International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.3 (2016), pp.57-66
http://dx.doi.org/10.14257/ijmue.2016.11.3.06

Model Checking of Non-Centralized Automaton Web Service with AMT Bounded Constraint

Zhang Rui-Min[1] and Li Xiao-Bin[1]

1. Lanzhou Institute of Technology, Department of Software, Gansu, Lanzhou, 730050, China

Abstract

In view of the web service model checking application, the combination mode of traditional finite state machines cannot guarantee the correctness of Web composite service. A web service model diction algorithm of non-centralized automaton based on satisfiability modulo theories (SMT) is proposed. First, SMT is used for bounded model checking of time automaton, and the time automaton model is directly converted into logical formula that can be identified by SMT, to make solution; secondly, the proposed SMT time automaton theory is used to achieve the employee travel arrangements web services for modeling and verification; finally, through an example analysis, the effectiveness of the algorithm on the termination of the path deadlock and the optimization of network parameters.

***Keywords**: Satisfiability modulo theories (SMT); Automaton; Web service; Model checking; Big data*

1. Introduction

The available Web services presented in the Internet can make use of integrate distributed mode to achieve the service system structure of distributed application [1]. Usually based on the workflow implementation to the value-added combination of basic Web composite services, establish distribution service invocation engine of different servers. Traditional workflow is a centralized call, which needs to be called and forward to all communication information in the process of implementation, so it needs great communication overhead. Therefore, compared with centralized, non-centralized operation mode has better operating performance [1].

Time automaton is a special kind of limited automaton to describe the real-time system, with its real area time clock. This feature makes it widely used in reality, such as elevator, network, aviation and other fields. In this paper, the composition optimization of Web services can also be applied, and because of its higher security requirements, the significance of research is even higher. Although automaton has a good description of real-time system, but due to the existence of the clock leading to the infinite migrating domain, it has the problem of low efficiency especially in the web service composition optimization, and it needs to be pre-processed to the clock variables, to make it transformed into a limited migrating area [2].

Early research results for the study of the problem were concentrated in dividing the time domain into the equivalent regions. For example, Literature [3] made the use of inconsistent constraint matrix to describe the equivalent partition areas, in order to build a finite area map. Literature [4] proposed the method for the detection of the finite model region diagram. But the problem is that the number of divisions increases with the number of the clock to increase exponentially. The Literatures [5-7] were proposed to solve the problem, which were all based on the clock preprocessing, but the problem is that the need of memory capacity is high, which increased the computing burden, especially the

ISSN: 1975-0080 IJMUE

practical application of big data aiming to Web service combination optimization is limited.

2. Description of Web Composite Service Problem

2.1. Finite Automaton

The key problem of Web service composition optimization in Internet is to reduce the artificial participation. Data in the implementation transfer among multiple basic services, and Web service composition optimization can be expressed based on a group of state transition with the specific process is: to obtain input information from other services. After processing, it transfers the information for broadcasting, which is similar to a kind of input and output sequence information transmission chain, so it is very suitable for using finite automaton for model checking and expression [8]. The relevant definitions are as follows: [8-9]:

Definition 1 ： The expression form of the finite automaton five-tuple is ： $A=\left(S,S^0,I,O,\Delta\right)$， among which， S is automaton status set， $S^0 \subseteq S1$ is the initial state of collection， I 和 O is automaton's input and output set respectively, the conversion collection can be expressed as $\Delta \subseteq S\times \bar{I} \times \bar{O} \times S$ ， $\Delta = \left(s,i,o,s'\right)$ means the initial state of the transformation as $s\in S$ ， $i\in \bar{I}$ is input， $o\in \bar{O}$ is output， $s'\in S$ is the purpose conversion state, can be expressed as ： $s\xrightarrow{i/o}s'$， $\left(s,i,o,s'\right)\in\Delta$ 。

Figure 1 shows the non-centralized implementation of the finite automaton. Service $s3$ can give a basic service for $r1,r2,r3$, indicating that the non-centralized implementation of automaton regards the input or output of the entire automaton as input or output of a target automaton.

