreover, the increasing daily complexity of the systems has created many challenges that made it difficult for the software engineer to meet the user's demand for high quality and continuous software. These challenges prompted the software engineer to pay more attention to the design process than To better understand, apply and disseminate well-known design principles, processes and practices to overcome these challenges, including quality challenges and the required quality of performance, usability, security, etc.

To help solve this problem, the EAFMT tool, whose requirements from the point of view of the software engineer, was to eliminate the problems referred to in this paragraph, was proposed.

The EAFMT tool helps the software engineer to provide information on the quality of the class schemas that contribute to increasing the comprehension of the software, especially the weak or flawed during the design phase, the collection of requirements, analysis, design and representation, and then to

improve these schemes by proposing to reorganize these Schemes to improve the quality of the drawings, simplify maintenance and increase reusability.

The work of the EAFMT tool involves four phases, illustrated by the Block Diagram in Fig. 2. The EAFMT's work is described in detail. Each of these four stages contains a number of steps and the steps of these stages may depend on each other at times. The following are the stages of the EAFMT tool:

• Phase 1: Convert the class schemas to the XMI document for detailed information describing them.

• Phase II: building a meter that determines the number of items, qualities and processes.

• Phase 3: Programming the standards in an object oriented programming method.

• Stage 4: Building the expert system that determines the quality of the product plan.

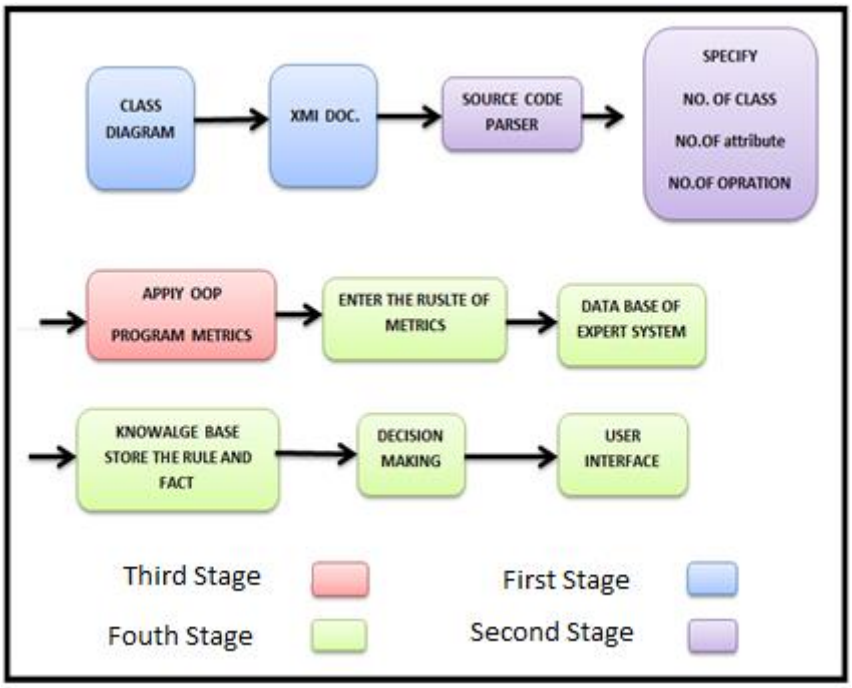

Figure (2) diagram to describe the four stages of the EAFMT tool

The threshold for each attribute is shown in the following table:

Table (1) threshold for each characteristic

Recommendations if the threshold is not exceeded	Recommendations if the threshold is exceeded	threshold	Character
It indicates that there is no reuse in the class	High re-use requires careful examination of the variety because it depends on a large number of varieties	1 to 6	**number of children**
Without recommendations	The complexity of the variety as a whole increases and there is difficulty in	1 to 5	**Depth of inheritance tree**

	calculating the behavior of the variety		
Without recommendations	The class fulfills different purposes and should be divided into several subcategories	Greater than 1	**Lack of coherence in functions**

The analysis phase of the EAFMT tool is described using the usage status chart as shown in Figure (3).

