Spatio-temporal Hybrid Automata for Cyber-Physical Systems

Zhucheng Shao and Jing Liu*

Shanghai Keylab of Trustworthy Computing
East China Normal University
Shanghai, 200062, P.R. China
shaozhucheng@ecnu.cn, jliu@sei.ecnu.edu.cn

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 modeling CPS systems. However, how we can capture temporal and spatial information into CPS models that allow describing the logical properties and constraints is still an unsolved problem in the CPS. In this paper, a spatio-temporal logic is provided, including the syntax and semantics, for describing the logical properties and constraints. Based on the logic, we propose an extended hybrid automaton, spatio-temporal hybrid automaton for CPSs. The automaton increases the ability to express spatial variables, spatial expression and related constraints on spatial terms. Then, we define formal semantics of spatio-temporal hybrid automata based on labeled transition systems. At the end of this paper, a Train Control System is introduced as a case study to show how to model the system behavior with spatio-temporal hybrid automata.

Keywords: Spatio-temporal logic, CPS, Hybrid automata.

1 Introduction

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 [3]. Therefore, CPS requires close interactions between the cyber and physical worlds both in time and space. These interactions are usually governed by events, which have time, location and observer attributes. An unsolved important problem is how to capture location and timing information into CPS models in a way that allow for validation of the logical properties of a program against the constraints imposed by its physical (sensor) interaction [24]. Thus, a new logic is needed for describing the constraints on location and time information, and for specifying the logical properties of CPSs, since the traditional models (e.g. hybrid automata, UML, CSP) are without consideration of location information.

* Corresponding author.

Z. Liu, J. Woodcock, and H. Zhu (Eds.): ICTAC 2013, LNCS 8049, pp. 337–354, 2013.

To model a CPS with location and time, a new semantic model is needed for the purpose of designing the unified system.

Temporal logic has found an important application in formal methods, where it is used to specify requirements of real-time systems on rules and symbolism for representing and reasoning about propositions qualified in terms of time. Propositional temporal logic (PTL) is one of the best known temporal logics, which has found many applications in CS and AI [1, 7–9]. In [1], Zohar 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 in Cyber-Physical Systems. The reason is that 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 $S4_u$ [10, 11, 13]. For modeling the truth-values of spatial propositions, spatial logic should be taken into consideration in our constructed logic.

The next apparent and natural step is attempting to construct spatio-temporal hybrids. For example, in [12], 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 [14], 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 [15]. Following their way, we will construct our spatio-temporal logic by enriching PTL with $S4_u$ for CPSs.

Cyber-physical systems can be usefully modeled as hybrid automata combining the physical dynamics within modes with discrete switching behavior between modes. However, the location information can not be captured into models, especially spatial constraints. In this paper, we extend spatial variables and spatial expressions of a hybrid automaton to increase the ability of expression. Naturally, spatial expressions and other expressions with connections in spatio-temporal hybrid automata can be interpreted in our spatio-temporal logic. We give the formal syntax and semantics (includes state, transition, trace and parallel composition) of the spatio-temporal hybrid automata.

This paper is organized as follows. Section 2 gives formal syntax and semantics of the spatio-temporal logic. Section 3 classify variables and expressions. Then, we give the formal syntax and semantics of the spatio-temporal hybrid automata. In Section 4, a Train Control System is introduced as a case study to show the efficiency of our approach.

2 Spatio-Temporal Logic

In CPS, the attributes of an event are application-independent. All CPS events have time, locations and observer attributes. Therefore, the logic, which will

be defined, should have the ability of expression on locations and observer attributes. Based on the propositional temporal logic and the construction method of logics [12], our spatio-temporal logic will be constructed by extending PTL with spatial temporal logic $S4_u$. By the way, part of our work on the logic can be found in [4]. The syntax of the spatio-temporal logic is defined as follows:

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

where

- p 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;
- $\boxbslash \tau$ means that τ occupies the whole space (all points belong to τ). We write $\diamondbslash \tau$ to say that the part of space represented by τ is not empty (sc. there is at least one point in τ). Obviously, $\diamondbslash \tau = \neg \boxbslash \overline{\tau}$.
- $\mathcal{U}$ 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; w is a nonempty set of time points; U is a nonempty set, the universe of the space.

