

# Monitoring of Software Project Progress Base on Automata Theory

Funchun Liu<sup>1, a</sup>, Gengbing Zhao<sup>2, b</sup>

School of Computers, Guangdong University of Technology, Guangzhou 510006, China;

<sup>a</sup>liufuch@gdut.edu.cn, <sup>b</sup>727689016@qq.com

**Keywords:** Automaton, Software Project, Progress Monitoring.

**Abstract.** Since the monitoring of software project progress involves many uncertain factors, it is crucial to choose an effective modeling method to monitor the software project progress. In this paper, we propose a novel method to monitor the software project progress based on automata theory. The main contributions are threefold. First, an extended automaton, namely assigned finite automaton, is formalized to model the software project progress. Then, a new system structure for monitoring software project progress is presented based on the assigned finite automaton. Finally, an algorithm to realize the monitoring is developed. In addition, an example is provided, in which it is illustrated that the proposed method may effectively monitor the software project progress.

## 1. Introduction

The complement of software project, to a great extent, depends on the effective management. As an important part of management of software project, monitoring software project progress is significant to complete the whole project on time[1,2].

Recently, the monitoring of software project was investigated by many scholars. A monitoring model of software project progress based on measurement was proposed by WenJie Zhao[3], which may effectively track and monitor the development of the each unit of the project. The practical progress vanguard line method[4] proposed by Xiaoming Cui can effectively monitor software project progress. A kind of web models of software developing based on finite automata [5] proposed by Liming Ao conveniently requires analysis and description. Xian Cheng Zhu[6] binding characteristics of industrial control system presents a software development model based on finite automata, which is applicable for system development in the field of industrial control. Besides, the Team Software Process(TSP) method based on Temporal Parallel Automata(TPA) proposed by Bing Zhang[7] is beneficial to the activity planning, resource allocation and schedule control of software process.

The above research has made some achievements in the software project progress monitoring, but their methods do not consider the characteristics of software project progress. Therefore, in this paper, combining with characteristics of software project progress, we propose a monitor method of software project progress based on automata theory.

## 2. Extension of finite automata

To improve the modeling ability of finite automaton, this paper will extend the automaton to assigned finite automaton(AFA). AFA is denoted by  $G=(Q,\Sigma,\delta,q_0,F)$ , where  $Q=X\times I$ ,  $\Sigma=E\times W$ ,  $X$  is the finite set of states,  $E$  is the finite set of events,  $I$  and  $W$  are all the sub sets of Natural number,  $q_0$  is the initial state,  $F$  is the set of accepting states final state,  $q_0$  belong to  $Q$ ,  $\delta:Q\times\Sigma\rightarrow Q$  is the transition function. Note, every elements in  $Q$  or  $\Sigma$  is a two-tuple.

Example 1: An AFA is noted by  $G_1=(Q,\Sigma,\delta,q_0,F)$ , where  $X=\{a,b,c\}$ ,  $I=\{0,1,2,3\}$ ,  $E=\{x,y,z\}$ ,  $W=\{3,5,8\}$ ,  $q_0=(a,0)$ ,  $F=\{(c,3)\}$ ,  $\delta((a,0),(x,3))=(a,3)$ ,  $\delta((a,3),(y,5))=(b,2)$ ,  $\delta((b,2),(y,8))=(b,3)$ ,  $\delta((b,3),(z,3))=(c,3)$ , the state transition diagram of this AFA shows in Fig.1.

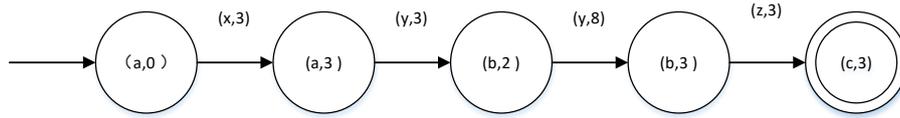


Fig. 1 State transition diagram

### 3. Software project process monitoring based on automata

#### 3.1 Monitoring system structure

The essence of software project progress monitoring is to adjust the schedule according to the deviation of the actual project completion degree and the plan completion degree, and finally achieve the project timely delivery or as short as possible to reduce the delivery time. Since each module of the software project has a deviation value of the actual and planned degree of completion and completion, therefore, in the establishment of software project progress monitoring system need to consider the progress of each module. The monitoring system structure diagram is shown in Fig. 2. The next section will analyze the structure of the monitoring system.

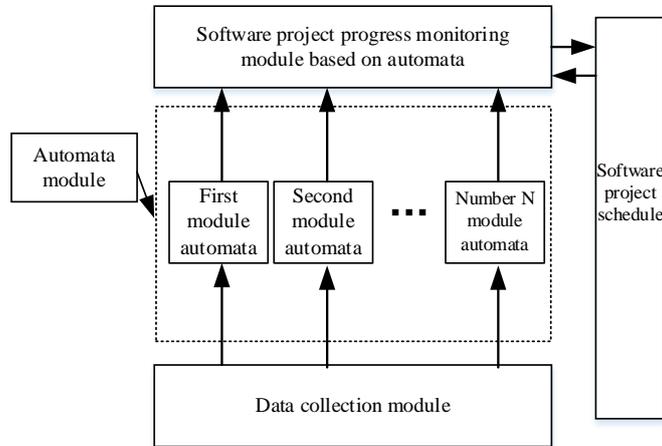


Fig. 2 Monitoring system structure figure

#### 3.2 Monitoring system structure analysis

##### (1) Data collection module

Data collection module is to collect basic data which can reflect changes of software project progress, and the collected data was stored in table called module schedule change list. The table contains five data items including the module number, develop events, work, module current progress (S, P) and update times, where develop event represent the actions of software development, module current progress (S, P) describe the result of project develop progress.

