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 10 (2): 367-375, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.367.375

Cross-organization Task Coordination Patterns of Urban Emergency Response Systems

Decun Meng, Qingtian Zeng, Faming Lu, Jian Sun and Jiufang An
College of Information Science and Technology,
Shandong University of Science and Technology, 266510, Qingdao, China

Abstract: In this study, Petri net is used to model and analyze cross-organization task coordination patterns of Urban Emergency Response Systems (UERS). A kind of Petri net extended with time, resource and message factors is firstly proposed to model tasks and emergency response process within a single organization. Based on this the proposed model, four kinds of cross-organization task coordination patterns within a UERS are summarized, including the sequential task pattern, the task synchronization pattern, the resource sharing pattern and the message exchanged pattern. A UERS can be modeled according to the task and emergency response process modeling methods of a single organization and the cross-organization task coordination patterns.

Key words: UERS, emergency response process, petri net, XML, synchronization

INTRODUCTION

Urban Emergency Response System (UERS) is a modernization symbol of a city (Zhong *et al.*, 2010), integrating the emergency directions and the scheduling, which deals with unexpected incidents into a management system through the integrated information network and the communications system. The unexpected incidents include the public security, the fire prevention, the hygienic first aid, the transportation, the public utility, the natural disaster and so on. A UERS realizes the unification to answer alarm, the unified command, the combined action, the rapid reaction by sharing the platform and the foundation information, providing a more convenient emergency rescue service and the related services for the resident and technical assistances for government to exactly deal with all kinds of disasters and technical supports for the public security of cities (Wei *et al.*, 2006). Based on the Markov Chain (MC) of the model, Zhong *et al.* (2010) analyzed the performance of typical UERS. Xia *et al.* (2008) constructed the framework of urban emergency response system based on Multi-Agent System and analyzed these agent groups of the system. Li and Wei (2008) presented a new model and framework of UERS based on decision-system. Tang and Li (2010) used UML to design a complete UERS model, including case diagram, class diagram, timing diagram, state diagram, component diagram and deployment diagram of the UERS. Zhong *et al.* (2003) analyzed the performance of the UERS by the Petri net. In the research on modeling and performance of the UERS, there is a little work on the coordination problem between organizations, although an emergency response process usually includes several organizations. To deal with the coordination problem between organizations is important to improve the efficiency of a complex system. There are many research results on the coordination problem between organizations in workflow systems or business process. For example, Kappel *et al.* (1998) believed that the so-called coordination is the joint effects of each independent component, so they insisted on the importance of coordination between the components. Van der Aalst (1998) modeled inter organizational workflows in terms of Petri nets and focus on techniques to verify the correctness of these workflows. In this study, we use Petri net as formal model to analyze cross-organization task coordination patterns of Urban Emergency Response Systems (UERS). A kind of Petri net extended with time, resource and message factors is firstly proposed to model tasks and emergency response process within a single organization. Based on the proposed model, four kinds of cross-organization task coordination patterns within a UERS are summarized and modeled. A UERS can be modeled according to the task and emergency response process modeling methods of a single organization and the cross-organization task coordination patterns.

TASK MODEL OF A SINGLE ORGANIZATION

Here, we first present the formal definition of an emergency task and the modeling approach for a task using Petri net.

Corresponding Author: Qingtian Zeng, College of Information Science and Technology,
Shandong University of Science and Technology, 266510, Qingdao, China

Definition 1: An emergency task is a 7-tuple:

$$< Name, Duration, Organization, MessagesReq, MessagesSent, Resources, PostTasks >$$

where:

- Name is the ID of a task and points out the content of a task
- Duration represents implementation time of a task
- Organization represents the organization to complete this task
- MessagesReq represents the message set required before the implementation of a task. In this study, if MessagesReq = ϕ, it means that the implementation of a task requires no messages
- MessageSent represents the message set sent after the implementation of a task. If MessageSent = ϕ, it means that the implementation of a task does not send any messages
- Resources represent the resource set required during the implementation of a task. If Resources = ϕ, it means that the implementation of a task requires no resources
- PostTasks represent the post task set. For any task in PostTasks of a task t, it should not be implemented until t is completed. If the post task set of t is empty, i.e., PostTasks $\subset \phi$, it means that t is an end task

In definition 1, we assume that the resources are exclusively used. If a resource is occupied or locked by one task, then the other tasks requiring this resource have to wait until the resource becomes available.

