Spatio-Temporal Properties Analysis for Cyber-Physical Systems

Zhucheng Shao[1], Jing Liu*[1], Zuohua Ding[2], Mingsong Chen[1], Ningkang Jiang[1]
1 Shanghai Keylab of Trustworthy Computing, East China Normal University, Shanghai, China
shaozhucheng@ecnu.cn, jliu@sei.ecnu.edu.cn
2 Center of Math Computing and Software Engineering, Zhejiang Sci-Tech University, Hangzhou, Zhejiang, China

***Abstract*—Cyber-Physical Systems (CPSs) integrate computing, communication and control processes. Close interactions between the cyber and physical worlds occur in time and space frequently. Therefore, both temporal and spatial information should be taken into consideration when specifying properties of CPS systems for verification. However, how to formulate properties specifying spatial together with temporal features is still an unsolved problem in the CPS. In this paper, we propose an approach to analyze the spatio-temproal properties of CPS. A spatio-temporal logic is developed, including the syntax and semantics of the logic. With that logic, properties of both states, transitions and global systems could be specified, paving the way for further verification. To show the efficiency of the approach , a Train Control System is introduced as a case study. Meanwhile, more details about how to specifying properties of CPS systems with our method are elaborated.**

***Keywords*-spatio-temporal logic, CPS, property, specification**

I. Introduction

The Cyber-Physical Systems (CPSs) are envisioned as heterogeneous systems of systems, which involve communication, computation, sensing, and actuating through heterogeneous and widely distributed physical devices and computation components [5], [6]. Therefore, CPS requires close interactions between the cyber and physical worlds in aspects of both in time and space. There are some research works on designing and modeling CPS, such as a CPS event model, which incorporates the spatio-temporal attributes and observer information into the event definition, which has been proposed by Tan.Y and et al [4]. An extended UML statechart, Spatio-Temporal UML statechart for CPSs (STUML Statechart) has been proposed in [3]. This statechart is based on the UML Profile for Modeling and Analysis of Real-time and Embedded (MARTE) systems. In STUML Statecharts, they unified the logical time and the chronometric time variables, and extend the traditional events to CPS events. However, those work does not involve in formal verification for CPSs. Our work is try to specify properties of CPSs, as the prerequisite for formal verification.

Temporal logic is used to describe any system of rules and symbolism for representing, and reasoning about propositions qualified in terms of time. It has found an important application in formal verification, where it is used to specify requirements of real-time systems. Propositional temporal logic (PTL) is one of the best known temporal logics which has found many applications in CS and AI, e.g. program verification and specification [1], [10], [19]–[21], distributed and multi-agent systems [9] or temporal databases [8]. In [1], Zahar Manna and Amir Pnueli gave a detailed methodology for the specification, verification, and development of real-time systems using the tool of temporal logic. However, the existing approaches can not be used directly to specifying properties of Cyber-Physical Systems. The reason is the truth-values of spatial propositions can not be expressed.

Spatial logic is a number of logics suitable for qualitative spatial representation and reasoning, such as RCC-8, $BRCC$, $S4_u$ and other fragments of $S4_u$. The most expressive spatial formalism of them is $S4_u$, which extends by $S4$ with the universal modalities [13]–[15]. $S4$ was introduced independently be Orlov, Lewis and Gödel without any intention about space [11], [12]. In [13]–[15], they gave an interpretation of topological space. For modeling the truth-values of spatial propositions, the spatial logic should be taken into consideration in our constructed logic.

Spatio-temporal logic is the next apparent and natural step by combining these two kinds of reasoning. There have been attempts to construct spatio-temporal hybrids. For example, in [16], Finger and Gabbay introduced a methodology whereby an arbitrary logic system L can be enriched with temporal features to create a new system $T(L)$. The new system is constructed by combining L with a pure propositional temporal logic T. In [17], Wolter and Zakharyaschev constructed a spatio-temporal logic, based on RCC-8 and PTL, intended for qualitative knowledge representation and reasoning about the behavior of spatial regions in time. Nevertheless, RCC-8 is a fragment of $S4_u$ and has rather limited expressive power [18]. The syntax of RCC-8 only contains eight binary predicates. Nor can RCC-8 represent complex relations between more than two regions. Following their way, we will construct our spatio-temporal logic by enriching PTL with $S4_u$.

This paper is organized as follows. Section II gives the existent propositional logics(e.g. $S4$, $S4_u$, PTL) for modeling the space and time. In section III, we develop a spatio-temporal logic based on $S4_u$ and PTL, together with the syntax and semantics of the logic. Section IV presents

* Corresponding Author

the method for specifying properties of states, transitions and global systems. In section V, we give an application example of Train Control System, which shows the usage of our method in specifying properties of CPS systems.

II. Spatial logic and temporal logic

In this section, we introduce a spatial logic $S4_u$ and a temporal logic PTL as preliminaries.

A. Spatial logic

$\mathcal{S}4_u$ is the most expressive spatial formalism, which is generated by extending the proposition modal logic $\mathcal{S}4$ with the $universal$ and the $existential\ quantifiers$ $\boxtimes$ and $\diamondsuit$. $\mathcal{S}4$ was introduced independently by Orlov (1928), Lewis (1932), and Gödel(1933) without any intention to reason about space. The formulas of $\mathcal{S}4$ can be defined as follows:

$$\tau ::=\ p \mid \overline{\tau} \mid \tau_1 \sqcap \tau_2 \mid \mathrm{I}\tau$$

where the p are variables. There are two modal operators denoted by I(It is necessary or provable) and C(It is possible or consistent). $\mathrm{C}\tau = \overline{\mathrm{I}\overline{\tau}}$.

Topological space is the intended interpretation for the logics, which are suitable for qualitative spatial representation and reasoning. A $topological\ space$ is a pair $\mathcal{I} = \langle U, \mathbb{I}\rangle$ in which U is a nonempty set, the universe of the space, and $\mathbb{I}$ is the interior operator on U satisfying the standard $Kuratowski\ axioms$:

$$\mathbb{I}(X \cap Y) = \mathbb{I}X \cap \mathbb{I}Y,\ \mathbb{I}X \subseteq \mathbb{I}\mathbb{I}X,\ \mathbb{I} \subseteq X,\ \mathbb{I}U = U\ \ (X, Y \subseteq U)$$

The operator dual to $\mathbb{I}$ is called the $closure\ operator$ and denoted by $\mathbb{C}$: $\mathbb{C}X = \overline{\mathbb{I}\overline{X}} = U - \mathbb{I}(U - X),\ \ X \subseteq U$. Thus, $\mathbb{I}X$ is the $interior$ of a set X, while $\mathbb{C}X$ is its $closure$. X is called *open* if $X = \mathbb{I}X$ and *closed* if $X = \mathbb{C}X$. The complement of an open set is closed and vice versa. The $boundary$ of a set X is defined as $\mathbb{C}X - \mathbb{I}X$. In addition, X and $U - X$ have the same boundary.

$\mathcal{S}4$ could be interpreted as a topological space: if we interpret the propositional variables as subsets of a topological space, the Boolean connectives as the standard set-theoretic operations, and I and C as the interior and the closure operators respectively.