##### (2) Automaton module

The automata module is composed of a number of module automata, each automata module is represented by a changing progress of project module, the definition of first module automata is as follows, other module automata are similar to it.

##### Definition 1: First module automata

First module automata is denoted by  $G_1=(Q_1, \Sigma_1, \delta_1, q_0, F_1)$ , where  $Q_1=X_1 \times I_1=\{(x_1, v_1) | x_1 \in X_1, v_1 \in V_1\}$ ,  $X_1=\{RA, D, C, T\}$ , RA represents requirement analysis, D represents design phase, C represents coding phase, T represents testing phase,  $V_1=\{0, 1, 2, \dots, 100\}$ , the element in  $V_1$  represents completed degree of each phase,  $\Sigma_1=E_1 \times W_1=\{(e_1, w_1) | x_1 \in E_1, v_1 \in W_1\}$ ,  $E_1=\{ra, d, c, t\}$ , ra represents requirement analysis event, d represents design event, c represents coding event, t represents testing event,  $W_1=\{w_1 | w_1 \in N\}$ , the element in  $W_1$  represents the cost of event in  $E_1$ ,  $q_0=(RA, 0)$  is the initial state,  $F_1=\{(T, 100)\}$  is the set of accepting states final state,  $q_0$  belong to  $Q_1$ ,  $\delta_1: Q_1 \times \Sigma_1 \rightarrow Q_1$  is the transition function.

The purpose of the automata module is to handle the data of data of data collection module, and get the state sequence and the event sequence of module progress.

##### Definition 2: The state sequence of module progress

The state sequence of module progress is a string ,denoted by  $S_{\alpha}=s_1s_2...s_n$ ,where  $s_1,s_2,...,s_n \in Q_1$ , $Q_1$  is the set of  $G_1=(Q_1,\Sigma_1,\delta_1,q_0,F_1)$ .

Definition 3:The event sequence of module progress

The state sequence of module progress is a string ,denoted by  $S_{\Sigma}=\sigma_1\sigma_2...\sigma_n$ ,where  $\sigma_1,\sigma_2,...,\sigma_n \in \Sigma_1$ , $\Sigma_1$  is the set of  $G_1=(Q_1,\Sigma_1,\delta_1,q_0,F_1)$ .

(3) Software project progress monitoring module based on automaton

The main content of this module is to compare the actual completion degree with the plan completion degree according to  $S_{\alpha}$  and  $S_{\Sigma}$  from automata module, then adjust the module' s development with the compared result, and finally realize the progress monitoring of software project. The following sections will further illustrate the concepts involved in this module.

1) Deviation rate calculation formula

$$Deviation\ Rate(DR) = \frac{Actual\ Completion(AC) - Plan\ Completion(PC)}{Plan\ Completion(PC)} \times 100\% \quad (1)$$

$$Actual\ Completion = \sum_1^n \sigma_i.w \quad (2)$$

Note, $n$  is correspond to  $n$  in  $S$ ,due to  $\sigma_i=(e_1,w_1)$  is a two-tuple,  $\sigma_i.w$  represent the second tuple value.

2) Classification of deviation

How to adjust the project development depend on deviation rate of actual completion and plan completion, and different deviation rate correspond to differ adjusted schemes. In this paper, deviation rate was divided into three part ,the content of every part is as follows:

A. Acceptable level

In this level, the deviation rate is small, and it don't effect the delivery of software project. Thus,project management need not adjust schemes.

B.Adjustable level

In this level, the deviation rate may effect the delivery of software project, but project manager can reduce it to acceptable level throw motivating team members measurement.

C.Need to modify the schedule level

In this level, the deviation rate is large ,and it can't reduce it to adjustable level whatever measurement are took by project manager.

(4) Software project schedule

The software project schedule records the work and other content of each module of the software project. It is the basis of the software project progress monitoring. The work of the software project plan is usually done by project managers and experts.

### 3.3 Monitoring algorithm implementation

The main task of the monitoring algorithm of software project progress based on automata is handle  $S_{\alpha}$  and  $S_{\Sigma}$ , and calculate the deviation rate level of each module combining with Software project schedule.The implementing steps of it is as follows:

First step: get  $S_{\alpha}$  and  $S_{\Sigma}$  of from each module automata according to the running result of module automata,and put them into the set of  $V$  as two-tuple form;

Second step: if  $V$  is not empty, take an element of  $V$ , and calculate the DR of this module according to formula (1),(2) and schedule's content; if  $V$  is empty, then jump to (5);

Third step: determine the level of this module's DR belong to which kind level according to DR,and put the module number and DR of this module in set of  $R$ ;

Fourth step: delete  $S_{\alpha}$  and  $S_{\Sigma}$  of this module from set of  $V$ ,then jump to (2);

Fifth step: output the set of  $R$  which contains the module number and the deviation rate level information.