For example, a task $<T_i, D_i, O_i, \{M_{i1}\}, \{M_{i2}\}, \{R_{i1}, R_{i2}\}, \phi>$ can be represented in XML language which is shown in Fig. 1.

Petri net is a useful tool for system modeling and analysis (Zeng, 2008; Zeng and Duan, 2007; Zeng *et al.*, 2008; Wang and Zeng, 2008; Liu and Yang, 2010). In this study, Petri net is used to model tasks of a UERS. A task T_i is represented by a transition and two places of the transition respectively represent the start and the end of T_i. In order to distinguish the logic places in a Petri net, we use a double-line circle to represent a resource place or a message place. Before the implementation of a task, each resource place should contain a token, which means the resources are well prepared and available for the task. However, no message places hold a single token because the message tokens are produced during the implementation of the emergency response process. For example, Fig. 2 presents the Petri net model for the task

```
<task>
    <name>Ti</name>
    <duration>Di</duration>
    <organization>Oi</organization>
    <messagesReq>
        <message>Mi1</message>
    </messagesReq>
    <messagesSent>
        <message>Mi2</message>
    </messagesSent>
    <resources>
        <resource>Ri1</resource>
        <resource>Ri2</resource>
    </resources>
    <postTasks></postTasks>
</task>
```

Fig. 1: A task represented in XML

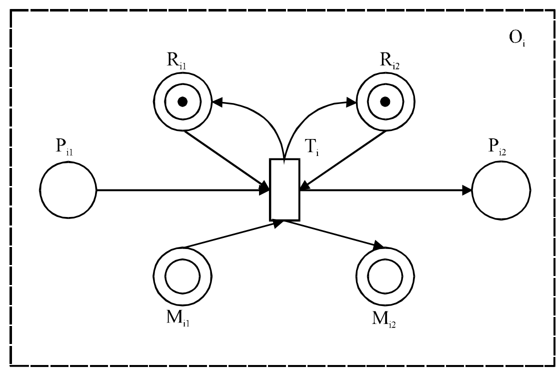

Fig. 2: Task model of a single organization

shown in Fig. 1. In Fig. 2, T_i represents the task which is corresponding to the element <name>, P_{i1} and P_{i2} represent the start and the end of the task, respectively. R_{i1} and R_{i2} represent the required resources which are corresponding to the element <resources>. M_{i1}, which is corresponding to the element <messagesReq>, represents the message received. On the other hand, M_{i2}, which is corresponding to the element <messagesSent>, represents the message sent. In Fig. 2, we can see that there are no input arcs to M_{i1}, which means M_{i1} is a message received from other tasks. And, there are no output arcs from M_{i2}, which means M_{i2} is a message sent to other tasks.

An emergency response process within an organization is composed of a set of emergency tasks, which can be formalized in an extended Petri net.

Definition 2: An emergency response process is an extended Petri net $\Sigma=<P, T, F, \tau, \lambda, M_0>$, where:

Table 1: Description of symbols in Fig 6

Name	Description	Name	Description	Name	Description
T_1	Alarm receipt	T_2	Report	T_3	Policemen race to the location
M_1	Demand handling	M_2	Instructed policemen race to the location	R_1	Police car
O_1	Police station				

```
<task>                                   <task>
    <name>T1</name>                          <name>T2</name>
    <duration>D1</duration>                  <duration>D2</duration>
    <organization>O1</organization>          <organization>O1</organization>
    <messagesReq></messagesReq>              <messagesReq></messagesReq>
    <messagesSent></messagesSent>            <messagesSent></messagesSent>
    <resources>                              <resources>
        <resource>R1</resource>                  <resource>R2</resource>
    </resources>                             </resources>
    <postTasks>                              <postTasks></postTasks>
        <postTask>T2</postTask>          </task>
    </postTasks>
</task>
```

Fig. 3: Two tasks of a single organization

- $P = P_L \cup P_R \cup P_I$, such that P_L represents the logic places between tasks, P_R represents the resource set and P_I represents the message set
- The transition set T represents the task set of an emergency response process
- $F \subseteq P \times T \cup T \times P$
- τ: $T \rightarrow D$, which represents the duration of each task, where D is a time set
- λ: $T \rightarrow O$, which represents the organization of each task, where, O is the organization
- M_0: $\forall p \in P$,

$$M_0(p) = \begin{cases} 1 & {}^{\bullet}p = \varnothing \vee p \in P_R \\ 0 & \text{Otherwise} \end{cases}$$