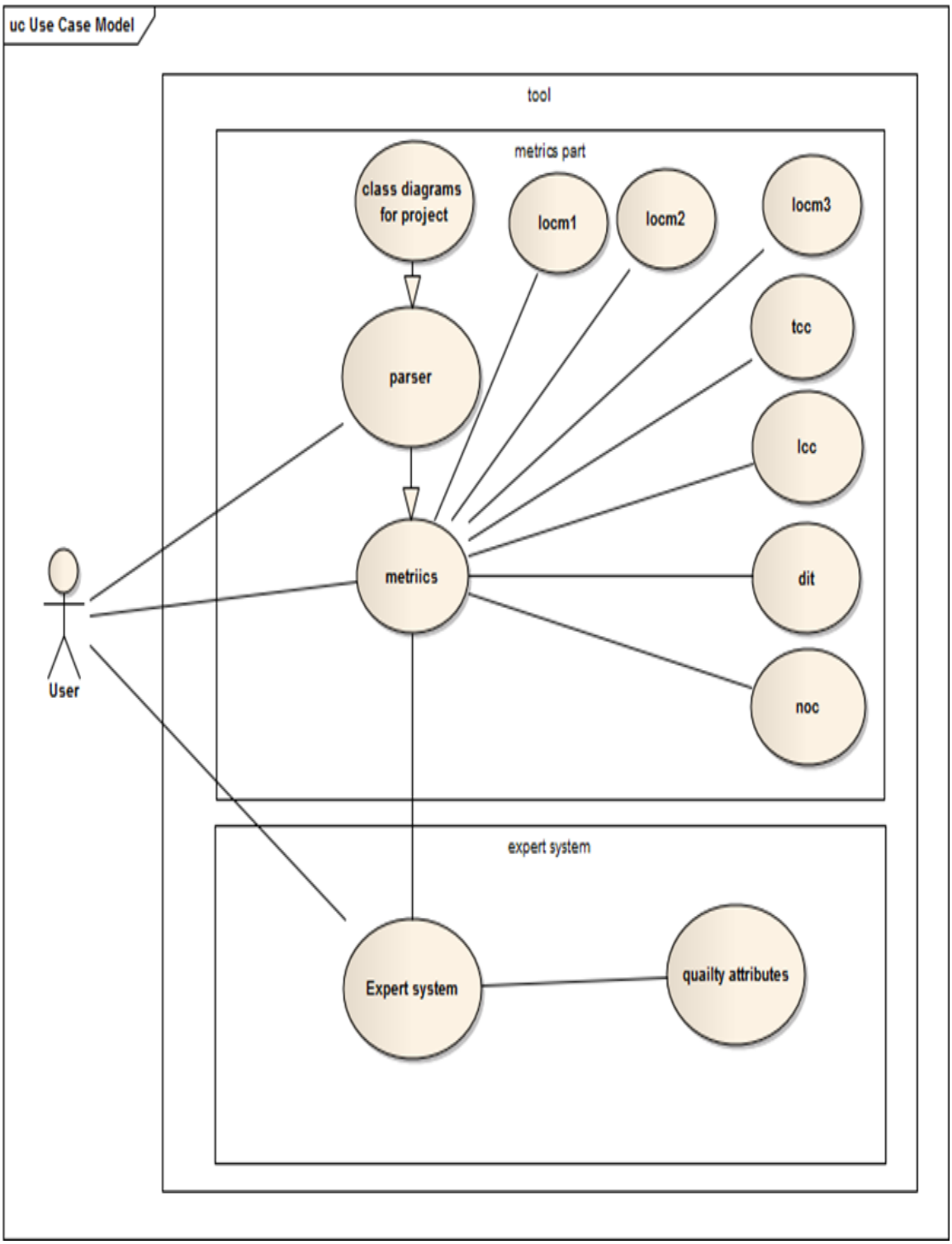

Figure (3) EAFMT usage status chart

The design phase of the EAFMT tool is illustrated using the Sequence Diagram (Figure 4).