Then we can get the following definitions:

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

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

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

- $(\mathfrak{M}, w) \vDash p \quad \text{iff} \quad w \in \mathfrak{U}(p)$,
- $(\mathfrak{M}, w) \vDash \boxbslash \tau \quad \text{iff} \quad \mathfrak{U}(\tau, w) = U$,
- $(\mathfrak{M}, w) \vDash \neg\varphi \quad \text{iff} \quad (\mathfrak{M}, w) \nvDash \varphi$,
- $(\mathfrak{M}, w) \vDash \varphi_1 \wedge \varphi_2 \quad \text{iff} \quad (\mathfrak{M}, w) \vDash \varphi_1$ and $(\mathfrak{M}, w) \vDash \varphi_2$,
- $(\mathfrak{M}, w) \vDash \varphi_1 \mathcal{U} \varphi_2 \quad \text{iff}$ 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 $u \in (v, w)$.

A formula of spatio-temporal logic φ is said to be satisfiable if there exists a topological temporal model $\mathfrak{M}$ such that $(\mathfrak{M}, w) \vDash \varphi$ 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 $\boxdot \tau$ 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

$$\Phi_w = \{\boxdot \tau | (\mathfrak{N}, w) \vDash p_\tau, p_\tau \doteq \boxdot \tau\} \cup \{p | (\mathfrak{N}, w) \vDash p_\tau, p_\tau \doteq p\}$$

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 [18, 19] for PTL-model to check satisfiability of Φ_w, which can be done using polynomial space.

Proof 1. *Let φ be a formula of our spatio-temporal logic. Based on the general proving frames(in [2], Lemma B.1 or in [20], 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 $\forall A \subseteq U$ ($\mathcal{I}A \in x_1 \rightarrow A \in x_2$). 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 $x \in V$,*

$$(\mathfrak{R}, \langle w, x \rangle) \vDash \tau \quad \textit{iff} \quad \mathfrak{U}(\tau, w) \in x,$$
$$(\mathfrak{R}, \langle w, x \rangle) \vDash p \quad \textit{iff} \quad \mathfrak{U}(p, w) \in x.$$

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 $\boxdot \tau$ 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 $\boxdot \tau$ 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*

$$\Phi_w = \{\boxdot \tau | (\mathfrak{N}, w) \vDash p_\tau, p_\tau \doteq \boxdot \tau\} \cup \{p | (\mathfrak{N}, w) \vDash p_\tau, p_\tau \doteq p\}$$

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 $\langle \mathfrak{P}, \mathfrak{U}_w \rangle \vDash \Phi_w$ 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 design a decision procedure for our spatio-temporal logic, which uses polynomial space, based on the corresponding nondeterministic PSPACE algorithm [18, 19] 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,

$$DC(X, Y) = \neg \Diamond (X \sqcap Y)$$

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

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

- EQ(X,Y)— X and Y are equal,

$$EQ(X, Y) = \boxdot (X \sqsubset Y) \wedge \boxdot (Y \sqsubset X)$$

- PO(X,Y)— X and Y overlap partially,

$$PO(X, Y) = \Diamond (\mathrm{I}X \sqcap \mathrm{I}Y) \wedge \neg \boxdot (X \sqsubset Y) \wedge \neg \boxdot (Y \sqsubset X)$$

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

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

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

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

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

$$\begin{aligned} PO3(X,Y,Z) = \diamondsuit(\mathrm{I}X \sqcap \mathrm{I}Y \wedge \mathrm{I}X \sqcap \mathrm{I}Z \wedge \mathrm{I}Y \sqcap \mathrm{I}Z) \wedge \\ \neg\boxdot(X \sqsubset Y \vee X \sqsubset Z \vee Y \sqsubset Z \vee \\ Y \sqsubset X \vee Z \sqsubset X \vee Z \sqsubset Y \vee) \end{aligned}$$

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

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

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

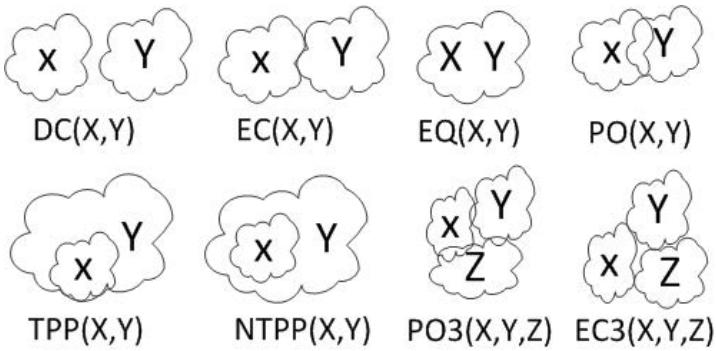

Fig. 1. Basic binary or ternary predicates

3 Spatio-temporal Hybrid Automata

A Spatio-temporal hybrid automaton is an extension of hybrid automaton based on spatio-temporal logic. Spatio-temporal logic brings a set of expressions into the hybrid automaton to represent the location or space related information of CPSs.

