An Architecture with Mixed Complex Event Processing Technology For CPS

Cheng Zhou\textsuperscript{1,a} Youqian Feng\textsuperscript{2,b} Yaru Jiang\textsuperscript{3,c}

\textsuperscript{1}Department of Basic, Air Force Engineering University
Xi’an, China
\textsuperscript{2}Department of Basic, Air Force Engineering University
Xi’an, China
\textsuperscript{3}Graduate School, Shaanxi Normal University
Xi’an, China
\textsuperscript{a}zhou417461659@163.com \textsuperscript{b}417461659@qq.com \textsuperscript{c}951421687@qq.com

ABSTRACT: Cyber-Physical System (CPS) is an emerging intelligent systems, which is considered as a combination of computers and physical world. Events are usually as a natural way for interactions and communications, which occur in the physical world and can be responded in the cyber world. Additionally, using complex event processing (CEP) technology, some complex events can be processed that are better suited for complicated and intelligent CPS. At first, a component-model framework is proposed for modeling CPS architecture based on event-driven pattern operational semantics, which provides a tool to understand the transition based on event-driven pattern in CPS. In order to improve the immense workload on centralized CEP, a mixed master-slaves CEP is proposed. Meanwhile, an architecture of CPS with master-slaves CEP is constructed and its execution mechanism is introduced in detail.

1. INTRODUCTION

The Cyber-physical system is an emerging cooperating system combining with both physical and cyber system, where those two system have access to connect and communicate with each other, usually with feedback loops [1]. As a very promising intelligent system recently, CPS has extensively been used in many fields about sensor-based system, such as autonomous cars, smart grid, distributed robotics, and wireless sensor networks [2].

As a result of integrate computation, communication and control, CPS requires rigorous interactions between the cyber and physical worlds both in time and space [3]. Hence, events are usually as a natural way for interactions and communications. Events occur in physical world and can be responded in the cyber world. However, it seemed that all events should be processed accurately and timely to guarantee temporal and spatial correctness, where a flexible architecture and task representation framework need to be studied[4,5,6,7]. Thus, a temporal and spatial events model has been introduced by YingTan et al.[5] in 2009. In [5], the temporal and spatial properties of events are discussed where events are a function of attribute-based, temporal, and spatial conditions. Furthermore, Imen Graja al. propose a time pattern-based modeling approach for CPS processes[8]. Its aim is to provide time supports to cater temporal and time-dependent physical properties.

Additionally, all mentioned events can be analyzed, aggregated and formed to a context using complex event processing (CEP) technology[9]. CEP has the capacity to process large amounts of
events which are sensed and detected by numerous sensors [10,11]. At present, a lot of researches focus largely on centralized CEP engine[9], where all events are centralized to a single CEP engine. Absolutely, it will lead to immense workload and coordination overhead on centralized CEP [9]. Hence, some scholars have committed to seeking several changed CEP technology and methodology to be fitted to any complicated and intelligent system[9]. There are three aspects for requirements of novel CEP. First, it would be possible to have several CEP engines working on distributed processing to improve the efficiency of processing numerous events. Second, large amounts of independent physical components should communicate with respective CEP engines what CEP engines need to do. Third, CEP engines should know functions and requirements of others engines.

In this paper, we propose a component-model framework for modeling CPS architecture based on event-driven pattern operational semantics, where all events are processed by mixed CEP engines. The mixed CEP engines contain two kinds of engines which are designed to process numerous events cooperatively, called master-slaves CEP pattern. The master engine knows about functions and requirements of other slaves engines, and all slaves engines know the requirements of respective components. It executes cyclically and atomically the following three-step. 1) all events are centralized to master engine and will be processed to be atomic events. 2) those relevant atomic events will be published to respective slaves CEP engines and will be processed to be interaction events which are required by components. 3) all interaction events are transferred to respective components to accomplish transitions.

The paper is structured as follows. A component-model framework for modeling CPS architecture based on event-driven pattern operational semantics is described in Section 2. In Section 3, an architecture of CPS is constructed with mixed CEP pattern, and its works is introduced in Section 4. Last section concludes and discusses future work directions.

