International Journal of Database Theory and Application
Vol.67 No.5 (2014), pp.109-122
http://dx.doi.org/10.14257/ijdta.2014.7.5.08

A Petri Net Processing Model of STeCEQL

Huiyong Li, Yixiang Chen and Kangli He

Software Engineering Institute, East China Normal University, Shanghai, China
lihuiyongchina@126.com

Abstract

The internet of vehicles is an important internet of things. It is an urgent problem that how to real-time processes these spatial and temporal data of the internet of vehicles. The complex event processing technology can filter the concerned data to the event by the complex event query language (EQL) and the system can responds effectively. The STeCEQL is a complex event language for the internet of vehicles, which constraint with spatial and temporal. The processing model of the complex event language is a core issue of the complex event processing technology. In this paper, we established a Petri Net processing model of STeCEQL. We give all kind of processing model of STeCEQL expressions. And these processing models are synthesized by two types of basic Petri Net model: the sequence structure and the logic and structure. Finally, we proved the Petri Net processing model of STeCEQL is structural boundedness and structural conservativeness, but it is not structural repetitiveness.

Keywords: *internet of things, mobile system, complex event query language, complex event processing model, Petri Net*

1. Introduction

The internet of vehicles is an important achievement of the internet of things [1]. There are not only a large number of non-mobile agents in the internet of vehicles, but also a lot of fast-moving mobile agents. In the system, the various sensors carried by each agent produce a continuous flood of data: time, position, direction, speed, temperature, *etc* [2, 3]. However, the conventional database technology can not deal these data effectively.

In recent years, the complex event processing technology has been successfully applied in the internet of vehicles projects. In internet of vehicles, the large amount of raw data generated by various sensors is seen as basic events. Through the complex event query language, the system will filter these basic events into meaningful complex events. Therefore, the massive amounts of data of the system can be effectively processed in real time. Moody K proposed a complex event of query language of internet of vehicles: SpaTec. This EQL has been applied to the London's bus monitoring system [4, 5]. Jin B proposed a complex event query language of internet of vehicles: CPSL [6]. We also proposed a complex event query language: STeCEQL. And it can describe a variety of temporal and spatial attributes of the events.

The processing model of the complex event query language is the core mechanism for dealing with massive data of the system. There are four types processing model of the EQL: Finite Automata Model [7,8], Matching Tree model[9,10], Directed Acyclic Graph model[11,12], Petri Net model[13,14].Petri Net can be used to model the systems with concurrent, asynchronous, distributed, non-deterministic, parallel and other characteristics. And it is considered to be the most powerful tool for studying discrete event systems. Meanwhile, Petri Net is a strictly defined mathematical model. The Petri Net methods can

ISSN: 2005-4270 IJDTA

analyze the structure of the system and the behavior of the dynamic characteristics of the system. In the internet of vehicles, complex event processing system is a typical distributed asynchronous concurrent system, so we use the Petri Net models.

In this paper, we have made the processing model of STeCEQL by Petri Net and analyzed the relevant properties of the model. The remainder of this paper is organized as follows: Section 2 reviews the STeCEQL. Section 3 simply introduces the Petri Net theory. Section 4 gives the various processing models of STeCEQL expressions. Section 5 analyzes the characteristics of the Petri Net model and proves the related properties. The last Section concludes this paper.

2. STeCEQL Language

The STeCEQL is complex event query language for the internet of vehicles of event driven architecture. It can express the basic events and the complex events. The syntax of STeCEQL is as follows:

ABexp:

$$attribute ::= true \mid false \mid x_a = a \mid x_a != a \mid x_a > a \mid x_a < a$$

$$\mid attribute_0 \wedge attribute_1 \mid attribute_0 \vee attribute_1$$

TBexp:

$$time ::= true \mid false \mid x_t \ BEFORE \ \ t \mid x_t \ AFTER \ \ t$$

$$\mid x_t \ EQUAL \ t \mid x_t \ OVERLAP \ t \mid x_t \ DURING \ t$$

$$\mid time_0 \vee time_1 \mid time_0 \wedge time_1$$

LBexp:

$$location ::= true \mid false \mid x_l \ EQ \ l \mid x_l \ OP \ l \mid x_l \ IN \ l$$

$$\mid x_l \ NORTH \ l \mid x_l \ SOURTH \ l \mid x_l \ EAST \ l \ \mid x_l \ WEST \ l$$