## 4. Case analysis

### 4.1 Experimental Data

Experimental data is from a software project that author took part in, and this project contains interface module numbered “001” and data processing module numbered “002”. The updating frequency of module progress list is two times a day, after a period of development, the contents of this module progress list see table 1 and table 2, the contents of software project schedule see table 3.

Table 1 The result of interface module schedule change list

Update times	Module number	Development events	Work (man day)	Module current progress (S, P)
0	001	-	-	(requirement analysis ,0)
1	001	requirement analysis event	6	(requirement analysis, 60)
2	001	requirement analysis event	4	(requirement analysis, 100)
3	001	design event	4	(design phase, 60)
4	001	design event	3	(design phase, 100)
5	001	coding event	8	(coding phase, 30)
6	001	coding event	12	(coding phase, 80)

Table 2 The result of data processing module schedule change list

Update times	Module number	Development events	Work (man day)	Module current progress (S, P)
0	002	-	-	(requirement analysis ,0)
1	002	requirement analysis event	6	(requirement analysis ,100)
2	002	design event	8	(design phase,100)
3	002	coding event	16	(coding phase,60)
4	002	coding event	8	(coding phase,100)
5	002	test event	6	(Testing phase,30)
6	002	test event	8	(Testing phase,100)

Table 3 Software project schedule list

Module number	Module name	Work(unit: man day)			
		Requirement analysis	design phase	Coding phase	Testing phase
001	Interface module	9	8	22	12
002	data processing module	6	7	20	16

### 4.2 Deviation rate level setting

Due to the actual situation of each project team is different, the deviation rate level setting can be differ. The setting of deviation rate level see Table 4.

Table 4 The setting of deviation rate level

Deviation rate level	Corresponding deviation rate range
Acceptable level	$DR \leq 10\%$
Adjustable level	$10\% < DR \leq 20\%$
Need to modify the schedule level	$DR > 20\%$

### 4.3 Results and evaluation

The monitoring results of the project are processed by the software project progress monitoring module based on automata, in which the table 5 is the monitoring result of the interface module, and the table 6 is the result of the data processing module.

Table 5 The monitoring results of interface module

Module number	Update times	Deviation rate (%)	Deviation rate level
001	1	10.0	Acceptable level
001	2	11.1	Adjustable level
001	3	1.4	Acceptable level
001	4	0.0	Acceptable level
001	5	5.9	Acceptable level
001	6	6.9	Acceptable level

Table 6 The monitoring results of data processing module

Module number	Update times	Deviation rate (%)	Deviation rate level
002	1	0.0	Acceptable level
002	2	7.7	Acceptable level
002	3	19.0	Adjustable level
002	4	15.2	Adjustable level
002	5	5.8	Acceptable level
002	6	6.1	Acceptable level

The content of table 5 and table 5 show, this monitoring method can real-time obtain changes of software project progress, and project managers can adjust the development of software project according to information of deviation rate and deviation rate level, ensuring the complete of software project on time. The experimental data show that the monitoring methods of software project progress based on assignment automata has active effect.

## 5. Summary

The monitoring of Software project progress is a complex problem. In this paper, combining with changing characteristics of the software project progress, a monitoring method of software project progress based on automata theory is proposed. This method throws assigned finite automata to model software project progress. Meanwhile, a system structure of monitoring software project progress based on assigned finite automaton and monitoring algorithm are given. The case shows, this method can monitor software project progress effectively.

This work was supported in part by the National Natural Science Foundation (61273118), the Provincial Major Program of Guangdong Province (2014KZDXM033), the Public Welfare Research and Capacity Building Project of Guangdong Province (2015A030402006) of China.

## References

- [1]. Wallae L, Kei l M. Software Project Risks and There Effect on Outcomes. Communications ACM. Vol. 47 (2004) No. 4, p. 68-73.
- [2]. Bing Kuang. Study on The Method of Software Project Management Plan and Control (Master, Guizhou University, China 2006). p 20-25.
- [3]. Wenjie Zhao. A Software Project Supervision and Control Model Research and Application. Computer CD Software and Applications. Vol.1(2015), No.1:p.77-79.
- [4]. Xiaoming Cui, Li Ma. Research of Software Project Schedule Controlling Methods. Computer Engineering and Design. Vol.31 (2010) No.12, p. 2754-2761.
- [5]. LiMing Ao, Shuda Wang, Yonghong Guo. A Developing Model for Software Workflow Based Finite Automation. Computer Engineering and Applications. Vol.40 (2004) No.28, p. 208-211.
- [6]. Xiancheng Zhu: Research on Software Development Model Based on Finite Automata in the Field of Industrial Control (Master, Huazhong University of Science and Technology, China 2007).

[7]. Bing Zhang: Research on Team Softy Process Modeling Methods and Process Definition Reuse(Master, Harbin Engineering University, China 2009).