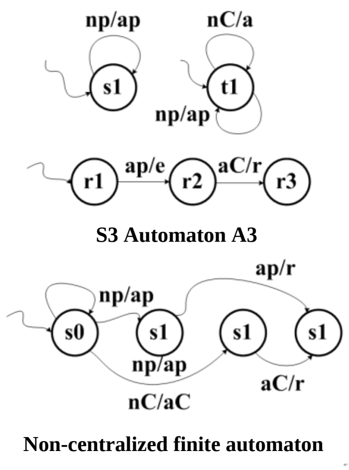
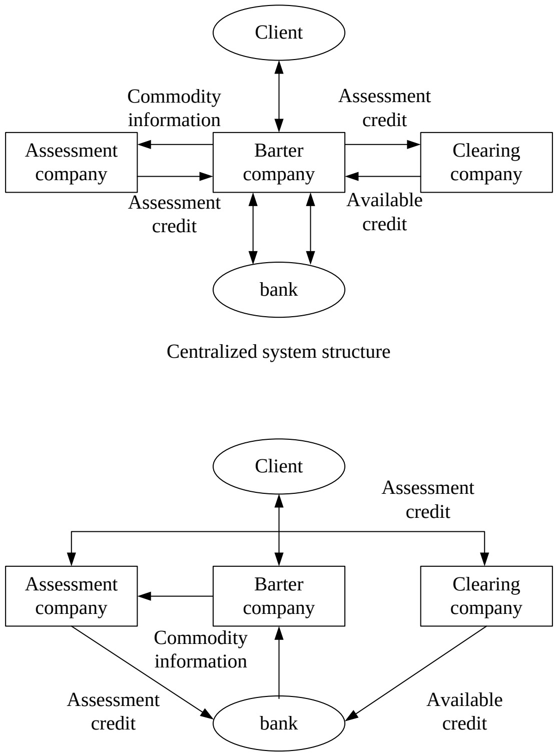


Figure 1. Non-Centralized Finite Automaton

2.2. Web Service Barter Transaction System Structure

Barter transaction system is a typical web service composition, and the non-centralized execution mode is shown as in Figure 2, with specific implementation steps: clients release goods in a barter corporate node, and products are reviewed by the barter company. The first step is to transmit commodity information through the client side to the assessment company for the amount evaluation and feedback to corporate barter. The second step is to forward the amount information through the barter company to the Clearing Corporation, after the deduction of commission, feedback the availability of credit to the barter company. The last step is for barter company to transfer the assessment and the availability of credit to the bank, for developing a line of credit for the customers. The above system is composed of three parts of Web services of the assessment companies, clearing companies and banks, and can be used as a module for the barter company to call [10].



Centralized system structure

Non-centralized system structure

Figure 2. Barter System Structure

Figure 2 shows the centralized and decentralized implementation of a web service composition to barter transactions. By comparing, it can be seen that the amount of information transmission of non-centralized barter system structure is significantly less than the centralized barter system structure, so this way of Web service composition form has higher execution efficiency.

3. SMT Time Automaton

3.1. SMT Bounded Model Checking

Given the limited model M and the nature of the model f , the model checking problem can be formulated as the model M to satisfy the model property f . The traditional model generation method generally gets larger state space, which is constrained by the hardware conditions. The traditional model checking method is not practical for the large data form of Web composite service. In order to solve this problem, the bounded detection model can effectively solve the problem. The implementation of this model detection is given by Literature [11], that is, for the problem of the common model detection $M|=Ef$, an example can be found in the limited implementation steps to meet model M to satisfy the property f . For a given k , the above model checking can build up the problem to satisfy the problem of the logic formula. If the condition $[[M, f]]_k$ is satisfied, it is proved that the above mentioned k steps are established to find a satisfying instance, and the model M is synchronously proved to satisfy the model property f . Otherwise, the above k steps cannot find satisfying example to meet the conditions, but cannot prove whether to meet the conditions of the case when $k' > k$.

Usually, the value k is gradually increased until it finds an instance to meet the condition, or to achieve the maximum execution step. This detection method does not need to carry out in the whole state space, which can effectively save the computational memory space. Therefore, in the verification of time automaton, it does not need clock preprocessing to get limited space of time automaton. In this way, it is necessary to make solution based on the logic formula, so it is necessary to transform the automaton model into a logical model, and it needs to make variables Boolean formalized, which is not conducive to the simplification of the algorithm.

Satisfiability Modulo Theories (SMT) can deal with non-Boolean variables, such as integer or real number. To this issue, this paper proposed a non-variable Boolean bounded time automaton model checking, and applied it to the optimization of Web composite services.