In detail, a $topological\ model$ of $\mathcal{S}4$ is a two-tuples $\mathcal{M} = \langle \mathcal{I}, \mathfrak{U}\rangle$, where $\mathcal{I} = \langle U, \mathbb{I}\rangle$ is a topological space and a $valuation$ $\mathfrak{U}$ is a mapping of every variable p onto a set $\mathfrak{U}(p) \subseteq U$. Then we can get the following characters:

$$\mathfrak{U}(\overline{\tau}) = U - \mathfrak{U}(\tau),\ \mathfrak{U}(\tau_1 \sqcap \tau_2) = \mathfrak{U}(\tau_1) \cap \mathfrak{U}(\tau_2),\ \mathfrak{U}(\mathrm{I}\tau) = \mathbb{I}\mathfrak{U}(\tau)$$

The spatial formulas of $\mathcal{S}4_u$ can be defined as follows:

$$\varphi ::= \boxtimes\tau \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$$

where the τ are spatial terms. Given a spatial term τ, we write $\diamondsuit\tau$ to say that the part of space represented by τ is not empty (sc. there is at least one point in τ); $\boxtimes\tau$ means that τ occupies the whole space (all points belong to τ). $\diamondsuit\tau = \neg\boxtimes\overline{\tau}$.

B. Temporal logic

The temporal logic, which is intended to introduce, is the propositional temporal logic PTL. PTL is interpreted in various flow of time which are modeled by strict linear orders $\mathfrak{T} = \langle W, <\rangle$, where W is a nonempty set of time points and $<$ is a connected, transitive and irreflexive precedence relation on W. The set of PTL-formulas is defined in the following way:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U} \varphi_2 \mid \varphi_1 \mathcal{S} \varphi_2$$

where p is propositional variable, $\neg$ and $\wedge$ are the Booleans, $\mathcal{U}$ and $\mathcal{S}$ are the binary temporal operators.

PTL-models are two-tuples $\mathfrak{M} = \langle \mathfrak{T}, \mathfrak{U}\rangle$, where $\mathfrak{T} = \langle W, <\rangle$ is a flow of time and $\mathfrak{U}$ is a map associating with each variable p a set $\mathfrak{U}(p) \subseteq W$ of time points (where p is supposed to be true). The $truth-relation$ $(\mathfrak{M}, w) \vDash \varphi$ is defined as follows, where (u, v) denotes the open interval $\{w \in W | u < w < v\}$:

- $(\mathfrak{M}, w) \vDash p$ iff $w \in \mathfrak{U}(p)$,
- $(\mathfrak{M}, w) \vDash \neg\varphi$ iff $(\mathfrak{M}, w) \nvDash \varphi$,
- $(\mathfrak{M}, w) \vDash \varphi_1 \wedge \varphi_2$ iff $(\mathfrak{M}, w) \vDash \varphi_1$ and $(\mathfrak{M}, w) \vDash \varphi_2$,
- $(\mathfrak{M}, w) \vDash \varphi_1 \mathcal{U} \varphi_2$ iff there is $v > w$ such that $(\mathfrak{M}, v) \vDash \varphi_2$ and $(\mathfrak{M}, u) \vDash \varphi_1$ for all $u \in (w, v)$,
- $(\mathfrak{M}, w) \vDash \varphi_1 \mathcal{S} \varphi_2$ iff there is $v < w$ such that $(\mathfrak{M}, v) \vDash \varphi_2$ and $(\mathfrak{M}, u) \vDash \varphi_1$ for all $u \in (v, w)$.

III. Spatio-temporal logic

In CPS, the attributes of an event are application-independent. All CPS events have time, locations and observer attributes. Observer attributes include two kinds of variables: 1) discrete variables with discrete values independent on time.e.g.$x = 1, 2, 3$. 2) continuous variables which can be represented as a continuous function dependent on time(with its initial condition).e.g

$$F \triangleq \begin{cases} x = x(t) \\ x(0) = 0 \end{cases}$$

where t is the current reading of specific clock. Therefore, the logic, which will be defined, should own the ability of expression on locations and observer attributes.

The syntax of our spatio-temporal logic is defined as follows:

$$\begin{aligned} \tau &::= s \mid \overline{\tau} \mid \tau_1 \sqcap \tau_2 \mid \mathrm{I}\tau \\ \varphi &::= p \mid \boxtimes\tau \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U} \varphi_2 \mid \varphi_1 \mathcal{S} \varphi_2 \end{aligned}$$

where

- p are normal propositional variables e.g. $p_0,\ p_1,\ p_2, \ldots$ in relation to observer attributes;
- τ are spatial terms in relation to location; $\overline{\tau}$ is the complementary terms of τ;

- $\tau_1 \sqcap \tau_2$ is the intersection of spatial terms τ_1 and τ_2, including any point which belongs to term τ_1 and belongs to term τ_2 too;
- $\mathrm{I}\tau$ is the modal operator on spatial term τ;
- $\neg$ and $\wedge$ are the Booleans;
- $\boxed{\forall}\,\tau$ means that τ occupies the whole space (all points belong to τ). We write $\Diamond_{\exists}\tau$ to say that the part of space represented by τ is not empty (sc. there is at least one point in τ). Obviously, $\Diamond_{\exists}\tau = \neg\boxed{\forall}\,\overline{\tau}$.
- $\mathcal{U}$ and $\mathcal{S}$ are the binary temporal operators.

Certainly, the semantics of our spatio-temporal logic can be interpreted by a topological temporal model. The topological temporal model is a triple of the form

$$\mathfrak{M} = \langle \mathfrak{T}, \mathcal{I}, \mathfrak{U} \rangle$$

where $\mathfrak{T}$ is a flow of time, $\mathcal{I}$ is a topological space and a *valuation* $\mathfrak{U}$, as an overloaded function, on the one hand, is a map associating with every spatial term s and every time point $w \in W$ onto a set $\mathfrak{U}(s, w) \subseteq U$–the space occupied by s at moment w; on the other hand, is a map associating with each normal propositional variable p a set $\mathfrak{U}(p) \subseteq W$ of time points.

Then we can get the following characters:

$$\mathfrak{U}(\overline{\tau}, w) = U - \mathfrak{U}(\tau, w), \quad \mathfrak{U}(\mathrm{I}\tau, w) = \mathbb{I}\mathfrak{U}(\tau, w)$$

$$\mathfrak{U}(\tau_1 \sqcap \tau_2, w) = \mathfrak{U}(\tau_1, w) \cap \ \mathfrak{U}(\tau_2, w)$$

The truth-values of spatio-temporal logic are defined as follows:

- $(\mathfrak{M}, w) \models p \quad \text{iff} \quad w \in \mathfrak{U}(p)$,
- $(\mathfrak{M}, w) \models \boxed{\forall}\tau \quad \text{iff} \quad \mathfrak{U}(\tau, w) = U$,
- $(\mathfrak{M}, w) \models \neg\varphi \quad \text{iff} \quad (\mathfrak{M}, w) \not\models \varphi$,
- $(\mathfrak{M}, w) \models \varphi_1 \wedge \varphi_2 \quad \text{iff} \quad (\mathfrak{M}, w) \models \varphi_1$ and $(\mathfrak{M}, w) \models \varphi_2$,
- $(\mathfrak{M}, w) \models \varphi_1 \mathcal{U} \varphi_2$ iff there is $v > w$ such that $(\mathfrak{M}, v) \models \varphi_2$ and $(\mathfrak{M}, u) \models \varphi_1$ for all $u \in (w, v)$,
- $(\mathfrak{M}, w) \models \varphi_1 \mathcal{S} \varphi_2$ iff there is $v < w$ such that $(\mathfrak{M}, v) \models \varphi_2$ and $(\mathfrak{M}, u) \models \varphi_1$ for all $u \in (v, w)$.

