

GENETIC-FUZZY PROCESS METRIC MEASUREMENT SYSTEM FOR AN OPERATING SYSTEM

¹Okpor D.Mand *²Obi J.C

¹Department of Computer Science, Delta State, Polytechnic, Ozoro

²Department of Computer Science, Faculty of Physical Sciences, University of Benin, Benin City. Nigeria.

Abstract

Operating system (Os) is the most essential software of the computer system, deprived of it, the computer system is totally useless. It is the frontier for assessing relevant computer resources. Its performance greatly enhances user overall objective across the system. Related literatures have tried different methods and techniques to measure the process matrix performance of the operating system but none has incorporated the use of genetic algorithm and fuzzy logic in their varied techniques which indeed is a novel approach. Extending the work of Michalis, this research focuses on measuring the process matrix performance of an operating system utilizing set of operating system criteria's while fusing fuzzy logic to handle impreciseness and genetic for process optimization.

Keywords

Os, Fuzzy Logic, Genetic Algorithm, Genetic Fuzzy System

1.INTRODUCTION

The operating system (OS) is the most important software that runs on a computer. Every system be it a smart phone, or other mobile device are centralized through the OS which is the overall functioning hub of the system. The basic task of the system such as switch between mobile or web pages, re-inputting and output of data, allow other peripheral essence necessary system resources are handle by the OS. The OS is responsible for the logical security of the devices or system as it integrates with other necessary segment of the system.

Every general-purpose computer must have an operating system to run other programs. Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files, directories on the disk, and controlling peripheral devices such as disk drives and printers.

For large systems, the operating system has even greater responsibilities and powers. It is like a traffic cop it makes sure that different program and users running at the same time do not

interfere with each other. The operating system is also responsible for *security*, ensuring that unauthorized users do not access the system. Operating systems can be classified as follows (Pfleeger, 2004):

- a. **Multi-user:** Provide an avenue where several user can process information or data at the same time.
- b. **Multiprocessing:** Supports running a program on more than one CPU.
- c. **Multitasking:** Allows more than one program to run concurrently.
- d. **Multithreading:** Allows different parts of a single program to run concurrently.
- e. **Real time:** Responds to input instantly. General-purpose operating systems, such as DOS and UNIX, are not real-time.

Software application are usually OS dependent because these application are written to run on certain class of OS

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world so as to describe such entities according to clearly defined rules, (Fenton and Pfleeger, 2004). In software development, measurements are conducted by using metrics. A *metric (measure)* is an empirical assignment of a value to an entity aiming to describe a specific characteristic of this entity (Fenton and Pfleeger, 2004). Measurements have been introduced into various software development process activities in order to satisfy the need to control software development and produce higher quality results.

In software development metric could be classified into three main areas, (Michalis, 2006)

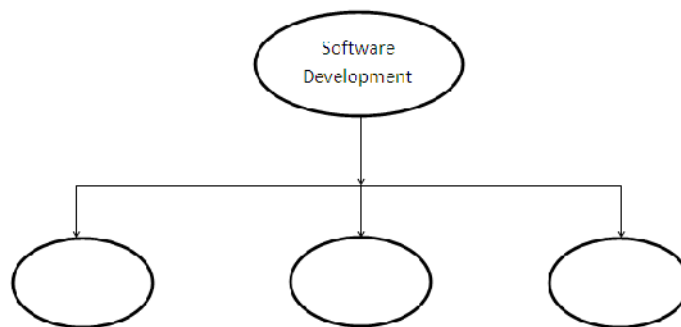


Fig.1: Architecture of Software Development Metrics

Product metrics: These metrics are related to a particular product and its attributes associated with such product such as code statement, delivered executable codes (Pressman, 2004).

Resource metrics: measuring the relevant resources for software development and their individual performance are tied to resources metric (Pressman, 2004).

Process metrics: The measures that focus on detecting problem or attaining successful best practice are known as process metric. These metric are utilized to pinpoint (Pressman, 2010).

Although Fuzzy Logic introduced by, Zadeh (1965) has been applied to most real-life situations to handle ambiguity and impreciseness of various criteria's and dataset, its application in the area of software metric is indeed novel.

Genetic Algorithm (GA) are necessary optimization and search gradient modeling natural selection process while fuzzy logic help draw up boundary classes at several point of uncertainties. This research centers on Process metric measurement of operating system utilizing Genetic-fuzzy system (fusion of genetic algorithm and fuzzy logic). This is largely based on the extension of Michalis work on software metric and measurement.