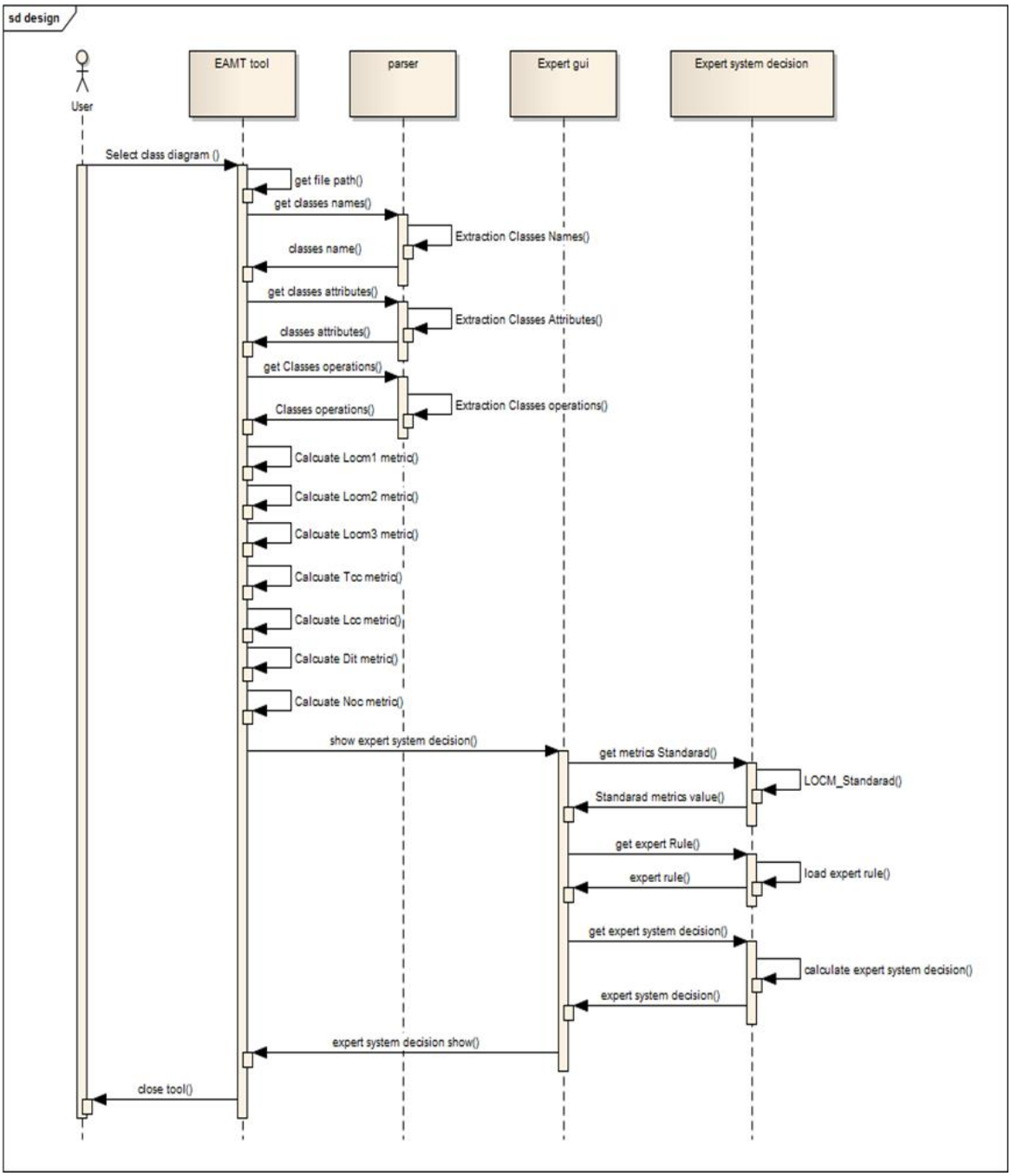

Figure (4) The sequences diagram of the EAFMT tool

5.1 Design suggested tool Algorithms

5.1.1 Algorithm of calculation quality metrics of Class diagram

The algorithm for calculating the metrics was designed according to the constructor that was built after converting the class diagram to the XMI document and the requirements of those measurements.

