View metadata, citation and similar papers at core.ac.uk brought to you by CORE
provided by Bulletin of Electrical Engineering and Informatics

**Bulletin of Electrical Engineering and Informatics**
Vol. 4, No. 3, September 2015, pp. 210~216
ISSN: 2089-3191 

# The Error Control Methods of Information System in Sensor Networks

**Liu Wei**[1], **Lu Xing Hua**[2]
[1]Guangzhou Panyu Polytechnic, Guangzhou, 511483 P.R. China
[2]Huali College Guangdong University of Technology, Guangzhou Zengcheng, 511325, P. R. China
E-mail: 157207002@qq.com, 44680189@qq.com

***Abstract***

*Information System Error Discovery took a lot of interests of information system experts. Indeed, this concept has been interpreted in more than one way. This paper describes the connections between 1) the Field-to-Scope Proportion and the Identifying and Managing way of Error Discovery and 2) the (Transmission, Influence and Implementing) TII model and Error Scope. The main goal of the work described here, is to seek deeper understanding of Error Discovery in general and, at last, to find easier ways of confirming it. Second, in this paper we try to model the TII evaluating using Work Expressing System which is Relational Graph. We also use this process model to present the relationship between Error Discovery, TII, (Field-To-Scope Proportion) FSP and Error Scope. After explaining the relationship between Semantic Error and Error Discovery, we improve the Relational Graph model by adding standard. We then offer instructions of the Relational Graph.*

***Keywords***: *Error Discovery; Process Logic; Work Expressing; Relational Graph*

## 1. Introduction

Error discovery represents one of the important quality attributes that could affect run-time behavior and system design. It is defined as the easiness of creating test standard for the system and its components, and to execute these tests in order to determine if the standard are met. Good Error Discovery measure makes it easier to isolate errors in a timely and effective manner. It represents area of concern that has the potential for application wide impact across the development phases. The importance of this concept attracted the researchers to relate it to other ordinary concepts to provide an estimate of the Error Discovery in an easier and a faster way. Error discovery received different interpretations from the researchers [1-4]. These interpretations suggested some relationships between the Error Discovery and other related concepts. Freedman [5] suggested that in order to make the behavior of the component both observable and controllable, the Error Discovery could be directly related to the inputs and outputs of a given program component. W. Junior and colleagues [6-8] defined the test-ability as the ease with which program errors may be exposed if they are present in the concerned program. They refer to this definition as transmission, influence and implementing (TII) model [9]. They found that it is difficult and expensive to measure program Error Discovery with this technique. Therefore, they tried to relate it to other easier concepts that could provide an indication to the program Error Discovery and could guide the researchers to the best way to design and develop the information system product.

To fulfill this role, I. M. Atakli [10] explored the relationship between program Error Discovery and other concepts and made ordinary observations. Their proportion was restricted to the creation of idealized model of programs as functions. To show this functional view of information system, they used a framework in which they related the program Error Discovery to the Dynamic Scope Proportion (FSP) and semantic error scope. For clarity and flow of information, the concepts are revisited.

In this paper, we introduce a new Background Way for Error discovery. It aims to respond to the following limits of traditional methods: they do not cover all Error Discovery way and they lack flexibility and instruction. The whole work consists of the study of TII Error Discovery process using an error control method. This paper describes the different types of

*Received February 3, 2015; Revised May 3, 2015; Accepted May 25, 2015*

instruction which are provided by the approach: 1) Instruction in the selection of the most appropriate process-model, 2) Instruction in the selection of the most suitable approach.

In fact, the TII process is modeled using a work Expression system named Relational Graph. This presentation allows us to represent the goals through the process and the strategies involved into it. Getting a good visual expressing of our process can be beneficial to the better understanding of the problem encountered during the test process. The map contains a finite number of paths, each of them prescribing a way to elaborate Error Discovery steps.

The remainder of this paper is organized as follows: Sections 2-5 of this paper present the revisited work of in terms of a functional view of information system, Field-To-Scope Proportion (FSP), Domain Error discovery and FSP, the Semantic Error scope, the relationship between Semantic Error scope and Error discovery, Semantic Error scope and FSP. Section 6 demonstrates the process Semantics and how to formalize it with Relational Graph. The paper finishes with a discussion of some other related work followed by some concluding remarks.

## 2. A Functional View of Information System

Every item of information system at its most primitive level may be viewed as a function or mapping according to some specification, S, from a set of input values (its field, F) to a set of output values (its scope, R).

