

A Petri Net Representation of a Web-Service-Based Emergency Management System in Railway Station

Suparna Karmakar, Ranjan Dasgupta

Abstract—Railway Stations are prone to emergency due to various reasons and proper monitor of railway stations are of immense importance from various angles. A Petri-net representation of a web-service-based Emergency management system has been proposed in this paper which will help in monitoring situation of train, track, signal etc. and in case of any emergency, necessary resources can be dispatched.

Keywords—Business process, Petri net, Rail Station Emergency Management, Web service based system

I. INTRODUCTION

WEB service based application system that takes the advantages of resource sharing is a very useful technique to solve more complex operation. Many researches have been done to find the method relevant to web services such as- WSDL[3] (Web Service Description Language), WSCL[3] (Web Services Conversation Language), BPEL4WS[3] (Business Process Execution Language for Web Services), WFSL[3] (Workflow Specification Language) etc. But all these approaches are xml-based description methods, which focuses on the description of composite web services from physical perspective. And as a result the logical details of web service based systems may be ignored and the correctness of syntax, semantic and pragmatic level is not reliable. Hence it is very much important to describe the web services logically before describing them physically.

In the paper [3], the authors have defined the way to model a web service based system using Petri net [6] for describing the composite web services logically. Also they have stated the usefulness of modeling web service based system and describing web services and their composition logically using Petri net[7] over other methods such as UML based method[10] and Workflow based methods[8,9]. And at the end of this paper they have shown the use of modeling method which they have suggested in a web service based

Suparna Karmakar, is with Guru Nanak Institute Of Technology (under West Bengal University Of Technology), 157/F Nilgunj road Panihati, Kolkata-700114, West Bengal, India. (phone: 09732625277; e-mail: karmakarsuparna27@gmail.com).

Dr. Ranjan Dasgupta is with National Institute of Technical Teachers' Training & Research, Block – FC, Sector – III, Salt Lake City, Kolkata 700 106, INDIA (e-mail: ranjandasgupta@hotmail.com).

application namely Airport Emergency Management System and using the reachability analysis technique of Petri net they have proved the correctness of their model.

When modeling the airport emergency management system they have not described about the emergency event in details, i.e. what kind of events are happening there & among them which events can be called as emergencies are not specified. Only they have stated that the system is monitoring the emergency events and in parallel they are preparing for handling this situation. And as they have not specified the kinds of emergency events it cannot be ensured whether the dispatched resources (such as medical treatment resources, resources for extinguishing fire, anti-hijacking resources, cleaning spot resources) are enough to handle the emergency situation or not.

In our research work we propose a web service based Rail station emergency management system. In this system we have considered extensively different monitoring system like-monitoring the train, monitoring the track, monitoring the signal, monitoring the weather condition, monitoring the electrical connection, monitoring the station & surroundings and it is checked whether there is any abnormalities happening in those monitored events. And to handle emergency situation, resources (such as rescue team resources, medical treatment resources, repairing team resources, cleaning spot resources) are dispatched. The decision taken for dispatching resources are taken according to the emergency occurred.

II. PETRI NET THEORY

Petri nets were developed in the early 1960s by Carl Adam Petri [1]. The formalized definition can be found in [2] [4]. Petri nets are a graphical and mathematical modeling tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, parallel, nondeterministic and/or stochastic. As a graphic instrument, it owns a graphically depicting function and simulates the dynamic behaviors of the system through the flow of tokens. And as a mathematical tool, it depicts system behaviors through building state equation (analysis of dynamic behavior can be found in [1][5]). Petri nets are defined by a quadruple (S, T, F, W) , where

1. S is a finite set of places, represented by circle or ovals.
2. T is a finite set of transitions, represented by squares or rectangles.

3. $F \subseteq \{S \times T\} \cup \{T \times S\}$ is the flow relation. Arcs connect places and transitions. Inward arcs go from a place to a transition and an outward arc goes from a transition to a place.
4. $W: F \rightarrow \mathbb{N} - \{0\}$ is the weight function, which associates a nonzero natural value to each element of F . If no weight value is explicitly associated with a flow element, the default value 1 is assumed for the function.

Tokens reside in the places. They are represented by a solid circle or by a dot inside of a place. The execution of Petri net is controlled by the position and movement of tokens. A Petri net is a kind of directed graph with initial marking M_0 . A marking (state) assigns to each place a positive integer. If a marking assigns a non-negative integer k to a place s , then s is said to be marked with k tokens. A marking is denoted by M , a m -vector. The s^{th} component is denoted by $M(s)$, which represents the number of tokens at place s [1].