Step 1: Build the product schema using the STAR UML language.

• Step 2: Read the product chart as an XMI document.

• Step 3: The interpreter who extracts the information from the XMI document and converts it into tables of information on the processes, classes, and attributes of the schema.

• Step 4: Calculate the scales based on the information provided by the interpreter. Each scale is calculated according to its law and as explained in paragraph (4).

• Step 5: Enter the results of the metrics into the expert system.

The activity chart for the algorithm for calculating the quality parameters of the product diagram is shown in Figure (5) using the UML activity diagrams.

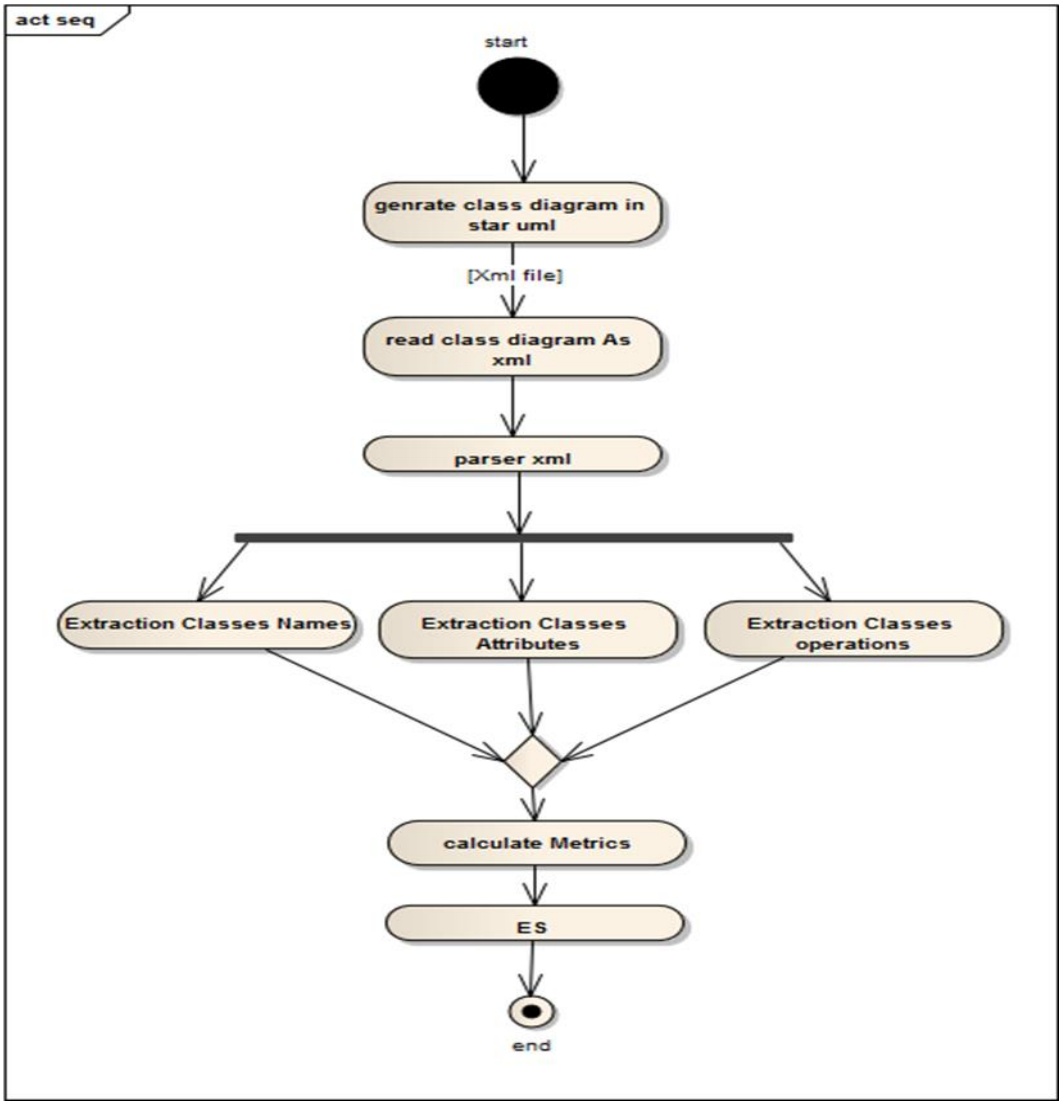

Figure (5) The activity diagram of the algorithm for calculating the quality of the product diagram

5-1-2: Algorithm of expert system

The expert system algorithm works to find the quality of each metrics as it is explained as follows:

• Step 1: Take the output of each metrics (resulting from the previous algorithm) and insert it into the expert system.

• Step 2: Create a rule for each metrics as the rule condition is the type of measure and the conclusion is the quality of that measure in the database.

• Step 3: The inference engine takes the result of each metrics and searches the knowledge base through its own laws to compare it. The forward chaing method is used to reach the final result. The data takes the results of each metrics to reach the target. :

0 <= x <= 0.2 The result of quality is very good. or

0.2 <= x <= 0.4 The result of quality is good. or

0.4 <= x <= 0.6 The quality score is average. or

0.6 <= x <= 0.8 The quality score is low. or

0.8 <= x <= 1 The quality score is very low.

Where x represents the output of each scale entered on the switch_case statement containing the previous quality comparisons, and after finding the quality output of each scale, the resulting quality values are compared to LCOM1, LCOM2, LCOM3 and output storage in variable eg y and take the output of that variable and compare it with the quality output TCC, LCC then we will get the final result of the quality of the product scheme.

• Step 4: Display the quality score on the user interface.

The activity diagram of the algorithm of the expert system is shown in Figure (6) using the UML activity diagrams.