3.2. M Model K Step Path Logic Formula

Suppose $M = \langle A_0, A_1, \text{n}, A_n \rangle$ is time automaton model, among which, $A_i = \left(\Sigma^i, S^i, S_0^i, X^i, I^i, T^i\right)$. Make proposition be $f = EFp|AGp|$, among which $p = a|\neg p|p1 \wedge p2|p1 \vee p2$, a is the primary sub proposition in the formula. The logical formula of building time automaton is $[[M, f]]_k$, make $[[M]]_k$ expressed as a feasible path for time automaton, and the model property f expressed as the form of $[[f]]_k$.

Then under the time automaton model, finite k step sequence state form is: $(s_0, v_0) \xrightarrow{d_0} (s_1, v_1) \xrightarrow{d_1} \text{n} \xrightarrow{d_{n-1}} (s_k, v_k)$, in which, (s_0, v_0) is the initial path point of finite sequences, and the components of $[[M]]_k$ are as follows:

3.2.1. Initial Path Point

(s_0, v_0) : $Init = Init_0 \wedge \text{n} \wedge Init_{n-1}$, among which:

$$Init_i = \left[\underset{s \in s_0^f}{\vee} (s_0 = s)\right] \wedge \left[\underset{t \in X^f}{\vee} (t = 0)\right] \quad (1)$$

$$Init_i = \left[\underset{s \in s_0^f}{\vee} (s_0 = s)\right] \wedge \left[\underset{t \in X^f}{\vee} (t = 0)\right] \quad (1)$$

In formula (1), given initial path point of automaton j , and set the initial value of clock state as 0.

3.2.2. Intermediate State

(s_i, v_i) **:**

$$\begin{cases} Const_i = Const_i(0) \wedge \text{n} \wedge Const_i(n-1) \\ Const_i(j) = \left[\underset{s \in s_0^f \wedge S \neq S_\mu}{\vee} (s_0 \neq s_v)\right] \wedge I^i(S_\mu) \end{cases} \quad (2)$$

$$\begin{cases} Const_i = Const_i(0) \wedge \mathrm{n} \wedge Const_i(n-1) \\ Const_i(j) = \left[\underset{s\in s_0^f \wedge S\neq S_\mu}{\vee} (s_0 \neq s_v) \right] \wedge I^i(S_\mu) \end{cases} \tag{2}$$

In formula (2), state j automaton $Const_i(j)$ is constrained by variable i. t_i is the i state value of clock t. $t_i \geq 0$ shows non negative of the clock state. S_i is the i position of system state. The rest is used to guarantee the sole position of the automaton state and the invariance of the state $I^i(S_\mu)$.

3.2.3. Steps of Sequence i

$(s_i, v_i) \xrightarrow{d_i} (s_{i+1}, v_{i+1})$: $step_i = step_i(j) \wedge \mathrm{n} \wedge step_i(n-1)$, $i \in [0, k-1]$, in which the variables:

$$step_i(j) = \underset{T_\mu}{\vee}\left(Exl_\mu^j(i) \wedge T_\mu^j(i) \wedge T_\mu^j \wedge d_i > 0\right) \tag{3}$$

$$Exl_\mu^j(i) = \underset{T_\mu}{\wedge} \neg a_v^i \tag{4}$$

$$T_\mu^j(i) = \neg a_v^i \wedge (s_i = s_\mu) \wedge \underset{t\in X^f}{\wedge} (t_{i+1} = t_i + d_i) \tag{5}$$

$$T_\mu^j = a_v^i \wedge (s_i = s_\mu) \wedge \underset{t\in\lambda_\mu^f}{\wedge} (t_{i+1} = 0) \underset{t\notin\lambda_\mu^f}{\wedge} (t_{i+1} = t_i + d_i) \tag{6}$$

$$step_i(j) = \underset{T_\mu}{\vee}\left(Exl_\mu^j(i) \wedge T_\mu^j(i) \wedge T_\mu^j \wedge d_i > 0\right) \tag{3}$$

$$Exl_\mu^j(i) = \underset{T_\mu}{\wedge} \neg a_v^i \tag{4}$$

$$T_\mu^j(i) = \neg a_v^i \wedge (s_i = s_\mu) \wedge \underset{t\in X^f}{\wedge} (t_{i+1} = t_i + d_i) \tag{5}$$

$$T_\mu^j = a_v^i \wedge (s_i = s_\mu) \wedge \underset{t\in\lambda_\mu^f}{\wedge} (t_{i+1} = 0) \underset{t\notin\lambda_\mu^f}{\wedge} (t_{i+1} = t_i + d_i) \tag{6}$$

In the above formulas, $step_i$ is the migrating process i of the model, which is actually the union of migrating $step_i(j)$ of all automatons in the step i. d_i is the clock delay of the system. Formula (4) is the exclusive internal migration of all automatons, and this limit ensures that each automaton can only move one step in a cycle. While the formula (5) and the formula (6) gives the delay migration and the action migration respectively. According to $step_i$, the path of automaton on the k step can be got as $Path_k = \wedge \underset{i\in[0,k]}{\wedge} step_i$.