A formula of spatio-temporal logic φ is said to be satisfiable if there exists a topological temporal model $\mathfrak{M}$ such that $(\mathfrak{M}, w) \models \varphi$ for some time point w.

Theorem 1. *The satisfiability problem for spatio-temporal logic formulas in topological-temporal models based on arbitrary flows of time is PSPACE- complete.*

For proving the theorem, we can construct a PTL-formula $\varphi*$ by replacing every occurrence of subformulas $\boxed{\forall}\,\tau$ and normal propositional variable p in φ with a fresh propositional variable p_τ. Then given a PTL-model $\mathfrak{N} = \langle \mathfrak{T}, \mathfrak{U} \rangle$ for $\varphi*$ and a time point w, we get the set

$$\Phi_w = \{\boxed{\forall}\,\tau | (\mathfrak{N}, w) \models p_\tau, p_\tau \doteq \boxed{\forall}\,\tau\} \cup \{p | (\mathfrak{N}, w) \models p_\tau, p_\tau \doteq p\}$$

It is easy to see that if Φ_w is satisfiable for every $w \in \mathfrak{T}$ in a PTL-model, there is a topological-temporal model satisfying φ based on the flow of time $\mathfrak{T}$. Then, we can use the suitable algorithm [22], [23] for PTL-model to check satisfiability of Φ_w, which can be done using polynomial space.

Proof: Let φ be a formula of our spatio-temporal logic. Based on the general proving frames(in [2], Lemma B.1 or in [24], Theorem 10.36), we only extend those spatio-temporal logical formula with the normal propositional variable p. The corresponding valuation $\mathfrak{U}$ and topological space on P are added to the topological-temporal model $\mathfrak{M} = \langle \mathfrak{T}, \mathcal{I}, \mathfrak{U} \rangle$ and the topological space $\mathcal{I} = \langle U, \mathcal{I} \rangle$. For any two ultrafilter $x_1, x_2 \in V$ (V is the set of all ultrafilters over U), put $x_1 R x_2$ (R is a quasi-order on V) iff $\forall A \subseteq U$ ($\mathcal{I}A \in x_1 \to A \in x_2$). Given an Aleksandrov topological-temporal model $\mathfrak{R} = \langle \mathfrak{T}, \mathfrak{B}, \mathfrak{Q} \rangle$, where $\mathfrak{B} = \langle V, R \rangle$, $\mathfrak{Q}(p, w) = \{x \in V | \mathfrak{U}(p, w) \in x\}$. Such that, for all $w \in W$ and $x \in V$,

$$(\mathfrak{R}, \langle w, x \rangle) \models \tau \ \text{ iff } \ \mathfrak{U}(\tau, w) \in x,$$
$$(\mathfrak{R}, \langle w, x \rangle) \models p \ \text{ iff } \ \mathfrak{U}(p, w) \in x.$$

Therefore, it is satisfiable in a topological-temporal model iff it is satisfiable in an Aleksandrov topological-temporal model based on the same flow of time $\mathfrak{T}$.

With every spatial subformula $\boxed{\forall}\,\tau$ and normal propositional variable p, we rewrite them with a fresh propositional variable p_τ. The PTL-formula $\varphi*$ could be obtained from φ by replacing all its subformulas of the form $\boxed{\forall}\,\tau$ and normal propositional variables p with p_τ.

We could claim that φ is satisfiable in an Aleksandrov topological-temporal model on a flow of time $\mathfrak{T} = \langle W, < \rangle$ iff

- there exists a temporal model $\mathfrak{N} = \langle \mathfrak{T}, \mathfrak{U} \rangle$ satisfying $\varphi*$;
- for every $w \in W$, the set

$$\Phi_w = \{\boxed{\forall}\,\tau | (\mathfrak{N}, w) \models p_\tau, p_\tau \doteq \boxed{\forall}\,\tau\}$$
$$\cup \{p | (\mathfrak{N}, w) \models p_\tau, p_\tau \doteq p\}$$

is satisfiable.

It is not hard to see that the implication($\Rightarrow$) is feasible. Conversely, suppose that we have a temporal model $\mathfrak{N} = \langle \mathfrak{T}, \mathfrak{U} \rangle$, which could satisfy those conditions above.

Let the union of Φ_w: $\Gamma = \bigcup_{w \in W} \Phi_w$. For every satisfiable $\Phi \subseteq \Gamma$, construct a model based on a finite quasi-order $\mathfrak{P}_\Phi = \langle V_\Phi, R_\Phi \rangle$ and satisfying Φ. Let n =max$\{|V_\Phi| : \Phi \subseteq \Gamma, \Phi$ is satisfiable$\}$ and $\mathfrak{P}$ is the disjoint union of n full n-ary trees of depth n whose nodes are clusters of cardinality n. It is not hard to see that every $\mathfrak{P}_\Phi$ is a p-morphic image of $\mathfrak{P}$. Therefore, every satisfiable $\Phi \subseteq \Gamma$ is satisfied in an Aleksandrov model based on $\mathfrak{P}$.

Thus there is a finite quasi-order $\mathfrak{P}$. Then, for every $w \in W$, we can get $\langle \mathfrak{P}, \mathfrak{U}_w \rangle \models \Phi_w$ for some valuation $\mathfrak{U}_w$. It is obvious that φ is satisfied in the Aleksandrov topological-temporal model $\langle \mathfrak{T}, \mathfrak{P}, \mathfrak{U}^* \rangle$, where $\mathfrak{U}^*(p, w) = \mathfrak{U}_w(p)$, $\mathfrak{U}^*(\tau, w) = \mathfrak{U}_w(\tau)$, for every spatial term τ, normal propositional variable p and every $w \in W$.

Finally, we can design a decision procedure for our spaio-temporal logic, which uses polynomial space , based on the corresponding nondeterministic PSPACE algorithm [22], [23] for PTL. We modify it as follows. Firstly the algorithm constructs a temporal model $\mathfrak{N} = \langle \mathfrak{T}, \mathfrak{U} \rangle$ for the formula φ^*. For every time point $w \in W$, it produces a state. In addition, it checks that whether the set Φ_w is satisfiable. Obviously, the extra check can also be performed by a PSPACE algorithm, which doesn't increase the complexity of the complete algorithm.