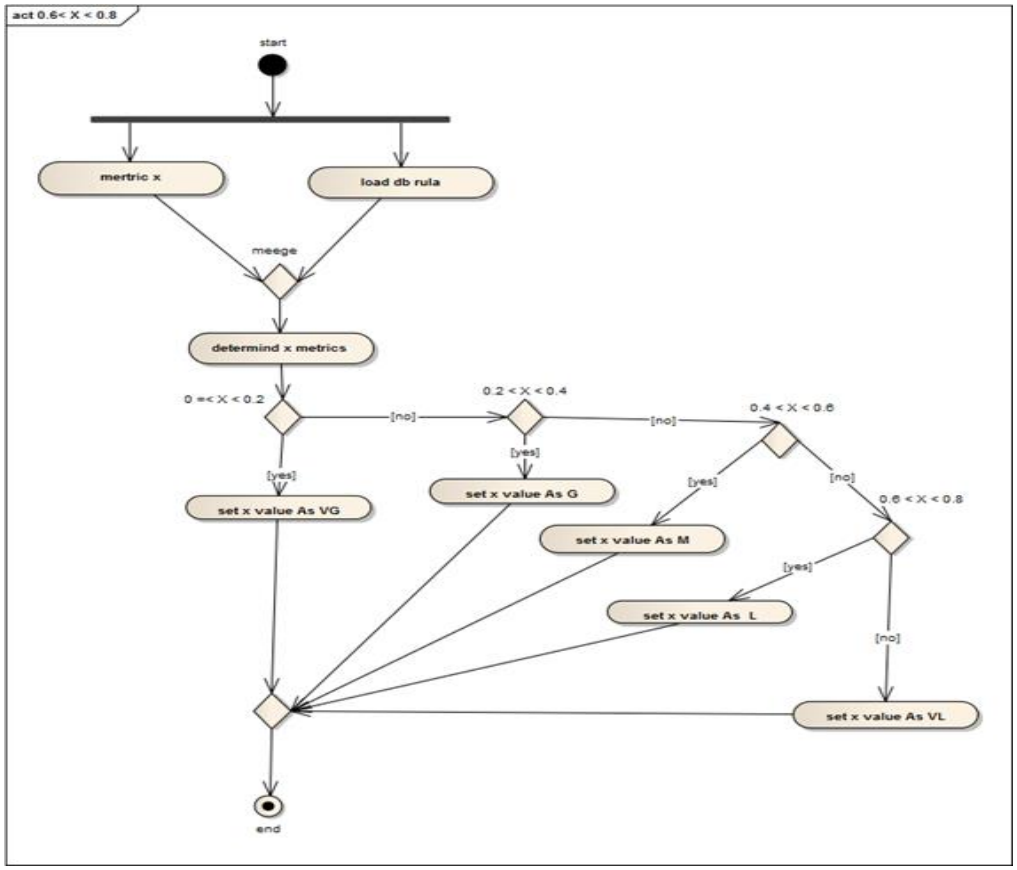

Figure (6) The activity diagram of the algorithm algorithm

The tool was tested on the ATM (Automated Teller Machine) product matrix consisting of 15 categories. After the information was extracted from the XMI document using the parameter, the quality parameters of the product diagram and the results of these measures were applied as shown in Figure 7.

Quality parameters of the product diagram and the results of these measures

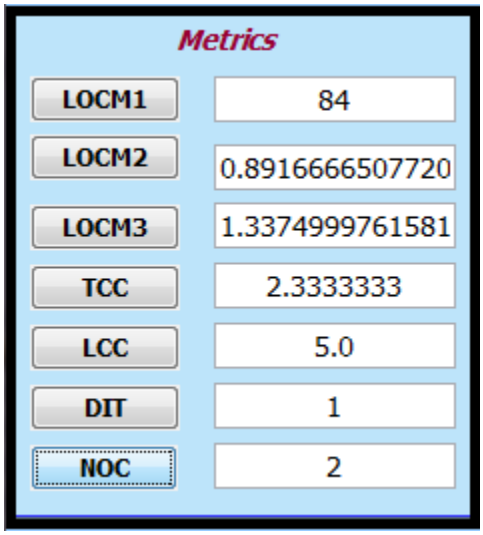

Figure (7) Results of application of quality standards of the product diagram on the ATM machine

The LOCM3, TCC, and LCC scales have been removed from the threshold and this indicates the possibility of errors in design. The DIT, NOC and