$$\mid x_l \ NORTHWEST \ l \mid x_l \ NORTHEAST \ l$$

$$\mid x_l \ SOURTHWEST \ l \mid x_l \ SOURTHEAST \ l$$

$$\mid location_0 \vee location_1 \mid locaiton_0 \wedge location_1$$

DBexp:

$$direction ::= true \mid false \mid x_d = d \mid x_d != d$$

EBexp:

$$e ::= agent^{time}(attribute_1; attribute_2; attribute_3 \cdots)$$

$$\mid agent_{location}(attribute_1; attribute_2; attribute_3 \cdots)$$

$$\mid agent^{time}_{location}(attribute_1; attribute_2; attribute_3 \cdots)$$

$$\mid agent^{time}_{(location,\ direction)}(attribute_1; attribute_2; attribute_3 \cdots)$$

CEexp:

$$e ::= e_1 \wedge e_2 \mid e_1 \vee e_2$$

$$\mid e_1 \ BEFORE \ e_2 \mid \mathrm{e}_1 \ AFTER \ e_2$$

$$\mid e_1 \ EQUAL \ e_2 \mid e_1 \ OVERLAP \ e_2 \mid e_1 \ DURING \ e_2$$

$$\mid e_1 \ EQ \ e_2 \mid e_1 \ OP \ e_2 \mid e_1 \ IN \ e_2$$

$$\mid e_1 \ NORTH \ \ e_2 \mid e_1 \ SOURTH \ \ e_2 \mid e_1 \ EAST \ \ e_2 \ \mid e_1 \ WEST \ \ e_2$$

$| e_1\ NORTHWEST\ e_2 \mid e_1\ NORTHEAST\ e_2$

$| e_1\ SOURTHWEST\ e_2 \mid e_1\ SOURTHEAST\ e_2$

3. The Petri Net Theory

Definition1 Meet the following conditions triples N = (S, T; F) is called Network:

(1) $S \cup T \neq \varnothing$

(2) $S \cap T = \varnothing$

(3) $F \subseteq S \times T \cup T \times S$

(4) $dom(F) \cup cod(F) = S \cup T$

$dom(F) = \{x \mid \exists x : (x, y) \in F\}$, $cod(F) = \{y \mid \exists x : (x, y) \in F\}$.

S and T are two disjoint sets, which are the basic elements of the network set N. S elements is called the S- or library (place), T elements called T- or changes (transition), F is the flow of the network N.

Definition 2 Let N = (S, T; F) is a net. Mapping M: S→ {0,1,2, ...} is called a marking of net N. The tuple (N, M) is called a marking net.

Definition 3 A Petri net system is a marking net Σ = (S, T; F, M) and has the following transition firing rule:

(1)For transition t∈T, if $\forall s \in S : s \in {}^{\bullet}t \rightarrow M(s) \geq 1$, then says that the transition t have the firing rule, denoted by M[t>.

(2)If M[t>, the under the marking M, the transition t can fire: the transition t fired from the marking M to the new marking M', denoted by M[t> M'.

And $\forall s \in S, \quad M'(s) = \begin{cases} M(s) - 1, & if\ s \in {}^{\bullet}t - t^{\bullet} \\ M(s) + 1, & if\ s \in t^{\bullet} - {}^{\bullet}t \\ M(s), & else \end{cases}$.

Definition 4 Let Σ = (S, T; F, M) is a Petri net, S={s1,s2,⋯,sm},T={t1,t2,⋯,tn}, then the structure of the Petri net ∑(S, T; F) can be expressed by a matrix of n rows and m columns A=$[a_{ij}]_{n\times m}$. $a_{ij} = a_{ij}^{+} - a_{ij}^{-}$, i∈(1,2,⋯,n), j∈(1,2,⋯,m),

$a_{ij}^{+} = \begin{cases} 1, & if\ (t_i, s_j) \in F; \\ 0, & else; \end{cases}$, $a_{ij}^{-} = \begin{cases} 1, & if\ (s_j, t_i) \in F; \\ 0, & else; \end{cases}$,

A is called incidence matrix of the Σ (or net N = (S, T; F)).

Theorem1 Let A is the incidence matrix of the network N = (S, T; F), then the necessary and sufficient conditions of the N is structural boundedness are: there is a m(m=|S|) dimensional vector of positive integers Y make AY≤0.

