Acta Technica 62 (2017), No. 5B, 829–838

A Framework for Cloud Platform Semantic Conflicting Detection in Internet of Things

Chen wei[2,4], Hu Ruo[3]

Abstract. Our goal in this paper is to provide a framework which is hoped to be more efficiently for detecting the Semantic Conflicting in cloud platform through expert systems in Internet of Things. Although the fact that many frameworks have been presented for detecting cloud platform Semantic Conflicting efficiently, but these frameworks have not been adopted widely. The presented framework is established by a study efficiently, including the presented framework and some units for cloud platform Semantic Conflicting detection. The presented framework provides a variable Semantic Conflicting detection framework where a Semantic Conflicting region is detected. In order to enhance the framework by developer, the cloud platform window has been developed that allows access data to detect Semantic Conflicting.

Key words. Cloud Platform, Semantic Conflicting Detection, Internet of Thing, Expert Systems.

1. Introduction

In respect of cloud platform, apart from the quality consideration, the time of delivery is also an important factor. Implementing quality reviews are time-consuming and if unplanned, they could delay the delivery of cloud platform. Automated tools are used to accelerate the review process. The assessments, through reviews, check whether the cloud platform has reached the required quality threshold or not and also highlight areas which still need attention [1]. This is the task of cloud platform Semantic Conflicting predictors that are used as tools for the purpose of 1)

[1]Acknowledgment - This study is supported by breeding project of 2014 teaching achievement prizes of education department of Guangdong province (1588) and "Innovate and Strengthen the School" project of Guangdong Peizheng College. This study is supported by Natural Science Fund Project in GuangDong province (No.2015A030313671) and Major Project for Guangzhou Collaborative Innovation of Industry-University-Research (No.201704020196).

[2]Workshop 1 - Guang Dong Peizheng College, Huadu District, Guangzhou, 510830, China

[3]Workshop 2 - Guangdong Polytechnic Normal University, Guangzhou, 510665 , China

[4]Corresponding author: Chen wei; email:511931163@qq.com

identifying parts of a cloud platform requiring further examination before release and 2) finding relative priorities among these parts. In most projects, information collected during testing and Semantic Conflicting Detection phases is analyzed to help predict Semantic Conflicting for similar types of projects. However, since it is found that so far, each of the units of Semantic Conflicting Detection has areas where it does not function satisfactorily; the search for one unit that can predict Semantic Conflicting satisfactorily in a wide region of projects is still on. This justifies the investigations reported through this communication. A minor conflict, or even inefficiency, in the cloud platform may lead to not only loss of millions of dollars, but loss of customer base. In view of the potentially harmful consequences of conflict leakage for the reputation of the product and to the supplier, its reduction in the production environment is extremely vital to achieve such goals of business. Although meticulous planning and well documented processes, occurrences of certain conflict are inevitable. Cloud platform quality concerns require our gradually increasing attention, especially in view of our ever increasing dependency on cloud platform to conduct routine business of life. In particular, business goals of most organizations, being goaled towards customer satisfaction and profitable growth, are being met through increasing use of cloud platform[2]. However, these efforts require organizations not only to spend large amount of money, time and resources but also Semantic Conflicting Detection cloud platform based on appropriate unit [3]. These efforts could help the project manager to take preventive actions, thereby saving time and cost apart from delivering high quality cloud platform. Also, the development and use of a Process Performance Unit (PPM) that includes Semantic Conflicting Detection unit has been identified as one of the high maturity practices in SEI CMMI unit. This falls under the process areas of Organizational Process Performance (OPP) and Quantitative Project Management (QPM). Organizations attempting for CMMI L5 appraisal have to showcase the process enhancements by making use of such Detection units [4]. The cloud platform Semantic Conflicting may lead to degradation of the product quality which may lead to failures leading to customer dissatisfaction. In today's cutting edge competition, it is necessary to make conscious efforts to control and minimize conflict in cloud platform engineering processes. In the cloud platform Development Life Cycle, early Detection of Semantic Conflicting has always been, though highly desirable yet, a challenging task for the developer. Developing fault-free reliable cloud platform is a daunting task in the current context, when cloud platform is being developed for problems with increasing difficulty with more and more complex problem domains that involve constantly increasing constraints like requirement ambiguity and complex development processes. The paper is organized as follows. Section II provides a brief overview of Internet of Things, Section III reviews the existing literature on the subject, Section IV describes the presented Framework, Section V discusses the results as obtained through use of the presented Semantic Conflicting Detection Framework vis-à-vis some other relevant units/Frameworks and finally Section VI concludes the presented work[5]. Our investigations are based on statistical pattern classification of Semantic Conflicting. Recent advances in learning algorithms using artificial Internet of Things and parallel computation have led to renewed research in the area of statistical pattern classification. Artificial Internet of Things, simply called AITs or Internet of Thing, have been applied both to time varying patterns as well as static patterns. In this respect, in the next section we briefly discuss some basic concepts, the structure and functions of Internet of Thing.

