Fault Detection in Embedded System using Rough and Fuzzy Rough Sets

BALACHANDRA PATTANAIK[1], CHANDRASEKARAN SUBRAMANIAM[2]
[1]Electrical and Electronics Engineering, [2]Computer Science Engineering
[1]Sathyabama University, [2]Rajalakshmi Engineering College, Chennai
INDIA
balapk1971@gmail.com, chandrasekaran_s@msn.com

Abstract: - The objective of the paper is to propose a fault detection technique for an embedded system using rough set classifiers. In a tightly coupled information system where the data dependency is high and embedded, the need for a strong mathematical model for fault detection in embedded system is needed. The system faults may be due to the constituent components including the hardware, application software and the operation environment at the target. The indiscernibility relation is generated due to the lack of information about the correct functionality or behavior of the components in the system. The properties or attributes of the components and the dependency between them give a tool for the detection of faulty components in an embedded application. The decision through Boolean reasoning and reduction of attributes excluding the necessary redundant components enhance the detection capability. The rough set approximation theory with the fuzzy membership functions helps not only to remove the vagueness in the detection method but also to locate the faulty components.

Key-Words: - Fault detection,Data dependency,Rough sets, Fuzzy rough sets,Embedded faults,Operation environment.

1 Introduction

A set is a collection or a number of things of the same kind that belong or are used together. It is a collection of things which are somehow related to each other but the nature of the relationship is not specified in the dictionary definitions. The antinomies or contradictions show that a set cannot be a collection of arbitrary elements [1]. In classical set theory, a set is crisp meaning that every element must be uniquely classified as belonging to the set or not. As per the bald man paradox, a small decrease in the value of an object will not turn the object into a failed object. That is the notion of the faultiness component is vague concept. This indiscernibility relation helps to construct the rough set approach. For any property, the indiscernibility with respect to that property is always sure to be a determining property relevant to the application of any predicate for which said property is a determining one. As per Dummett, the indiscernibility is an observational relation and one cannot make sense of tolerance for relations. The idea of the indiscernibility relation between two pairs of objects cannot be tolerated. The internal relation of similarity between the constituent objects and the external relation of indiscernibility between pairs can be made [2].The α and β level fuzzy rough sets are proposed and the fuzzy approximation space are studied to improve the computational efficiency in decision making process [3].The earlier Motor Current Signature Analysis (MCSA) has been carried out by the application of rough sets towards a classifier in fault diagnosis [4]. The fault detection in embedded system composed of hardware, application software and the input output devices can be enhanced through mathematical reduction of the knowledge about the possible and probable faults using rough sets and their approximation techniques. Generalizations to rough set theory regarding the approximation of fuzzy sets can be used to quantity ambiguities due to both fuzzy boundaries and rough resemblance [5]. If objects cannot be distinguished from one another, it is possible to group equal objects together. This can be done to reduce a set of data objects to its basic structure without losing any information. A group of equal objects can thus be represented by a proxy of the group. Another argument for this step is that presumably equal objects contain redundant information which can lead to inconsistency problems [6]. A rough set based approach is proposed to handle the uncertainty of the values of the parameters in the power model [7].

Rough set theory has an overlap with many other theories dealing with imperfect knowledge, e.g., evidence theory, fuzzy sets, Bayesian inference and others [8]. The fuzzy rough set attributes technique reduces the dataset without much loss in the information when compared with the conventional rough set attribute reduction (RSAR) [9].

The main focus of the paper is to apply the rough set and fuzzy rough set attribute reduction technique for the fault detection in an embedded system in terms of its detection capability for combined and mixed faults with diversified attributes and their values. The various faults are categorized according to the functionality, behavior and the target environment of the embedded system. The system faults and the hardware faults can be detected by the proposed hybrid techniques using different reduction techniques in the detection mechanism.