So the logic formula model of time automaton can be expressed as:

$$[[M, f]]_k = Init \wedge \underset{i\in[0,k]}{\wedge} Const_i \wedge Path_k \tag{7}$$

$$[[M, f]]_k = Init \wedge \underset{i\in[0,k]}{\wedge} Const_i \wedge Path_k \tag{7}$$

3.3. The Solution of SMT Logical Formula

Based on the reachability of logic formula model, $f = EFp$ is the set path of automaton running state, thus the following conditions $[[f]]_k = \underset{i\in[0,k]}{\vee} [[p]]_k$ can be set up, in which, $[[p]]_k$ can be based on logic transform, such as $[[\neg a \wedge (b \vee c)]]_k = \neg a_i \wedge (b_i \vee c_i)$. While for the security $f = AGp$ of the model, it is just the opposite of the reachability of model, which shows that all states p of the path are established. And this can be used to realize the mutual transformation between the reachability and the security of the model through reverse operation. If the model property $f' = EF\neg p$ is established, then the path satisfies the model property $\neg p$, indicating that the model property f is not established. On the contrary, the dissatisfaction f' synchronously proves the establishment of automaton model property f .

SMT is a practical tool to solve the logical proposition, which can be used to solve the logical problem with non-Boolean variables. For instance, SMT algorithm not only can be used to solve the logical problem like $\neg a \wedge (b \vee c)$, but also can solve the logical problem of non-Boolean variables such as $(a+b>3)\wedge(a>2)$ containing a integer, to simplify the model test.

The logic verification process takes the model M, the model property f and the upper bound of the execution steps K as the input, for $k<K$, the formula for its corresponding logic problem is $[[M,f]]_k = [[M]]_k \wedge [[f]]_k$; based on SMT tools to solve, if the output of the result is unsat, then $k=k+1$, until the output as sat is found or satisfied the upper bound K of execution steps. Is the result of sat cannot be found, then it indicates that in the process of K execution, the security is satisfied while the reachability is not satisfied; conversely, it indicates that the reachability is satisfied while the security is not satisfied.

4. Experiment and Result Analysis

4.1. Algorithm Model Checking

The algorithm of Literature [12] was selected to compare with the experiment, which is characterized by the synchronous modeling of the automaton and clock. Its application is quite extensive, such as the system with the clock which can realize the sharing mechanism of multi thread system variables. The computer hardware conditions for the experiment comparison are set as follows: CPU processor is Intel i3 2.4GHz; internal memory is 6G RAM, operating system is Ubuntu, and the test object is selected as shown in Figure 3. The comparison results of running time is seen in Table 1.