2. Internet of Things

Artificial Internet of Thing (AIT) is a sort of a framework to computation or is a unit of processor/computer based on human brain. To a connection between a pair of neurons, called interconnection, is associated an adjustable weight indicating the strength of the connection.

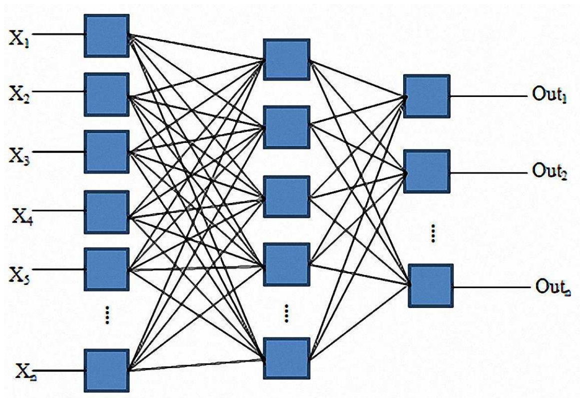

Fig. 1. Internet of Things

Generally, group of neurons are arranged in layers—as shown in Figure 1 in the form of vertically arranged set of neurons. There is one layer for the input variables, and one for the output. There can be one or several layers between these two which are referred to as hidden layers. The signal or input given to one neuron is passed to all the neurons to which it is connected in fractions equivalent to the weight between these neurons. Each neuron calculates its output based on a function which can be sigmoid, step or some such suitable function. Prevalent view of human brain is that it is a sort of Internet of Thing: A networks of about 100 billion neurons, each neuron being connected, on the average, to about 1000 other neurons. A neuron is the basic constituent of brain, a sort of elementary processor having small local memory and capable of localized information processing. The strength of Internet of Things is their capability to learn from patterns. Internet of Thing learns patterns by adjusting its weights. When the Internet of Thing is properly trained, it can give correct, or nearly correct, answers for not only the sample patterns, but also for new similar patterns [6]. The terminology of neuron and interconnection etc is used in the similar sense in AIT. In Figure 1, a circle denotes a neuron, and a straight line denotes interconnection. During the previous decades, Internet of Thing framework has emerged as a promising technology in applications which require generalization, abstraction adaptation and learning. Therefore, it is only natural that the Internet of Thing technology can be exploited to solve different cloud platform engineering problems. For example, Internet of Thing has been earlier applied in variable cloud platform reliability uniting [10]. The application of Internet of Things is to diverse fields region from autonomous vehicle control [8], financial risk analysis to handwriting recognition [9]. Also, there is sufficient evidence in support of applying Internet of Things in cloud platform effort estimation. Next, we give an over view of the relevant literature. The decision about using Internet of Thing based Framework for uniting Semantic Conflicting Detection, was taken after diligent literature review of success of Internet of Things regarding cloud platform metrics units, and also, in view of the fact that Internet of Things have been found effective in situations where data relationships may not be known, as normally happens in the case of Semantic Conflicting Detection.