A program which implements specification S should also map from F to R. However, if an error f exists in the program, there will be some subset of the field, Ff say, on which the erroneous program Pf computes an error result. The set of error results, denoted Rf, may contain values both in R and outside R. The effect of Pf on values outside of F remains unspecified. See Figure 1. Note that the field and the scope can be considered for an entire program, an individual program component, a program path or simply a single program location.

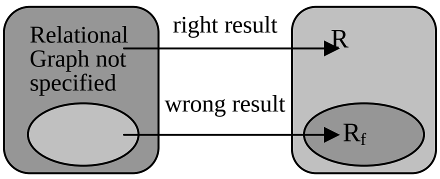


Figure 1. View of a wrong program

## 3. Field-to-Scope Proportion (FSP)

The field-to-scope proportion (FSP) has been proposed by M. Yu [1] as a specification metric. Put simply:

$$Field-to-Scope-proportion = \frac{|F|}{|S|} \quad (1)$$

where |F| is the cardinality of the field of the specification and |S| is the cardinality of the scope. FSP can be determined for mathematical or computational functions. We consider, for example, the function f(d) = d mod 2, where the input d is a member of the set of natural numbers not deeper than 100, i.e. F={1,2,3,…,100F}. Clearly the function generates only two possible outputs, namely 0 when d is even and 1 when d is odd, so that R = {0,1} and FSP = 100/2 = 50. Difficulties arise when the field and the scope, are infinite.

FSP metric provides an approximate measure of in- formation loss. Information loss may become manifest as “internal data state collapse” which occurs when two different data states are input in a program and produces the same output state. M. Yu remarks a connection between FSP and state collapse, and implies that the Error Discovery of a program is correlated with the FSP. High FSP is thought to lead to low Error Discovery and vice versa.

## 4. Field Error Discovery and FSP

Field Error Discovery involves use of the concepts of identifying and managing [5]. Information system component is observable, if a test input is repeated, the output is the same. If the outputs are not the same, the component is dependent on hidden states not identified by the tester and Freedman calls this an “input inconsistency”. A software component is controllable, if an output identifier is specified to be a certain scope of values and there are particular instances of values that cannot be generated by any test input values, those are termed “output inconsistencies”.

Most functions and procedures are not a priori observable and controllable. The modifications required to achieve field Error Discovery are called extensions.

Observable extensions are achieved by introducing new input variables so that the component becomes observable, i.e. distinct outputs can only arise from distinct inputs. Controllable extensions are achieved by modifying outputs for the given component so that it becomes controllable, i.e. all claimed outputs are attainable with some input. Managing is achieved by an appropriate reduction of the scope. Identifying and Managing can be measured.

In order to consider the relationship between field Error Discovery and field-to-scope proportion, the field and scope of the component after modification with observable and controllable extensions can be written as:

$$DRR' = \frac{|D'|}{|R'|} = \frac{|D| + |\Delta D|}{|R| - |\Delta R|} \tag{2}$$

Field-to-scope proportion of a program component, after modification to make it field testable, is the field-to-scope proportion of the component before modification multiplied by one plus the relative size of the field extension and divided by one minus the relative size of the scope reduction.

## 5. Semantic Error scope

The experts drew a distinction between the syntactic and the semantic nature of errors. The syntactic nature can be described by the syntactical differences between the error program and the correct program. The semantic nature of an error, on the other hand, results from the view that for some subset of the input field an error computation takes place producing incorrect output. Corresponding to the syntactic size of an error, Offutt and Hayes defined the semantic size of an error as “the relative size of the sub field of F for which the output mapping is incorrect”. It should be obvious that there is no reason why there should be a link between syntactic error scope and semantic error scope. Indeed it is perfectly possible to find situations where a syntactically small error results in a very large semantic error scope, and vice versa.

### 5.1. Semantic Error Scope and Error Discovery

Experts suggested that semantic error scope is closely related to Error Discovery in the sense of M Yu et al. If a statement in the subject program has low Error Discovery, then any error associated with that statement might be expected to have small semantic size and any statement containing an error with large semantic size could be expected to exhibit high Error Discovery.

To explore this connection between semantic size and Error Discovery further, consider the Transmission, Influence and Implementing (TII) model that provides the basis for test-ability evaluating. According to the TII model, the probability of failure is a combination of the individual probabilities: 1) that the error is executed (E = implementing); 2) that implementing of the error causes corruption of the data state (I = influence); and 3) that the error data state propagates to the output (P = transmission).