2. REVIEW OF RELATED LITERATURE

The development of large, complex, high quality and cost effective software system by integrating varied software developers working in phases (modules) simultaneously required the collection and analysis of measurable data which guide estimation, decision making and assessment, drove, Michalis (2006) to his research work titled: Software Metrics and Measurement.

The scope and objective of his work focuses on utilizing product matrix as an aid in designing, prediction and assessment of the final software product quality, provided data used for decision-making, cost and effort estimation, fault prevention, testing reduction and consequently aid in producing better software for E-commerce and E-government System.

The methodology of Michalis work geared toward developing a model embedding both the internal and external metric (attributes) on which he could assess a product for determining the quality both prior and after project manager and user perception of the product.

The limitation and unresolved problem fall square on his inability to focus on the remaining two matrices which are process and resource matrix.

An extension of process metric for measuring the performance of an operating system utilizing its underlining processes forms the basis of this research.

3. METHODOLOGY

In order for us to develop a software process matrix model for measuring the overall performance of an operating system, we must design a model embedding the basic processes associated with an operating system, Here, we are using 7 basic and major parameters) presented in Table 1.

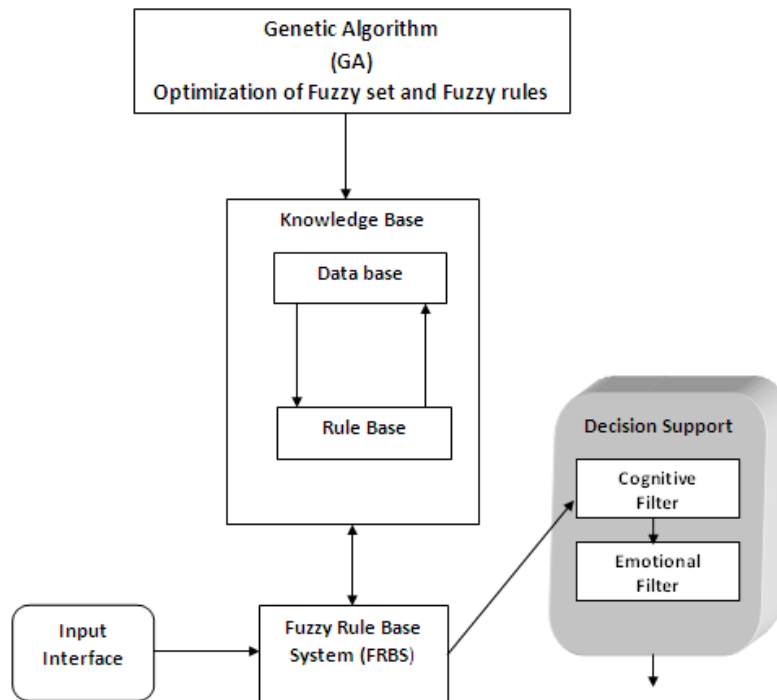
Table 1: Process matrix parameters of an Operating System (Os)

PMPC	OS Process Matrix Parameters
P01	Interrupt Handling
P02	Memory utilization
P03	Process creation/destruction/State Switching
P04	Dispatching
P05	Process synchronization
P06	Inter-process Communication
P07	Support of I/O Processes

* PMPC: Process Matric Parameters Codes

The Model is made up of several segments, all working in unionism to achieve the overall objectives.

- a. **Knowledgebase:** The linguistic variable (fuzzy set) for Os measurement and the fuzzy-if-then rules resides in the knowledgebase of our proposed model.
- b. The **Inference Engine** is the heart of the proposed model which combine the respective components to achieve the set-down objective.
- c. The **Genetic Optimizer** handles optimization of our generated membership function with the aim of arriving at a central value. The genetic optimizer comprises of three main operators namely selction: provides a means of moving toward promising region within our search space (membership functions). Individuals (lingusitic variable) with the highest fitness are selected and have the higher probability of surviving into the next generation in our search space. Mutation; a genetic operator in our proposed model exchanges one or more gene values in a chromosome, thereby preventing stagnation at any local optima.
- d. **Fuzzy Logic:** the impreciseness (vagueness) associated with the criteriaare resolved utilizing the rich facilities of fuzzy logic in the proposed model.
- e. **Decision support** predicate output result based on emotional filter and cognitive filter in the proposed model. The cognitive filter of the decision support engine takes as input the output report of knowledge-base of our model and applies objective rules to ranking. The emotional filter takes as input the output report of the cognitive filter in our model and applies the subjective rules in the domain of studies.