Therefore, the satisfiability problem for spatio-temporal logic formulas in topological-temporal models based on arbitrary flows of time is PSPACE- complete. ■

In addition, for describing truth values of relations between spatial terms, there are some basic binary or ternary predicates on spatial terms in Fig.1, such as

- DC(X,Y)—spatial terms X and Y are disconnected,

$$DC(X, Y) = \neg \Diamond_{\exists}(X \sqcap Y)$$

- EC(X,Y)— X and Y are externally connected,

$$EC(X, Y) = \Diamond_{\exists}(X \sqcap Y) \wedge \neg \Diamond_{\exists}(\mathrm{I}X \sqcap \mathrm{I}Y)$$

- EQ(X,Y)— X and Y are equal,

$$EQ(X, Y) = \Box_{\forall}(X \sqsubset Y) \wedge \Box_{\forall}(Y \sqsubset X)$$

- PO(X,Y)— X and Y overlap partially,

$$PO(X, Y) = \Diamond_{\exists}(\mathrm{I}X \sqcap \mathrm{I}Y) \wedge \neg \Box_{\forall}(X \sqsubset Y) \wedge \neg \Box_{\forall}(Y \sqsubset X)$$

- TPP(X,Y)— X is a tangential proper part of Y ,

$$TPP(X, Y) = \Box_{\forall}(X \sqsubset Y) \wedge \neg \Box_{\forall}(Y \sqsubset X) \wedge \neg \Box_{\forall}(X \sqsubset \mathrm{I}Y)$$

- NTPP(X,Y)— X is a nontangential proper part of Y,

$$NTPP(X, Y) = \Box_{\forall}(X \sqsubset \mathrm{I}Y) \wedge \neg \Box_{\forall}(Y \sqsubset X)$$

- PO3(X,Y,Z)— spatial terms X, Y and Z overlap partially,

$$\begin{aligned} PO3(X, Y, Z) = {} & \Diamond_{\exists}(\mathrm{I}X \sqcap \mathrm{I}Y \wedge \mathrm{I}X \sqcap \mathrm{I}Z \wedge \mathrm{I}Y \sqcap \mathrm{I}Z) \wedge \\ & \neg \Box_{\forall}(X \sqsubset Y \vee X \sqsubset Z \vee Y \sqsubset Z \vee \\ & Y \sqsubset X \vee Z \sqsubset X \vee Z \sqsubset Y \vee) \end{aligned}$$

- EC3(X,Y,Z)— spatial terms X, Y and Z are externally connected,

$$\begin{aligned} EC3(X, Y, Z) = {} & \Diamond_{\exists}(X \sqcap Y \wedge X \sqcap Z \wedge Y \sqcap Z) \wedge \\ & \neg \Diamond_{\exists}(\mathrm{I}X \sqcap \mathrm{I}Y \vee \mathrm{I}X \sqcap \mathrm{I}Z \vee \mathrm{I}Y \sqcap \mathrm{I}Z) \end{aligned}$$

Without doubt, there are many other complex predicates, which could be expressed using our spatio-temporal logic.

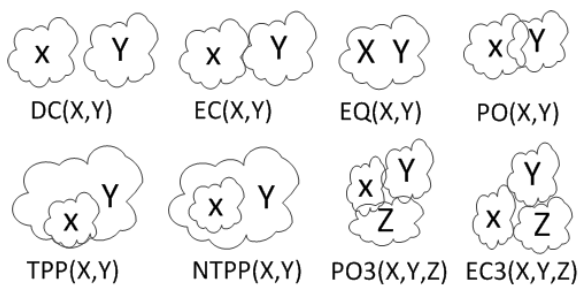

Figure 1. Basic binary or ternary predicates

IV. Modeling CPS and Analyzing Properties

In this section, we will illustrate the use of our spatio-temporal logic for specifying properties of cyber-physical systems. To introduce some structure in the extensive set of program properties, properties are clarified into six types: safety properties, guarantee properties, obligation properties, response properties, persistence properties, reactive properties.

Firstly, we provide the system description language, which is appropriate for expressing the behavior of system. The properties of single states and transitions are described at the same time.

A. CPS state models

A CPS state model is used to describe the behavior of system. The behavior of a system is abstracted to the transitions between various states. The states are classified to several modes satisfying specific constraints. The mode is one of the basic units of the CPS state model. The system execution is considered as traveling among these modes. Time flow only exists in modes, not including any transitions.

A CPS state model is a tuple [3],

$$M = (n, Var, Sub, E, avt, init, inv, elf)$$

where

- A unique name n which identifies itself. Given a CPS state model, we use a set Modes to collect all mode names.
- A set of variables Var is governed by this mode. It includes a set of discrete variables $dVar$, a set of continuous variables $cVar$, a set of clock variables $ckVar$ and a set of signals S.
- Sub is a set of names of its immediate submodes. This attribute reflects the hierarchical structure of the CPS state model. Given a CPS state model, we define a global function $Children : Mode \rightarrow 2^{Mode}$ to represent the relationship between a mode and its immediate submodes. The function subjects to the constraints of the tree structure and every mode has one father or none. If a mode has no children, it is called an atomic mode. Otherwise, it is called a composite mode. In each

submode (mode), we use a location function to record the location of these modes. $Location : Mode \rightarrow L$ where $L = \{(x, y, z)|x, y, z \in R\}$. The function $Location$ records at the current reading of global clock r, the location situation of all the submodes.

- E is a set of edges connecting its submode in Sub. The labels marked on edges specify the trigger events(evt), guard conditions(grd) and actions of a transition behavior(act).
- avt is called the activity of a mode which is the conjunction of several $Differential\ Expressions$. The activity specifies the varying patterns of continuous variables dependent on time within the mode. It is the main constraint to the continuous variables in a mode.
- $init$ is the initial requirement of the mode. It restricts the variables to the appropriate values which guarantees the valid execution. The execution flow is permitted to enter the mode if its initial requirement is satisfied.
- inv is an invariant condition which must hold in this mode. The invariant specifies the global constraints to the variables. Whenever the invariant condition is violated, the execution flow must exit this mode. If there is no valid transition for the mode, i.e. the inv of target mode is not satisfied, system will transfer to a halt.
- elf is a special guard condition. When elf is satisfied, it will trigger a self transition of this mode, and within this transition an event can be generated. This is used to describe the situation that some changes are caused by the changes of continuous activity.

The behavior of a mode is formalized as a labeled transition system.

A labeled transition system S is a tuple

$$(States, Labels, \rightarrow, S_0)$$

where $States$ is a set of states, S_0 is the set of initial states and $S_0 \subseteq States$, $Labels$ is a set of labels which identify the transitions, $\rightarrow$is a ternary relation over States specifying the transitions between states.

In the CPS state model, a state of a system can be formalized as a structure [3]

$$(m, v, \vec{\gamma}, \iota)$$

where

- $m \in Modes$ is a mode name. It specifies the execution flow is currently in mode m.
- $v : Var \rightarrow Value$ is a function representing the evaluation for every state variable of the system. $Value$ stands for the evaluation set of Var. For $S \rightarrow Var$, when a signal s is generated, we add s into the set S; if signal s is expended, we drop the s from the set S.
- $\vec{\gamma}$: the sequence of default physical clock whose reading is converted by current clock defined within the transition.
- $\iota : \iota_{\vec{\gamma},m} = Location_{\vec{\gamma}}(m)$. ι records the location of mode m with respect to default physical clock $\vec{\gamma}$. For each reading in γ, function will record a system location.