In this study, we assume that the tasks within a single organization are executed in sequence and there are no resource conflicts and message exchange among tasks. If two tasks $<T_1, D_1, O_1, MessagesReq_1, MessagesSent_1, Resource_1, PostTask_1>$ and $<T_2, D_2, O_2, MessagesReq_2, MessagesSent_2, Resource_2, PostTask_2>$ satisfy (1) $O_1 = O_2$ and (2) $T_2 \in PostTask_1$, it means task T_2 is one of the post tasks of T_1. In other words, we can say that T_1 and T_2 are sequential tasks within a single organization. For example, Fig. 3 shows two sequential tasks within a single organization.

To model two sequential tasks within a single organization, T_1 and T_2, we can use the model shown in Fig. 4, where the end place of T_1 and the start place T_2 are merged as one place.

If a task is not the post of any task, then it is the start of the whole emergency response process. Likewise, if a task is not followed by any post tasks, then it is the end of the whole emergency response process. The start place of a start task should contain tokens representing the start of the process and the end place of an end task is the end place of the whole process. For example, if we assume the T_1 and T_2 in Fig. 4 as the start and the end tasks of an emergency response process respectively, then the corresponding Petri net model can be shown in Fig. 4.

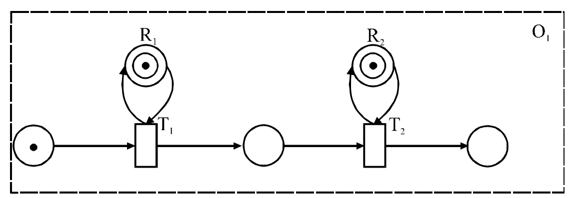

Fig. 4: Tasks model of a single organization

For example, the tasks set of a local police station in an emergency response process are shown in Fig. 5.

According to the modeling method, Fig. 6 shows the Petri net model for the emergency response process shown in Fig. 5. The meanings of symbols in Fig. 6 are shown in Table 1.

CROSS-ORGANIZATION TASK COORDINATION PATTERNS

In a UERS, there are requirements for organizations to collaborate with each other to complete a complex emergency response process. By analyzing a large number of practical cases, four kinds of basic cross-organization task coordination patterns are given

```
<taskSet>
    <task>
        <name>alarm receipt</name>
        <duration>2 minutes</duration>
        <organization>***local police station</organization>
        <messagesReq></messagesReq>
        <messagesSent></messagesSent>
        <resources></resources>
        <postTasks>
            <postTask>report</postTask>
        </postTasks>
    </task>
    <task>
        <name>report</name>
        <duration>3 minutes</duration>
        <organization>***local police station</organization>
        <messagesReq></messagesReq>
        <messagesSent>
            <message>demand handling</message>
        </messagesSent>
        <resources></resources>
        <postTasks>
            <postTask>policemen race to the scene</postTask>
        </postTasks>
    </task>
    <task>
        <name>policemen race to the scene</name>
        <duration>5 minutes</duration>
        <organization>***local police station</organization>
        <messagesReq>
            <message>instructed policemen race to the scene</message>
        </messagesReq>
        <messagesSent></messagesSent>
        <resources>
            <resource>police car</resource>
        </resources>
        <postTasks></postTasks>
    </task>
</taskSet>
```

Fig. 5: Tasks set of a local police station

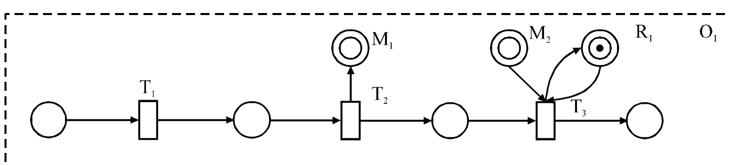

Fig. 6: The Petri net model of the local police station

here, which are the sequential task pattern, the task synchronization pattern, the resource sharing pattern, the message transmission pattern. The formal definitions and models of the four coordination patterns are presented here. For simplicity, the four coordination patterns are illustrated with only two organizations, as the patterns with more organizations are similar with it.

