Component-based event composition modeling for CPS

Zhonghai Yin and Yanan Chu

Mathematics Department, Air Force Engineering University, Xi'an Shanxi 710038, China

Abstract. In order to combine event-drive model with component-based architecture design, this paper proposes a component-based event composition model to realize CPS's event processing. Firstly, the formal representations of component and attribute-oriented event are defined. Every component is consisted of subcomponents and the corresponding event sets. The attribute "type" is added to attribute-oriented event definition so as to describe the responsiveness to the component. Secondly, component-based event composition model is constructed. Concept lattice-based event algebra system is built to describe the relations between events, and the rules for drawing Hasse diagram are discussed. Thirdly, as there are redundancies among composite events, two simplification methods are proposed. Finally, the communication-based train control system is simulated to verify the event composition model. Results show that the event composition model we have constructed can be applied to express composite events correctly and effectively.

1. Introduction

Cyber Physical System (CPS) is a large-scale network system which is composed of sensors, actuators, physical entities and computing units. By deep integration of physical and computing processes, it is aimed at controlling physical entities efficiently and reliably [1,2,3]. As a distributed complex system, CPS tends to adopt event-driven architecture design. As a matter of fact, event-driven model has been widely applied to procedure developing, such as event-driven based windows operating system, event-based Active Database [4], object-oriented DBMS [5], and so on. The above models are instructive but cannot be applied to CPS design. The reason is that event defined in these systems is simple, independent, of single attribute, and of object and subject. But in CPS, knowledge about time, space and other attributes from kinds of nodes needs to be managed, which means event fusion and combination [6]. This paper proposes that CPS is composed of components which are modelled by object-oriented method. This component-based design scheme can build good software structure, improve reuse of component and support software maintenance [7]. We define formal representation which is multi-tuple to describe event type. Concept lattice-based algebra system is constructed to implement component-based event composition which is represented by algebraic expression.

2. Algebra model of event composition

Tan Ying [8,9] first proposes CPS temporal-spatial event concept, and builds temporal-spatial event composition model. In view of the situation that, if at least one event arrives among $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$, the composite event will occur, we add operation "+" to express "or" relation on base of Tan's event model. As different components are sensitive to different events, we add attribute elements "component ID" and "event type" to the formal representation of event type.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

2.1. Formal representations for component and event

The system is divided into varieties of component. Each component has individual event set. The formal representation of component is defined as:

$$C=\{Comp_{id}, SubCompSet_{id}, EventSet\}$$

- $Comp_{id}$ denotes component instance ID, which is consisted of $SubCompSet_{id}$.
- $SubCompSet_{id}$ denotes the set of atomic components, which may be physical entities or processes.
- $EventSet$ denotes the set of event types of interest of $Comp_{id}$.

If component C is composed of atomic components C_1, C_2,..., C_n, the event sets of which are $EventSet_1$, $EventSet_2$, ..., $EventSet_n$ respectively, the corresponding $EventSet$ of C is $Op(EventSet_1 \bigcup EventSet_2 \bigcup \cdots \bigcup EventSet_n)$, where Op denotes atomic events composition.

On base of Tan's event model, we define the formal representation of event as follows:

$$\varepsilon_{cps} = \Gamma\{Comp_{id}, Obs_{id}, Phy_{id}(Envi_{id}),$$
$$Type_{id}, Attr(\mathrm{Re}\,la), \mathrm{T}^g, \mathrm{L}^g, \mathrm{T}^o, \mathrm{L}^o, \rho_{id}\}$$

- Γ denotes the type of event ε_{cps}.
- $Comp_{id}$ denotes component instance ID.
- Obs_{id} denotes the observer ID.
- $Phy_{id}(Envi_{id})$ denotes an observed physical entity ID or environment ID or component ID.
- $Type_{id}$ denotes the event trigger type, when $Type_{id}$ equals 0, ε_{cps} is an event of synchronization type, which means $Comp_{id}$ is not sensitive to event ε_{cps}; when $Type_{id}$ equals 1, it is an event of trigger type, which means event ε_{cps} has some effect to $Comp_{id}$. 0 is assigned to $Type_{id}$ as default value.
- $\mathrm{T}^g, \mathrm{L}^g, \mathrm{T}^o, \mathrm{L}^o$ [8] are temporal-spatial attributes, and $\mathrm{T}^g, \mathrm{L}^g$ are time and location when and where ε_{cps} is generated, $\mathrm{T}^o, \mathrm{L}^o$ are time and location when and where ε_{cps} is monitored by observer.
- ρ_{id} denotes confidence level of ε_{cps}, which varies with time t.