3.1 Expressions

After expansion of expression, there are five kinds of variables in spatio-temporal hybrid automaton :

- discrete variables with discrete values independent of time and location. e.g. $x = 0, 1, 2$.

- continuous variables which can be represented as a continuous function dependent on time (with its initial condition), e.g. $F \triangleq \begin{cases} x = x(t) \\ x(0) = 0 \end{cases}$
- clock variables with a special clock type. A *Clock* c can be defined as follows: $c ::= (I, \rho)$, where
  - I is a set of time instants. A time instant represents an observation point of a clock, either physical or logical, reflecting specific events dependent on time.
  - ρ is a function, $\rho : I \to \mathbb{R}_{\geq 0}$ maps every time instant in I to a non-negative real number. We call ρ the sampling function that discretizes the continuous time to a series of observation points on demand.
- spatial variables which can be represented by locations. The location attribute is given by the coordinate in the form of (x, y, z). So the spatial variables can be represented in the form of $((x, y, z), r)$, r stands for the radius of an object.
- action variables which have a boolean value specifying whether the corresponding CPS event occurs or not, e.g. θ, η. All CPS events $(L, \langle attr, \Gamma, \iota \rangle)$ have observer attributes, time and location, where L is a symbol and used to label the event. It is used to record the attributes, generate time and location of the event.

Corresponding to those kinds of variables, there are five kinds of expression: Algebraic Expression, Differential Expression, Clock Expression, Spatial Expression and Action Expression. *Algebraic Expressions* can be used to describe the relation between discrete variables. *Differential Expressions* are capable to describe activities of continuous variables. *Clock Constraint Expressions* are used to define clock-related expressions. *Spatial Expressions* are used to define expressions on spatial terms. *Action Expressions* are used when defining events and actions. In our formalization, every expression occurring is considered as a predicate over variables with specific types.

- $A(x)$ 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.
- The clock constraint expression can be represented in terms of either an equation or inequation over time instants. Based on the clock structure, the current reading of a clock c could be represented by the first entry of the time instant sequence $c.I$. After a clock tick, $c.I$ moves to next instant.
- 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 $PO(\tau_1, \tau_2)$, $TPP(\tau_1, \tau_2)$, $NTPP(\tau_1, \tau_2)$.
- Action expression is an expression in terms of action variables. Every action variable acts as a atom in predicate calculus.

In our formalization, algebraic expressions, differential expressions, clock constraint expressions and action expressions are considered as normal propositional variables p in our spatio-temporal logic. The semantics for logical connectives with these four kinds of expressions are interpreted by our spatio-temporal logic like the first order logic language. For example, Let $p \triangleq v_1(x) \wedge v_2(x)$, the variable $x \vDash p$ iff $x \vDash v_1(x)$ and $x \vDash v_2(x)$. Therefore, all of the expressions could be expressed by our spatio-temporal logic.

3.2 Spatio-temporal Hybrid Automata

A spatio-temporal hybrid automaton is described by a tuple

$$(M, X, A, Var, E, Inv, Act)$$

where

- M is a finite set, called the set of discrete states or modes. There are the vertices of a control graph. Every mode has a unique name to identify itself in the set.
- X is the continuous state space of the hybrid automaton. Generally, $X \subset \mathbb{R}^n$ or X is an n-dimensional manifold.
- A is a finite set of symbols which is used to label the edges.
- A set of variables Var is governed by modes. It includes a set of discrete variables $dVar$, a set of continuous variables $cVar$, a set of clock variables $ckVar$, a set of spatial variables $sVar$ and a set of signals S.
- E is a set of edges called events(or transitions). Every edge is defined by a 5-tuple:

  $$(m, a, grd_{mm'}, jmp_{mm'}, m')$$

  where $m, m' \in M, a \in A$, A is a finite set of symbols which is used to label the edges. $grd_{mm'}$ is the guard condition which specifies when the transition from m to m' is enabled. $jmp_{mm'}$ is a relation defined by a subset $X \times X$. The transition from mode m to $m'$is enabled when the condition satisfies $grd_{mm'}$, while the continuous state x jumps to a new value x' denoted by $(x, x') \in jmp_{mm'}$ during the transition.
- Inv is a mapping from the modes M to the subset of X, that is $Inv(m) \subset X$ for all $m \in M$. Whenever the system in mode m, the continuous states x must satisfy $x \in Inv(m)$. The subset $Inv(m)$ for $m \in M$ is called the invariant of mode m. The invariant specifies the global constraints to the variables. Whenever the invariant condition is violated, the system must exit the relevant mode.
- Act is called the activity of a mode which is the conjunction of several differential expressions. Each mode is assigned an Act. The activity of one mode specifies the changing of continuous variables depending on time within the mode.