2 Faultiness and Vagueness of Faults

The hardware components characterized by the same information or knowledge are similar with respect to the available information. For example the hardware components like the buffer, demultiplexer and memory are characterized by the information about the data size that they can handle considered as similar based on that information but they are dissimilar in the case of number of control signals or power dissipation per operation. The dissimilarity arises due to many factors like environment and the operation performed. If one particular component exhibits different behavior through some parameters change during an intended operation, they may be classified as dissimilar under that operation. In other words, the faulty behavior can be detected by the dissimilarity of the component through the different values for the same attributes during the operation. If the values differ, then the component may be classified as either normal or faulty. Fault is an observational relation and so the fault detection method is not a single determining property but a complete group of interdependent determining properties. There are many properties or attributes through which the component fault can be detected based on the dissimilarity between the values. The typical set of hardware component faults that can be detected through the values of the attributes are Interrupt Bus Faults, Refresh faults, Cache memory faults, Synchronization faults, Input output faults and Buffer faults. The vagueness or impreciseness in selecting the property is to be dealt with utmost care so as to include the observational tolerance or measurement error. The determining property has to be correctly selected from a number of attributes so as to reduce the set of components to detect the faulty components.The fuzzy concepts can be applied to decide the nature of faults and their location based on rough set theory.

3 Rough Sets of Faults in Embedded System

Let *M* be an embedded machine with a set of all objects related to hardware and software and the operating environment. The faultiness in the objects can be considered as a relation ***'F'*** representing the vagueness or imprecision in knowing the faulty condition of the components of the machine ***'M'***. The faultiness may originate from the hardware and software category that result the machine faulty. It can be represented as

$$F := M \times M \qquad (1)$$

assuming any two instances of faults make the system faulty. *F is* the set of fault conditions due to one object in hardware and one object in software or the object in environment.

The lower approximation of a set *H* with respect to *F* is a set of all hardware modules which can be certain and positive as such with respect to *F*. The upper approximation of set *X* with respect to *F* is the set of all objects which can be possible classified as *H* with respect to *F*. The boundary region of set *H* with respect to *F* is set of all components or modules which can be classified neither as *H* or not *H* with respect to *F*. The set *H* is crisp with respect to *F* is the boundary region is empty. The set *H* is rough or imprecise if the boundary region of *H* is non empty. Hence a rough set in an embedded system is having non empty boundary region. The fault class of *'F'* is represented by *F (h)* where $h \in H$. The approximates and the boundary regions for hardware and software and operating environment of embedded system can be identified with respect to the fault class *F*. The lower approximation of the hardware components can be written as

$$\underline{F}(H) = \underset{h \in M}{U}\{F(h) : F(h) \subseteq H\} \qquad (2)$$

i.e., lower approximation of the certain of the fault F in the machine *M* with respect to the fault F is the union of all the faulty hardware elements in the embedding system. The upper approximation of the hardware fault *F* of *H* can be represented as

$$\overline{F}(H) = \underset{h \in M}{U}\{F(h) : F(h) \cap H \neq \phi\} \qquad (3)$$

It indicates that the upper approximation of faulty hardware components in an embedding system is the union of all possible hardware faults such that there is no hardware element which belongs to both the system faulty and set. The boundary region for hardware faults is the difference between the upper and lower approximation of hardware set with respect to fault class.

$$FN_F(H) = \overline{F}(H) - \underline{F}(H) \quad (4)$$

The rough sets *S* and *E* are the sets of faulty software components in the embedded software *S* and the set of faulty components due to the operational environment *E* including the battery, sensors and actuator devices and the target operating system. Their lower, upper approximations and their boundary conditions of *S* can be represented as equations from 5, 6 and 7 shown below:

$$\underline{F}(S) = \underset{s \in M}{U}\{F(s) : F(s) \subseteq S\} \quad (5)$$

$$\overline{F}(S) = \underset{s \in M}{U}\{F(s) : F(s) \cap S \neq \phi\} \quad (6)$$

$$FN_F(S) = \overline{F}(S) - \underline{F}(S) \quad (7)$$

The boundary regions for the hardware, software and environment faults are identified not only to detect but also locate them in the respective operational domain ignoring the inter dependency between some rare faulty conditions. Their lower, upper approximations and their boundary conditions of *E* can be represented as equations from 8, 9 and 10 shown below:

$$\underline{F}(E) = \underset{e \in M}{U}\{F(e) : F(e) \subseteq H\} \quad (8)$$