Sequential task pattern: If two tasks

$$< T_1, D_1, O_1, MessagesReq_1, MessagesSent_1, Resources_1, PostTasks_1 >$$

and

$$< T_2, D_2, O_2, MessagesReq_2, MessagesSent_2, Resources_2, PostTasks_2 >$$

satisfy, (1) $O_1 \neq O_2$ and (2) $T_2 \in PostTask_1$, then two organizations O_1 and O_2 are collaborated with sequential tasks T_1 and T_2.

For example, two tasks shown in Fig. 7 illustrate T_1 differs from T_2 in <organization> and T_2 belongs to the element<postTasks> of T_1. So, they are two sequential tasks between O_1 and O_2.

```
<task>                                          <task>
    <name>T1</name>                                 <name>T2</name>
    <duration>D1</duration>                         <duration>D2</duration>
    <organization>O1</organization>                 <organization>O2</organization>
    <messagesReq></messagesReq>                     <messagesReq></messagesReq>
    <messagesSent></messagesSent>                   <messagesSent></messagesSent>
    <resources></resources>                         <resources></resources>
    <postTasks>                                     <postTasks></postTasks>
    <postTask>T2</postTask>                     </task>
    </postTasks>
</task>
```

Fig. 7: Two sequential tasks

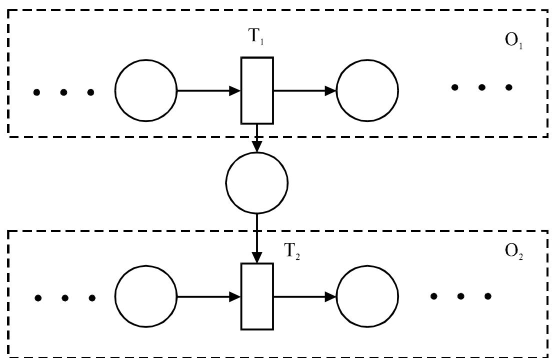

Fig. 8: Petri net model of sequential task pattern

```
<task>                                          <task>
    <name>T1</name>                                 <name>T2</name>
    <duration>D1</duration>                         <duration>D1</duration>
    <organization>O1</organization>                 <organization>O2</organization>
    <messagesReq></messagesReq>                     <messagesReq></messagesReq>
    <messagesSent></messagesSent>                   <messagesSent></messagesSent>
    <resources></resources>                         <resources></resources>
    <postTasks></postTasks>                         <postTasks></postTasks>
</task>                                         </task>
```

Fig. 9: Two synchronous tasks

To model two organizations O_1 and O_2 collaborated with sequential tasks T_1 and T_2 (T_2 is the post task of T_1), a new output place of the corresponding transition of T_1 is added and it is added as one input place of T_2 at the same time. The model of the sequential task pattern is shown in Fig. 8.

Task synchronization pattern: If two tasks:

$$< T_1, D_1, O_1, MessagesReq_1, MessagesSent_1, Resources_1, PostTasks_1 >$$

and

$$< T_2, D_2, O_2, MessagesReq_2, MessagesSent_2, Resources_2, PostTasks_2 >$$

satisfy (1) $T_1 = T_2$; (2) $D_1 = D_2$ and (3) $O_1 \neq O_2$, then two organizations O_1 and O_2 are collaborated with synchronous tasks T_1 and T_2.

For example, the two tasks shown in Fig. 9 illustrate T_1 differs from T_2 in <organization> and have the same <name> and <duration>, so they are two synchronous tasks between O_1 and O_2.

To model two organizations O_1 and O_2 collaborated with synchronous tasks T_1 and T_2, the two tasks are merged into one task in the Petri net model and are represented by one transition T_{12}, but the start places and the end places of T_1 and T_2 are connected to T_{12}. The model of the task synchronization pattern is shown in Fig. 10.