2. BASIC CONCEPTS

In this section, a component-model is formally described. It provides a valid method to understand the transitions which are generated by interaction events among components in CPS.

**Definition 1.** The set of atomic events \( E = \bigcup_{i=1}^{n} e_i = \{e_1, e_2, \ldots, e_n\} \) is the collection of atomic events which are detected by sensors, where \( n \) is the degree of collection.

**Definition 2.** A component is a subsystem or physical entity in CPS, and it can be described as a tuple of \(\text{COM}=[E_{\text{Cyb}}, Q, T, \{|C\}_{\text{Cyb}}, \{|A\}_{\text{Cyb}}]\), where

- \( E_{\text{Cyb}} \) is the set of events which are processed by CEP engine.
- \( Q \) is the collection of states.
- \( T \subseteq Q \times E \times Q \) is the collection of transitions.
- \( \{|C\}_{\text{Cyb}} \) is the collection of transition conditions. It can be empty, a single condition, or combination of several conditions.
- \( \{|A\}_{\text{Cyb}} \) is the collection of actions which are triggered by events.

An arbitrary transition \( \tau=(q, e, q') \in T \) represents that state \( q \) transfers to \( q' \) under event \( e \). Given the transition condition \( C \), and the action \( A \), a transition also can be specified as \( (q, e, C, A, q') \).

**Definition 3.** A composite event \( f \) is composed of the combination of atomic events through CEP engine, denoted by \( (e, r, c) \).

- \( e = [e] \in 2^{E_{\text{Cyb}}} \) is the Power Set of atomic events in component \( \text{COM} \);
- \( r = \{\land, \lor, \neg\} \) is the operators, representing synchronization occurrence, parallelizing occurrence, and nonoccurrence;
- \( c_{\text{Cyb}} \) are the conditions of occurrence for composite event.

Thus, composite event can be described as combination of any atomic events through operators \( r \). Those heterogeneous and numerous composite event will be more suitable for intelligent and complex CPS. An example are as following:
Example 1. (1) Composite event $e_i$ represents both $e_i$ and $e_j$ occurrence of synchronization, and the time of occurrence of $e_i$ is three seconds later than $e_j$, denoted by

$$e_i = e_i[\text{attr, tim, loc}, c, sr] \land e_j[\text{attr, tim, loc}, c, sr], (\text{tim} - \text{tim}_j = 3s).$$

(2) Composite event $e_i$ represents synchronization of occurrence of $e_i$ and nonoccurrence of $e_j$, and the time of occurrence of $e_i$ is three seconds later than $e_j$, denoted by

$$e_i = e_i[\text{attr, tim, loc}, c, sr] \lor \neg e_j[\text{attr, tim, loc}, c, sr], (\text{tim} - \text{tim}_j = 3s).$$

Definition 4. Numerous composite events can be produced by composite operation. However, only several composite events are significant to system. Interaction event $In$ is introduced to describe the special composite event which can generate transitions. The set of interaction is denoted by $I$. 

Definition 5. A system is composed of many components, where interaction event can be regarded as the communication unit between components, denoted by

$$\Gamma[(\text{COM})], \text{Cond}, \text{Act}, \text{Init}, i \in [1,n]=I,$$

where $\Gamma[(\text{COM})]$ are components in CPS, $\text{Cond}_i$ is a set of transition conditions, $\text{Act}_i$ is a set of actions for each state under relative interaction events. $\text{Init}$ is set of initial states $q = (q_1, \cdots, q_r)$. The interaction between components can be described as a transitions system $(\Omega, E, \rightarrow)$,

- $Q = \prod_{i=1}^{n} \text{COM}_i, Q$ are all states to components;
- $E_i = \bigcup_{j \in I} \{ In \in I \}_{i}$ are interaction collection of events;
- $\rightarrow$ is the state transition. It meets the rule as following,

$$\forall e_i \in \text{In} : q_{i} = \begin{cases} \text{cond}, & q_i = q_i \land \text{true} \\ \uparrow \text{Cond}, & q_i \neq q_i \end{cases} \\ q_i \rightarrow q_i,$$

It means that a transition can be executed only if an interaction event $e_i$ is obtained; otherwise, the state remains constant.