its results were Very good showing that there is no complexity in the layout design and provides us with the possibility of reuse.

The results of the application of the expert system are illustrated in Figure (8) which illustrates the result of the final scheme of the product.

The results of the application of the expert system

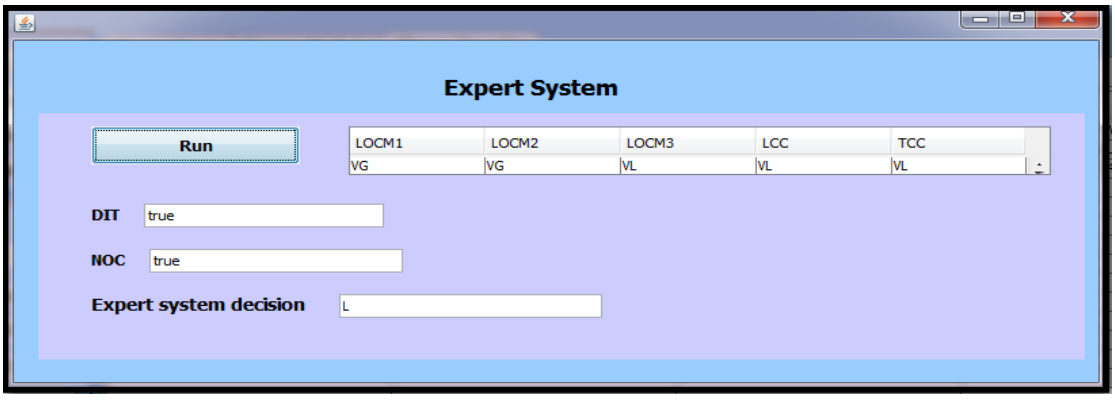

Figure (8) Results of the expert system of the ATM machine

After finding the result of the quality of each scale, the decision is taken by the expert system based on those results. The quality of the product line is very small. The plan is not reliable in generating the code for the ATM system. This will affect reuse and work. In order to improve the quality of this scheme.