Referring to Figure 2, where, as before, F represents the entire input field of the subject program, there will be some subset E of F such that all test values in E cause the error to be executed. Amongst those input values that cause error implementing, some will result in data state influence, as represented by the region I. Finally amongst those input values that cause data state influence, some will propagate the error state to the output, as represented by the region P.

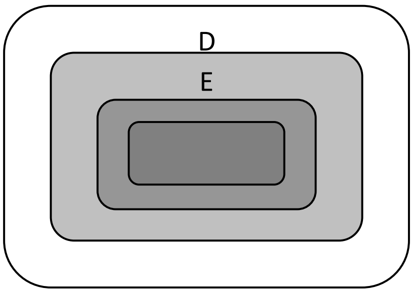


Figure 2. Input field view of the TII

In practice M Yu [1] suggests estimating Error Discovery at a location by separate evaluating processes for the three individual components of the model. These processes are presented in Section 6.

An alternative Error Discovery evaluating procedure could be based on considering versions of the chosen program with location L mutated. The mutation change, provided it does not generate an equivalent mutant, can be regarded as a seeded error that has a semantic size in the same way as naturally occurring errors. The smallest semantic size of such mutants, being a worst case, could provide an estimate for Error Discovery at the location L.

A traditional (strong) mutation testing tool could be used. It requires establishing a large number of input test cases chosen randomly from the input field and then confirming for each mutant generated by the tool, the proportion of test cases that kill that mutant. This is different from normal usage where, once a mutant is killed with some test case, no further test cases are applied to that mutant. Experts did adopt this procedure to estimate the semantic size of all mutants created by the same mutation operator in an attempt to measure the size of given error types. The aim is to determine the minimum semantic size of all mutations at a location. Although still an expensive process, this has the merit, superficially at least, of being considerably more straightforward than using separate evaluating procedures for the three components of the TII model.

It is noted in passing that since transmission analysis is akin to strong mutation testing, and influence analysis is akin to weak mutation testing, a similar distinction could be made for semantic error scope. On the other hand, weak semantic error scope can be considered as the proportion of the input field that merely results in an infected data state immediately after executing an error, i.e.

$$Weak-semantic-fault-size=\frac{|I|}{|D|} \quad (3)$$

### 5.2. Semantic Error Scope and FSP

Semantic can be related to the field-to-scope proportion (FSP). However, since semantic error scope depends solely on the input field, whereas FSP depends on both the field and the scope, there is unlikely to be a direct connection. What can be deduced is a relationship involving error scope, measured in terms of input and output, and FSP both for the correct program and also for an error version when executed over just that portion of the field that exposes the error.

Then denoting FSP for the correct program **P** with input field *F* by $FSP_{PD}$ and for error program $\mathbf{P}_f$ with just the error-exposing input field $F_f$ by $FSP_{PfFf}$ the following is obtained:

$$DRR_{PD}=DRR_{P_fD_f}\times\frac{output-fault-size}{input-fault-size} \quad (4)$$

This equation captures the (admittedly) rather limited connection between FSP and semantic error scope.

## 6. The Process Semantics

Error control method is considered today as a key issue by both the Information System Logic (ISL) and the Information Systems Logic (ISL) communities. Recent interest in error control method is part of the shift of focus from the product to the process view of systems development. There is already considerable evidence for believing that there shall be both: improved productivity of the software systems industry and improved systems quality, as a result of improved development processes. Recent in depth studies of information system development practices, however, demonstrate that we know very little about the development process. Thus, to realize the promise of systems development processes, there is a great need for "a conceptual process model framework".

Error control method is a rather new research area. Consequently there is no consensus on what is a good presentation to represent processes or even on what the final objectives. Process models may be constructed for a number of different reasons, to fulfill different purposes. One purpose may be purely descriptive, that is, to record how some process or class of processes is actually performed.

Alternatively, models may be constructed to guide, support and provide advises or instructions to developers, *i.e.*: to be prescriptive. The ISL expert has focused on descriptive models more than the ISL experts. Yet another way of looking at process models is in terms of the process aspect that they address: some focus on managerial way of the development process whereas others have technical concerns. We propose in this paper a well-defined and repeatable approach to generate well-formed instruction centered process models. For instruction centered process models to be well-formed, we have identified a list of requirements and purposes.