Definition 6. The concept of priority is introduced to solve the phenomenon of conflict for response of interaction events. For any interaction events $\text{In}, \text{In}' \in I$, the priority $P = (\text{In}, \text{In}')$ is a rigorous partial order, representing $\text{In} \prec \text{In}'$ or $\text{In} < \text{In}'$. It means that the priority of $\text{In}'$ is higher than $\text{In}$, i.e., the transition generated by $\text{In}'$ will be prioritized when different transitions are triggered by interaction $\text{In}$ and $\text{In}'$ simultaneously. Thus transitions with priority can be represented as a transition system $(\Omega, E, \rightarrow, \rightarrow')$ for components $P[(\text{COM})]. \rightarrow$ is the transition with a higher priority, specifying as:

$$\exists f' \in \Gamma: f = \begin{cases} e_i, & \forall \text{In}, \text{In}' \in I, \exists q_i, q_i' \in Q \\ \forall e_i \in \text{In} : q_i = \begin{cases} \text{cond}, & q_i = q_i \land \text{true} \\ \uparrow \text{Cond}, & q_i = q_i' \end{cases} \\ \rightarrow q_i, q_i \rightarrow q_i' \end{cases}$$

For arbitrary two interaction events $\text{In}, \text{In}' \in I$, the transition from $q_i$ to $q_i'$ can be generated under $\forall e_i \in \text{In}$ and $\text{Cond}_i = \text{true}$ if any higher priority of transition from $q_i$ to $q_i'$ under $\forall e_i' \in \text{In}$ and $\text{Cond}_i' = \text{true}$ does not exist.

3. ARCHITECTURE

In this section, a mixed CEP is designed to better suit for CPS, called master-slaves CEP pattern. There are two categories of CEP engines to process large numbers of events cooperatively.

All events are centralized to master CEP engine and will be processed to be atomic events through fusion operation. Then those relevant atomic events will be delivered to respective slaves CEP engines and will be processed to be interaction events. At last, the whole relevant interaction events are
transferred to respective components in which a transition can be accomplished through response of actions. The architecture of CPS with master-slaves CEP pattern is shown in Fig. 1.

Figure 1. Architecture of CPS with master-slaves CEP.

Sensors Group (SR(IDi)) : Sensors are the interface of physical and cyber world. They can measure significant information from physical word and convert them to be identifiable cyber-physical events.

Master CEP engine (Engine(ID,O)) : Master CEP engine has the ability of storage, fusion operation, classification, and publishing. From previous section, there probably exist several redundant events which are detected by diverse categories of sensors. Thus, those redundant events can be processed to be atomic events by master engine. Actually, in many cases the states are given for every component so that the interaction events on current state can be determined. All interaction events are the power set of atomic events, and they are significant composite events. Hence, it is bound to affect the efficiency of execution for making the composite events aimlessly. Hence, numerous relevant atomic events can be picked through the function of classification by master engine for the requirement of interaction events of current state in each component. Those atomic events will be published to respective slaves engines for further processing.

Furthermore, some specials signal events can be responded to perfect the execution of system from the participation of human. For example, given a temperature control of smart home system, the cyber-physical event high temperature which is detected by temperature sensor will be processed by CEP engine and generates an action of refrigeration. However, it is not appropriate for such a continuous refrigeration owning to the sickness like fever for human. Thus it is necessary for human to participate in the system to deliver the signal event keeping temperature for a comfortable indoor environment.