A. Transition Enabling Rule:

A transition t is said to be enabled if each input place s of t is marked with at least $w(s, t)$ tokens, where $w(s, t)$ is the weight of the arc from s to t . For a given marking the firing of an enabled transition results in a successor marking M' ,

Where,

$$M'(s) = M(s) - w(s, t) + w(t, s) \quad (1)$$

$w(s, t)$ is the weight of the arc between s and t . $w(t, s)$ is the weight of the arc between t and s .

III. COMPOSITION OPERATION OF PETRINET:

A. Sharing Composition Operation: Suppose that a net system is defined as $N_i = (S_i, T_i, F_i)$, where $i=1,2$.

If $S_1 \cap S_2 \neq \emptyset$ & $T_1 \cap T_2 = \emptyset$, then we call $N = (S_1 \cup S_2, T_1 \cup T_2, F_1 \cup F_2)$, is the sharing composition net [3] of net N_1 and net N_2 , which is represented by $N = N_1 \text{ C}_s N_2$.

The composite initial marking $M_0(s)$ for the composed net is represented as

$$M_0(s) = \begin{cases} M_{0i}(s), & s \in S_i - (S_1 \cap S_2), (i=1,2) \\ M_{01}(s) + M_{02}(s) - 1, & s \in S_1 \cap S_2 \end{cases} \quad (2)$$

where, $M_{0i}(s)$ is the initial marking of the i^{th} Net

B. Synchronous Composition Operation: Suppose that a net system is defined as $N_i = (S_i, T_i, F_i)$, where $i=1,2$.

If $S_1 \cap S_2 = \emptyset$ & $T_1 \cap T_2 \neq \emptyset$, then we call $N = (S_1 \cup S_2, T_1 \cup T_2, F_1 \cup F_2)$ is the synchronous composition net [3] of net N_1 and net N_2 . The composite initial marking $M_0(s)$ for the composed net is represented as

$$M_0(s) = \begin{cases} M_{0i}(s), & s \in S_i - (S_1 \cap S_2), (i=1,2) \\ M_{01}(s) + M_{02}(s) - 1, & s \in S_1 \cap S_2 \end{cases} \quad (3)$$

where, $M_{0i}(s)$ is the initial marking of the i^{th} Net

IV. DEFINING VARIOUS COMPOSITION OPERATION USED FOR MODELING WEB SERVICE BASED SYSTEM

Here, some simple examples are given to introduce various composition operation that are useful in modeling the web service based system. Here circle, transition, arc and a black dot is used to represent a resource, a web service and information flow respectively.

A. Sequential Operation: The way two or more services can be executed sequentially and what will be their sequence ordering can be represented using sequential operation. For example in fig1 transition t_2 can fire only after the firing of t_1 . This imposes the precedence constraint “ t_2 after t_1 .” And by this way two transitions fire in sequence.

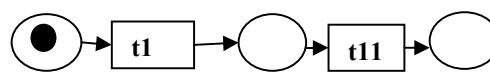


Fig. 1 Sequence operation

B. Concurrent Execution: Parallel activities or concurrency can be easily expressed in terms of Petri nets. For example, in the Petri net shown in Fig. 2, the parallel activities represented by t_2 and t_3 starts at the firing of t_1 and ends at the firing of t_4

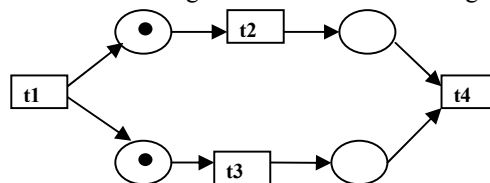


Fig. 2 Concurrent operation

C. Choice Operation: If it is required to execute a particular service among a collection of services at a time, then it is represented using choice operation of the Petri net. For example in fig3 at a time $t_1/t_2/t_3$ can be fired as their input place has token. But together they cannot fire at a time.

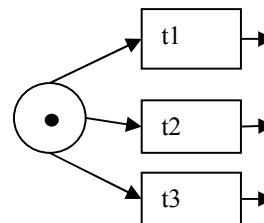


Fig. 3 Choice operation

V. REPEATING OPERATION

Sometime it is required to execute a certain transition or a group of transitions repeatedly under certain condition, then it is represented using repeating operation of the Petrinet. For example, in fig4 the service t2 is executing repeatedly under certain condition.