Theorem2 Let A is the incidence matrix of the network N = (S, T; F), then the necessary and sufficient conditions of the N is structural conservativeness are: there is a m(m=|S|) dimensional vector of positive integers Y make AY=0.

Theorem3 Let A is the incidence matrix of the network N = (S, T; F), then the necessary and sufficient conditions of the N is structural repetitiveness are: there is a n(m=|T|) dimensional vector of positive integers X make ATX≥0.

These are some basic concepts and key theorems of Petri net models. In the modeling process, the calculation of Petri net is also important. These calculations include: insert, delete, replace, synthesis and decompose. Here is the definition of combination computing.

Definition5 Let Ni=(Si,Ti;Fi), i=1,2, $S_1 \cap S_2 \neq \varnothing, T_1 \cap T_2 = \varnothing$, then $N = (S_1 \cup S_2, T_1 \cup T_2; F_1 \cup F_2)$ is called a shared combination net of N1 and N2, denoted by N=N1CsN2.

Theorem4 Let net N is the shared combination net of N1 and N2, N=N1CsN2. then:

(1) If the net N1 and N2 is structural boundedness, then the net N is also structural boundedness.

(2) If the net N1 and N2 is structural conservativeness, then the net N is also structural conservativeness.

(3) If the net N1 and N2 is structural repetitiveness, then the net N is also structural repetitiveness.

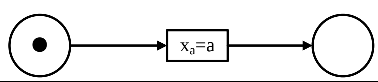
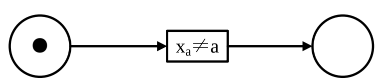
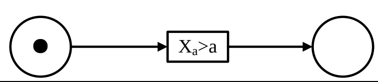
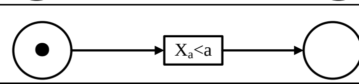
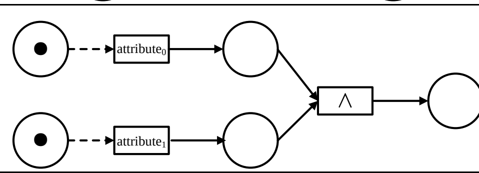
4. The Petri Net Processing Model of STeCEQL

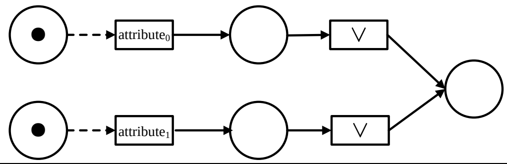
In the Petri net processing model of complex event query language, we use the input library of Petri net indicate the basic events and the output library of Petri net express the complex events. The number of the tokens in the library represents the number of event instances. When an event instance is coming, the number of the token in the library will be changed. Through calculating the transition function, the model can determine whether the transition conditions are met. If the transition rule is satisfied, the transition will fire and the status of the input and the output library will change.

We establish the Petri net processing model of all kinds of STeCEQL expressions. Assuming that the capacity of each library is limited and the number of token in libraries represents the number of the event instances.

ABexp:

Table 1. The Model of ABexp

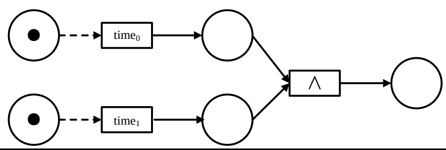
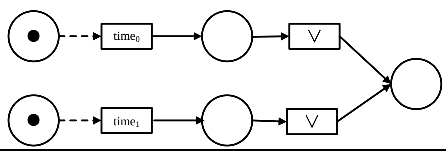
ABexp	Petri net model
x_a=a	
x_a!=a	
x_a>a	
x_a<a	
attribute0∧ attribute1	

attribute0∨ attribute1	

The Petri net processing models of ABexp of STeCEQL are in Table 1. From the table, we can conclude that the four kind expressions: x_a=a、x_a!=a、x_a>a and x_a<a have the similar basic net structure. The dotted lines of the logic and operation and the arithmetic operation model indicate that there are many libraries and transitions in the front sets of the model.