$$\overline{F}(E) = \underset{e \in M}{U}\{F(e) : F(e) \cap E \neq \phi\} \quad (9)$$

$$FN_F(E) = \overline{F}(E) - \underline{F}(E) \quad (10)$$

4 Fuzzy Rough Sets

The set of various faulty conditions of different components in an embedded system can be defined using approximations as in rough sets. But at the same time, the same set may also be defined in terms of membership functions and their cardinalities. The rough membership function expresses the conditional probability that an object belongs to a set given the relation and can be interpreted as a degree that the object belongs to the set in view of information about the object expressed by the relation. The fuzzy rough sets can be constructed using the upper and lower bounds with fuzzy membership functions or combining the fuzzy and rough sets based on the cutting of fuzzy sets or fuzzy relations [3].The fuzzy rough sets for the hardware, software and the environment faults are represented through equations 11,12 and 13.

$$\mu_h^F(H) = \frac{|H \cap F(h)|}{F(h)} \quad (11)$$

$$\mu_s^F(S) = \frac{|S \cap F(s)|}{F(s)} \quad (12)$$

$$\mu_e^F(E) = \frac{|H \cap F(e)|}{F(e)} \quad (13)$$

The rough sets in an embedded system can be formed by the classification of disjoint categories of objects from each of three segments shown in Figure1. The objects or components belonging to the same category, say hardware characterized by the same attributes or properties or features are not distinguishable. For example, the core processor and timer component are belonging to the same hardware category and characterized by the attributes like data size and clock frequency are indistinguishable embedded hardware components. Similarly the configuration file and the initialization file are not distinguishable based on the attributes like return value of system call and type of memory. In the environment category, the sensors and actuators are not distinguishable based on the data size and minimum threshold supply voltage. The rough sets can be formed using the lower and upper approximations with respect to many attributes and their values in each and every category say, rough sets in hardware or rough sets in the environment.

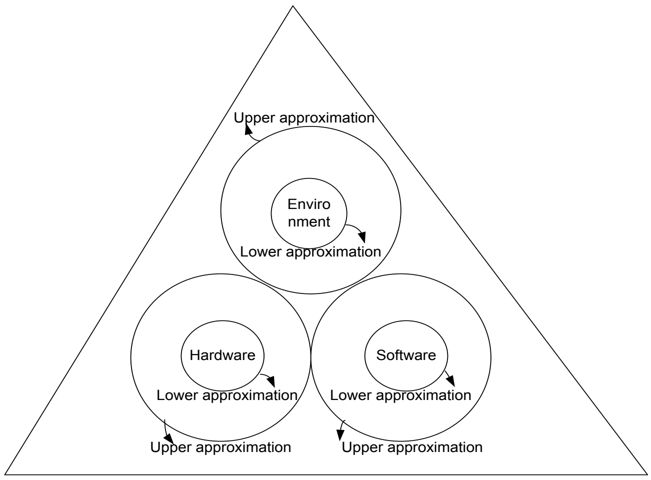

Fig. 1 Rough Sets in Embedded System

An embedded system in the context of fault diagnosis can be represented as an information system *S* which consists of information regarding attributes and their software and the environment represented by *H*, *S* and *E* respectively. It can be represented as values in the three different categories say hardware, $S = (U, A, F, \delta, \mu, \sigma)$ Where *U* is a non-empty set of finite objects or entities of hardware, software and operation environment entities.The set $(U = H \cup S \cup E)$ and *A* is a non empty finite set of attributes. The set of conditional attributes *C* and fault detection attributes *D* are distinct say,

$A = \{C \cup D\}$ and $C \cup D = NULL$ set *F*' is non empty set which is the union of the sets representing the values *V* for the attributes of the hardware components, processes *P* in the application software components and entities *N* in the operating environment such that

$$F' \cong (V \cup P \cup N).$$

The transformation functions are needed for an embedded system since it involves three categories of attributes. They map the attributes to their corresponding values in the case of hardware components, processes in the case of application software modules and the needed entities in the operating environment. The functions δ, μ, and σ are representing the value, process and device transformation functions respectively and defined as follows:

$\delta : H \times A \rightarrow V$ (Discrete values)

$\mu : S \times A \rightarrow P$ (Distributed Values)

$\sigma : E \times A \rightarrow N$ (Fuzzy values).