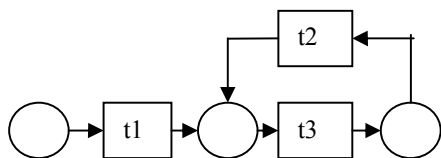


Fig. 4 Repeating operation

VI. MAPPING RELATIONSHIP BETWEEN THE MODEL AND THE PETRI NET SYSTEM

The mapping relationships [3] between the model and Petri net are described as follows:

In the process of modeling a web service based system using Petrinet,

- i. a transition of a Petri net has been used to indicate web services which are defined as the accessible programs and devices distributed on the web and can be invoked and employed by other application programs,
- ii. places are used to represent resource, which is the input/output state or pre-condition / post- condition of web services. If there are any other pre-condition in addition, then it must be specified beside the arrow linking the web service and the resource,
- iii. token represents the presence of resource and also represent whether this resource is ready to be processed or not,
- iv. flow relation has been used to represent the information flows between a web service and a resource,
- v. synchronous composition operation has been used to represent concurrent operation,
- vi. sharing composition operation having the restriction that the number of the sharing resource should not be larger than one has been used to represent choice operation. This restriction need to maintain through the entire design process, and
- vii. a composite operation by the sharing composition operation and the synchronous composition operation has been used to represent the repeating operation.

VII. CONSTRUCTION OF THE PETRINET MODEL FOR EMERGENCY MANAGEMENT IN RAILWAY STATION

For constructions of the Petrinet model for emergency management in railway station, the business process is analyzed first. Then top business process is mapped to the service flow of the top level and then top service flow has been decomposed to identify the partial service flows of the lower level according to business process. This recursive procedure will continue until the basic service flow has been identified. Here the business activities of the rail station emergency

management have been identified using this procedure and encoded. For example, t1 denotes the first service of the first level. t2 denotes the second service which can be made by decomposing the service t1. We construct the service flows of every level by the composition operations such as sequential, concurrent, choice, etc. explained before.

A. Identification of the business process of the emergency management in railway station

The business process of the rail station emergency management consists of some sub processes. These are emergency preparation, emergency response and emergency restoration. Now these sub processes need to decompose to find partial service flow. The business process of the emergency preparation consists of monitoring the events (like - position of train, track, signal, weather condition, station & surroundings), analyzing the monitoring information and preparing for the emergency scheme. The business process of the emergency response consists of deciding on the emergency action, dispatching resources and succoring the spot. The business process in the restoration stage consists of compensating the victim, releasing the information related to the emergency event, evaluating the event and performing feedback. Resources dispatching consists of dispatching the medical treatment resource, rescue team resources, repairing team resources and spot-cleaning resource.

1. Service flow of the first level

The services of the first level are shown in table1. Services t1(Emergency Preparation), t2 (Emergency Response) and t3(Emergency Restoration) will be executed in sequence. And thus the service flow of the first level has been constructed using the sequence operation as shown in fig.5.

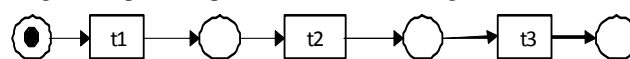


Fig. 5 The service flow of the first level

2. Service Flow Of The Second Level

In this level for finding the partial service flow, the services of the first level has been decomposed. The services by decomposing t1 are shown in table1. Service t11(Monitoring the event) and t12(Analyzing the monitoring event)will be executed in sequence. Hence these services can be composed using sequence operation. The composition result of t11 and t12 will be executed by repeating operation if the analysis result shows that the event occurred is not the emergency event. Otherwise, the composition result of t11 and t12 will be composed by concurrent operation with the service t13 (Preparing for the emergency). The service flow constructed is shown in fig 6 where t1c1, t1c2 and t1c3 are auxiliary services. The services by decomposing t2 are shown in table1. Services t21(Deciding on the emergency action), t22(Dispatching resources) and t23(Succoring the spot) will be executed in sequence, so we can construct the service flow using sequence operation as shown in Fig.7.

The services by decomposing the service t3 are shown in table1. Services t33(Evaluating the event) and t34 (Performing feedback)are composed by sequence operation and the result is composed with t31(Compensating the victim) and t32(Releasing the information) by concurrent operation. The service flow constructed is shown in Fig8 where t3c1 and t3c2 are two auxiliary services.

