http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (3): 396-398, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.396.398

Using Fuzzy Clustering Method to Classify the Component in the Process of Software Evolution

[1]Jian Wang, [2,3]Na Zhao, [1]Wei Du, [4]Yang Zhao, [2,3]Ye Qian and [2,3]Zuo Jiang
[1]College of Information and Automation Engineering,
Kunming University of Science and Technology, Kunming 650091, China
[2]School of Software, Yunnan University, Kunming 650091, China
[3]Key Laboratory in Software Engineering of Yunnan Province, Kunming 650091, China
[4]Yunnan Normal University, Kunming 650091, China

Abstract: In this study, we assume a software process model with independent functions as a component. The vaguer and ambiguity aspects of human thinking and reasoning process lead people to utilize fuzzy theories to solve certain problems. The goal of this paper is to employ the method of fuzzy mathematic to classify the components in component library during the process of software evolution. To classify component, the target parameters extracted from components are selected out before the standardization and the clustering is performed after the establishment of fuzzy equivalent matrix. This paper outlines the overall processes of the classification of a component and finally an example is discussed as a case study.

Key words: Petri net, software evolution process, component model, fuzzy mathematics

INTRODUCTION

During software evolution, the changes at various granularities occur continuously or discontinuously. An evolution process model must embody the properties of evolution and be able to define more dynamic components than with traditional development so that the changes can be described. By observation and analysis, it is found that the following properties exist in software evolution processes: Iteration (Yang and Ward, 2003; Lehman, 1997), Concurrency, Interleaving of continuous and discontinuous change (Aoyama, 2001), Feedback-driven system (Chatters *et al*., 2000; Lehman and Ramil, 1999) and Multi-level framework.

The goal of this paper is provide an effective solution for component classification. The key to the solution is that how to generate a powerful evaluation mechanics that could be applied to components during the classification. We already knew that the evaluation-related process is a process full of ambiguity and uncertainty, thus, the intuitive feel for the ambiguity led us to the correspondent concept in mathematics: fuzzy clustering.

RELATIVE WORK

In the area of software evolution based on metrics, Lanza proposed an approach based on a combination of software visualization and software metrics which have already been successfully applied in the field of software reverse engineering. Using this approach they discussed a simple and effective way to visualize the evolution of software systems that helps to recover the evolution of object-oriented software systems (Lanza, 2001). Gustafsson *et al.* (2002) showed how software metrics and architectural patterns can be used for the management of software evolution. The quality of a software system is assured in the software design phase by computing various kinds of design metrics from the system architecture, by automatically exploring instances of design patterns and anti-patterns from the architecture and by reporting potential quality problems to the designers (Gustafsson *et al*., 2002).

THE STEPS OF COMPONENT CLASSIFICATION USING FUZZY CLUSTERING

Step 1: The selection of statistical indicators: In previous research, we explored the component's modeling process from evolution perspective. It is discovered that the component classification can be performed from various angle base on the model we already get. Actually, the following six parameters are available: The size of a component, time of life cycle,

Corresponding Author: Na Zhao, School of Software, Yunnan University, Kunming 650091, China

the risk, the cost, the number of resource consumed and the priorities of component.

Step 2: Data standardization on the statistical indicators extracted from component: After the selection of statistical data, the process of expert grading is performed, followed by the process of standardization or normalization. The process can treated as follows:

$$x=\frac{x'-\overline{x'}}{c}, x'_{ik}=\frac{x_{ik}-\overline{x_k}}{s_k}, (i=1,2,\cdots,n;\quad k=1,2,\cdots,m)$$

where, x_{ik} stands for the raw data; $\overline{x_k}$ is the average of the raw data, s_k is the standard deviation of raw data. $\overline{x_k}$ and s_k are written out as follows:

$$\overline{x_k}=\frac{1}{n}\sum_{i=1}^{n}X_{ik}, s_k=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ik}-\overline{x_k})^2}$$

If we compress the standardized data into a closed interval [0,1], the extreme value standardization formula can be used:

$$x''_{ik}=\frac{x'_{ik}-\min\{x'_{ik}\}}{\max\{x'_{ik}\}-\min\{x'_{ik}\}}, (k=1,2,\cdots,m)$$

Evidently, x = 1 when $x'_{ik}=\max\{x'_{ik}\}$, x = 0 when $x'_{ik}=\min\{x'_{ik}\}$

Step 3: Establishment of fuzzy equivalent matrix: The purpose of this step is to calculate out the statistical quantity that is used to evaluate the similarity of two components to be classified, the statistical quantity being referred here is r_{ij} (i, j 1, 2 ..., n) where, n is the number of the component to be classified. Thus, we get the resembling relations on domain U:

$$\underline{R}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix}$$

To calculate here, there are a number of ways: the dot product method, the angle cosine method, the correlated coefficient method, exponential similar coefficient method, max-min method, minimum of arithmetic average method and minimum of geometric average method (Li, 2001). Here, we use exponential analysis:

$$r_{ij}=\frac{1}{m}\sum_{k=1}^{m}\exp\left[\frac{3(x_{ik}-x_{jk})^2}{4S_k^2}\right]$$

$$s_k^{\,2}=\frac{1}{n}\sum_{i=1}^{n}(x_{ik}-\overline{x}_k)^2$$

$$\overline{x_k}=\frac{1}{n}\sum_{i=1}^{n}X_{ik}\ ,\ (k=1,2,\cdots m)$$

Step 4: Clustering: To carry out clustering, r must be a fuzzy equivalent relationship, that is, must meet the condition of reflexivity, symmetry and transitivity. And when these conditions are not satisfied, then we need to use transitive closure method to construct fuzzy equivalent relationship.