Slave CEP engine (Engine(IDj,COMj)) : Slave CEP engine belongs to respective component and can be utilized for further processing about relevant atomic events which are subscribed from master engine, containing subscribing atomic events, processing of composite events, detection of interaction events, and modeling of priority. At first, all relevant atomic events which belongs to \( COM_j \) are subscribed to slave engine. According to the requirement of interaction events for current state, large amounts of composite events \( E_{\text{preconj}} \) can be obtained from composite operation. Meanwhile, those composite events are the power set of atomic events, denoting \( E_{\text{preconj}} = 2^{\mathcal{E}_{\text{pre}}} \). Clearly, just several interaction events belonging to \( E_{\text{preconj}} \) are significant to component. Hence, it is necessary to detect the feasible interaction events \( E_{\text{feas conj}} \subseteq E_{\text{preconj}} \) for current state. It is possible that a state can obtain two or
more interaction events. Thus, the model of priorities will be constructed to solve the conflict that current state can be triggered to different states by several interaction events.

Component \((Com_i(ID)\) : It is obvious that a component is subsystem or entity in system. It contains two parts: computation unit and actuator, which are utilized to compute and execute control signal to complete a state transition respectively.

4. PROCESS OF EVENTS

From previous section, an architecture with mixed master-slaves CEP engines is introduced briefly. This section specifies how the whole architecture works.

4.1 Runtime of Lifecycle

At first, the concept of global clock is introduced to unify the time of whole system so that every events have its own order of occurrence under the system time. It is clear that those events which are detected during the same runtime lifecycle can be processed by CEP engine. Meanwhile, most of sensors detect events in local coordinate system, while it is possible that some events require global information. Thus, another concept of global coordinate system is introduced to unify the position information of whole system. An example is introduced to specify the runtime lifecycle of the architecture.

Example 2. The runtime lifecycle is shown in Fig. 2. A system has four layers, sensors, mixed CEP engine, computation unit, and actuator. In layer of sensors, \(SR = \{sr_1, sr_2, sr_3\} \) are three kinds of sensors, depicted by solid circle (●), curved triangle (▲), and triangle (▲). The sample periods are \(T_{sr_1}, T_{sr_2}, T_{sr_3}\) respectively. All these sensors detect the physical world periodically, and process those
significant information to be cyber-physical events. In the layer of mixed CEP engines, all cyber-physical events will be processed to be cyber events after $t^n$. Sensors will continue to detect events after $t^n$ while those events will be processed in next lifecycle by mixed CEP engines. Then, those cyber events will be send to computation unit after $t^n$. In the third layer, computation unit converts those cyber events to be control signal after $t^n$ and sends them to actuator after $t^n$. At last, the actuator responds actions based on control signal during $t^n$. Thus, one runtime lifecycle is completed. Meanwhile, recycling it again and again, the interaction between physical and cyber world can be executed.

4.2 Flow of Events
In this section, the processing flow of events will be introduced in detail for previous architecture.

1) Sensors Group
Large numbers of cyber-physical events will be sensed and detected by sensors. However, most of those events are possible redundant. The set of cyber-physical events $E_{SR}$ from $N$ sensors can be represented by:

$$E_{SR} = \bigcup_{i=N} E \{sr \}_{i} = \bigcup_{i=N} \{ e[\text{attr}, \text{tim}(t), \text{loc}(\text{local}), c_{p}, sr_{i}] \},$$

Where $\text{attr}$ is the attribution of events, which is determined by the categories of sensors. $\text{tim}(t)$ is the time of occurrence with the time stamp of sensors. $\text{loc}(\text{local})$ is the location of occurrence in local coordinate system of sensors. $c_{p}$ are conditions of occurrence, and $sr_{i}$ is the sensor.

2) Master engine
All cyber-physical events will be centralized to process through the operation of storage, fusion, classification, and publishing by the master engine.

- **Storage.** Two categories events need to be storage. One is the set of cyber-physical events $E_{SR}$ which are delivered by $N$ sensors. Another is that the set of atomic events $E_{COM}$ which involves in interaction events of current state for component $comj$.