In the single states and transitions, there are some equation or inequation expressions acting as predicates for observer variables. The predicates on spatial terms could be expressed as shown in Section III. Spatial terms could be generated from the location variables by giving a radius as a brief method or generated by curvilinear equations based on lots of locations.

- $A(x)$ is an algebraic expression in terms of either an equation or inequation over algebraic objects. The expression is satisfied when the evaluation of every discrete variable makes the equation or inequation hold.
- $F(x, \frac{dx}{dt})$ is a differential expression with its initial condition describing the properties involving continuous variables(x) dependent on time. $w(t)$ is a continuous variable, the differential expression is satisfied when $w(t)$ is a solution of $F(x, \frac{dx}{dt}) = 0$, and acts on the domain of continuously differentiable function.
- Giving a radius r, for the location ι, we can get the area occupied by them, named $\tau = (\iota, r)$. For the location attributes ι_1 and ι_2, we can get the area occupied by them, spatial terms τ_1, τ_2. The relation between of them can be described using basic binary predicates or some other predicates which are expressed by our spatio-temporal logic, such as $PO(\tau_1, \tau_2)$, $TPP(\tau_1, \tau_2)$, $NTPP(\tau_1, \tau_2)$.

Those predicates are used to describe properties of single states and transitions in Cyber-Physical systems.

B. *Classified properties of CPS*

After introducing the method that describes properties of single states and transitions, based on the system description language, we proceed to study more complex properties that can be expressed by formulas.

- Safety Properties

  Safety property claims that all positions in a computation satisfy some properties. We define a basic safety formula to be a formula of the form

  $$\neg(\top\ \mathcal{U}\ \neg\varphi)$$

  where $\top$ is the logical constant 'true', φ is a spatio-temporal logical formula, the same below. A safety formula is any formula that is equivalent to a basic safety formula. A property that can be specified by a safety formula is called a $safety\ property$.

  An example of safety property is the formula

  $$\neg(\top\ \mathcal{U}\ \neg DC(X, Y))$$

  which specifies that spatial terms X and Y are disconnected in all states of the system. For the normal propositional variables, there is an example

  $$\neg(\top\ \mathcal{U}\ \neg(x \geq 0))$$

which specifies that x is nonnegative in all states of the system.

- Guarantee Properties
We define a basic $guaranteeformula$ to be a formula of the form
$$\top\ \mathcal{U}\ \varphi$$
This formula claims that at least one position in a computation state satisfies φ. A guarantee formula is any formula that is equivalent to a basic guarantee formula. A property that can be specified by a guarantee formula is called a $guarantee\ property$.
An example of guarantee property is the following formula
$$\top\ \mathcal{U}\ PO(X,Y)$$
which specifies that spatial terms X and Y overlap partially in some state of the system.

- Obligation Properties
Some properties cannot be expressed by either safety or guarantee formulas alone and must be expressed by a boolean combination of such formula. We therefore consider the class of such properties.
A basic simple obligation formula is a formula of the form
$$\neg(\top\ \mathcal{U}\ \neg\varphi_1) \quad \vee \quad (\top\ \mathcal{U}\ \varphi_2)$$
where φ_1 and φ_2 are spatio-temporal logical formulas. This formula claims that either φ_1 holds at all positions of a computation or φ_2 holds at some position. A simple obligation formula is any formula that is equivalent to a basic simple obligation formula. A property that can be specified by a simple obligation formula is called a $simple\ obligation\ property$.
Another normal form of obligation formulas is
$$(\top\ \mathcal{U}\ \varphi_3) \rightarrow (\top\ \mathcal{U}\ \varphi_2)$$
which claims that if some state satisfies φ_3 then some state satisfies φ_2. For example, if spatial terms X and Y are disconnected in the initiation, they will eventually become external connected. This can be described by the obligation formulas
$$(\top\ \mathcal{U}\ DC(X,Y)) \rightarrow (\top\ \mathcal{U}\ EC(X,Y))$$

- Response Properties
Usually, response properties ensure that some event happens infinitely many times. They can express the property of a system, stating that every stimulus has a response. A basic response formula is a formula of the form
$$\neg(\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ \varphi))$$
It claims that infinitely many positions in the computation satisfy φ. A response formula is any formula that is equivalent to a basic response formula. A property that can be specified by a response formula is called a $response\ property$.
Another form of response formula is
$$\neg(\top\ \mathcal{U}\ \neg(\varphi' \rightarrow (\top\ \mathcal{U}\ \varphi)))$$
The formula claims that every φ'-position is followed by a φ-position. Thus, φ is a guaranteed response to φ'.
For example, considering the following situation, a system S includes variable v, spatial terms X, Y, and statuses p, q. In the status p, the variable v is more than 10 noted as $v > 10$. Spatial terms X and Y are disconnected noted as $DC(X,Y)$ in the status q. When each status p appears, the status q will be occur as a respond to p. This can be described by the respond formula
$$\neg(\top\ \mathcal{U}\ \neg((v > 10) \rightarrow (\top\ \mathcal{U}\ DC(X,Y))))$$

- Persistence Properties
Usually, persistence formulas are used to describe the eventual stabilization of some state or past property of the system. They allow an arbitrary delay until the stabilization occurs, but require that once it occurs it is continuously maintained.
A basic persistence formula is a form of the form
$$\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ \neg\varphi)$$
The formula claims that all but finitely many positions in the computation(all positions from a certain point on) satisfy φ. A persistence formula is any formula that is equivalent to a basic persistence formula. A property that can be specified by a persistence formula is called a $persistence\ property$.
Another form of persistence is
$$\neg(\top\ \mathcal{U}\ \neg(\varphi \mathcal{S} \varphi'))$$
which specifies the eventual stabilization of φ as being caused by φ'.
For example, consider a system S with an input variable x and an out put variable y. Consider a stronger requirement than respond Properties, by which once y is set to 1 in response to $x > 5$, it remain so permanently. The property can be expressed by the persistence formula
$$\neg(\top\ \mathcal{U}\ \neg((y = 1)\mathcal{S}(x > 5)))$$

- Reactive Properties
A basic simple reactivity formula is a formula formed by a disjunction of a response formula and a persistence formula
$$\neg(\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ \varphi_1)) \ \vee\ (\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ \neg\varphi_2))$$
This formula claims that either the computation contains infinitely many φ_1-positions or all but finitely many of its positions are φ_2-positions. A reactivity formula is any formula that is equivalent to a basic reactivity formula. A

property that can be specified by a reactivity formula is called a $reactivity\ property$.

Usually, we specify such properties using another formula of the form,

$$\neg(\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ \varphi_3))\ \rightarrow\ \neg(\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ \varphi_1))$$

which is obviously equivalent to the basic simple reactivity formula.

This formula claims that if the computation contains infinitely many φ_3-positions it must also contain infinitely φ_1-positions. It is used to describe a response of a more complicated type, which does not guarantee a response to single stimuli. It is only when we have infinitely many stimuli that we must respond by infinitely many responses. This is a convenient abstraction to a situation in stimuli, but not specify a bound on how many stimuli may happen before the eventual response.