Output (Range of Process Matric)

Figure 3: Genetic- Fuzzy Process Metric Model for an Operating System

4. RESULT AND DISCUSSIONS

The fuzzy partition for each input feature consists of the parameters of operating system. However, it can occur that if the fuzzy partition of operating system is not set up correctly, or if the number of linguistic terms for the input features is not large enough, then some patterns will be misclassified. The rules that can be generated from the initial fuzzy partitions of the classification of are thus:

- a. Low performance (Class: C_1)
- b. Moderate performance (Class: C_2)
- c. Optimal performance (Class: C_3)

If the operating system is performing less than or equal to two (2) of the parameters of operating system process matric parameter *THEN* (C_1), if the operating system is performing three (3) of the parameters of operating system process matric parameters *THEN* (C_2) and if the operating system is performing four (4) or more parameters of the parameters of operating system process matric parameters *THEN* (C_3)

The Fuzzy IF-THEN Rules (R_i) for operating system process matric is thus:

R1: IF the operating can perform Interrupt Handling *THEN* it is in class C_1 .

- R2:** IF the operating can perform Interrupt handling and memory utilization THEN it is in class C_1 .
- R3:** IF the operating can perform Interrupt handling, memory utilization and process creation/destruction/State Switching THEN it is in class C_2 .
- R4:** IF the operating can perform Interrupt handling, memory utilization, process creation/destruction/State switching and dispatching THEN it is in class C_3 .
- R5:** IF the operating can perform Interrupt handling, memory utilization, process creation/destruction/State switching, dispatching and process synchronization THEN it is in class C_3 .
- R6:** IF the operating can perform Interrupt handling, memory utilization, process creation/destruction/State switching, dispatching, process synchronization and Inter-process communication THEN it is in class C_3 .
- R7:** IF the operating can perform Interrupt handling, memory utilization, process creation/destruction/State switching, dispatching, process synchronization and Support of I/O Processes C_3 .

A typical data set that contains the seven parameters is presented in Table 2. This shows the degree of intensity (membership) of Operating system process matrix.

Table 2: Data Set showing the Degree of membership of Operating System Process

Parameters Or Fuzzy Sets Of Operating System Process	Code s	Degree of Membership		
		Cluster 1 (C_1)	Cluster 2 (C_2)	Cluster 3 (C_3)
Interrupt Handling	R01	0.50	0.15	0.35
Memory utilization	R02	0.20	0.20	0.60
Process creation/destruction/State Switching	R03	0.10	0.80	0.10
Dispatching	R04	0.20	0.10	0.70
Process synchronization	R05	0.30	0.60	0.10
Inter-process Communication	R06	0.05	0.05	0.90
Support of I/O Processes	R07	0.00	0.50	0.50

The algorithm terminates when the stop criterion is met. The Genetic algorithm utilizes the following conditions to determine when to stop: Generations or Fitness limit. In this case, we used the number of generation (4th generation) to determine the stopping criterion.

Genetic Algorithm Inference:

- R1:** IF R01 THEN $C_1 = 0.50$
- R2:** IF R01 AND R02 THEN $C_2 = 0.18$
- R3:** IF R01, R02 AND R03 THEN $C_2 = 0.38$
- R4:** IF R01, R02, R03 AND R04 THEN $C_3 = 0.44$
- R5:** IF R01, R02, R03, R04 AND R05 THEN $C_3 = 0.37$
- R6:** IF R01, R02, R03, R04, R05 AND R06 THEN $C_3 = 0.46$

R7: IF R01, R02, R03, R04, R05, R06 AND R07 THEN C3 = 0.46

We then convert these resolved values into whole numbers and imply them to be the fitness function (*f*) of the initial generation (Parents)

R1: 50, **R2:** 18, **R3:** 38 **R4:** 44 **R5:** 37 **R6:** 46
R7: 46

Table 3: 1st and 2nd Generation Table

S/N	Selection	Chromosomes (Binary; 0 or 1)			Fitness function
		Parent (1 st Gen)	Crossover	Parent (2 nd Gen)	
1	50	110010	1 & 6	110101	53
2	46	101110	2 & 4	101100	44
3	46	101110	Mutation	101100	44
4	44	101100	2 & 4	101110	46
5	38	100110	5 & 7	100010	34
6	37	100101	1 & 6	100010	34
7	18	010010	5 & 7	010110	22