TBexp:

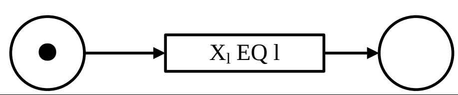
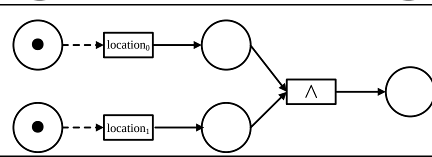
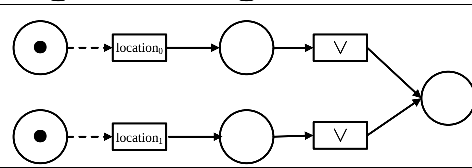
Table 2. The Model of TBexp

TBexp	Petri net model
x_t BEFORE t	
$time_0$∧$time_1$	
$time_0$∨$time_1$	

The Petri net processing models of TBexp of STeCEQL are in table 2. We only list the model of expression x_t BEFORE t and the other expression's models are similar. The dotted lines of the logic and operation and the logic or operation model indicate that there are many libraries and transitions in the front sets of the model.

LBexp:

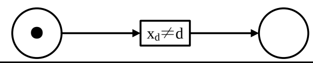
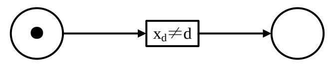
Table 3. The Model of LBexp

LBexp	Petri net model
x_l EQ l	
$location_0$∧ $location_1$	
$location_0$∨ $location_1$	

The Petri net processing models of LBexp of STeCEQL are in Table 3. We only list the model of expression x_l EQ l and the other expression's models are similar. The dotted lines of the logic and operation and the logic or operation model indicate that there are many libraries and transitions in the front sets of the model.

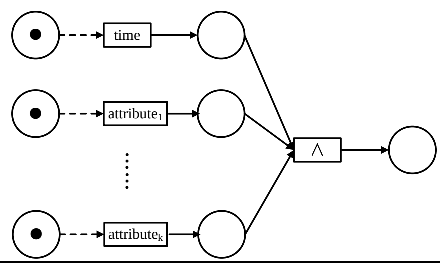
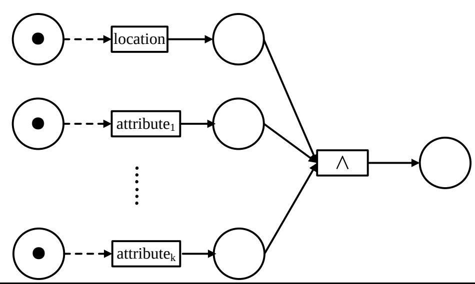
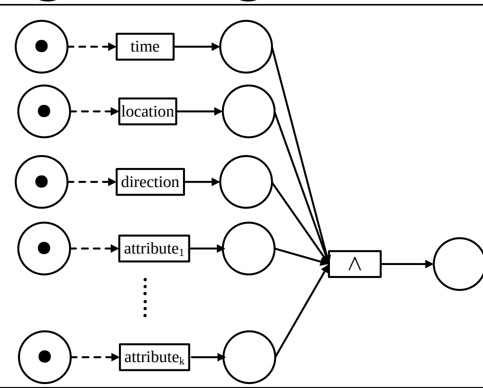
DBexp:

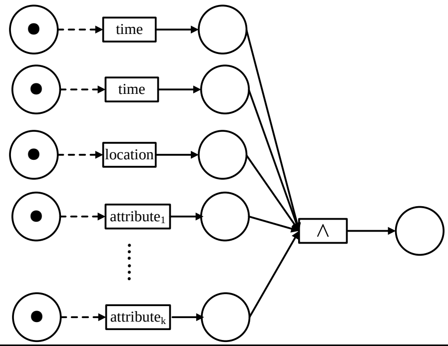
Table 4. The Model of DBexp

DBexp	Petri net model
x_d=d	
x_d!=d	

EBexp:

Table 5. The Model of EBexp

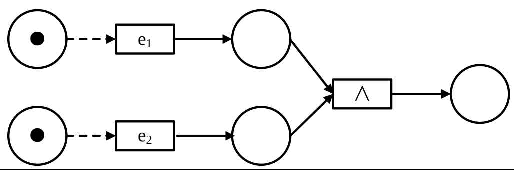
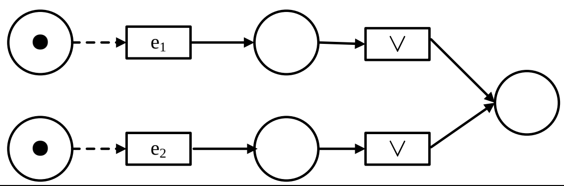
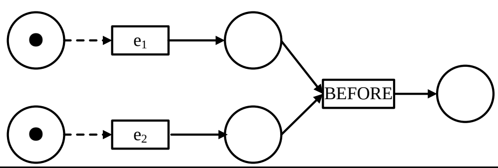
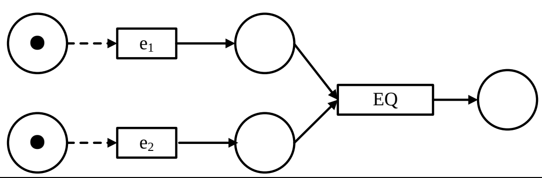
EBexp	Petri net model
$agent^{time}(attribute_1;attribute_2;attribute_3\cdots)$	
$agent_{location}(attribute_1;attribute_2;attribute_3\cdots)$	
$agent_{location}^{time}(attribute_1;attribute_2;attribute_3\cdots)$	