For example, the subset H_1 consists of attributes of the hardware components mentioned as below:

H_1 = {data_size, output_voltage, power_consumption, buffer_size} and it is known that $H_1 \subseteq H \subseteq U$.

The values for the above attributes are discrete quantities as per the data sheet specification of the hardware involved. They may be permitted with a little tolerance and if they not, they bring a binary relation thereby it is possible to detect the fault in the corresponding component. The accuracy measure or the fault detection capability with the available information can be determined by the cardinality of the attribute in the respective lower and higher approximation sets of the hardware with respect to the attribute "data size"

$$\alpha_{data\ size}(H_1) = \frac{\text{Number of Occurrences in Lower Approximation}}{\text{Number of Occurrences In Upper Approximation}} \qquad (14)$$

Many subsets can be defined using the attributes and the corresponding rough sets are used to implicitly detect the faults. The decision table for hardware components. and their attributes can be enumerated based on the processor, random access memory and buffer and so on. Due to the existence of the three different subsystems or categories of components in an embedded system, the number of attributes and their values are infinite. That too in the components and external components are to be derived. A discrete distributed fuzzy rough set theory has to be modeled so as to detect the faults in the system. The indivisible relationship R_H, R_S and R_E can be declared for hardware, software and the entities separately as shown below where h_1 and h_2 are elements in the hardware set which are indistinguishable as per the attribute '*a*' belongs to a set *A* through the mapping function δ. Similarly the software components s_1 and s_2 are elements in the software set which are indistinguishable as per the attribute '*a*' belongs to a set *A* through the mapping function μ and the environment components e_1 and e_2 are elements in the environment set which are indistinguishable as per the attribute '*a*' belongs to a set *A* through the mapping function σ. Which are represented as follows: faulty conditions, the values are nondeterministic and random in the case of hardware but the software components may yield either faulty or not based on the function calls with the underlying real time operating system. The environment entities in the embedded system may be many intermediate values so it will into fuzzy rough sets. Hence a hybrid rough sets comprising of individual components, distributed

$$R_H : Indivisible(R_H) = \{(h_1, h_2) \in H \times H \mid a \in A, \delta(h_1, a) = \delta(h_2, a)\}$$

$$R_S : Indivisible(R_S) = \{(s_1, s_2) \in S \times S \mid a \in A, \mu(s_1, a) = \mu(s_2, a)\}$$

$$R_E : Indivisible(R_E) = \{(e_1, e_2) \in E \times E \mid a \in A, \sigma(e_1, a) = \sigma(e_2, a)\}$$

The rows in the decision table cannot be differentiated based on any attribute subset using the relevant relationships. It is possible to arrive at *H/D*, *S/D* and *E/D* where *H*, *S* and *E* are categories of sets of embedded components and *D* is the set of detection attributes. For all three categories, the component weight age in the fault detection strategy may be calculated as the weightage of an attribute is calculated based on its cardinality and similarity of the attribute with other attributes in each category. That is the number of times; the attribute is uniquely identified in its own category and other categories and its relationships. The sets *H/D*, *S/D* and *E/D* and the $U/C_{11}...U/C_{1N}$ where $C_{11} = U - \{a_{11}\}$ can be determined. The overall fault detection capability or accuracy of the rough set approach can be calculated as S (h, s, e) = $\alpha (C_{ij})(\delta + \mu + \sigma) / \text{Value}(C_{ij})$

If the Data size (a_{11}) = 8 bits, output (a12) = 2.4 V, Power consumption (a_{13}), Buffer size (a14) = 1 byte, the values of attributes for the normal and faulty conditions are known from the Tables 1.a and 1.b respectively.