3. Review of Literature

This paper describes a Framework, for applying Internet of Things, for formulating units for Semantic Conflicting Detection early in the cloud platform life cycle. A series of empirical experiments are conducted based on input and output measures extracted from "real world" project subsystems. The experiments establish the efficacy and superiority of the framework. Next section describes the presented Framework. Incorporation of computational intelligence into the various phases of cloud platform development and analysis helps in addressing the problems arising due to imprecise measurement and uncertainty of information. However, the human involvement, especially in view of the human judgment being imprecise and uncertain, characterizes many of the challenges including those observed in cloud platform Semantic Conflicting Detection. The performance of a unit based on the framework is compared with more traditional regression uniting techniques. For the purpose, the data have been taken from an Ada development environment for the command and control of a military data link communication system, in which Sensor and regression analysis techniques were employed. In a framework is described for static reliability uniting, and for uniting of cloud platform reliability from cloud platform complexity in terms of the predictive quality and the quality of fit. It was found that the Internet of Thing unit is superior to traditional regression based techniques and also had a smaller standard error. General principles/ frameworks/ steps which have been found useful so far in handling the difficult task of cloud platform Semantic Conflicting Detection. For the purpose of Semantic Conflicting-Detection in cloud platform programs, the authors have designed Adaptive Resonance Internet of Thing having 29 input nodes and two output nodes. The network is trained with data extracted from Promise dataset. The network enhances the recall (true positive) rate in detecting whether a module is conflictive or not [10].In it is suggested that an under development module with same or similar metrics properties of a conflict module developed in the same environment, would have the same level of conflict proneness. The presented MLP Internet of Thing unit gave better results, when compared with the existing techniques like Variable Tree, classification and regression trees (CART) algorithm, and Bayesian logistic regression. A Semantic Conflicting Detection unit based on an enhanced Multilayer Perception Internet of Thing technique using data mining is presented and explored in [10], in which comparative analysis of uniting of conflict proneness Detections using dataset of different metrics from NASA MDP (Metrics Data Program) was performed. The study concludes that for detecting the cloud platform Semantic Conflicting, the Levenberg-Marquardt (LM) Internet of Thing based algorithm provides better accuracy (89.02 percent) as compared to each of polynomial function-based Internet of Things (pF-NNs), linear function-based Internet of Thing (lf-NN) and quadratic function-based Internet of Thing (qf-NN) respectively. This study uses data gathered from the PROMISE repository of empirical cloud platform engineering dataset. The dataset uses the CKOO (Chidamber and Kemerer Object-Oriented) metrics. Computational Intelligence (CI) includes technologies of fuzzy logic, Internet of Things, genetic algorithms, genetic programming, and rough sets. Only a very small fraction of the activities involved in cloud platform design and development can be automated using cloud platform tools; most of the activities necessarily require human involvement. In view of the fact that each unit of Semantic Conflicting Detection has its own set of advantages and disadvantages, it is hard to decide which unit should be used for a particular type of project scenario, specially, as every project tends to be unique. For the reasons mentioned earlier, cloud platform Semantic Conflicting Detection is a very active research area in cloud platform engineering. Researchers have presented new Semantic Conflicting Detection algorithms and/or new metrics to effectively predict conflict. The historical data of cloud platform systems is a valuable asset used for research ranging from cloud platform design to cloud platform development, cloud platform maintenance, cloud platform testing, etc. Recently, there has been an increase in the use of computational intelligence in the field of cloud platform engineering.