For example, for the variables x, y and spatial terms X, Y, if $x > 200 \wedge y > 150$ exists in infinitely many positions, $DC(X, Y)$ will also exists in infinitely many positions. This can be described by the reactive formulas

$$\neg(\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ (x > 200 \wedge y > 150)))\ \rightarrow \\ \neg(\top\ \mathcal{U}\ \neg(\top\ \mathcal{U}\ DC(X, Y)))$$

V. Case study: Train Control System

Intelligent Transportation System is the development direction of future transportation system. It integrates Electronic sensor technology, Data communication transmission technology, System control technology and Computer technology to manage the transportation system. It is a real-time, accurate, efficient and integrated transportation management system. In this section, we will only illustrate a preliminary Intelligent Transportation System, an communication based train control (CBTC) system [7] as a case study.

Communication Based Train Control System is the trend of development of rail train control system in the future. The core of CBTC system is Vehicle On Board Controller (VOBC) subsystem, which mainly achieves three functions on control: Automatic Train Protection (ATP), Automatic Train Supervision (ATS) and Automatic Train Operation (A-TO). ATP is the core subsystem of VOBC system. The train functions on acceleration, coasting, deceleration, stopping, and door opening are supervised by the ATP system. But its most important responsibility is to protect the system from over speed and avoid crashing, that is what we will discuss in the follow.

A. Requirements

In this system we focus on two trains, which construct a global system. The ATP devices of these two trains are used to protect the train from over speeding and avoid train crash. Therefore, there are two components: speed supervision unit (SSU) and distance supervision unit (DSU) in this system. The behavior and interaction of them can be described as follows:

- After the train finishes self-detection, ATP device is initialized.
- At the same time, distance supervision unit is initialized to observe global events on the location of trains.
- After every fix period, speed-sensor sends current speed to the speed supervision unit .
- According to the current driving mode and speed curves sent by wayside equipment, the speed supervision unit calculates the current limit speed.
- Then, SSU calculates the difference between the limit speed and the current speed $DiffSpeed$.
- 1) If $DiffSpeed$ is less than a critical speed $CriticalSpeed$ and more than zero, SSU will send a warning message to the Train Operation Display(TOD) to inform the driver of deceleration. After warning, SSU will send a normal brake message to Braking Equipment (BE). Then the BE will apply the normal brake until the speed is more than $CriticalSpeed$. All these operations should be done in 150ms.
  2) If $DiffSpeed$ is less than zero, SSU will send an e-mergency message directly to BE. Then BE will apply the emergency brake until velocity $v = 0$. These operations should be done in 100ms.
- As the trains moving in their tracks, location related events of trains are observed by distance supervision unit (DSU), i.e. locations ι_1 and ι_2, the distance between T1 and T2 Dis.
- 1) If SDU observers that the distance is more than a emerge distance $EmergeDistance$ and less than a safe distance $SafeDistance$, SDU will send a warning message to the T1 and T2 to inform the driver of deceleration. After warning, SDU will send a normal brake message to Braking Equipment (BE). Then the BE will apply the normal brake until the distance is more than $SafeDistance$. All these operations should be done in 150ms.
  2) If SDU observers that the distance is less than $EmergeDistance$, SDU will send an emergency message directly to BE. Then BE will apply the emergency brake until velocity v=0. These operations should be done in 100ms.

B. Behavior of system

Based on the syntax and semantics of CPS state model , we can model the behavior of the Protection functions of AT-P system. The most important components are Component SSU, Component DSU and Component BE . We use CPS state machines to model the behavior of them as follows.

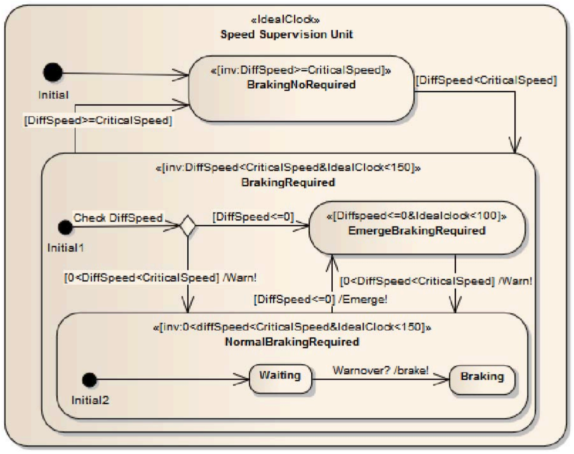
«IdealClock»
Speed Supervision Unit
Initial
«[inv:DiffSpeed>=CriticalSpeed]»
BrakingNoRequired
[DiffSpeed<CriticalSpeed]
[DiffSpeed>=CriticalSpeed]
«[inv:DiffSpeed<CriticalSpeed&IdealClock<150]»
BrakingRequired
Check DiffSpeed
Initial1
[DiffSpeed<=0]
«[Diffspeed<=0&Idealclock<100]»
EmergeBrakingRequired
[0<DiffSpeed<CriticalSpeed] /Warn!
[0<DiffSpeed<CriticalSpeed] /Warn!
[DiffSpeed<=0] /Emerge!
«[inv:0<diffSpeed<CriticalSpeed&IdealClock<150]»
NormalBrakingRequired
Initial2
Waiting
Warnover? /brake!
Braking

Figure 2. Behavior of Speed Supervision Unit

Fig.2 describes the behavior of speed supervision unit. The unit is responsible for calculating data and sending protection commands. The unique clock of them is $IdealClock$. In the following transition system we all refer the current $IdealClock$ as c for convenience. For lack of space, we only give a transition from $BrakingNoRequired$ to $BrakingRequired$ as an example.

$evt(tr) \wedge grd(tr) = \epsilon \wedge (DiffSpeed < CriticalSpeed)$,
$act(tr) = IdealClock := 0$,

When $v, c, \iota \models grd(tr)$ and $(v, c, \iota, v, 0, \iota') \models act(tr)$,
$tr \triangleq (BrakingNotRequired, v, c, \iota)$
$\quad \rightarrow (BrakingRequired, v, 0, \iota')$,
$\lambda(tr) = (\epsilon, DiffSpeed < CriticalSpeed, IdealClock := 0)$.
When braking is required, no matter what kind of braking, the braking duration should be less than 150 ms. Hence in this mode

$$inv(BrakingRequired) = (IdealClock < 150) \wedge \\ (DiffSpeed < CriticalSpeedDif)$$