$agent^{time}_{(location,\, direction)}(attribute_1; attribute_2; attribute_3 \cdots)$	

The Petri net processing models of EBexp of STeCEQL are in Table 5. We list the models of the four basic expressions. The dotted lines of the model indicate that there are many libraries and transitions in the front sets of the model.

CEBexp:

Table 6. The Model of CEBexp

CEBexp	Petri net model
e1 ∧ e2	
e1 ∨ e2	
e1 BEFORE e2	
e1 EQ e2	

The Petri net processing models of CEBexp of STeCEQL are in table 6. We only list the model of expression e1 EQ e2 and e1 BEFORE e2, the other expression's models are similar. The dotted lines of model indicate that there are many libraries and transitions in the front sets of the model.

According to the above Petri net models, we can compose various models of all kind of STeCEQL expressions. The following are two examples:

Example1

ce1: $car1^{(x_t\ DURING\ \{(14{:}38{:}30, 14{:}38{:}40)\})}_{(x_l\ EQ\ \{(500,1300)\},\ x_d = NORTH)}(x_v > 124)$.

The Petri net processing model is as Figure 1.

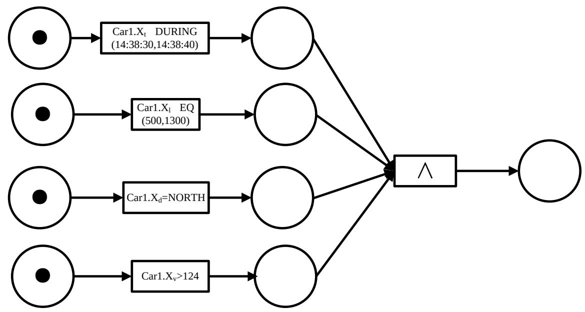


Figure 1. The Processing Model of e1

Example 1

e1: $car5^{(x_t\ DURING\ \{(8:26:30,8:26:35)\})}_{(x_l\ IN\ \{(695,85),(696,85),(697,85)\},x_d=EAST)}(x_v > 60 \wedge x_v < 95)$,

e2: $car6^{(x_t\ BEFORE\ \{(9:30:42,9:40:03)\})}_{(x_l\ EQUAL\ \{(1324,90)\},x_d=NORTHEAST)}(x_v \leq 120)$,

ce2: $e1\ OVERLAP\ e2$.

The Petri net processing model is as Figure 2.