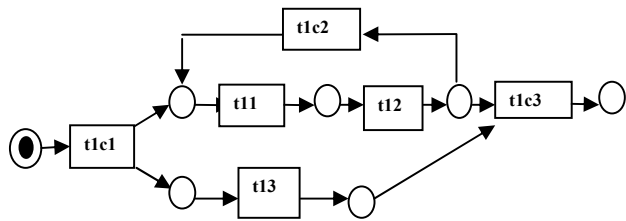


Fig. 6 The service flow by decomposing t1

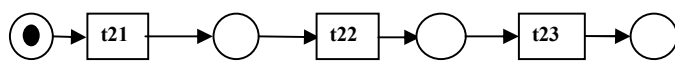


Fig. 7 The service flow by decomposing t2

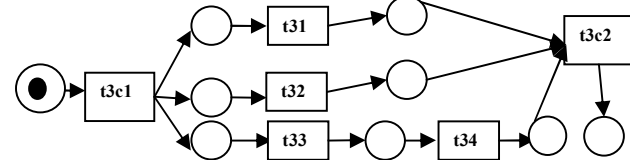


Fig. 8 The service flow by decomposing t3

3. Service Flow of The Third Level

The services by decomposing t11 are showing in table1. Here services t111(Monitoring the train), t112 (Monitoring the signal), t113(Monitoring the railway track), t114(Monitoring the electrical connections), t115 (Monitoring the weather), t116(Monitoring the station & surroundings), will be executed in sequence with t117(Monitoring whether accident occurred), t118(Monitoring whether need to repair signal), t119(Monitoring whether need to repair railway track), t11A(Monitoring whether need to repair connection), t11B(Monitoring whether need to control against the weather condition), t11C(Monitoring whether need to control against any unwanted situation in station & surroundings.) respectively. Hence, we can compose them by sequence operation. All theses sequence operations are executing concurrently. Thus the service flow constructed is showing in figure 9.The services by decomposing t22 are shown in table1.The logical relationship is to choose one or more services executed from t221(Dispatching medical treatment resources),t222(Dispatching rescue team resources), t223(Dispatching repairing team resources) and t224(Dispatching cleaning spot resources), so we can construct the service flow shown as in Fig. 10.

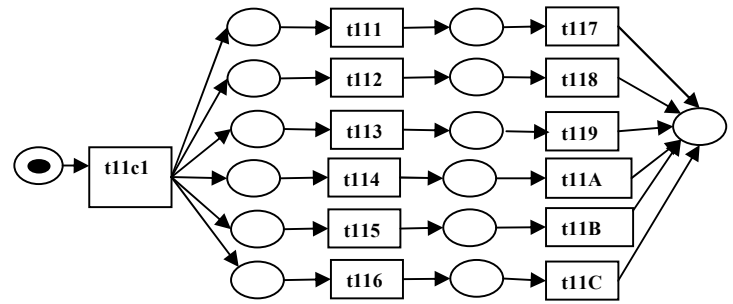


Fig: 9 The service flow by decomposing t11

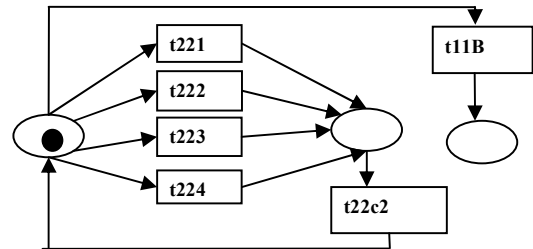


Fig: 10 The service flow by decomposing t22

4. Service Flow of The Fourth Level

The services by decomposing t111 are shown in table1 .The logical relationship is to choose one or more services executed from t1111(Monitoring the entering train at station), t1112(Monitoring the train at station), t1113(Monitoring the train leaving the station). So we can construct the service flow shown as in Fig. 11.

The services by decomposing t112 are shown in table1. The logical relationship is to choose one or more services executed from t1121 (Monitoring whether the signal invisible), t1122(Monitoring whether the signal damaged),so we can construct the service flow shown as in fig. 12.

The services by decomposing t115 are shown in table1.The logical relationship is to choose one or more services executed from t1151(Monitoring heavy rainfall), t1152(Monitoring heavy fog), t1153 (Monitoring heavy snowfall) , so we can construct the service flow shown as in Fig. 13.