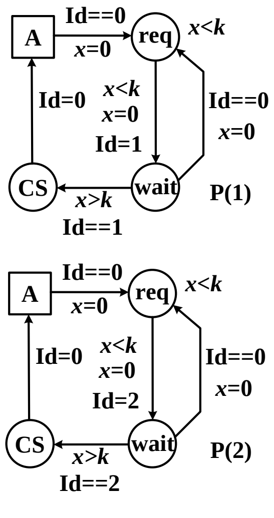


Figure 3. Test Object

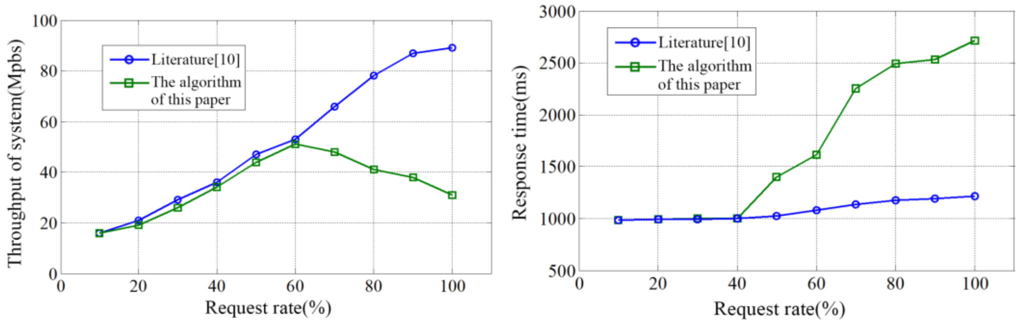
Table 1. Comparison of Running Time

Synchronization Process	Algorithm of this paper	Algorithm of Literature [10]
3	0m0.324s	0m0.025s
4	0m0.236s	0m0.053s
5	0m0.323s	0m0.055s
6	0m0.442s	0m0.112s
7	0m0.712s	0m0.298s
8	0m1.464s	0m1.357s
9	0m3.367s	0m5.795s
10	0m8.258s	0m24.653s
11	0m25.580s	1m37.951s
12	2m49.638s	6m43.834s
13	9m14.344s	Time>60m

Table 1 gives the comparison of the results obtained by the experimental results of this algorithm with the results of the Literature [12] algorithm. The experimental results show that the computation time of time automaton model checking based on SMT algorithm is less than that of the Literature [12] algorithm. The advantages are not obvious under the condition when parallel processes is less, but with the increase of the process numbers of automaton model, the computation time of time automaton model checking based on SMT algorithm is obviously better than Literature [12] algorithm. This shows that the algorithm of this paper is more suitable for such large data problem processing of Web service composition optimization.

4.2. Web Service Application Experiment

Simulation examples use the barter system structure like Figure 2. In Web service combination optimization problems, service response time and network throughput are two commonly used evaluation indexes. In comparison with the performance of the web service composition optimization model and the Literature [12] timed automaton model in the barter transaction shown in Figure 2, the results of simulation comparison are shown as in Figure 4.



(a) Throughput of System (b) Response Time

Figure 4. Comparison of Response Indexes in Barter System

Figure 4 shows the comparison of the system throughput and response time by the algorithm of this paper and the Literature [12] algorithm in the barter transaction system. From Figure 4 (a), it can be seen that there is little difference of system throughput index between the proposed algorithm of this paper and the Literature [12] algorithm in the low request rate. When the request rate reaches 60%, the processing performance of the Literature [12] algorithm reaches the limit and then the latter requests are in the waiting state; and with the increase of requests, the amount of system throughput is gradually reduced. Conversely, the system throughput of this paper is gradually increasing with the increase of the requests, showing the smooth of the network from one side.

In the comparison indexes of network response time, when the number of requests is small, the network response time indexes of the algorithm of this paper and the Literature [12] algorithm have little difference; when the request rate reaches the 40% of the system capacity, the response time of the Literature [12] algorithm is rapidly increased with the increase of request rate, while the response time of this paper's algorithm has little change, which indicates the running smooth of the network system.

5. Conclusion

This paper is based on SMT to carry out a bounded model checking method for time automaton; the web service composition problem is optimized, and based on the non-centralized control mode, the finite automaton model is implemented by the non-centralized model. The basic communication process of Web services is realized through the process of the automaton state migration. Based on the closure of automaton transfer mode to achieve the simulation of each state, and the service, the target and the communication cost are taken as input parameters of the model. By comparing the simulation results, it shows the effectiveness of the proposed algorithm in the barter model application, verifies the performance advantages of the algorithm of this paper. Because there is still the problem of access rights among the services in the practical application, it has certain effect of uncertainty on the actual performance. In the future, the research direction is the composition optimization of a large number of Web services.

References

[1] Z. Lv, C. Esteve, J. Chirivella and P. Gagliardo, "Clinical Feedback and Technology Selection of Game Based Dysphonic Rehabilitation Tool", 2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth2015), IEEE, **(2015)**.

[2] W. Ke, "Next generation job management systems for extreme-scale ensemble computing", Proceedings of the 23rd international symposium on High-performance parallel and distributed computing. ACM, **(2014)**.

[3] L. Tonglin, "Distributed Key-Value Store on HPC and Cloud Systems", 2nd Greater Chicago Area System Research Workshop (GCASR), **(2013)**.

[4] T. Su, W. Wang, Z. Lv, W. Wu and X. Li, "Rapid Delaunay Triangulation for Random Distributed Point Cloud Data Using Adaptive Hilbert Curve", Computers & Graphics, **(2015)**.