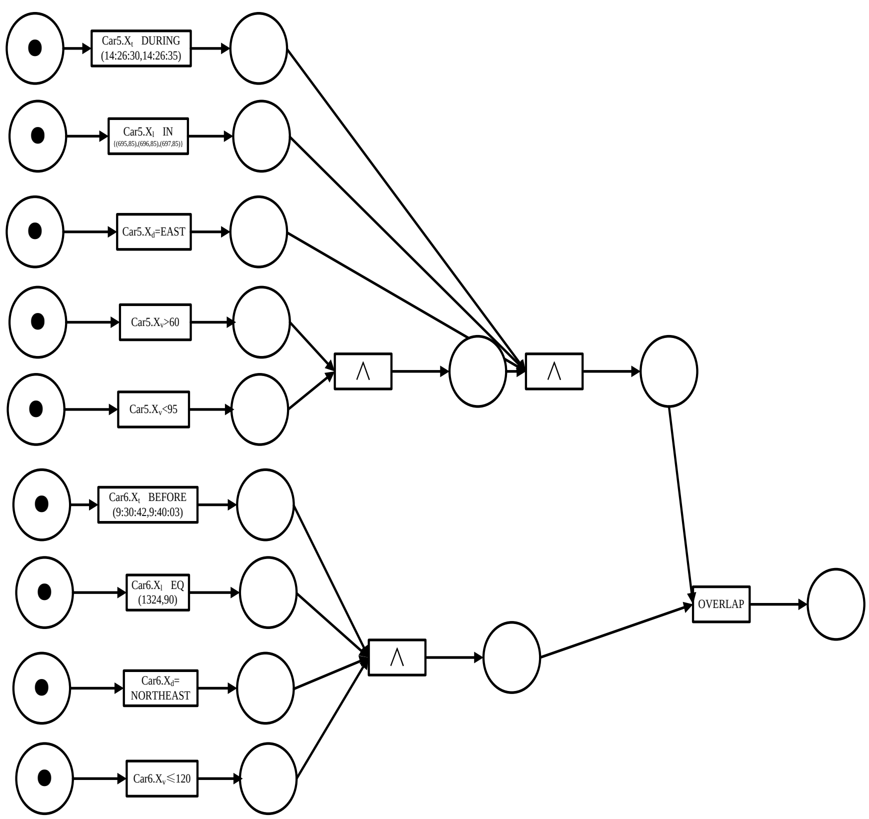


Figure 3. The Processing Model of ce1

5. Analysis of the Petri Net Model of STeCEQL

5.1. The Characteristic of the Petri Net Model of STeCEQL

From the models of tables 1 to 6, the processing models of STeCEQL consist of the sequence structure model and the logic and structure model. The sequence structure model is as Figure 3. The logic and structure model is as Figure 4.

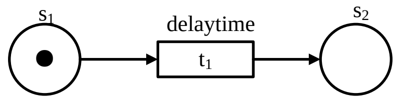


Figure 3. The Petri Net Model of Sequence Structure

As the Figure 3 shows, the Petri net model consists of two libraries s1 and s2, a transition t1. According to this model, its incidence matrix as below:

The output matrix: $a_{ij}^{+} = [0 \quad 1]$, and the input matrix: $a_{ij}^{-} = [1 \quad 0]$,

So the incidence matrix is $A_1 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad 1]$.

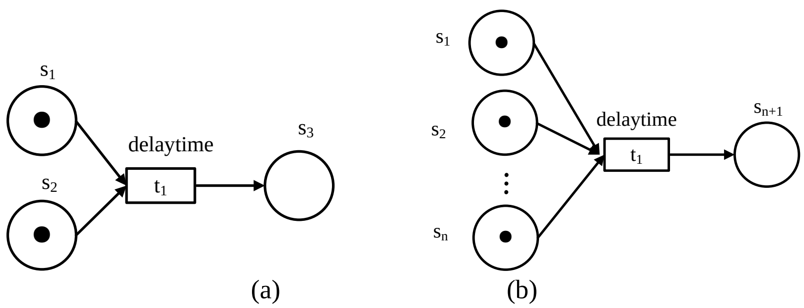


Figure 4. The Petri Net Model of Logic and Structure

As the Figure 4 shows, the Petri net model consists of multiple libraries and a transition t1. According to this model, its incidence matrix as below:

As the Figure 4 (a) shows, if there are two branches in the model. The output matrix: $a_{ij}^{+} = [0 \quad 0 \quad 1]$, the output matrix: $a_{ij}^{-} = [1 \quad 1 \quad 0]$, so the incidence matrix: $A_2 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad -1 \quad 1]$.

As the figure4 (b) shows, if there are multiple branches in the model. The input matrix: $a_{ij}^{+} = [0 \quad 0 \quad \cdots \quad 1]$, and the output matrix: $a_{ij}^{-} = [1 \quad 1 \quad \cdots \quad 0]$, so the incidence matrix: $A_3 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad -1 \quad \cdots \quad 1]$.

5.2. Analysis of the Petri Net Model of STeCEQL

According to the Petri net models of STeCEQL, we analyze the properties of these structures.

Property1 the Petri Net models of the STeCEQL are structural boundedness.

Proof: Let N0=(S, T; F) is the basic net of a STeCEQL expression,

Case1: when N0 only include the basic sequence structure as Figure 3, the incidence matrix: $A_1 = a_{ij}^+ - a_{ij}^- = [-1 \quad 1]$, then exist Y= $\begin{bmatrix} 3 \\ 1 \end{bmatrix}$ make AY=-3+1=-2≤0.

According to theorem1, N0 is structural boundedness.