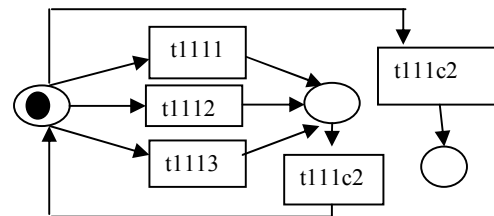


Fig. 11 The service flow by decomposing t111

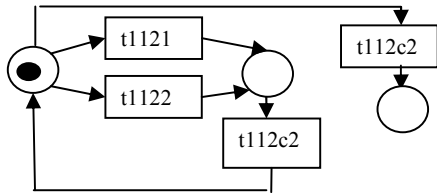


Fig. 12 The service flow by decomposing t112

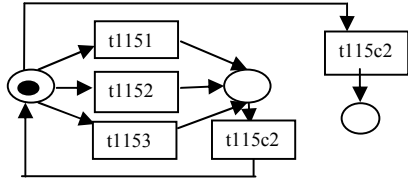


Fig. 13 The service flow by decomposing t115

B. Constructing the logical model of the web service-based rail station emergency management system

By replacing the upper services by the corresponding lower service flows respectively, the logical model of the web-service based rail station emergency management system has been constructed, shown as in Fig. 14. The next step is to delete some auxiliary services in order to simplify the model by ensuring the original logical relationship correctly. The simplified model is shown in Fig.15