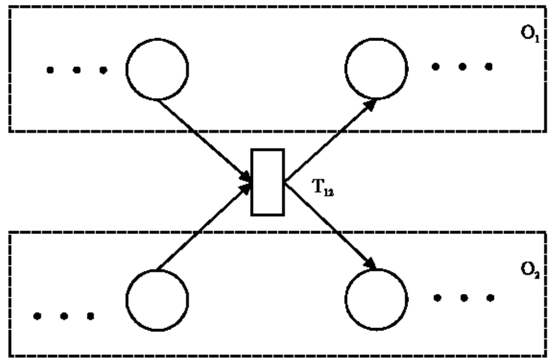

Fig. 10: Petri net model of task synchronization pattern

```
<task>                                      <task>
    <name>T1</name>                             <name>T2</name>
    <duration>D1</duration>                     <duration>D2</duration>
    <organization>O1</organization>             <organization>O2</organization>
    <messagesReq></messagesReq>                 <messagesReq></messagesReq>
    <messagesSent></messagesSent>               <messagesSent></messagesSent>
    <resources>                                 <resources>
    <resource>R1</resource>                         <resource>R1</resource>
    </resources>                                </resources>
    <postTasks></postTasks>                     <postTasks></postTasks>
</task>                                     </task>
```

Fig. 11: Two tasks sharing resource

Resource sharing pattern: If two tasks:

$$< T_1, D_1, O_1, MessagesReq_1, MessagesSent_1, Resources_1, PostTasks_1 >$$

and

$$< T_2, D_2, O_2, MessagesReq_2, MessagesSent_2, Resources_2, PostTasks_2 >$$

satisfy (1) $O_1 \neq O_2$ and (2) $Resources_1 \cap Resources_2 \neq \phi$, then two organizations O_1 and O_2 are collaborated with sharing resources between T_1 and T_2.

For example, the two tasks shown in Fig. 11 illustrate T_1 differs from T_2 in <organization> and theirs <resource> have the same resource R_1, so they are two tasks with sharing resources between O_1 and O_2.

To model two organizations O_1 and O_2 collaborated with sharing resources between tasks T_1 and T_2, the same resource used by T_1 and T_2 is represented by a resource place. The model of the resource sharing pattern is shown in Fig. 12.

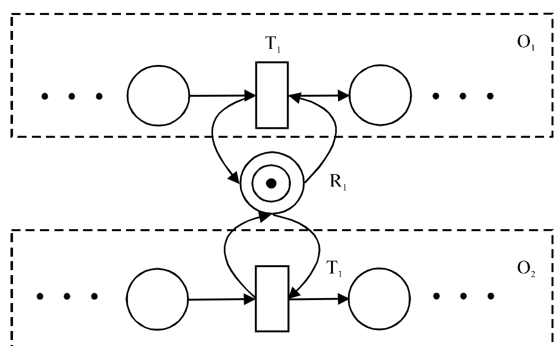

Fig. 12: Petri net model of resource sharing pattern

Message exchanged pattern: If two tasks:

$$< T_1, D_1, O_1, MessagesReq_1, MessagesSent_1, Resources_1, PostTasks_1 >$$

and

$$< T_2, D_2, O_2, MessagesReq_2, MessagesSent_2, Resources_2, PostTasks_2 >$$

satisfy, (1) $O_1 \neq O_2$ and (2) $MessagesReq_1 \cap MessagesSent_2 \neq \phi$ or $MessagesReq_2 \cap MessagesSent_1 \neq \phi$, then two organizations O_1 and O_2 are collaborated with message exchanged between tasks T_1 and T_2. For example, the two tasks shown in Fig. 13, the <messageSent> of T_1 and the <messageReq> of T_2 have the same message M_1, so there is a message exchanged between O_1 and O_2.

To model two organizations O_1 and O_2 collaborated with message exchanged between tasks T_1 and T_2, the message place exchanged is represented by one message

```
<task>                                        <task>
    <name>T1</name>                               <name>T2</name>
    <duration>D1</duration>                       <duration>D2</duration>
    <organization>O1</organization>               <organization>O2</organization>
    <messagesReq></messagesReq>                   <messagesReq>
    <messagesSent>                                <message>M1</message>
    <message>M1</message>                         </messagesReq>
    </messagesSent>                               <messagesSent></messagesSent>
    <resources></resources>                       <resources></resources>
    <postTasks></postTasks>                       <postTasks></postTasks>
</task>                                       </task>
```