2.2. Component-based event composition model

Simon Bliudze [10] shows component-based algebra of interaction and defines interface type. Here the unary typing operator "$[\cdot]^*$" is introduced to denote event type, and binary operation "+" and "·" are introduced to denote the relations between events. This paper proposes concept lattice-based event algebra system which is built on base of component architecture and logic relations between events.

Let $subcomp_1$, $subcomp_2$,..., $subcomp_n$ be atomic components of component C, the corresponding atomic event type sets are $eventset_1, eventset_2, \cdots, eventset_n$. Let E_{comp} be the event type set of component C.

(1) Let $RadixSet = \{eventset_1, eventset_2, \cdots, eventset_n\} \bigcup \{0,1\}$ be radix of event type set E_{comp}. Element 0 denotes that it is an unreasonable event relative to component C. Oppositely, 1 denotes it's reasonable to component C. All of the composite event can be represented by event radix.

(2) If events $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$ which are of types $\Gamma_1, \Gamma_2, \cdots, \Gamma_n$ respectively arrive, and they can be integrated to ε_{cps}, the composite event ε_{cps} is called the conjunction of $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$, and the binary operation "·" is defined to express events composition relation. According to the definition of CPS concept lattice[8], if events $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ all arrive, it is uncertain that composite event ε_{cps} will be generated. When ε_{cps} is generated, the relation is expressed by $\varepsilon_{cps} = \varepsilon_1 \cdot \varepsilon_2 \cdots \varepsilon_n$.

If at least one event arrives among $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$ which are of types $\Gamma_1, \Gamma_2, \cdots, \Gamma_n$ respectively, and any of them can be extended to ε_{cps}, event ε_{cps} is accepted as the extraction of $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$. This kind of event composition relation can be represented by defining binary operations "+". Similarly, if at least one event arrives among $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$, it's uncertain that composite event ε_{cps} will be generated. When ε_{cps} is generated, the relation is expressed by $\varepsilon_{cps} = \varepsilon_1 + \varepsilon_2 \cdots + \varepsilon_n$. No matter $\varepsilon_{cps} = \varepsilon_1 \cdot \varepsilon_2 \cdots \varepsilon_n$ or $\varepsilon_{cps} = \varepsilon_1 + \varepsilon_2 \cdots + \varepsilon_n$, the partial order relations $\varepsilon_1 \prec \varepsilon_{cps}, \varepsilon_2 \prec \varepsilon_{cps}, \cdots, \varepsilon_n \prec \varepsilon_{cps}$ hold.

(3) Events are divided into two types: trigger and synchronization. Here unary typing operator "$[\cdot]^*$" is defined, where $[\cdot]^0$ denotes events of synchronization type which means the component isn't sensitive to the event, and $[\cdot]^1$ denotes events of trigger type which means the component is sensitive to it. The typing operator is used to convert event type explicitly. Note that the same event in different composite events can be typed diversely.

(4) $\forall x, y, z \in E_{comp}$, the binary operations satisfy the following axioms:

Commutativity: $x+y=y+x$; $x\cdot y=y\cdot x$;
Associativity: $(x+y)+z=x+(y+z)$; $(x\cdot y)\cdot z=x\cdot(y\cdot z)$;
Distribution: $x\cdot(y+z)=x\cdot y+x\cdot z$;
Idempotence: $x\cdot x=x$ (x is a monomial) ; $x+x=x$;
$\forall x,y,z\in E_{comp}$, the composition operations satisfy the axioms [10] as follows:

$$\begin{aligned}
&[0]^1=[0];[x+y]^\alpha=[x]^\alpha+[y]^\beta;\\
&[x]^1[0]^1=[x]^1;[[x][0]]^\alpha=[0];\\
&[x]^1[y]^1=[x]^1[y]+[y]^1;\\
&[x]^1[y]=[[x]^1[y]]^1+[0]^1[y];\\
&[[x]^1[y]]^\alpha=[x]^\alpha+[[x][y]]^\alpha;\\
&[x][y]=[[x][y]]+[0]^1[x][y];\\
&[x]^1[y][z]=[x]^1[[y]^1[z]^1];\\
&[[[x][y]][z]]^\alpha=[[x][y][z]]^\alpha;
\end{aligned}\qquad (1)$$

According to the above definition and axioms, E_{comp} is obtained by binary and unary operations based on *RadixSet* , and concept lattice-based event algebra system (E_{comp}, $[\,]^*$, +, ·) is built. The operations including binary and unary operations follow the computing priority rule that $[\cdot]^*$ has higher level than "·" which is prior to "+".