Table 1.a: Hardware Components Normal Data

H/W Component	Data Size	O/P Volt	Power Consumption	Clock
C_1 Core	8	2.4	0.25 W	60 MHZ
C_2 RAM	16	2.4	0.25 W	48 MHZ
C_3 Port	16	3.6	0.1 W	48 MHZ
C_4 Timer	8	2.4	0.1 W	60 MHZ

Table 1.b: Hardware Components Faulty Data

Component	Very Large	Low	Minimum	Typical	Exact
Buffer	0.1	0.3	0.6	0.8	1
Crystal	0.1	0.2	0.6	0.8	1
configuration	0.1	0.3	0.5	0.8	1
Function call	0.1	0.2	0.5	0.9	1
RTOS	0.1	0.2	0.6	0.9	1

The critical hardware objects in an embedded system are identified that are based on the application and their conditional attributes are tabulated as in Table 2.The values of these attributes are scaled into boolean values to deicide the first level of fault detection within the hardware components family. The values are so selected that they may be within the operating conditions as per the device electrical and thermal characteristics according to the design data sheet. The determining property is selected as the minimum current rating since power consumption is the foremost indication for any embedded system operation irrespective of other parameters. The selection of the set of conditional and determining attributes is application and domain specific. In the case of embedded domain and fault tolerant application, the threshold values for all the attributes and their critical relations are considered to bring the optimal size of individual sets. For hardware components, the Boolean relationships as well as reasoning are considered to detect the component fault but in the case of environment entities, the fuzzy relations and reasoning are applied to locate correctly the faulty items.

Table 2: Reduced Rough Set for Minimum Current in Hardware

Objects	Operating Voltage	External Interface	Temp. Sensitivity	Duty Cycle	Min. Current
Core	Low	no	yes	fixed	Low
Flash RAM	Low	yes	yes	fixed	Low
Timer	Low	yes	yes	custom	Low
Port	Low	yes	yes	fixed	High
ADC	High	yes	yes	fixed	High
DAC	High	yes	yes	fixed	High
Rx	High	yes	yes	fixed	High
Tx	High	yes	yes	fixed	High
PWM	Low	no	yes	custom	No

Table 3: Reduced Rough Set for Fuzzy Faults in Environment

Object	Size	Positive Region	Reducts
Battery	2	1	{level, current}
Board Sensor	2	1	{level, size]
Convertor	2	1	{size, EMI}
PCB	2	1	{output type, size}
Sensor 1	3	1	{repairable, output type, current}
Sensor 2	3	1	{repairable, current, EMI}
Activator1	3	1	{interoperability, output type, current}
Activator2	3	1	{interoperability,current,EMI}

The rough sets are explored with a master table for each category of components. The critical attributes which affect not only the functionality but also the behavior of the components are considered in the exploration system. The rules and the reduced sets for the system are found out based on the determining attributes. In hardware one deciding attribute was considered whereas in the application software category, two attributes are considered to detect the faults. The reduced rough set for fuzzy faults in the environment and reduced rough set for fuzzy faults in the application software components are shown in Table 3 and Table 4 respectively. The corresponding cut sets are derived from the master and sub table where the threshold values for the operating system components are obtained from thevendor specification sheet. The detection results are separated from the table and it is flexible to partition the master table into many for further of fine tuned detection processes as shown in Figure 2.The decomposition trees for the rough sets in individual category of components of the embedded system are shown in Figure 3 & Figure 4.

When the entire system is considered as a single entity, the lower and upper approximations of the set covering all objects in three categories are determined. The memory related fault is dominant in the hardware side whereas the OS compatibility issues are the main reason for system faults in the software side as shown in Table 5. The environment entities are easily. detectable in the case of rough sets approach as per the determining attributes and conditional attributes selected in the problem. The number of internal nodes, leaf nodes and total nodes needed for the fault detection are shown in the Figure 5. The number of elements in the rule and reduce set for the discussed embedded system are shown in Figure 6.