Fig. 13: Two tasks with message exchanged

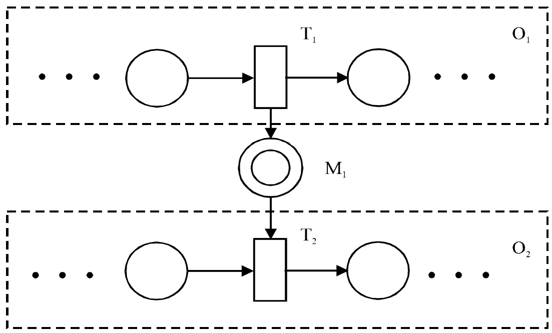

Fig. 14: Petri net model of message exchanged pattern

Table 2: A serial tasks of the bomb attack in a city

Name	Duration	Organization	MessagesReq	MessagesSent	Resources	PostTasks
T_1	D_1	O_1	ϕ	ϕ	ϕ	$\{T_2\}$
T_2	D_2	O_1	ϕ	$\{M_1\}$	ϕ	$\{T_3\}$
T_3	D_3	O_1	$\{M_2\}$	ϕ	$\{R_1\}$	ϕ
T_4	D_4	O_2	$\{M_1\}$	$\{M_2\}$	ϕ	$\{T_5\}$
T_5	D_3	O_2	ϕ	$\{M_3, M_4, M_5\}$	ϕ	$\{T_{12}\}$
T_6	D_5	O_3	$\{M_3\}$	ϕ	$\{R_2\}$	$\{T_9\}$
T_7	D_6	O_4	$\{M_4\}$	ϕ	$\{R_2\}$	$\{T_{10}\}$
T_8	D_4	O_5	$\{M_5\}$	ϕ	$\{R_3\}$	$\{T_{11}\}$
T_9	D_7	O_3	ϕ	$\{M_6\}$	$\{R_4\}$	$\{T_{10}, T_{14}\}$
T_{10}	D_7	O_4	ϕ	$\{M_7\}$	$\{R_5\}$	$\{T_{15}\}$
T_{11}	D_8	O_5	ϕ	$\{M_8\}$	$\{R_6\}$	$\{T_{16}\}$
T_{12}	D_9	O_2	$\{M_6, M_7, M_8\}$	ϕ	$\{R_7\}$	$\{T_{13}\}$
T_{13}	D_{10}	O_2	ϕ	$\{M_9, M_{10}, M_{11}\}$	$\{R_8\}$	$\{T_{17}\}$
T_{14}	D_7	O_3	$\{M_9\}$	ϕ	$\{R_9\}$	ϕ
T_{14}	D_7	O_4	$\{M_{10}\}$	ϕ	$\{R_9\}$	ϕ
T_{14}	D_7	O_5	$\{M_{11}\}$	ϕ	$\{R_9\}$	ϕ
T_{14}	D_7	O_2	ϕ	ϕ	$\{R_9\}$	ϕ

place and the transitions of T_1 and T_2 connect to this message place. The model of the message exchanged pattern is shown in Fig. 14.

Application case: According to the emergency response process modeling method of a single organization given in here and the task coordination patterns between organizations given in here, a serial of cross-organization tasks within UERS can be modeled. In this section, an example is given for illustration. For example, a set of tasks in an emergency response process are shown in Table 2, which includes 14 tasks completed by five organizations. During the implementation of the emergency response process, there are nine kinds of resources required and eleven kinds of messages exchanged. The meanings of symbols in Table 2 are shown in Table 3.

According to the emergency response process modeling method and the task coordination patterns between organizations, the Petri net model for the tasks presented in Table 2 is shown in Fig. 15.

According to Fig. 15, the emergency response process for each organization is very clear. The task coordination patterns between these five organizations are observed and presented in Table 4. From the table, we can see that O_1 only cooperates with O_2, but O_2, O_3, O_4 and O_5 are working in a very close collaboration with each