Table 4: 2nd and 3rd Generation Table

S/N	Selection	Chromosomes (Binary; 0 or 1)			Fitness function
		Parent (2 nd Gen)	Crossover	Parent (3 rd Gen)	
1	53	110101	1 & 3	110100	52
2	46	101110	2 & 6	101010	42
3	44	101100	1 & 3	101101	45
4	44	101100	4 & 5	101010	42
5	34	100010	4 & 5	100100	36
6	34	100010	2 & 6	100110	38
7	22	010110	Mutation	010100	20

Table 5: 3rd and 4th Generation Table

S/N	Selection	Chromosomes (Binary; 0 or 1)			Fitness function
		Parent (3 rd Gen)	Crossover	Parent (4 th Gen)	
1	52	110100	Mutation	110110	54
2	45	101101	2 & 3	101110	46
3	42	101010	2 & 3	101001	41
4	42	101010	6 & 4	101000	40
5	38	100110	5 & 7	100100	40
6	36	100100	6 & 4	100110	38
7	20	010100	5 & 7	010110	22

To create our 2nd and 3rd generation from the parents (1st generation) we chose the third bit from the left to be our crossover point. In the 4th generation each bold bit signifies the cross-over bits, a single bold bit signifies mutation of that bit and an italicized chromosomes signifies elitism. The best fourth generation (stopping criterion) is that with the best fitness function, 54. This implies that the clusters of the various parameters has been searched and optimized to 0.54. Therefore combination of parameters which produce a membership function $< 0.54 = C1$, $0.54 = C2$ and $0.54 = C3$ as presented in Figure 4.

Table 6:Data Set showing the Degree of membership of Operating System Process

Parameters Or Fuzzy Sets Of Operating System Process	Cod	Degree of Membership		
		Cluster 1 (C ₁)	Cluster 2 (C ₂)	Cluster 3 (C ₃)
Interrupt Handling	R01	0.50	0.15	0.35
Memory utilization	R02	0.20	0.20	0.60
Process creation/destruction/State Switching	R03	0.10	0.80	0.10
Dispatching	R04	0.20	0.10	0.70
Process synchronization	R05	0.30	0.60	0.10
Inter-process Communication	R06	0.05	0.05	0.90
Support of I/O Processes	R07	0.00	0.50	0.50
Result		Low Performa nce	Moderate Performan ce	Optimal Performan ce

5. CONCLUSION

A genetic-fuzzy system for the measurement of operating system process has been developed using the rich approach of genetic algorithm and fuzzy logic. The system will bring about optimality along with boundary-range precision. The implementation of such system for other human endeavor will handle and provide optimal solutions to human varied problems.

Reference

- [1] Christos S. And Dimitros S. (2008) "Neural Network", retrieved from <http://www.docstoc.com/docs/15050/neural-networks>.
- [2] Fenton N.E. and Pfleeger S.L. (2004), "Software Metrics: A Rigorous & Practical Approach (2nded. Revisited Printing)", London: International Thomson Computer Press.
- [3] Kasabov N. K. (1998), "Foundations of neural networks, fuzzy systems, and knowledge engineering", A Bradford Book, The MIT Press Cambridge, Massachusetts London, England, ISBN 0-262-11212-4.
- [4] Leondes C (2010), "The Technology of Fuzzy Logic Algorithm retrieved from Suite101.com/examples-of-expert-System-application-in-artificial-Intelligence.
- [5] Pressman A. (2010), "Product that improve your software engineering practices", retrieved online from <http://www.rspa.com/spi/metrics-process.html#general>
- [6] Robert F. (2000) Introduction to Neuro-Fuzzy Systems, Advances in Soft Computing Series, Springer-Verlag, Berlin/Heidelberg, 289 pages. (ISBN3-7908-1256-0)(MR1760972).
- [7] Michalis X. (2006), "Software Matric and Measurement" School of Sciences and Technology, Hellenic Open University, 23 SaktouriStr, Patras, GR 262 22, Greece, In "Encyclopedia of E-

- Commerce, E-Government and Mobile Commerce”, Mehdi Khosrow-Pour(Ed.), Idea Group Publishing, ISBN: 1-59140-799-0, pp. 1029-1036, 2006.
- [8] Stephen H. K. (2002), “Software Quality Matric Overview” Retrieved online from <http://www.informit.com/articles/article.aspx?p=30306>
- [9] Webopedia (2011), “Operating System” retrieved online <http://www.webopedia.com/TERM/O>
- [10] Zadeh L. A. (1965), “Fuzzy sets, Information and Control”, Vol.8, pp.338-353