According to the expression of event composition, there must exist some relations between subevents, maybe partial order relation by "$\prec$", conjunction by "·", or extraction by "+". Concept lattice-based Hasse diagram is built as follows:

(1) The top level of Hasse diagram is the global observer which is a type of observer event.

(2) The expression of composite event "$\varepsilon_{cps}=\varepsilon_1\cdot\varepsilon_2\cdots\varepsilon_n$" describes that ε_{cps} is generated, iff $\varepsilon_1,\varepsilon_2,.....,\varepsilon_n$ all arrive. Operator "·" is denoted in Hasse diagram by arc. Expression "$\varepsilon_{cps}=\varepsilon_1+\varepsilon_2\cdots+\varepsilon_n$" describes that ε_{cps} is generated iff at least one event among $\varepsilon_1,\varepsilon_2,.....,\varepsilon_n$ arrives. Operator "+" is denoted in Hasse diagram without arc.

(3) The event ε_{cps} is of trigger type, if the component is sensitive to it. Under this circumstance, event ε_{cps} is marked with filled circle (●) in the Hasse diagram. Else it is marked with hollow circle (○).

(4) Mass of events can be composed by traversing Hasse diagram. The events which are marked with filled circle are recorded when traversal trace passes through them. These recorded events have some effect on the component.

3. Events fusion process

CPS receives mass of events all the time, which can be represented by event algebra. It's necessary to simplify events so as to decrease the running time and avoid waste of resource. Simplification process means fusion of events which are the same or have the same effect, so as to avoid response to repeated events. This paper proposes two methods for events fusion: Concept lattice-based algebraic expression simplification process and Hasse diagram-based event composition simplification process.

(1) Concept lattice-based algebraic expression simplification process

The received events received during one timeslot may be described redundantly by algebraic expression. By translating algebraic expression of events into normalized forms, the same events can be fused to one.

The normalized formalization of event algebra expression is as follows [10]:

$$x=\sum_{i=1}^{n}[x_i]'+\sum_{i=1}^{n}[y_i]+[0]'\sum_{i=1}^{n}z_i \qquad (2)$$

Where x_i,y_i,z_i are of synchronization type. The terms of the first summand indicate composite events have effects to the component. The terms of the second summand indicate composite events which are effective to the component cannot be generated by combining subevents. The terms of the third summand denotes that composite events or subevents don't belong to the event set of the component.

(2) Hasse diagram-based event composition simplification process

Concept lattice-based event algebra model is shown in the above section. There exists some relation between any two composite events. They can form partial order relation, or they have the same

conjunction event (or extraction event). Mass of events can be fused into one event, if there are partial order relations between them, or they have the same effect to system.

If there exists concept lattice-based event composition relation " $\varepsilon_{cps} = \varepsilon_1 \cdot \varepsilon_2 \cdots \varepsilon_n$ " or " $\varepsilon_{cps} = \varepsilon_1 + \varepsilon_2 \cdots + \varepsilon_n$ ", it means ε_{cps} is the direct descendant event of $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$, and $\varepsilon_1 \prec \varepsilon_{cps}, \varepsilon_2 \prec \varepsilon_{cps}, \cdots, \varepsilon_n \prec \varepsilon_{cps}$. If $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ are of the same type relative to the same component, reserve ε_{cps} and discard $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$.

On basis of algebra system, Hasse diagram and events simplification method proposed in the above sections, we summarize events composition process as follows:

a. Composite events are generated by gathering the received events during one timeslot along Hasse diagram.
b. Present the event algebra expression.
c. Translate the expression from b into the form as equation (2) by following composition operation rules as equation (1).
d. Simplify composite events. If there are partial order relations between events, and they are of the same type relative to the same component, retain the higher level events.

4. Simulation

Here we simulate that how the communication-based train control system avoids overspeeding and driving with close distance from other trains. Let *diffV* be the difference value between the highest speed and running speed, *critiV* andξbe the critical value of the speed difference and fluctuate range of it, *diffD* be the distance between train T1 and T2, *emergD* be the distance for carrying on emergency brake, and *safeD* be the safe distance.