ISBN: 978-1-61804-019-0

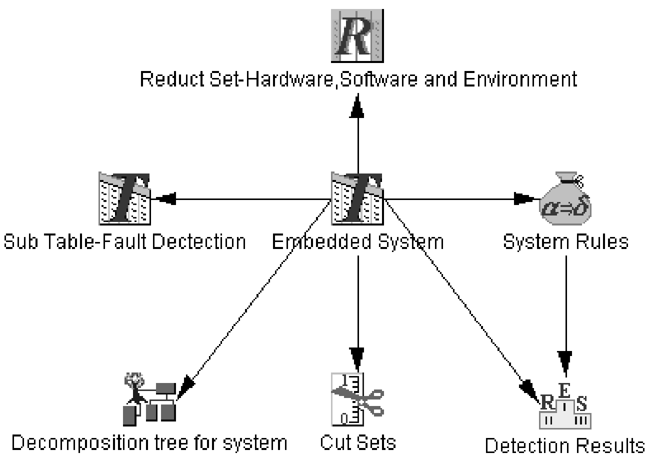

Fig.2. Embedded system Faults as Rough Sets

Table 4: Reduced Rough Set for Fuzzy Faults in Application Software Components

Object ID	Size	Positive Region	Reducts
1	2	1	{Output type, Current}
2	2	1	{level, Output type}
3	2	1	{level, Current}
4	2	1	{Current, EMI}
5	2	1	{Output type, Size}
6	2	1	{Size, Current}
7	2	1	{Interoperability, Size}
8	2	1	{Level, Size}
9	2	1	{Size, EMI}
10	3	1	{Repairable, Output type, EMI}
11	3	1	{Repairable, Interoperability}
12	3	1	{Level, Repairable, Interoperability}
13	3	1	{Interoperability, Output type}
14	3	1	{Level, Interoperability, EMI}

Table 5: Reduced Rough Set for Embedded System

Ob ject ID	Siz e	Positive Region	Reducts
1	2	0.625	{OS Compatibility, RTOS Problem}
1	2	0.625	{RTOS Problem, Device Support}
1	2	0.625	{OS Compatibility, Device Support}
1	2	0.625	{Os Compatibility, Memory Level}
1	2	0.625	{Memory Level, Device Support}