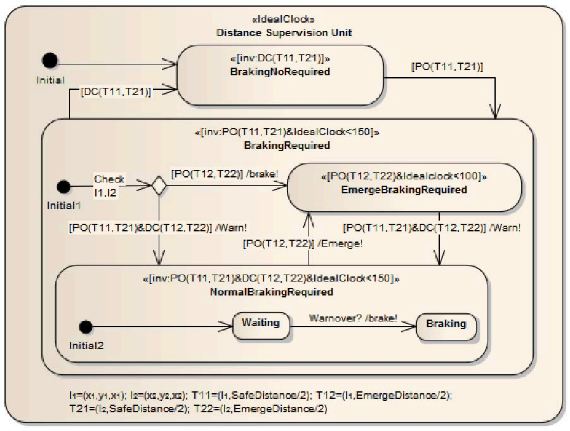
«IdealClock»
Distance Supervision Unit
Initial
«[inv:DC(T11,T21)]»
BrakingNoRequired
[PO(T11,T21)]
[DC(T11,T21)]
«[inv:PO(T11,T21)&IdealClock<150]»
BrakingRequired
Check I1,I2
Initial1
[PO(T12,T22)] /brake!
«[PO(T12,T22)&Idealclock<100]»
EmergeBrakingRequired
[PO(T11,T21)&DC(T12,T22)] /Warn!
[PO(T11,T21)&DC(T12,T22)] /Warn!
[PO(T12,T22)] /Emerge!
«[inv:PO(T11,T21)&DC(T12,T22)&IdealClock<150]»
NormalBrakingRequired
Initial2
Waiting
Warnover? /brake!
Braking
I1=(x1,y1,x1); I2=(x2,y2,x2); T11=(I1,SafeDistance/2); T12=(I1,EmergeDistance/2);
T21=(I2,SafeDistance/2); T22=(I2,EmergeDistance/2)

Figure 3. Behavior of Distance Supervision Unit

Fig.3 describes the behavior of distance supervision unit(SDU). SDU observers the system events after it has initialized. It could protect the components in the system from crashing by capturing the information from events. The transition system of SDU begins from the initial mode. At once, observer got an event of train T1 and an event of train T2 at the same time, where

$$E1 = (L_1; < attr_1; t; (x_1, y_1, z_1) >)$$
$$E2 = (L_2; < attr_2; t; (x_2, y_2, z_2) >).$$

SDU calculates spatial terms occupied by trains based on two events and ensures the distance between these two trains.

$$\tau_1 = ((x_1, y_1, z_1), r_1)$$
$$\tau_2 = ((x_2, y_2, z_2), r_2)$$

where r_1 and r_2 are radiuses, which can be generated based on the environment. Then spatial terms τ_1 and τ_2 will be used to calculate the guard and then SDU will control the signal generate through the statechart model.

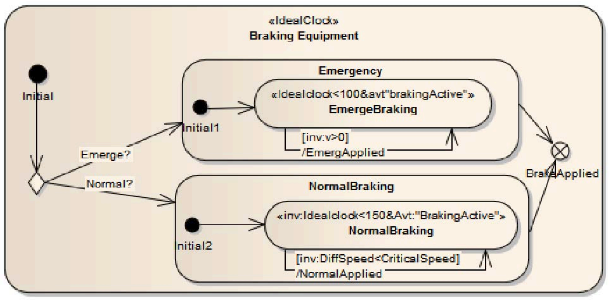

Figure 4. Behavior of component Braking Equipment

Fig.4 describes the Braking Equipment state machine under the condition of velocity $v \geq 0$. We list a transition for a short specification as follows.
$evt(tr) \wedge grd(tr) = Emerge? \wedge TRUE$,
$act(tr) = IdealClock := 0$,

When $v, c, \iota \models grd(tr)$ and $(v, c, \iota, v, 0, \iota') \models act(tr)$,
$tr \triangleq (initial_1, v, 0, \iota) \rightarrow (EmergeBraking, v, 0, \iota')$
Edge tr is fired immediately after entering submode $Emergency$.
$inv(EmergeBraking) = (IdealClock < 100) \wedge (v \geqslant 0)$
$avt(EmergeBraking) = \begin{cases} -fW = \frac{W}{g} \cdot \frac{\mathrm{d}v}{\mathrm{d}t} \\ v \mid_{t=0} = v_0 \end{cases}$.
If f is a solution of equation $avt(EmergeBraking)$, f satisfies
$inv(EmergeBraking)[f/v] = TRUE \ \wedge$
$avt(EmergeBraking)[f/v, \dot{f}/\dot{v}] = TRUE.$

In the submode $EmergeBraking$, the continuous variable v is changing when time passes, following the $avt(EmergeBraking)$. According to Newton second law of motion, fW is the external force to brake, $\frac{W}{g}$ is the weight of the train and $\frac{dv}{dt}$ is the acceleration of the train.

C. *Properties of system*

There are some properties in single states or transitions about speed such as follows.

$$\varphi_1 = diffspeed \leq 0$$
$$\varphi_2 = diffspeed \geq CriticalSpeed$$
$$\varphi_3 = diffspeed < CriticalSpeed \wedge diffspeed > 0$$
$$\varphi_3 = \neg\varphi_1 \wedge \neg\varphi_2$$

where φ_1 states that the current speed is more than the speed limit. The train is in a serious dangerous condition and need to apply the emergency bark until velocity $v = 0$; φ_2 states that the difference between the speed limit and the current speed is more than the limited value $CriticalSpeed$. When a state or a transition satisfies φ_2, the train is in the safe condition; φ_3 states that the speed is less than the speed limit and more than the safe speed. So SSU will send a warning message to TOD and send a normal brake message to Braking Equipment to apply the normal brake until satisfying φ_2.

In the train control system, every train has their own location (x, y, z) at a certain time instant. Giving a radius r of the location, we think the location occupied a spatial term $\tau = ((x, y, z), r)$, i.e. r is half of the distance between two trains.

Therefore, we can specify the properties in single states or transitions about location as follows.

$$\varphi_4 = DC(\tau_{11}, \tau_{21})$$
$$\varphi_5 = PO(\tau_{12}, \tau_{22})$$
$$\varphi_6 = DC(\tau_{12}, \tau_{22}) \wedge PO(\tau_{11}, \tau_{21})$$

where

$$\tau_{11} = ((x_1, y_1, z_1), SafeDistance/2),$$
$$\tau_{21} = ((x_2, y_2, z_2), SafeDistance/2),$$
$$\tau_{12} = ((x_1, y_1, z_1), EmergeDistance/2),$$
$$\tau_{22} = ((x_2, y_2, z_2), EmergeDistance/2),$$

φ_4 states that spatial τ_{11} and τ_{21} are disconnected. It also means that the distance of train T_1 and T_2 is more than $SafeDistance$. φ_5 states that τ_{12} and τ_{22} partially overlap. It also means that the distance of train T_1 and T_2 is less than $EmergeDistance$. Component SDU need to send an emergency message directly to BE. Then BE applies the emergency brake until velocity $v = 0$. φ_6 states that τ_{11} and τ_{21} partially overlap and τ_{12} and τ_{22} are disconnected. It also means that the distance of train T_1 and T_2 is less than $SafeDistance$ and more than $EmergeDistance$. SDU should send a warning message to T_1 and T_2. Then the BE will apply the normal brake until satisfying φ_4.