To realize and adapt this approach we adopted a goal-perspective, the Map-driven error control method. The *Map* approach is expressing system based on *purposes* and *strategies*. In this system, purposes abstract from organizational tasks and the different ways in which tasks are performed are purpose-achievement strategies. The map is capable of abstracting from the detail of business processes to highlight organizational goals and their achievement. In this section we first introduce the *key concepts* of a map and their relationships. Then we define *map components* as process to for modeling the Error Discovery process.

This process is modeled using Relational Graph presentation which is a process model. This model is a work Express system based on a non-deterministic ordering of goals and strategies. A map can be represented as a labeled directed graph. The nodes represent goals and the links between nodes correspond to strategies. The directed nature of the graph shows the order of the different goals.

A Relational Graph is defined as a semantics model which allows designing several processes under a single expressing. It is a labeled directed graph with purposes as nodes and strategies as edges between purposes. A Relational Graph is composed of one or more portions. A portion is a triplet < source purpose I, target purpose J, strategy Sij> that captures the specific manner to achieve the purpose J beginning from the purpose I with the strategy Sij. An purpose is expressed in natural language and is composed of a verb followed by parameters. Each Relational Graph has two special purposes "Begin" and "End" to respectively begin and end the navigation in the Relational Graph. Each purpose can only appear once in a given Relational Graph. Each portion is associated a direction that can be one of the following three types: Ordinary, Strategy or Strategic. There are three directions associated with a Relational Graph: IAG, SSG and PSD. IAG can be one of the aforementioned types namely strategy or ordinary or strategic while SSG and PSD are always strategy directions. These directions are further explained below.

1) A direction named "Purpose Achievement Direction" (IAG) is associated to each portion providing a proportional mean to satisfy the target purpose of the portion.
2) "Strategy Selection Direction" (SSG) determines which strategies connect two purposes and helps to choose the most appropriate one according to the given situation. It is applied when more than one strategy exists to satisfy a target purpose from a source one.
3) "Purpose Selection Direction" (PSD) determines which purposes follow a given one and helps in the selection of one of them. It results in the selected purpose and the corresponding set of either IAGs or SSGs. The former is valid when there is only one portion between the source and target purposes, whereas the latter occurs when there are several portions.

A map has two special goals, *Begin* and *End* which represent the beginning and the ending of the process respectively. A goal represents a state that is expected to be reached and a strategy corresponds to how to achieve a goal. To *estimate the Error Discovery*, the process consists of the evaluating of the probability of transmission, influence and implementing. We try through this Relational Graph to model our process. We can find the principal goals which are the evaluating of the probability of transmission, influence and implementing. Achieving these goals allows the evaluating of the Error Discovery.

Also, the process semantics for the Error discovery formalized using Relational Graph. It contains four core purposes “Estimate Propagation probability” and “Estimate Execution Probability”, “Estimate Infection Probability” and “Estimate Error discovery” in addition to “Begin” and “End” purposes. We use also this process model to elaborate and to present the relationship between Error Discovery, TII, and FSP. The main purpose of using the Relational Graph presentation is to simplify the relationship between Error Discovery, FSP and semantic error scope. The Relational Graph model was introduced in this paper in order to model processes in a flexible way.

To allow tester or user to go through the different purposes of the map, the approach provides a set of factors called Background Factors.

Estimating Error Discovery involves the use of identifying, managing concepts and some extensions which are modifications required to achieve field Error Discovery. The relationship between Error Discovery and semantic error scope is important where in case of low Error Discovery we expect to have small semantic size and in case of high Error Discovery we expect to have large semantic error scope. Error discovery is correlated with the field/scope proportion. Adapting do- main-to-scope proportion needs two ways: to invert the proportion so that it becomes the Scope-to-Field Proportion (SFP) or to calculate the scope-to-field proportion dynamically.

The proposed Background factors characterize current situation and then, help designer to choose the appropriate strategy among several presented in the map. We have identified the following factors*:* Application type, Application complexity, Similarity with others applications, User-application adaptation, User-application adaptation, Tester Experience [7].

Background factors guide and orient tester during testing through the design semantics process. When we say guide the tester we mean that this person can choose the appropriate goal to achieve the different options in TII process (transmission evaluating, influence evaluating and implementing evaluating). After the achievements of these goals, the tester can calculate the global Error Discovery.

## 7. Related Works

This section briefly mentions some of the most significant related work (besides that already cited) which is concerned with error models, error transmission and error-based testing.