- **Operation of fusion.** The operation of fusion can be utilized for two ways. First, fusion operation can eliminate those redundant events which usually have the same attribution. The other application of fusion operation is that compensates some information about relative events. Such as a velocity sensor detects an speeding event in local coordinate system and the information of global position must be obtained. Thus, it is necessary that another sensor records the position of occurrence and converts it in global coordinate system, labeling observer. The method of fusion operation can refer to the technology of data fusion. The set of atomic events $E_{\nu}$ which contains $I$ elements after the operation of fusion $\oplus$ can be denoted by:

$$E_{\nu} = \oplus_{E_{SR}} \bigcup_{i=N} E \{sr \}_{i} = \bigcup_{i=N} \{ e[\text{attr}, \text{tim}(t), \text{loc}(\text{SYS}), c_{p}, sr_{i}] \}$$

- **Classification.** The operation of classification is utilized to select the relevant atomic events which involves in interaction events of current state for respective components, denoting

$$\hat{E}_{COM} = \bigcup_{comj} \hat{E}_{COM} = \bigcup_{comj} \bigcup_{n=1} \hat{E}_{[\text{attr}, \text{tim}, \text{loc}, c_{p}, sr_{i}, \text{eg}_{j}]}.$$
Step 1: Take out the first element $e_i$ in $E_i$.
Step 2: Let $i = 1$.
Step 3: Let $j = 1$.
Step 4: If $e_j \in E_i, com$, let $\hat{e}_{i,j} = e_j$; then, go to step 5.
Step 5: If $j < J$, let $j = j + 1$ and return to step 4; else, go to step 6.
Step 6: If $i < I$, let $i = i + 1$ and return to step 3; else, stop iteration and return $\hat{E}_\text{com} = \bigcup_{a \in \mathbb{A}, \alpha \in \mathbb{A}, \beta \in \mathbb{A}} \{\hat{e}_{\alpha, \beta} \}$. 

- Publish. Publishing is utilized to send all atomic events $\hat{E}_\text{com}$ to respective slave engines. The classification is the premises and basis for publishing.

3) Slaves engines
All atomic events will be further processed to be interaction events by respective slave engines. Its processing specifies as following.

- Subscribe. Each slave engine will subscribe atomic events which involve in interaction events of current state for component.

- Storage. There are three categories of events will be storage in slaves engines. First is the set of atomic event $\hat{E}_\text{com}$ which are subscribed from master engine. Second is the set of composite events $E^\gamma_i$ which are generated by composite operation. Third is the set of interaction events $E_i, com$ of current state which are delivered by components.

- Composite operation. Composite operation is adopted to generate composite events through the operator $\{\gamma\}$. Make a rule that all atomic events $\hat{E}_\text{com}$ which are occurrence during the same period of runtime lifecycle can be processed to be composite events, denoting $E_i, com$

$$E_i, com = \bigcup_{a \in \mathbb{A}, \alpha \in \mathbb{A}, \beta \in \mathbb{A}} \{\hat{e}_{\alpha, \beta} \}.$$ 

- Detection of interaction events. The process of detection specifies in Tab. 2. As large amounts of composite events are no significant to systems, it is necessary to detect significant interaction events $\hat{E}_\text{com}$ from numerous composite events, where

$$\hat{E}_\text{in} = \bigcup_{a \in \mathbb{A}, \alpha \in \mathbb{A}, \beta \in \mathbb{A}} \{\hat{e}_{\alpha, \beta} \}.$$ 

Table 2. The processing of detection interaction events

Algorithm of detection interaction events

Step 1: Take out the first element $\hat{e}_i^{(n)}$ in $E_i$, and let $k = 1$.
Step 2: If $E_i^{(n)} \in E_i$, let $\hat{e}_i^{(n)} = E_i^{(n)}$; then, go to step 3.
Step 3: Let $k = k + 1$ and go to step 4.
Step 4: If $k > K$, stop iteration and return $\hat{E}_\text{com} = \bigcup_{a \in \mathbb{A}, \alpha \in \mathbb{A}, \beta \in \mathbb{A}} \{\hat{e}_{\alpha, \beta} \}$; else, return to step 2.