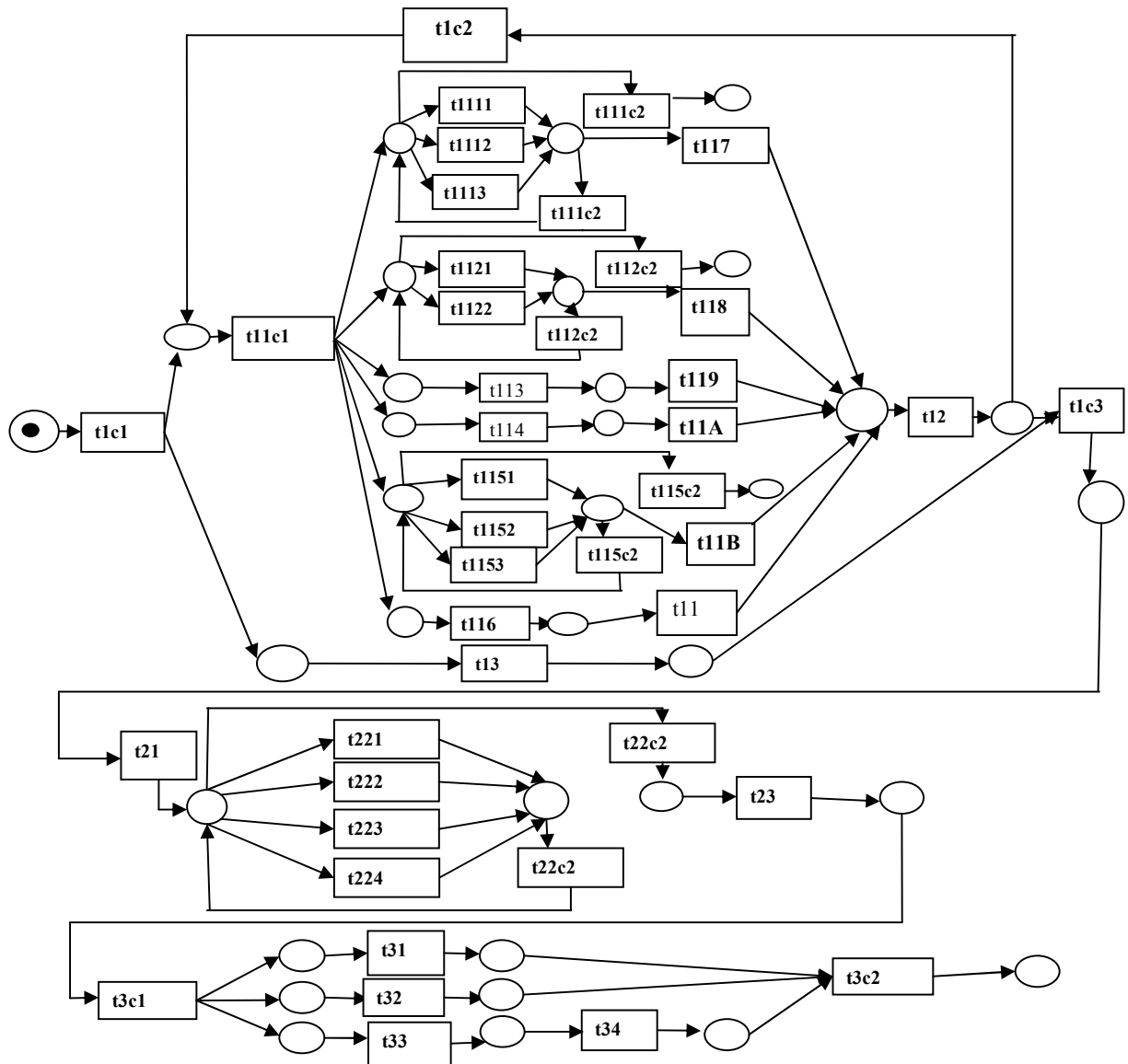


Fig. 14 The model of the emergency management in rail station

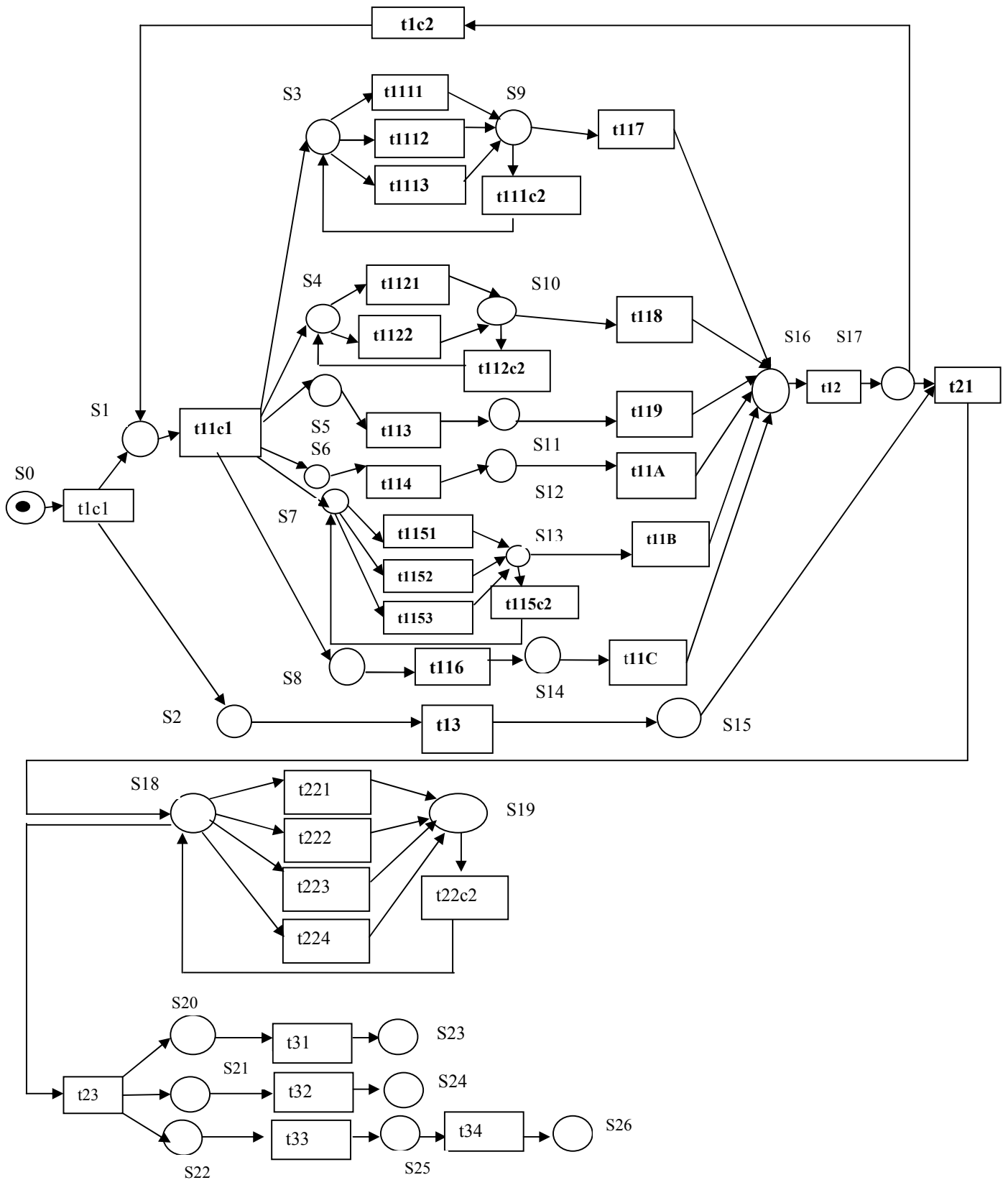


Fig. 15 The reduced model of the Emergency Management in Railway Station

TABLE I
SERVICES AND THEIR DENOTATIONS

<u>t_i</u>	<u>Service Name</u>
t1	Emergency Preparation
t2	Emergency Response
t3	Emergency Restoration
t11	Monitoring the event
t12	Analyzing the monitoring event
t13	Preparing for the emergency
t21	Deciding on the emergency action
t22	Dispatching resources
t23	Succoring the spot
t31	Compensating the victim
t32	Releasing the information
t33	Evaluating the event
t34	Performing feedback
t111	Monitoring the train
t112	Monitoring the signal
t113	Monitoring the railway track
t114	Monitoring the electrical connections
t115	Monitoring the weather
t116	Monitoring the station & surroundings
t117	Monitoring whether accident occurred
t118	Monitoring whether need to repair signal
t119	Monitoring whether need to repair railway track
t11A	Monitoring whether need to repair connection
t11B	Monitoring whether need to control against the weather condition
t11C	Monitoring whether need to control against any unwanted situation in station & surroundings.
t221	Dispatching medical treatment resources
t222	Dispatching rescue team resources
t223	Dispatching repairing team resources
t224	Dispatching cleaning spot resources
t1111	Monitoring the entering train at station
t1112	Monitoring the train at station
t1113	Monitoring the train leaving the station
t1121	Monitoring whether the signal invisible
t1122	Monitoring whether the signal damaged
t1151	Monitoring heavy rainfall
t1152	Monitoring heavy fog
t1153	Monitoring heavy snowfall

VII. CONCLUSION

Petri net being one of the most useful graph-model for representing various complex systems, we have chosen Petri net to represent web service based emergency management system in Railway station. In this paper we have designed the system and analysed possible emergency situations. Due to shortage of space we have not included the reachability analysis for ensuring the deadlock free-ness of the model.