3.3 Semantics

The execution of a spatio-temporal hybrid automaton results in continuous change and discrete change. The mixed discrete-continuous dynamic can be abstracted by a fully discrete transition system. In this paper, we formalize it as a labeled transition system.

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.
- $\rightarrow$: $States \times Labels \times States$ is a ternary relation over $States$ specifying the transitions between states.

State. In spatio-temporal hybrid automata, a state of a system can be formalized as a structure (m, v, i, ι) where

- $m \in M$ is a mode name. It specifies the current execution of system is in mode m.
- v is the current values of every variable of the system. Each variable in Var owns a value.
- i is a time instant which is the current reading of the default physical clock.
- $\iota = Loc_i(m)$ records the location of mode m with respect to the time instant i. For each reading of clock, function will record a system location.

Transition. A transition relation $\rightarrow$: $States \times Labels \times States$ is a relation from a source state s to a target state s' with specific transition. Generally, a concrete transition is written as $s \xrightarrow{\lambda} s'$ where λ is a label identifying this transition which contains three attributes:

1) a trigger event $evt(\lambda)$ specifies a significant occurrence in time or space.
2) a guard condition $grd(\lambda)$ specifies when the transition is enable.
3) an action $jmp(\lambda)$ which is performed when the transition occurs to change the values of variables.

A transition $s \xrightarrow{\lambda} s'$ can be triggered if a trigger event $evt(\lambda)$ is observed and the source state s satisfies the guard condition $grd(\lambda)$. While a transition is triggered, its action $jmp(\lambda)$ is performed, then the current system state transforms to the target state s'. We can use a transition rule to describe the transition $(m, v, i, \iota) \xrightarrow{\lambda} (m', v', i', \iota')$ as follows

$$\frac{v, i, \iota \models evt(\lambda) \wedge grd(\lambda)}{(m, v, i, \iota) \xrightarrow{\lambda} (m', v', i', \iota')} \quad jmp(\lambda)(v, i, \iota, v', i', \iota')$$

It states that a transition is a valid transition iff it holds the following conditions.

1) $v, i, \iota \models evt(\lambda) \wedge grd(\lambda)$
2) $\iota' = Loc_{i'}(m')$
3) $v, i, \iota, v', i', \iota' \models jmp(\lambda)$

Trace. For considering the infinite behavior (liveness) of a spatio-temporal automaton, we should pay close attention on infinite sequences of transitions.

Consider a labeled transition system S and a state s_0 of S. A s_0-rooted trajectory of S is a finite or infinite sequence of pairs $(a_i, s_i)_{i\geq 1}$ of labels $a_i \in Labels$ and states $s_i \in States$ thus $s_{i-1} \xrightarrow{a_i} s_i,\ i \geq 1$. A live transition system (S, L) is a pair, where S is a labeled transition system and L is a set of infinite initialized trajectories ($(a_i, s_i)_{i\geq 1}, s_0$ is an initial state of S) of S. If (S, L) is a live transition system, and $(a_i, s_i)_{i\geq 1}$ is either a finite initialized trajectory of S or a infinite initialized trajectory in L, such that the corresponding sequence $\langle a_i\rangle_{i\geq 1}$ is called a finite or infinite trace of a live transition system (S, L).

We associate with each transition of the label transition system S a duration in $\mathbb{R} \geq 0$. For trigger events $e \in evt$, the duration of $s \xrightarrow{e} s'$ is 0. For actions $j \in jmp$, the duration of $s \xrightarrow{j} s'$ is 0. For guard conditions with the clock constraint $grd := c.I_{s'} - c.I_s \geq \delta$, the duration of $s \xrightarrow{grd} s'$ is t. An infinite trajectory $\langle a_i, s_i\rangle_{i\geq 1}$ of the label transition system S diverges if the infinite sum $\sum_{i\geq 1} \delta_i$ diverges, where each δ_i is the duration of the corresponding transition $s_{i-1} \xrightarrow{a_i} s_i,\ i \geq 1$. Let L be be set of divergent initialized trajectories of the label transition system S. The spatio-temporal hybrid automaton H is nonzeno if L is *machine-closed* for S(The set L of infinite initialized trajectories is *machine-closed* for S if every finite initialized trajectory of S is a prefix of some trajectory in L.). Each trace of the live transition system (S, L) is called a timed trace of H.

Parallel Composition. Given two spatio-temporal hybrid automata we define a product automaton called the parallel composition. Conceptually, a run of the parallel composition is comprised of simultaneous runs of the component automata which are independent except that:

1) They must synchronize on shared events.
2) The only product states that are permitted are those for which the restrictions on conditions are jointly satisfiable [25].