6. Conclusions

The EAFMT software tool was designed to measure the quality of the product diagram by analyzing the product diagram and applying the quality standards. A number of conclusions have been recorded during the construction and testing of EAFMT tool :

1. Extensible Markup Language (XML) is very important in building any tool that needs graphical representation by converting graphic descriptions that are difficult to programmatically translate into textual descriptions that can be expressed programmatically after analysis.
2. The adoption of computer-aided software engineering tools in the area of software quality can reduce the effort and cost, and ensure that no burdens are imposed on the software development institution, if it is designed to operate automatically or semi-automatic.
3. The design of the EAFMT tool in three stages of the analysis of the class, attributes and processes can give more accurate results than taking the results of the analysis with one level of abstraction, which is the level of the project as a whole. However, the measurements used give accurate results of the planned quality, Depth of design for each category.
4. The process of linking the quality standards of the software with the expert systems gave more accurate results by combining the advantages of the standards with the expert systems in addition to that this method was not used earlier.
5. The adoption of this tool by programmers who write code will significantly improve the quality of software projects and reduce the chance of errors, and make the process of reuse and maintenance is easier to the software developer. The application of the EAFMT tool within student projects or industrial projects can give strong support to the functional requirements and non-functional requirements provided by the software.
6. The study of the possibility of finding a quality attribute to the quality standards of the scheme of the item enables the user of the tool to rely on this attribute, and reduce the reliance on software expertise, which is expensive and rare in some cases.

References:

1. عبد الله ، خبابة وعبد الوهاب ، جبّاري ، "*النظم الخبيرة ونظم دعم القرار كمدخل لاتخاذ القرار في المؤسسة*" ، كلية العلوم الاقتصادية وعلوم التسيير والعلوم التجارية ، جامعة المسيلة ـ الجزائر ، 2005 .
2. **Vizureanu , Petricğ, "*Expert Systems*" , 1st ed. , Intech , India, 2010.**
3. **Subramanyam .R and Krishnam .M.S,"*Empirical analysis of CK metrics for object oriented desigen complexity implications for software defects*",IEEE Transactions on software engineering ,Vol.29,No.4, 2003.**
4. **Gupta .A,Batra .G, "*Analyzing theortical basis and inconsistencies of object oriented metrics*" ,International journal on computer science and engineering,Vol.4,No.5,PP:803-808, 2012.**