REFERENCES

[1] Tadao Murata, "Petri Nets: Properties, Analysis and Application", Proceedings of the IEEE, Vol.77, No.4, April 1989

[2] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli, "Fundamentals of Software Engineering", Second Edition, Pearson Education, Inc., 2003

[3] "A Petri Net Theory-Based Method for Modeling Web Service Based Systems" Rendong Han; Kecheng Liu; Yanbing Ju; Jinghua Zhao, Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on ; Digital Object Identifier: 10.1109/ WiCom.2008.2839, Publication Year: 2008; Page(s): 1-7.

[4] Yuan Chongzhi, Principle and Application of Petri net[M]. Beijing: Publishing House of Electronic Industry, 2005:40-66

[5] "Petri Net Model of Session Initiation Protocol and Its Verification"-Yang Peng; Yuan Zhanting; Wang Jizeng; Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on Digital Object Identifier: 10.1109/WICOM.2007.466 Publication Year: 2007; Page(s): 1861-1864.

[6] Zhang J. Chang C K, Chung J Y, Kim S W. "WS-net: A Petri net-based specification model for web service," ICWS'04, 2004.

[7] Milanovic N and Malek M, "Current solutions for web service composition," IEEE Internet Computing(S1089- 7801), IEEE, 2004, 8(6):51-59.

[8] Karakoc E., Kardas K. and Senkul P., "A workflow-based Web Service composition system," 2006 IEEE/WIC/ACM International Conference on Web Intelligence International Intelligence Agent Technology Workshops, 2006, p 4.

[9] Shi Yuliang, Zhang Liang and Shi, Baile, "A workflow model based on cooperation of web services," Proceedings - 2006 10th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2006, 2006, p 740-745.

[10] Cambroner M. Emilia, Diaz Gregorio, Pardo J. Jose and Valero Valentin, "Using UML diagrams to model real-time Web services," Second International Conference on Internet and Web Applications and Services, 2007, pp. 4222926.