[5] J. Hu, Z. Gao and W. Pan, "Multi-angle Social Network Recommendation Algorithms and Similarity Network Evaluation", Journal of Applied Mathematics, vol. 2013, **(2013)**.

[6] Z. Lv, T. Yin, Y. Han, Y. Chen and G. Chen, "WebVR——web virtual reality engine based on P2P network", Journal of Networks, vol. 6, no. 7, **(2011)**, pp. 990-998.

[7] J. Yang, B. Chen, J. Zhou and Z. Lv, "A portable biomedical device for respiratory monitoring with a stable power source", Sensors, **(2015)**.

[8] Z. Dongfang, "FusionFS: Toward supporting data-intensive scientific applications on extreme-scale high-performance computing systems", Big Data (Big Data), 2014 IEEE International Conference on. IEEE, **(2014)**.

[9] S. Dang, J. Ju, D. Matthews, X. Feng and C. Zuo, "Efficient solar power heating system based on lenticular condensation", Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on, **(2014)**.

[10] X. Zhang, Y. Han, D. Hao and Z. Lv, "ARPPS: Augmented Reality Pipeline Prospect System", 22th International Conference on Neural Information Processing (ICONIP), Istanbul, Turkey. In press, **(2015)**.

[11] J. Hu and Z. Gao, "Distinction immune genes of hepatitis-induced heptatocellular carcinoma", Bioinformatics, vol. 2, no. 24, **(2012)**, pp. 3191-3194.

[12] W. Ke, "Overcoming Hadoop Scaling Limitations through Distributed Task Execution".
[13] Z. Su, X. Zhang and X. Ou, "After we knew it: empirical study and modeling of cost-effectiveness of exploiting prevalent known vulnerabilities across IAAS cloud", Proceedings of the 9th ACM symposium on Information, computer and communications security. ACM, **(2014)**.
[14] Z. Lv, A. Halawani, S. Feng, H. Li and S. U. Rehman, "Multimodal Hand and Foot Gesture Interaction for Handheld Devices", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 11, 1s, Article 10, **(2014)**, pp. 19.
[15] J. He, Y. Geng, F. Liu and C. Xu, "CC-KF: Enhanced TOA Performance in Multipath and NLOS Indoor Extreme Environment", IEEE Sensor Journal, vol. 14, no. 11, **(2014)**, pp. 3766-3774.
[16] S. Zhou, L. Mi, H. Chen and Y. Geng, "Building detection in Digital surface model", 2013 IEEE International Conference on Imaging Systems and Techniques (IST), **(2012)**.
[17] N. Lu, C. Lu, Z. Yang and Y. Geng, "Modeling Framework for Mining Lifecycle Management", Journal of Networks, vol. 9, no. 3, **(2014)**, pp. 719-725.
[18] Y. Geng and K. Pahlavan, "On the accuracy of rf and image processing based hybrid localization for wireless capsule endoscopy", IEEE Wireless Communications and Networking Conference (WCNC), **(2015)**.
[19] G. Liu, Y. Geng and K. Pahlavan, "Effects of calibration RFID tags on performance of inertial navigation in indoor environment", 2015 International Conference on Computing, Networking and Communications (ICNC), **(2015)**.
[20] J. He, Y. Geng, Y. Wan, S. Li and K. Pahlavan, "A cyber physical test-bed for virtualization of RF access environment for body sensor network", IEEE Sensor Journal, vol. 13, no. 10, **(2013)**, pp. 3826-3836.
[21] M. Zhou, G. Bao, Y. Geng, B. Alkandari and X. Li, "Polyp detection and radius measurement in small intestine using video capsule endoscopy", 2014 7th International Conference on Biomedical Engineering and Informatics (BMEI), **(2014)**.
[22] X. Song and Y. Geng, "Distributed community detection optimization algorithm for complex networks", Journal of Networks, vol. 9, no. 10, **(2014)**, pp. 2758-2765.
[23] J. Yang, S. He, Y. Lin and Z. Lv, "Multimedia cloud transmission and storage system based on internet of things", Multimedia Tools and Applications, **(2016)**.

Author

Zhang Rui-Min, She was born in 1978. Now she is lectrue in Lanzhou Institute of Technology. Her research interest is mainly in the area of Computer Software, Mechanical and Electrical Integration. She has published several research papers in scholarly journals in the above research areas and has participated in several books.