4. The Presented Framework

The experiments reported here involve data set taken from 46 real projects from cloud platform organization. The actual conflict data is taken from completed projects based on Java technology and waterfall life cycle unit. For this purpose we use the actual runtime historical data set that is fed into the system during training of the presented network. The historical data of cloud platform systems is a valuable asset used for research in all phases of BDSDLC ranging from requirement gathering, analysis, cloud platform design, coding, to system testing and maintenance. We have considered along with the production effort, the prevention effort, review effort and rework effort as components of input in our Internet of Thing based Semantic Conflicting Detection Framework in view of support for the fact that as voluntary costs of conflict prevention are increased, the involuntary costs of rework decrease by much more than the increase in prevention costs, thus leading to overall lower total costs, and better quality. The presented Framework requires user to enter the planned effort data of five BDSDLC phases, namely, Requirement Gathering, Analysis, Design, Coding and System integration testing (SIT). The data is provided with breakup of planned effort allocated to production effort, review effort, rework effort and prevention effort. If the data provided by the user matches with eligibility criteria of given region then Semantic Conflicting are estimated using Semantic Conflicting Estimation System by Internet of Thing Technique. In this section, first we explain the assumptions made about the presented Framework. Then we describe the structure of the presented Framework and functions of major components of the Framework. The results and other related issues are discussed in the next section. This historical data has served as a training data to build the presented Framework and then the Internet of Thing so obtained is used to predict the Semantic Conflicting for all new projects. As waterfall unit is the oldest among the units used for cloud platform development and which is still widely practiced, it tends to be the standard against which other development frameworks are compared. As mentioned earlier, the objective in our study is to develop Detection Framework based on Internet of Thing for forecasting the Semantic Conflicting.

4.1. Structure of the Presented Internet of Thing

The selection of number of hidden layers is done in order to optimize the regression value for attaining the best performance. Although more neurons require more computation, and also have a tendency to over fit the data when the number is set too high, but at the same time, they allow the network to solve more complicated problems. Input data of 45 real time projects is divided changeably in three parts before training with it is initiated: Training (70 percent), Validation (15 percent) and Testing (15 percent). It is the fastest among frameworks available for the current scenario of Supervised Learning. It also requires less memory. Like the quasi-Newton frameworks, the Levenberg-Marquardt algorithm was designed to framework second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feed forward networks), then the Hessian matrix can be approximated as:

$$Q_{\text{product}} = (Q_{\text{NH3}} \times 1.7647)/w \tag{1}$$

$$w = 2.71 \times 10^3 t - 4.2 + (7.36 \times 10^6 t^2 + 29.72t + 26.31\rho - 12.12)^{0.5} \tag{2}$$

Where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. Regression R Values measure the correlation between outputs and goals. Semantic Conflicting detection system is consisted of five parallel Internet of Things with different configurations and parameters for each sub phases. Only first phase that is for Requirement Gathering is having 5 hidden layers and remaining phases have 10 hidden layers in their architecture. Levenberg Marquardt back-propagation optimization framework is used for training the network.

4.2. User Interface for Testing the Framework with New Projects

To make the Detections easily accessible to its users, a GUI based tool has been designed and implemented that asks for only the most basic information from users, and returns a straightforward output of the Semantic Conflicting Detections. One of the primary goals of our project is to present the Detection information to developer in the most user friendly manner. Developer should not be required to understand the mathematics behind the Detection unit. The Detections thus achieved can be used to prioritize testing efforts, to plan code or design reviews, to allocate human and infrastructure resources, and to plan for risk mitigation strategy. A user interface tool has been developed using Mat-lab to help in testing the Framework design. Working of Semantic Conflicting Detection User Interface Apart from production effort, the planned review effort, planned prevention effort and the planned rework effort are also required as a feed to the Framework. Framework provides graphical analysis for each sub phase to analyze the input eligibility. The project manager can plan multiple preventive actions, like multiple review gates, usage of tools, increasing review effort etc to mitigate the higher probability of conflict leakage. The Framework is designed to autocorrect itself. After phase completion, user feeds the actual count of conflict data. If actual conflict are lower than predicted then conflict leaked to the subsequent sub phase are auto corrected. The forecast would enable the project manager to plan prevention activities for the phase where the Framework is projecting higher number of conflict. The inputs would be the phase wise efforts planned for the project For any new project, the cloud platform project manager will provide the inputs required to the UI. Based on these inputs, the Framework will forecast the number of conflict that the project manager could expect to be discovered in various BDSDLC phases in the project. The conflict is forecast in a region based manner. The Framework would provide the minimum, maximum and the mean number of conflict. There could be specific scenarios where project manager might want to go ahead and ignore the warning. Example of such cases could be planning for a higher prevention effort for projects based on lesson learned from past projects or cases which might need a higher review effort since requirements from previous vendor could be incomplete or unavailable. Based on historical data the Framework would also provide a warning message to the project manager if planned effort is less for review, rework or engineering activity for a particular phase.