Table 3: Description of symbols in Table 2

Name	Description	Name	Description	Name	Description
Tasks					
T_1	Alarm receipt	T_2	Report	T_3	Policemen race to the location
T_4	To hold a meeting	T_5	To start a emergency response process	T_6	Firemen race to the location
T_7	Paramedics race to the location	T_8	Regimental polices race to the location	T_9	To control the fire and rescue survivors
T_{10}	To give first-aid to the wounded	T_{11}	To disarm unexploded bombs	T_{12}	To assess the location
T_{13}	To catch criminals and investigate the details of the case	T_{14}	To hold a press conference		
Resources					
R_1	Police car	R_2	Bridge	R_3	Packer
R_4	Fire truck	R_5	Ambulance	R_6	Police dog
R_7	Camera	R_8	Pistol	R_9	Meeting room
Messages					
M_1	Demand handling	M_2	To instruct policemen race to the location	M_3	To instruct firemen race to the location
M_4	To instruct paramedics race to the location	M_5	To instruct regimental polices race to the location	M_6	Fire behavior
M_7	The message of casualty	M_8	The message of bombs	M_9	Leaders of fire station attend the press conference
M_{10}	Leaders of hospital attend the press conference	M_{11}	Leaders of regimental police attend the press conference		
Organizations					
O_1	Police station	O_2	Public security bureau	O_3	Fire station
O_4	Hospital	O_5	Regimental police		
Durations					
D_1	2 min	D_2	3 min	D_3	5 min
D_4	10 min	D_5	15 min	D_6	20 min
D_7	30 min	D_8	60 min	D_9	90 min
D_{10}	5 h				

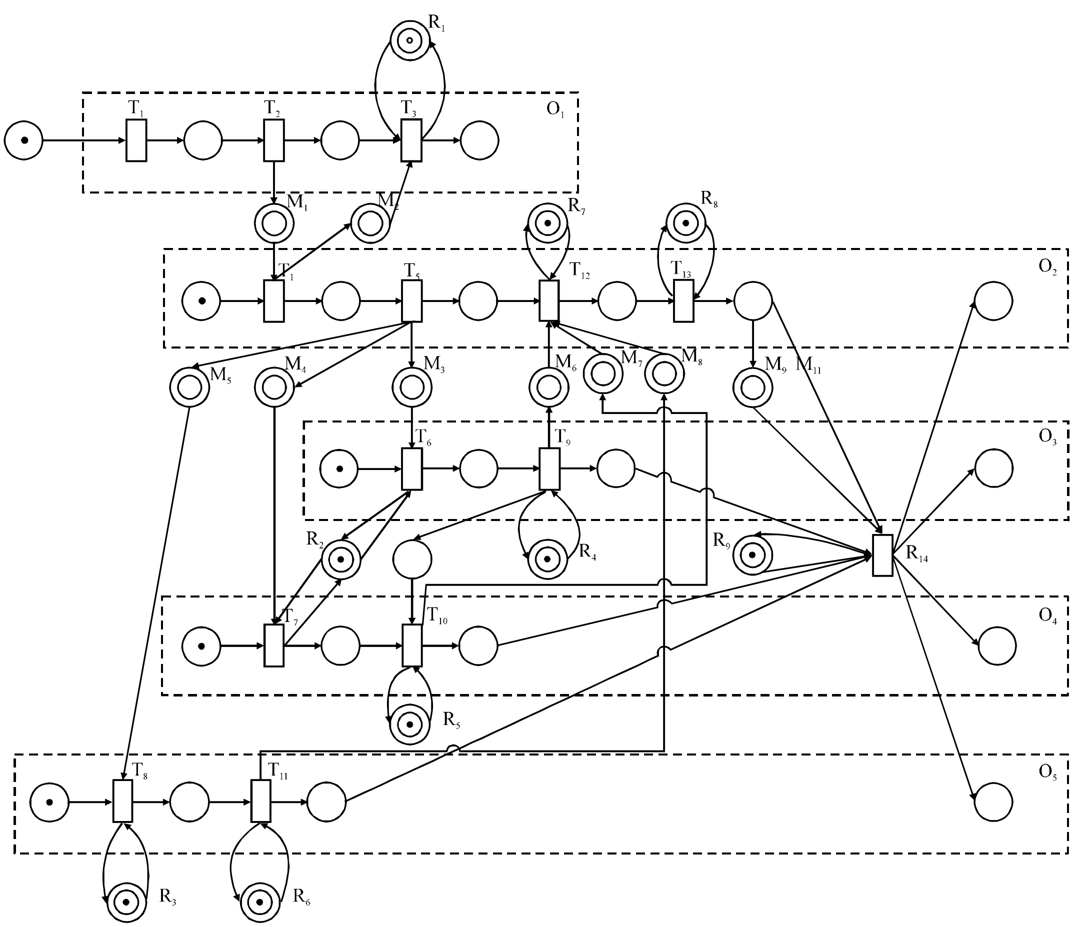

Fig. 15: Petri net model of tasks in Table 2

Table 4: Coordination patterns between organizations

Organization	O_1	O_2	O_3	O_4	O_5
O_1		MEP			
O_2	MEP		MEP, TSP	MEP, TSP	MEP, TSP
O_3		MEP, TSP		RSP, STP, TSP	TSP
O_4		MEP, TSP	RSP, STP,TSP		TSP
O_5		MEP, TSP	TSP	TSP	

other. Form Table 4, it can also be found that message exchanged pattern is the most common patterns between organizations.