The TII model bears some similarity to the RELAY model in which an error *originates* a potential failure that must then *transfer* through computations to produce a *state failure* and at last be revealed as an external *failure*. M. Yu [1] showed just how useful a TII Error Discovery estimate could be when used in conjunction with conventional reliability testing to provide, via so-called “squeeze play”, a confidence bound for the correctness of a program. On a more cautionary note however, they also provided a stark critique of the assumptions underlying the TII model.

We used the presentation Relational Graph to define the TII model. TII model is an error Discovery process. Relational Graph as process model allows us to better understand the TII process and presents the selection of the appropriate probability evaluating. Map as expressing system was originally defined and has been the subject of many uses that go beyond the expressing of process logic. We can find for example the use of maps for requirements logic and alignment of COTS products or customization of an ERP system to the needs of an organization.

Finally to validate our proposed approach, we have focused, after that, in describing how the approach guides through an empirical evaluation.

## 8. Conclusions

Error discovery is an important attribute of information system as far as the testing experts is concerned since its measurement leads to the prospect of facilitating and improving the testing process. Unfortunately Error Discovery has various guises. Two distinct and significant interpretations are due to Freedman [1-5] and M. Yu *et al.* [1]. Freedman's notion of Error Discovery has two facets, identifying and managing, both of which can be measured by the extent of certain modifications to a program component. M. Yu's notion of Error Discovery can be estimated by the computationally expensive TII technique and M. Yu himself has suggested a possible link with the rather ordinary concept of field-to-scope proportion.

By taking a functional view of information system, this paper has produced a succinct characterization of managing and identifying and developed a ordinary mathematical relationship involving them and the field-to-scope proportion. Semantic error scope has also been considered and its relationship with M. Yu's Error Discovery has been explored. A consequence of this is the suggestion that Error Discovery of a program location could be estimated more straight forwardly by a small adaptation of the traditional strong mutation testing process, to find the minimum semantic size of all mutants at the location. Finally some refinements of semantic error scope have been introduced and their relationship with FSP has been considered. To visualize the TII model, we model the process using the system of expressing Relational Graph. This presentation allows giving more importance to the goals and the strategies used in this process.

The authors recognize the desirability of validating the connections between the concepts as discussed here. Validation could take the form of empirical evidence, but could also consider a more analytical approach along the lines who has modeled information system as finite functions to deduce theoretical results concerning error coupling. In the meantime, this paper has made a limited begin at putting together the various separate pieces of what might be considered a rather complex jigsaw of related concepts.

## References

[1] M Yu, H Mokhtar and M Merabti. "Error Management in Neural networks". *IEEE Wireless Communications*. 2007; 6: 13-19.
[2] Ruo hu. Channel Access Controlling in Wireless Sensor Network using Smart Grid System. *Applied Mathematics & Information Sciences*. 2012; 11: 813-820.
[3] CR Li and CK Liang. "An error-Tolerant Incident Boundary Discovery Algorithm in Web Networks". *Information Networking: Towards Ubiquitous Networking and Services*. 2008; 5: 406-414.
[4] Ruo hu. Stability Analysis of Wireless Sensor Network Service via Data Stream Methods. *Applied Mathematics & Information Sciences*. 2012; 11: 793-798.
[5] Hu Ruo. New Network Access Control Method Using Intelligence Agent Technology. *Applied Mathematics & Information Sciences*. 2013; 2: 44-48.
[6] HS Hu and GH Qin. "Error Management Frameworks in Neural networks". *4th International Conference Intelligent Computation Technology and Automation*, Shenzhen, China. 2011.
[7] XH Xu, B Zhou and J Wan. "Tree Topology Based Error Diagnosis in Neural networks". *International Conference on Wireless Networks and Information Systems*. Shanghai, China. 2009.
[8] W Junior, T Figueiredo, H Wong and A Loureiro. "Intrusion behavior Discovery in Wireless Web Net-works". *18th International Parallel and Distributed Processing Symposium*, New Mexico, Mexico. 2004.
[9] XY Xiao, WC Peng, CC Hung and WC Lee. "Using Web Ranks for In Network Discovery of Error Text in Neural networks". *International Workshop Data Logic for Wireless and Mobile Access,* Beijing, China. 2007.
[10] IM Atakli, H Hu, Y Chen, WS Ku and Z Su. "Intrusion behavior Discovery in Neural networks Using Weighted Trust Estimating". Proceedings of Spring Simulation Multi-Conference, O ttawa, Canada. 2008.