Case2: when N0 only include the structure as figure4 (a),

the incidence matrix: $A_2 = a_{ij}^+ - a_{ij}^- = [-1 \quad -1 \quad 1]$, then exist Y= $\begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix}$ make AY=-3-3+1=-5≤0.

According to theorem1, N0 is structural boundedness.

Case3: when N0 only include the structure as figure4 (b),

the incidence matrix: $A_3 = a_{ij}^+ - a_{ij}^- = [-1 \quad -1 \quad \cdots \quad 1]$, then exist Y= $\begin{bmatrix} 3 \\ 3 \\ \vdots \\ 1 \end{bmatrix}$ make AY=-3n+1=1-3n≤0.

According to theorem1, N0 is structural boundedness.

Because the Petri net processing models of STeCEQL consist of the sequence structure and the logic and structure. And these models are shared combination net. According to theorem4, the Petri net processing models of STeCEQL are structural boundedness.

We finish the proof of this theorem.

Property2 the Petri Net models of the STeCEQL are structural conservativeness.

Proof: Let N0=(S, T; F) is the basic net of a STeCEQL expression,

Case1: when N0 only include the basic sequence structure as Figure 3,

the incidence matrix: $A_1 = a_{ij}^+ - a_{ij}^- = [-1 \quad 1]$, then exist Y= $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ make AY=-1+1=0.

According to theorem2, N0 is structural conservativeness.

Case2: when N0 only include the structure as figure4 (a),

the incidence matrix: $A_2 = a_{ij}^+ - a_{ij}^- = [-1 \quad -1 \quad 1]$, then exist Y= $\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$ make AY=-1-1+2=0.

According to theorem2, N0 is structural conservativeness.

Case3: when N0 only include the structure as figure4 (b), the incidence matrix: $A_3 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad -1 \quad \cdots \quad 1]$, then exist Y= $\begin{bmatrix} 1 \\ \\ \\ \vdots \\ \\ n \end{bmatrix}$ make AY=-n+n=0.

According to theorem2, N0 is structural conservativeness.

Because the Petri net processing models of STeCEQL consist of the sequence structure and the logic and structure. And these models are shared combination net. According to theorem4, the Petri net processing models of STeCEQL are structural conservativeness.

We finish the proof of this theorem.

Property3 the Petri Net models of the STeCEQL are not structural repetitiveness.

Proof: Let N0=(S, T; F) is the basic net of a STeCEQL expression,

Case1: when N0 only include the basic sequence structure as Figure 3,

the incidence matrix: $A_1 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad 1]$,

If there is a X=$[x_1 \quad x_2]$ make A1TX≥0,

Then $\begin{cases} -x_1 - x_2 \geq 0 \\ x_1 + x_2 \geq 0 \end{cases}$ there are positive integer solutions.

But $\begin{cases} -x_1 - x_2 \geq 0 \\ x_1 + x_2 \geq 0 \end{cases} \rightarrow x_1 + x_2 = 0$, so there are not positive integer solutions.

So, there are not X make A1TX≥0,

According to theorem3, N0 is not structural repetitiveness.

Case2: when N0 only include the structure as Figure 4 (a),

the incidence matrix: $A_2 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad -1 \quad 1]$,

If there is a X=$[x_1 \quad x_2 \quad x_3]$ make A2TX≥0,

Then $\begin{cases} -x_1 - x_2 - x_3 \geq 0 \\ -x_1 - x_2 - x_3 \geq 0 \\ x_1 + x_2 + x_3 \geq 0 \end{cases}$ there are positive integer solutions.

But $\begin{cases} -x_1 - x_2 - x_3 \geq 0 \\ -x_1 - x_2 - x_3 \geq 0 \\ x_1 + x_2 + x_3 \geq 0 \end{cases} \rightarrow x_1 + x_2 + x_3 = 0$, so there are not positive integer solutions.

So, there are not X make A2TX≥0,

According to theorem3, N0 is not structural repetitiveness.

Case3: when N0 only include the structure as Figure 4 (b),

the incidence matrix: $A_3 = a_{ij}^{+} - a_{ij}^{-} = [-1 \quad -1 \quad \cdots \quad 1]$, If there is a X=$[x_1 \quad x_2 \quad \cdots \quad x_{n+1}]$ make A3TX≥0,

Then $\begin{cases} -x_1 - x_2 - x_3 \geq 0 \\ -x_1 - x_2 - x_3 \geq 0 \\ x_1 + x_2 + x_3 \geq 0 \end{cases}$ there are positive integer solutions.