In Table 4, STP represents Sequential task pattern, TSP represents Task synchronization pattern, RSP represents Resource sharing pattern and MEP represents Message exchanged pattern.

CONCLUSION

The coordination problem is important within a complex UERS. In this study, we use Petri net as formal tool to model the task of a single organization and propose four task coordination patterns between organizations in UERS, which include the sequential task pattern, the task synchronization pattern, the resource sharing pattern and the message exchanged pattern based on Petri net. Finally, an example is used as application case to illustrate the proposed modeling approach. The coordination problem is a very complex research topic. In this study, we only propose the modeling method and coordination patterns of tasks between organizations. Based on the modeling method and the coordination patterns, there are many problems to be continued, for example, the resource conflict analysis between organizations, the performance analysis for a complex UERS cross organization and so on.

ACKNOWLEDGMENT

This study is supported partly by the NSFC under Grant No. 60603090, 90718011 and 50875158; Science and Technology Development Fund of Shandong Province of China (2010GSF10811); the Excellent Young Scientist Foundation of Shandong Province of China under Grant No. BS2009DX004; the Special Fund for Fast Sharing of Science Paper in Net Era by CSTD (20093718110008); the Open Project of Computer Architecture Lab of ICT, CAS, (No. ICT-ARCH200807); the Research Foundation of Shandong Educational Committee under Grant No. J08LJ77 and the Taishan Scholar Program of Shandong Province.

REFERENCES

Kappel, G., S. Rausch-Schott and W. Retschitzegger, 1998. Coordination in workflow management systems: A rule-based approach. Comput. Sci., 1364: 99-120.

Li, W.Z. and W. Wei, 2008. Study on model and framework of urban emergency response system. Proceedings of the International Conference on Computer Science and Information Technology, Aug. 29-Sept. 2, Khulna, pp: 845-849.

Liu, X. and L. Yang, 2010. Behavior-aware trust reasoning based on associate petri net. Inform. Technol. J., 9: 1178-1183.

Tang, Z.Q. and N. Li, 2010. Modeling of city emergency response system based on UML. Comput. Eng. Design, 31: 2044-2051.

Van der Aalst, W.M.P., 1998. Modeling and analyzing interorganizational workflows. Proceedings of the First International Conference on Application of Concurrency to System Design, March 23-26, Fukushima, pp: 262-262.

Wang, H.Q. and Q.T. Zeng, 2008. Modeling and analysis for workflow constrained by resources and non-determined time: An approach based on petri nets. IEEE Trans. Syst. Man Cybernetics Part A: Syst. Hum., 38: 802-817.

Wei, S., Y.H. Chen, D.Y. Hu and W. Su, 2006. Study and realization of urban emergency response system. Comput. Eng., 32: 267-269.

Xia, W., C.J. Ge and X.J. Guan, 2008. Research on constructing the framework of urban emergency response system based on multi-agent system. Proceedinns of the IEEE International Conference on Industrial Engineering and Engineering Management, Dec. 8-11, Singapore, pp: 2066-2070.

Zeng, Q., 2008. A construction method for the process expression of petri net based on decomposition. Inform. Technol. J., 7: 420-429.

Zeng, Q.T. and H. Duan, 2007. Behavior description for complex flexible manufacturing system based on decomposition of Petri net. Int. J. Comput. Syst. Sci. Eng., 22: 359-363.

Zeng, Q.T., H.Q. Wang, H. Duan and Y.B. Han, 2008. Conflict detection and resolution for workflow constrained by resources and non-determined duration. J. Syst. Software, 81: 1491-1504.

Zhong, M., C. Shi, T. Fu, L. He and J. Shi, 2010. Study in performance analysis of China Urban emergency response system based on Petri net. Safety Sci., 48: 755-762.

Zhong, M.H., T.M. Liu and G.Z. Liu, 2003. Performance analysis on urban emergency response system based on petri net. China Safety Sci. J., 13: 17-20.