- Modeling of priority. For definition 8, it is possible that there are several feasible interactions for current state. It will generate the problem of conflict. Thus, the modeling of priority can be utilized to void this conflict. For example, for two arbitrary interaction events $\forall \hat{e}_i^{(m)} \hat{e}_i^{(n)} \in \hat{E}_\text{com}$, their priority can be modeled as:

$$\text{Priority}_\text{com} = \{\hat{e}_i^{(m)} \hat{e}_i^{(n)}\}.$$ 

It means that $\hat{e}_i^{(m)}$ has a higher priority than $\hat{e}_i^{(n)}$, i.e., the transition generated by $\hat{e}_i^{(m)}$ will be executed firstly.
5. CONCLUSION AND FUTURE WORK

In this paper, a component-model framework is proposed for some preliminaries and concepts of CPS events. It provides a useful method to better understand the transition based on event-driven pattern. Meanwhile, a modeling of priority is introduced to avoid conflict.

Also, a mixed CEP framework is provided for CPS where two categories of CEP engines are designed to process large numbers of events cooperatively, called master-slaves CEP pattern. Thus, an architecture of CPS with mixed CEP engine is constructed. Additionally, the specific processing of events is also introduced.

Future work direction includes the study of derivation on interaction events in CEP, in particular by rigorous method of algebra. Also, a method on detection interaction events will be researched to enhance the efficiency for retrieval from larges amounts of composite events. Additional, the fusion operation will be further discussed to find a suitable utilization for event processing in CPS.

REFERENCES

[1] Edward A. Lee, The Past, Present and Future of Cyber-Physical Systems: A Focus on Models, Sensors, 15, 4837-4869, 2015.
[2] Jan Oliver Ringert, Bernhard Rumpe. Andreas Wortmann. Architecture and Behavior Modeling of Cyber-Physical Systems with MontiArcAutomaton. Aachener Informatik-Berichte, Software Engineering, Band 20. February 27, 2015.
[3] Edward A. Lee. Cyber Physical Systems: Design Challenges. Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on, 5-7 May 2008. Orlando, Florida, USA. pp, 363-369. DOI: 10.1109/ISORC.2008.25
[4] Keliang Zhou, Cen Ye, Jiafu Wan, etc. Advanced Control Technologies in Cyber-Physical System. Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on. 26-27 Aug. 2013. Hangzhou, China. Pp: 569 – 573. DOI : 10.1109/IHMSC.2013.284
[5] Ying Tan, Mehmet C. Vuran, Mehmet C. Vuran. Spatio-Temporal Event Model for Cyber-Physical Systems. Distributed Computing Systems Workshops, 2009. ICDCS Workshops’09, 29th IEEE International Conference on. 22-26 June 2009. Montreal, Quebec, Canada. pp, 44 - 50. DOI: 10.1109/ICDCSW.2009.82
[6] Xiaoyu Li, Yuying Wang, Xingshe Zhou. An Event-based Architecture for Cyber Physical Systems. Information Science and Technology (ICIST), 2014 4th IEEE International Conference on, 26-28 April 2014, Shenzhen, China. Pp, 96 – 99. DOI : 10.1109/ICIST.2014.6920340
[7] Fangfang Li, Jia Xu, Ge Yu. A Survey on Event Processing for CPS. Advances in Wireless Sensor Networks, 334, pp 157-166. Springer Berlin Heidelberg.
[8] Imen Graja, Slim Kallel, Nawal Guermouche et al. Time Patterns for Cyber-Physical Systems. Computers and Communication (ISCC), 2016 IEEE Symposium on, 27-30 June 2016, Messina, Italy. pp, 1208-1211. DOI: 10.1109/ISCC.2016.7543900
[9] Julius Ollesch. Doctoral Symposium: Adaptive Steering of Cyber-Physical Systems with Atomic Complex Event Processing Services. Proceeding DEBS ’16 Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems, June 20 - 24, 2016, Irvine, California. Pp. 402-405. DOI: 10.1145/2933267.2933427
[10] Cheng Zhou, Zhonghai Yin, Youqian Feng. An algebraic complex event processing method for cyber-physical system. Cluster Computing(2018): 1-9.
[11] Cheng Zhou, Zhonghai Yin, Youqian Feng. Events Algebra of Triggers. The Computer Journal, 1, (2017) 1-10.