The formal representation of events of speed type from train T1 is as follows:

$$\varepsilon_{cps1} = \{Comp_{id}, Obs_{id}, \{T1\}, Type_{id}, g(diffV), T^g, \mathrm{L}^g, \mathrm{T}^o, \mathrm{L}^o, \rho\} .$$

The formal representation of events of distance type from train T1 is as follows:

$$\varepsilon_{cps2} = \{Comp_{id}, Obs_{id}, \{T1\}, Type_{id}, f(diffD), T^g, \mathrm{L}^g, \mathrm{T}^o, \mathrm{L}^o, \rho\} .$$

g(diffV) denotes the constraint expression of speed, and *f(diffD)* denotes the constraint expression of distance. The corresponding control activity is shown as Table 1.

The speed event and distance event are combined, and the corresponding control activity is shown as Table 2.

Table 1. Speed\Distance attribute constraint of atomic event and control activity

Speed/Distance Constraint Attribute	Control Activity
$g(diffV): diffV >= critiV + \xi$	Acceleration
$g(diffV): critiV < diffV < critiV + \xi$	Constant Speed
$g(diffV): diffV >= critiV$	Non Brake
$g(diffV): 0 < diffV < critiV$	Regular Brake
$g(diffV): diffV <= 0$	Emergency Brake
$f(diffD): diffD >= safeD$	Non Brake
$f(diffD): emergD < diffD < safeD$	Regular Brake
$f(diffD): diffD <= emergD$	Emergency Brake

Table 2. Speed\Distance attribute constraint of composite event and control activity

Speed/Distance Attribute Constraint	Control Activity
$g(diffV):diffV <= 0$ *or* $f(diffD):diffD <= emergD$	Emergency Brake
$(g(diffV):0 < diffV < critiV$ *and* $f(diffD):diffD > emergD)$ *or* $(g(diffV):diffV > 0$ *and* $f(diffD):emergD < diffD < safeD)$	Regular Brake
$g(diffV):diffV >= critiV$ *and* $f(diffD):diffD >= safeD$	No Brake
$g(diffV)$: $diffV >= critiV + \xi$ *and* $f(diffD)$: $diffD >= safeD$	Acceleration
$g(diffV)$: $critiV < diffV < critiV + \xi$ *and* $f(diffD)$: $diffD >= safeD$	Constant Speed

Let *Eaccelerate* be acceleration event, *Econstant* be constant speed event, *Esafe* be non-brake event, *Eregular* be regular brake event, and *Eemerg_l* and *Eemerg_h* be emergency brake event. Hasse diagram of event composition is shown as Figure 1.

$$Esafe=[E12 \bullet E13]^1;$$
$$Eaccelerate=[[Esafe]^0 \bullet E11]^1=[E12 \bullet E13 \bullet E11]^1;$$
$$Econs\tan t=[[Esafe]^0 \bullet E14]^1=[E12 \bullet E13 \bullet E14]^1;$$
$$Eregular=[E15+E16]^1=[E15]^1+[E16]^1;$$
$$Eemerg\_l=[E17+E18]^1=[E17]^1+[E18]^1;$$
$$Eemerg\_h=[[Eregular]^0 \bullet [Eemerg\_l]^0]^1$$
$$=[(E15+E16) \bullet (E17+E18)]^1$$
$$=[E15 \bullet E17]^1+[E15 \bullet E18]^1+[E16 \bullet E17]^1+[E16 \bullet E18]^1;$$

The priority of responsiveness to event types is as follows:

$$Eregular \prec Eemerg\_h; Eemerg\_1 \prec Eemerg\_h;$$
$$Esafe \prec Eaccelerate; Esafe \prec Econs\tan t;$$

If distance event E15 from train T1 and T2 and speed event E18 from train T1 are received at one timeslot, event computing unit will traverse the Hasse diagram and generate composite events as follows:

$$[E15]+[E18] \rightarrow [Eregular]'+[Eemerg\_l]'$$
$$\rightarrow [Eregular]' \cdot [Eemerg\_l]'$$
$$=[Eregular]'+[Eemerg\_l]'+[Eregular \cdot Eemerg\_l]'$$
$$=[Eregular]'+[Eemerg\_l]'+[Eemerg\_h]'.$$

According to the priority of responsiveness to event types, retain $Eemerg\_h$ and discard $Eregular, Eregular\_l$.

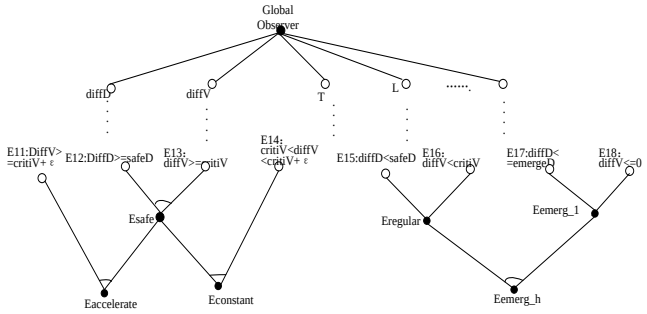

Figure 1. Hasse diagram of event composition

Figure 2 shows the change of velocity with time when different events are received by Train T1.