- Safety Properties
  In the train control system, SDU must ensure the trains are not able to collide during all the states. Therefore, we can get the following system properties.

  $$\neg(\top\ \mathcal{U}\ \neg\varphi)$$

  where $\varphi = DC(\tau_a, \tau_b)$, $\tau_a = ((x, y, z), 0)$, $\tau_b = ((x', y', z'), 0)$.
  The speed v of a train can not more than a limit speed LS too much. Therefore, there is a safety property on speed as follows.

  $$\neg(\top\ \mathcal{U}\ \neg\varphi')$$

  where $\varphi' = \varepsilon < diffSpeed,\ \ \varepsilon < 0$, which is equivalent to $\varepsilon < LS - v$.
- Guarantee Properties
  In the system, the train should at a safe speed and at a safe distance in most instances. Therefore, we can get the following guarantee property.

  $$\top\ \mathcal{U}\ (\varphi_v \wedge \varphi_p)$$

  where $\varphi_v = \varphi_2$, states that the difference between the speed limit and the current speed is more than the limited value $CriticalSpeed$. the train is moving at the safe speed; $\varphi_p = \varphi_4$ states that spatial τ_{11} and τ_{21} are disconnected. It also means that the distance of train T_1 and T_2 is more than $SafeDistance$. The trains are moving with the safe distance.
- Response Properties
  As the increase of speed, the braking distance will also be increased. Therefore, the control system should ensure the distance of trains longer. Giving a special speed $sSpeed$ of train T_1, the distance of train T_1 and T_2 (T_1 is behind the T_2) is more than dis. Then the spatial terms occupied by T_1 and T_2 can be described as follows

  $$\tau_1 = ((x_1, y_1, z_1), dis/2), \quad \tau_2 = ((x_2, y_2, z_2), dis/2)$$

  where $(x_i, y_i, z_i), i = 1, 2$ is the location of train T_1 and T_2. In the status p, the variable v is more than $sSpeed$ noted as $v > sSpeed$. Spatial terms τ_1 and τ_2 are disconnected noted as $DC(\tau_1, \tau_2)$ in the status q. When each status p appears, the status q will be occur as a respond to p. The properties of system can be described using the following response formula

  $$\neg(\top\ \mathcal{U}\ \neg((v > sSpeed) \rightarrow (\top\ \mathcal{U}\ DC(\tau_1, \tau_2))))$$

There are more properties can be described using our spatio-temporal logic. For lack of space, we do not list all of them in the paper. Though the case study, we only try to show the usage of our method on specifying properties of CPS. For the method on how to verify those properties, we can consider the following three ways: one is using the existing model checking tools verify those properties without spatial features; another way is transforming spatial related proposition to non-spatial proposition and using the existing model checking tools verify those properties, which are own simple spatial terms(e.g. point set, circle or rectangle) with limited predicates; the final and the most efficacious way is developing a new tool, which can support modeling arbitrary spatial terms and checking the truth-value of various predicates in the future work.

VI. Conclusion

There are logics for expressing spatial or temporal characteristics respectively e.g. $S4$, $S4_u$, PTL. For unified modeling spatial and temporal characteristics, a spatio-temporal logic was constructed based on $S4_u$ and PTL semantics. Then the spatio-temporal logic was used to describe properties of states, transitions and global systems after modeling system behavior of CPSs using the CPS state model. Finally, a Train Control System is employed as a case study to show a workflow of modeling systems behavior with a CPS state model and specifying properties of CPS systems with our spatio-temporal logic.

In the future, the related algorithms on satisfiability problems should be considered. Moreover, we will work on the verification and tool support of our spatio-temporal logic.

Acknowledgment

We would like to thank the reviewers for their valuable comments. This work is partially supported by the projects funded by the 973 program 2009CB320702, NSFC 61170084, Zuohua Ding is supported by the NSFC No.61170015. The Shanghai Key Lab is supported by Shanghai Knowledge Service Platform ZF 1213, NSFC Creative Team 61021004 and 863 Project 2011AA010101.

References

[1] Z. Manna, A. Pnueli. The Temporal Logic of Reactive and Concurrent Systems Specification. *Springer-Verlag,* 1992.

[2] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, M. Zakharyaschev. Combining Spatial and Temporal Logics. *Journal of Artificial Intelligence Research,* pages 167-243, 2005.

[3] Z. Liu, J. Liu, J. He, Z. Ding. Spatio-Temporal UML Statechart for Cyber-Physical Systems. *IEEE 17th International Conference on Engineering of Complex Computer Systems,* pages 137-146, 2012.

[4] Y. Tan, M. C. Vuran, S. Goddard, Y. Yu, M. Song, S. Ren. A concept lattice-based event model for cyber-physical systems. *in Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems,* pages 50-60, 2010.

[5] E. A. Lee. Cyber-physical systems-are computing foundations adequate. *in Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap,* pages 1-9, 2006.

[6] E. A. Lee. Cyber Physical Systems: Design Challenges. *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC),* pages 363-369, 2008.

[7] IEEE, IEEE Recommended Practice for Communications-Based Train Control (CBTC) System Design and Functional Allocations, *IEEE Std 1474.3-2008,* 2008.

[8] J. Chomicki. Temporal query language: a survey. *Temporal Logic, Lecture Notes in Computer Science,* Volume 827, pages 506-534, 1994.

[9] R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi. Reasoning about Knowledge. *MIT Press,* 1995.

[10] Z. Manna, A. Pnueli. Temporal Verification of Reactive Systems: Safety. *Springer-Verlag,* 1995.

[11] I. E. Orlov. The calculus of compatibility of propositions. *Mathematics of the USSR,* Volume 35, pages 263-286, 1928.

[12] C. I. Lewis, C. H. Langford. Symbolic Logic. *Appleton-Century-Crofts, New York,* 1932.

[13] T. Chen. Algebraic postulates and a geometric interpretation of the Lewis calculus of strict implication. *Bulletin of the AMS,* Volume 44, pages 737-744, 1938.

[14] M. H. Stone. Application of the theory of Boolean rings to general topology. *Transactions of the AMS,* Volume 41, pages 321-364, 1937.

[15] J. C. C. McKinsey. A solution of the decision problem for the Lewis systems S2 and S4, with an application to topology. *Journal of Symbolic Logic,* Volume 6, Issue 4, pages 117-134, 1941.

[16] M. Finger, D. M. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information,* Volume 1, Issue 3, pages 203-233, 1992.

[17] F. Wolter, M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. *Proceedings of the 7th Conference on Principles of Knowledge Representation and Reasoning (KR2000),* pages 3-14, 2000.

[18] M. J. Egenhofer, J. R. Herring. Categorizing topological relationships between regions, lines and points in geographic databases. *Tech.rep, University of Maine,* 1991.

[19] P. Wolper. Expressing interesting properties of programs in propositional temporal logic. *Proceedings of the 13th ACM SIGACT-SIGPLAN symposium on Principles of programming languages,* pages 184-192, 1986.

[20] M. Zhang, D. Hung and Z. Liu "Verification of Linear Duration Invariants by Model Checking CTL Properties" *UNU-IIST report 396* , 2008 .

[21] M. Zhang, Z. Liu, C. Morrssets, and A. P. Ravn, "Design and verification of fault-tolerant components", *Methods, Models, and Tools for Fault Tolerance 09*, pages 57-84, 2009.

[22] M. Reynolds. The complexity of the temporal logic with until over general linear time. *Journal of Computer and System Sciences,* Volume 66, Issue 2, pages 393-426, 2003.

[23] A. P. Sistla, E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM,* Volume 32, Issue 3, pages 733-749, 1985.

[24] C. Alexander, M. Zakharyaschev. Modal Logic. *Clarendon Press (Oxford and New York),* Vol.35 of Oxford Logic Guides, 1997 .