THE EXAMPLE OF COMPONENT CLASSIFICATION USING FUZZY CLUSTERING

We use U = (u1, u2, u3, u4, u5) to represent the five components need to be classified. Six parameters are selected to perform the task of component's classification, including: the size of component size, time cycle, risk, cost, the number of resources consumed and component's priority. After grading by experts, we go the measuring results as follows:

u1 = {94, 50, 21, 87, 66, 10}; u2 = {85, 34, 13, 93, 73, 6}; u3 = {71, 23, 12, 86, 51, 5}; u4 = {63, 17, 11, 65, 42, 9}; u5 = {84, 46, 14, 73, 41, 7}

We are now trying to classify these five components.

First of all, we use the approach of exponential similar coefficient to calculate quantity r:

$$r_{ij}=\frac{1}{6}\sum_{k=1}^{6}\exp\left[\frac{3(u_{ik}-u_{jk})^2}{4s_k^2}\right]\ (i,j=1,2,...,5)$$

Where:

$$s_k=\sqrt{\frac{1}{5}\sum_{i=1}^{5}(u_{ik}-\overline{u}_k)^2}, \overline{u_k}=\frac{1}{5}\sum_{i=1}^{5}u_{ik}, (k=1,2,\cdots 6)$$

And the results of calculation are as follows:

$$R=\begin{bmatrix} 1 & 0.423 & 0.239 & 0.153 & 0.328 \\ 0.423 & 1 & 0.572 & 0.209 & 0.553 \\ 0.239 & 0.572 & 1 & 0.538 & 0.429 \\ 0.153 & 0.209 & 0.538 & 1 & 0.453 \\ 0.328 & 0.553 & 0.429 & 0.453 & 1 \end{bmatrix}$$

Product matrix R by itself and repeat this process for a certain times until the equation $R_{2k} = R_k$ is satisfied. R_k

is referred to as fuzzy equivalent matrix, that is, $r_{ij} = 1$, $r_{ij} = r_{ij}$, $S_k * S_k = S_k$.

After three times of self-multiplication, we discover that R^8 equals to R^4 , so the fuzzy equivalent relation is: R^4. Now we arrive at the step of clustering.

We can perform component classification by setting the value of λ and standard of component according to user's requirements:

When $0 \le \lambda \le 0.423$, all elements in the matrix are 1, so U is divided into one and only one category: {u1, u2, u3, u4, u4, u5}, the classification we get in this step is the roughest;

When $0.423 < \lambda \le 0.538$, the level matrix of equivalent matrix is:

$$R_\lambda = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Evidently, U is divided into 2 categories: (u1), (u2, u4, u3, u5); when $0.538 < \lambda \le 0.553$, the level matrix of equivalent matrix is:

$$R_\lambda = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

U is divided into 3 categories: (u1), (u2, u3), (u4, u5); when $0.553 < \lambda \le 0.572$, the level matrix of equivalent matrix is:

$$R_\lambda = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

U is divided into 4 categories: (u1), (u2, u3), (u4), (u5); when $0.572 < \lambda \le 1$, it is found that only the diagonal elements are greater than or equal to λ while we examining the elements of the matrix, so all the diagonal elements here equals one, while the rest elements of the matrix are zero, making the matrix a identity matrix. The result is that U is divided into 5 categories: (u1), (u2), (u3), (u4), (u5), each element of the matrix is to be divided into one class, making it the most detailed classification.

ACKNOWLEDGMENT

This work has been supported by the National Science Foundation of China under Grant No. 60963007, by the Science Foundation of Yunnan Province, China under Grant No. 2007F008M, the Key Subject Foundation of School of Software of Yunnan University and the Open Foundation of Key Laboratory in Software Engineering of Yunnan Province under Grant No. 2010KS01, by the Science Foundation of Yunnan Province, China under Grant No. 2010CD026, 2008CD082, 2008ZC047M, 2010ZC030, by the Science Foundation of Yunnan Provinc-the Research of the component description and its evolution based on extended object-oriented Petri Net, by the Science Foundation of Yunnan University, No. 2009F36Q, by the Science Foundation of Yunnan Province Education Department of China under Grant No. 09C0109.

REFERENCES

Aoyama, M., 2001. Continuous and discontinuous software evolution: Aspects of software evolution across multiple product lines. Proceedings of the 4th International Workshop on Principles of Software Evolution, March 13, 2001, ACM Press, New York, pp: 87-90.

Chatters, B.W., M.M. Lehman, J.F. Ramil and P. Wernick, 2000. Modelling a software evolution process: A long-term case study. Software Process: Improvement Pract., 5: 91-102.

Gustafsson, J., J. Paakki, L. Nenonen and A.I. Verkamo, 2002. Architecture-centric software evolution by software metrics and design patterns. Proceedings of the 6th European Conference on Software Maintenance and Reengineering, (ECSMR? 02), IEEE Computer Society, pp: 108-115.

Lanza, M., 2001. The evolution matrix: Recovering software evolution using software visualization techniques. Proceedings of the 4th International Workshop on Principles of Software Evolution, March 13, 2001, ACM Press, New York, pp: 37-42.

Lehman, M.M., 1997. Laws of software evolution. Lecture Notes Comput. Sci., 1149: 108-124.

Lehman, M.M. and J.F. Ramil, 1999. The impact of feedback in the global software process. J. Syst. Software, 46: 123-134.

Li, H., 2001. Fuzzy Mathematical Foundations and Practical Algorithms. Science Press, Beijing.

Yang, H. and M. Ward, 2003. Successful Evolution of Software System. Artech House, United Kingdom, Pages: 101.