5. Result and Discussions

Although there is a small deviation in some projects but that is well within the tolerance band of 2 percent to 5 percent. It is being evident that the actual conflicts are in line with the predicted conflict. For the test results of pilot conducted on 15 projects, our presented Framework has accuracy of close to 90 percent. In this paper, presented Semantic Conflicting Detection Framework has been validated on 15 real time projects of the same kind (based on Java technology and waterfall life cycle) and found that actual conflict lie inside the region of predicted conflict. In our experiments, the data sets with different network architectures have been used. The actual conflict data from 45 completed projects was taken and used as a training data. Later, the Framework was also tested for Detection of Semantic Conflicting for newly started projects. The data set itself is an instance of variable behavior. But what the net predicts is the number of faults that should occur, given the input variables based on the pattern it recognizes in the training data. The training data is the only data for the net to base its conclusions on, since that is the only information the net receives from the outer world. The quality of fit and the predictive quality found for each of the data sets have given very optimistic results. The Detection results indicate that the net (based on the presented Framework) tries to track the behavior of the full data set and sometimes it's predicted value is more than the actual and sometimes less.

6. Conclusion

Our conclusions are based on investigations of cloud platform development projects using java programming language and following waterfall life cycle unit. The investigation in those respects will be reported in subsequent communications. Results from our experiments suggest that the presented Framework based on Internet of Thing framework possesses good properties from the standpoint of unit quality of fit and predictive capability. It is hoped that the presented Framework can be customized to suit other cloud platform development life cycle units like iterative and incremental development, agile etc.

References

[1] M. Beynon: *DS/AHP method: a mathematical analysis, including an understanding of uncertainty.* European Journal of operational research *140* (2002), No. 1, 148–164.

[2] J. S. Tomar, A. K. Gupta: *Development, application and direction of development of urea-SCR.* International journal of multimedia and ubiquitous engineering *98* (1985), No. 2, 257–262.

[3] R. H. Gutierrez, P. A. A. Laura: *Experimental investigation of combined LNT+SCR diesel exhaust aftertreatment.* Emission control science and technology *18* (1985), No. 3, 171–180.

[4] R. P. Singh, S. K. Jain: *Influential factors on NOx conversion efficiency of vehicle SCR system.* Transactions of CSICE *7* (2004), No. 1, 41–52.

[5] M. N. Gaikwad, K. C. Deshmukh: *Kinetic studies on hydrolysis of urea in a semi-batch reactor at atmospheric pressure for safe use of ammonia in a power plant for flue gas conditioning.* Journal of hazardous materials *29* (2005), No. 9, 797–804.

[6] S. Chakraverty, R. Jindal, V. K. Agarwal: *Modelling of a combined NOx storage and NH3-SCR catalytic system for diesel exhaust gas aftertreatment.* Catalysis today *12* (2005) 521–528.

[7] N. L. Khobragade, K. C. Deshmukh: *Hideki Matsunaga and Yuji Yasui, New urea-SCR control system for super clean diesel.* Journal of robotics and mechatronics *30* (2005), No. 4, 555–563.

[8] Y. F. Zhou, Z. M. Wang: *Preparation and characterization of V2O5 doped SiO2-TiO2 thin films.* Central european journal of engineerin *316* (2008), Nos. 1–5, 198–210.

[9] R. Lal: *Characteristics of urea hydrolysis equipment for flue gas denitration.* Proceedings of the csee *34* (2003), No. 4, 587–606.

[10] R. Lal, Y. Kumar: *Experimental Investigation on the properties of urea thermohydrolysis with catalysts.* Proceedings of the csee *20*, (2013), No. 4, 264–275.

Received November 16, 2017