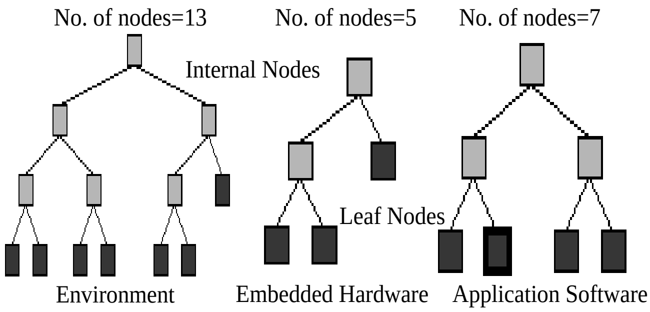

Fig.3.Decomposition Tree for Individual Rough Sets

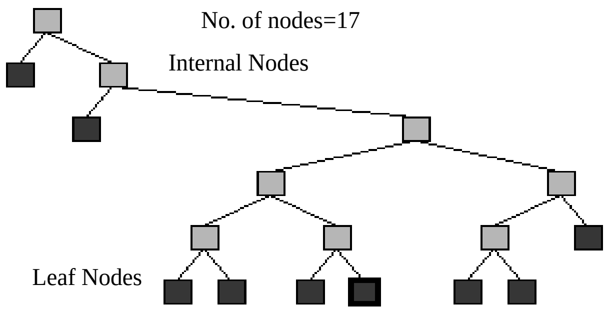

Figure 4: Decomposition Tree for Embedded System

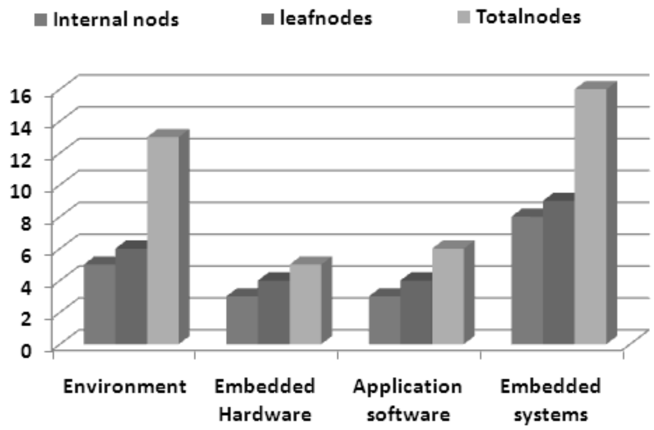

Figure 5: Number of nodes in Embedded System

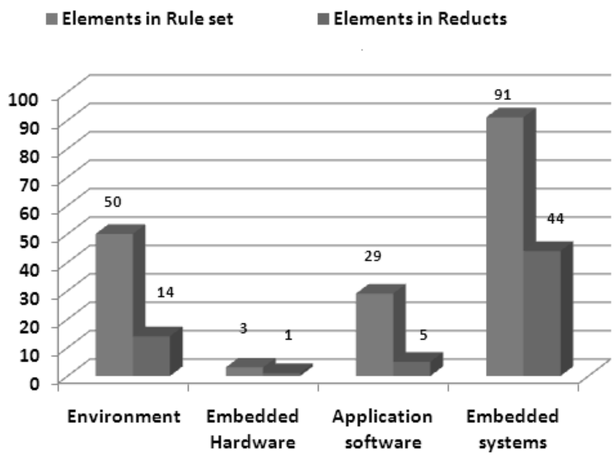

Figure 6: Number of Elements in the Rule and Reduce Set

5 Conclusion

The fault detection in an embedded system addresses various hardware components with configurable parameters and software application modules with standard functions in the library files and also the fuzzy values of the battery and connectors. This integrated approach in manipulating the rough sets for each category of components are considered in the paper. The fuzzy rough sets and the rough sets are used to detect the nature of faulty component or process depending on the decision parameters. Since the attribute sets and the number of states in which a normal or a faulty system exists increase the complexity of the detection process. The importance or weightage of each attribute and the detection capability from the reduced attribute sets can be determined using the given approach. It can be scaled to many configurations of the system including multiple cores and mixed signal components with or without redundant components under a real time operating system library functions. The limitation of the work is that it can be tuned to only application specific designs not as a general framework for fault diagnosis of the system. In the future, the exact location of the faults either due to application code and or due to unreliable third party hardware components will be identified to activate the necessary dynamic modules to realize a fault tolerant embedded system.

References:

[1] Zdzislaw Pawlak and Andrzej Skowron, "Rudiments of Rough Sets", *An Information Science International Journal,* pp.1-49, June, 2006.

[2] Achille C. Varzi, "Vagueness, Indiscernibility, and Pragmatics Comments on Burns", *The Southern Journal of Philosophy,* vol. 33, pp. 49-62, 1995.

[3] W. N. Liu, J.T. Yao and Y.Y.Yao, "Rough Approximations Under Level Fuzzy Sets", *Rough Sets and Current Trends in Computing: 4th International Conference*, Uppsala, Sweden,2004, pp. 1-6.

[4] E.L.Bonaldi, L.E.Borges da silva, Et.al, "Using Rough Sets Techniques as a Fault Diagnosis Classifier for Induction Motors", *28th Annual Conference of the IEEE Industrial Electronics Society,* Sevilla, Spain, 2002, pp. 1-7.

[5] Debashis Sen, Sanker K, Pal, "Generalized Rough Sets,Entropy and Image Ambiguity Measures" *,IEEE Transactions On Systems, Man, And Cybernetics,* vol.39, pp.117-120, February 2009.

[6] Maren scheffel, Martin Wolpers, "Analyzing Contextualized Attention Metadata with Rough Set Methodologies to Support Self-regulated Learning",*10th IEEE International Conference on Advanced Learning Technologies*, Germany, 2010, pp.125-128.

[7] Shu Xiao and Edmund M-K. Lai, "A Rough Set Approach to Instruction-Level Power Analysis of Embedded VLIW Processors", 2005, pp.1-5.

[8] Zdzistaw Pawlak, "Rough set theory and its applications", *Journal of Telecommunications and Information Technology,* pp.7-10,March, 2002.

[9] Richard Jensen and Qiang Shen, "Fuzzy-Rough Sets for Deseriptive Dimensionality Reduction", *Fuzzy Systems* 2002, *FUZZ-IEEE'02,* Honolulu, HI, USA, 2002, pp. 29-34.