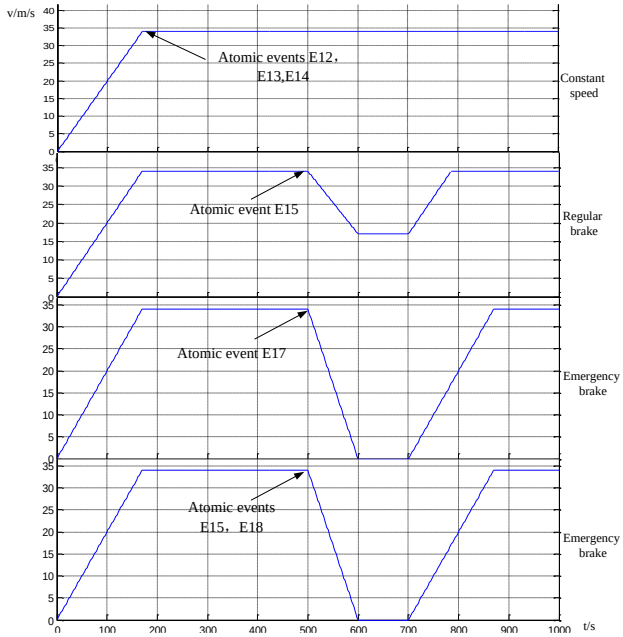

Figure 2. Train-running simulation

Event computing unit will traverse the Hasse diagram and generate composite event *Econstant*, if atomic events E12, E13, E14 are received by train T1. Then Actuator performs actions to keep an even speed. Similarly, composite event *Eregular* will be generated when atomic event E15 is received by train T1. Then the actuator performs regular brake. Composite event *Eemerg_l* will be generated when atomic event E17 is received by train T1. Then the actuator performs emergency brake. Composite event *Eemerg_h* will be generated when atomic events E15, E18 are received by train T1. Then the actuator performs emergency brake. The simulation verifies information processing mechanism and event algebra model.

5. Conclusion

This paper proposes the component-based event composition model, defines concept lattice-based event algebra system, and describes the rules for drawing Hasse diagram. Moreover, simplification methods of mass of composite events are analysed. These theories and methods are in favour of event composition and component coordination. The simulation verifies accuracy of the model and high efficiency of event processing. In the future, we will focus on two aspects: one is how to fuse events on basis of temporal-spatial consistency. The other one is how to express the priority of responsiveness in the Hasse diagram.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No.61472443).

References

[1] Shi Jianhua, Wan Jiafu, Yan Hehua, Suo Hui. 2011. “A Survey of Cyber-Physical Systems”, *International Conference on Wireless Communications & Signal Processing*, **49**, pp.1-6

[2] Shi Ling. 2014. “Analysis and design of secure cyber-physical Systems”, *Control Theory and Technology*, **12**, pp.413-414

[3] He Jifeng. 2010. “Cyber-physical system”, *Communications of China computer society*, **6**, pp.25-29

[4] R. Adaikkalavan, S. 2006. Chakravarthy. "SnoopIB: interval-based event specification and detection for active databases", *Data and Knowledge Engineering*, **59**, pp.139–165
[5] S. Chakravarthy. 1997. "Sentinel: an object-oriented DBMS with event-based rules", *Acm Sigmod Record*, **26**, pp.572–575
[6] Cugola G, Margara A. 2012. "Processing flows of information: From data stream to complex event processing", *ACM Computer Survey*, **15**, pp. 1–62
[7] Appel S, Frischbier S, Freudenreich T, Buchmann A. 2012. "Eventlets: Components for the Integration of Event Streams with SOA", *SOCA*
[8] Tan Ying, Vuran M C, Goddard S, et al. 2010. "A concept lattice-based event model for cyber-physical systems", *IEEE International Conference on Cyber-Physical Systems*, pp.50-60
[9] Tan Ying, Vuran M C, Goddard S. 2009. "Spatio-temporal event Model for cyber-physical systems", *IEEE International Conference on Distributed Computing Systems Workshops*, pp. 44-50
[10] Simon Bliudze, Joseph Sifakis. 2008. "The algebra of connectors structuring interaction in BIP", *IEEE Transactions on computers*, **57**, pp.1315-1330