But $\begin{cases} -x_1 - x_2 - x_3 \geq 0 \\ -x_1 - x_2 - x_3 \geq 0 \\ x_1 + x_2 + x_3 \geq 0 \end{cases} \rightarrow x_1 + x_2 + x_3 = 0$, so there are not positive integer solutions.

So, there are not X make A3TX≥0,

According to theorem3, N0 is not structural repetitiveness.

Because the Petri net processing models of STeCEQL consist of the sequence structure and the logic and structure. And these models are shared combination net. According to theorem4, the Petri net processing models of STeCEQL are not structural repetitiveness.

We finish the proof of this theorem.

6. Conclusions

In this paper, we propose the Petri Net processing model of the STeCEQL. We analyse the relevant properties of the model. And we proved the Petri Net processing model of STeCEQL is structural boundedness and structural conservativeness, but it is not structural repetitiveness.

We only analyse the Petri Net processing model of the STeCEQL. Next, we will design the processing algorithms based the model and implement these algorithms.

Acknowledgements

This work is supported by the National Basic Research Program of China (No. 2011CB302802) , the National Natural Science Foundation of China (No.61370100 and No.61021004) and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

[1] G. Dimitrakopoulos, “Intelligent transportation systems based on internet-connected vehicles: Fundamental research areas and challenges, IEEE”, **(2011)**.

[2] Y. Chen, “Stec: A location-triggered specification language for real-time systems, IEEE”, **(2012)**.

[3] H. Wu, Y. Chen and M. Zhang, “On Denotational Semantics of Spatial-Temporal Consistency Language--STeC IEEE”, **(2013)**.

[4]S Schwiderski-Grosche and K. Moody, “The SpaTeC composite event language for spatio-temporal reasoning in mobile systems ACM”, **(2009)**.

[5] K. Moody, J. Bacon, D. Evans and S. Schwiderski-Grosche, “From active data management to event-based systems and more Springer”, **(2010)**.

[6] B. Jin, W. Zhuo, J. Hu, H. Chen and Y. Yang, “Decision Support Systems 55”, **(2013)**.

[7] D. Gyllstrom, E. Wu, H.-J. Chae and Y. Diao *et al.*, arXiv preprint cs/0612128, **(2006)**.

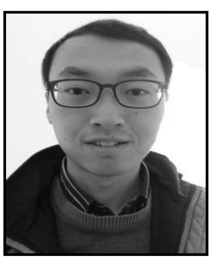
[8] A. J. Demers, J. Gehrke, B. Panda and M. Riedewald, *et.al.*, “Cayuga: A General Purpose Event Monitoring System”, **(2007)**.

[9] M. Mansouri-Samani and M. Sloman, "Distributed Systems Engineering 4", **(1997)** .
[10] R. E. Gruber, B. Krishnamurthy and E. Panagos, "The architecture of the READY event notification service, IEEE Computer Society", **(1999)**.
[11] S. Chakravarthy, E. Anwar, L. Maugis and D. Mishra, "Information and Software Technology 36", **(1994)** .
[12] A. Geppert, D. Tombros, "Event-based distributed workflow execution with EVE, Springer", **(1998)**.
[13] S. Gatziu, A. Geppert and K. R. Dittrich, "The SAMOS active DBMS prototype, Citeseer", **(1995)**.
[14] W. Hu, W. Ye, Y. Huang and S. Zhang, "Complex event processing in RFID middleware: A three layer perspective", IEEE, **(2008)**.

Authors

Huiyong Li, he received his M.Sc. in Computer Sciences (2011) from Taiyuan University of Science and Technology. Now he is PhD student of software engineering institute, East China Normal University. His current research interests include different aspects of Internet of Things and Real-time Distributed Systems.

Yixiang Chen, he received his PhD in Mathematics (1995) from Sichuan University. Now he is Chair of the MoE Key Research Center for Software/ Hardware Co-design Engineering, East China Normal University. His current research interests include different aspects of Real-time System, Formal Models, Formal Semantics of Programming, Foundation of Computations and Trustworthy Network.



Kangli He, he received his B.Sc. in Software Engineering (2013) from East China Normal University. Now he is a PhD student of software engineering institute, East China Normal University. His current research interests include different aspects of Internet of Things, Hardware/Software co-design and quantitative formal verification.