5. Qureshi.M.R.J and Qureshi .W.A,"*Evaluation of the design metric to reduce the number of defects in software development* ",I.J.information Technology and Computer Science ,Vol.10,No.5815,Published online in MECS(http://www.mecs_press.org /),2012.
6. Sharma.A.K. ,Kalia .A. and Singh .H*,"Empirical Analysis of object oriented quality suites*" ,International Journal of engineering and advanced technology .Vol.1,No.4,PP:2249-8958, 2012.
7. AL_Ahmar .M.A,"*Rule based expert system for selecting software development methodology*" ,Journal of Theoretical and applied information technology,2010.
8. Badiru , Adedeji B. and Cheung John Y., "*Fuzzy Engineering Expert Systems with Neural Network Applications*" , John Wiley & Sons Inc. , New York-USA, 2002.
9. Abed jader A., Hasson S., *"Design An Expert System for Information Classification Applicative in The Library of Administration and Economics College At The University of Mosul*", Dissertation for the Degree of Master of Department of Management Information System, Mosul University, 2011 .
10. Nayak A., Samanta D., "*Automatic Test Data Synthesis using UML Sequence Diagrams*"Journal of Object Technology, Vol. 09, No. 2, pages 75-104, from http://www.jot.fm/issues/issue 2010 03/article2 , 2010.
11. Miliauskaite J., "*Quality of Service: Concept Analysis*",Vilnius University, Institute of Mathematics and Information,2010.
12. ISO 9001:2000 " *Quality Management systems_Requirements*",ISO:2000.
13. P.Jalate ," *Aconcise introduction to software engineering* ",Springer ,London,2008.
14. Hanad.A.,"*Software quality analysis by object oriented approach in design phase using UML*", international Journal of computer technology and electronics engineering ,Vol.1,NO.3,PP:48_52,2011.
15. Qureshi R.,Qureshi W.,"*Evaluation of the design metric to reduce the number of defects in software development*" ,Information technology and computer science ,DOI:10.5815,PP:9-17,2012.
16. Sarica S, "*UML-Based functional system testing*", Dissertation for the Degree of Master of Science (M.Sc.) in Electrical and Electronics engineering, University of Middle East Technical, 2011.
17. Salem.A &Qureshi.A,"*Analysis of inconsistencies in object oriented metrics*" ,Journal of software engineering and applications,Vol.4,No.2,PP:123-128,2011.
18. R.S.Pressman, "*Software engineering :Apractitoner's Approach*", 6th ed,Mc Graw Hill, 2004.
19. Claudio.A,Alessandro.G,"*On the reuse and maintenance of a spect oriented software :An assessment framework*",Computer science department,Federal University of Bahia,Salvador,2003.

20. **Chidamber S.,kemerer C.,"*A metrics suite for object oriented design*",IEEE transaction software engineering ,Vol.20,No.6,PP:476-492,1994.**
21. **Rosenberg H.,Hyatt E.,"*Metrics for object oriented environment*",third annual software metrics conference,USA ,1997.**
22. **Rosenberg .L.H,"*Applying and interpreting object oriented metrics*",proceeding of second system of software technology conference ,USA,2010.**
23. **kaur A.,Singh S.,Kahlon K.,Singh P.,"*Empirical analysis of CK&MOOD metrics suite*",International journal of innovation management and technology ,Vol.1,No.5,2010.**

الملخص :

تم في هذا البحث تصميم وتنفيذ أداة برمجية تدعى بـ (Expert Analysis For Metrics Tool EAFMT)تصنف هذه الاداة بوصفها أداة من أدوات هندسة البرمجيات بمساعدة الحاسوب(CASE)خلال مرحلة التصميم بالاعتماد على واحدة من اهم تقنيات الذكاء الاصطناعي وهي الانظمة الخبيرة (Expert Systems)لإيجاد طريقة جديده تلقائية لتقييم وتحسين جودة النظام.

تقوم هذه الاداة بقياس جودة البرمجيات خلال مرحلة التصميم اذ تعتمد على تحليل مخططات الصنف(Class Diagram)احدى مخططات لغة النمذجة الموحدة ((Unified Modeling Language UMLاذ تم تصميم مُعرب (Parser) لاستخراج المعلومات المطلوبة لتنفيذ المقاييس وبعدها يتم اخذ ناتج كل مقياس ليتم تقييم جودة مخطط الصنف من قبل الانظمة الخبيرة .

تهدف هذه الأداة الى تقليل المصادر المستخدمة فضلا عن الوقت الهائل المستغرق في التأكد من خلو البرامج الخاصة بالتصميم من الاخطاء. وكذلك ضمان كون النظام الناتج بالجودة العالية المتوقعة من قبل الزبون والمستخدم.

فحصت كفاءة الأداة بتطبيقها على مخطط لمشروع ماكنة الصراف الالي بصورة كاملة باستخدام مبدأ البرمجة كيانية المنحى Object Oriented Programming (OOP) اذ تم تطبيق مقاييس جودة البرمجيات على المشروع المذكور و نجحت الاداة بتحليل وتقييم جودة المخطط اعتمادا على الأنظمة الخبيرة.