We define the parallel composition$\mathcal{A}\|\mathcal{B}$ of the spatio-temporal hybrid automata $\mathcal{A}$ and $\mathcal{B}$.

$$\mathcal{A} = (M^{\mathcal{A}}, X^{\mathcal{A}}, A^{\mathcal{A}}, Var^{\mathcal{A}}, E^{\mathcal{A}}, Inv^{\mathcal{A}}, Act^{\mathcal{A}})$$

$$\mathcal{B} = (M^{\mathcal{B}}, X^{\mathcal{B}}, A^{\mathcal{B}}, Var^{\mathcal{B}}, E^{\mathcal{B}}, Inv^{\mathcal{B}}, Act^{\mathcal{B}})$$

- $M = M^{\mathcal{A}} \times M^{\mathcal{B}}$
- $X = X^{\mathcal{A}} \times X^{\mathcal{B}}$
- $A = \alpha_1 \cup \alpha_2 \cup \alpha_3$ where

1) α_1 is a subset of $A^{\mathcal{A}}$. Each element in α_1 is used to label the edges such as $(m_1, m_2) \to (m_1', m_2),\ m_1, m_1' \in M^{\mathcal{A}}, m_2 \in M^{\mathcal{B}}$.
   $\alpha_1 = \{a | m_1 \xrightarrow{a} m_1'\}$.
2) α_2 is a subset of $A^{\mathcal{B}}$. Each element in α_2 is used to label the edges such as $(m_1, m_2) \to (m_1, m_2'),\ m_1 \in M^{\mathcal{A}}, m_2, m_2' \in M^{\mathcal{B}}$.
   $\alpha_2 = \{a | m_2 \xrightarrow{a} m_2'\}$.
3) α_3 is a set of symbols to label the edges such as $(m_1, m_2) \to (m_1', m_2'),\ m_1, m_1' \in M^{\mathcal{A}}, m_2, m_2' \in M^{\mathcal{B}}$.
   $\alpha_3 = \{a * b | m_1 \xrightarrow{a} m_1', m_2 \xrightarrow{b} m_2'\}$ $*$ is for simple connection of symbols.

- $Var = Var^{\mathcal{A}} \cup Var^{\mathcal{B}}$
- E is a set of edges called events. Every edge is a 5-tuple:

$$(m, a, grd(a), jmp(a), m')$$

  where $a \in A$ is a symbol to label the edges. The guard condition $grd(a)$ is constructed as follows

$$grd(a) = \begin{cases} grd(a) & \text{if } a \in A^{\mathcal{A}} \\ grd(a) & \text{if } a \in A^{\mathcal{B}} \\ grd(b) \wedge grd(c) & \text{if } a = b * c, b \in A^{\mathcal{A}}, c \in A^{\mathcal{B}} \end{cases}$$

  The action $jmp(a)$ is constructed as follows

$$jmp(a) = \begin{cases} jmp(a) & \text{if } a \in A^{\mathcal{A}} \\ jmp(a) & \text{if } a \in A^{\mathcal{B}} \\ jmp(b) \wedge jmp(c) & \text{if } a = b * c, b \in A^{\mathcal{A}}, c \in A^{\mathcal{B}} \end{cases}$$

- $Inv = \{inv | inv = inv_{m_1} \wedge inv_{m_2},\ inv_{m_1} \in Inv^{\mathcal{A}},\ inv_{m_2} \in Inv^{\mathcal{B}}\}$, for the mode $m = (m_1, m_2)$, $m_1 \in M^{\mathcal{A}}$, $m_2 \in M^{\mathcal{B}}$.
- $Act = \{act | act_{m_1} \wedge act_{m_1}\}$ where act_{m_1} is for specifying activity of the mode m_1, act_{m_2} is for specifying activity of the mode m_2, act is for specifying activity of the mode m, $m = (m_1, m_2)$, $m_1 \in M^{\mathcal{A}}$, $m_2 \in M^{\mathcal{B}}$.

4 Case Study: Train Control System

Intelligent Transportation Systems are the 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, a communication based train control (CBTC) system [5] 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 a CBTC system is a 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 (ATO). 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 following.

4.1 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 emergency 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 $Emerge\ Distance$ 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.

4.2 Behavior of the System

Based on the syntax and semantics of spatio-temporal hybrid automata model, we can model the behavior of the Protection functions of ATP system. The most important components are Component SSU, Component DSU and Component BE. We use spatio-temporal hybrid automata to model the behavior of them as follows.

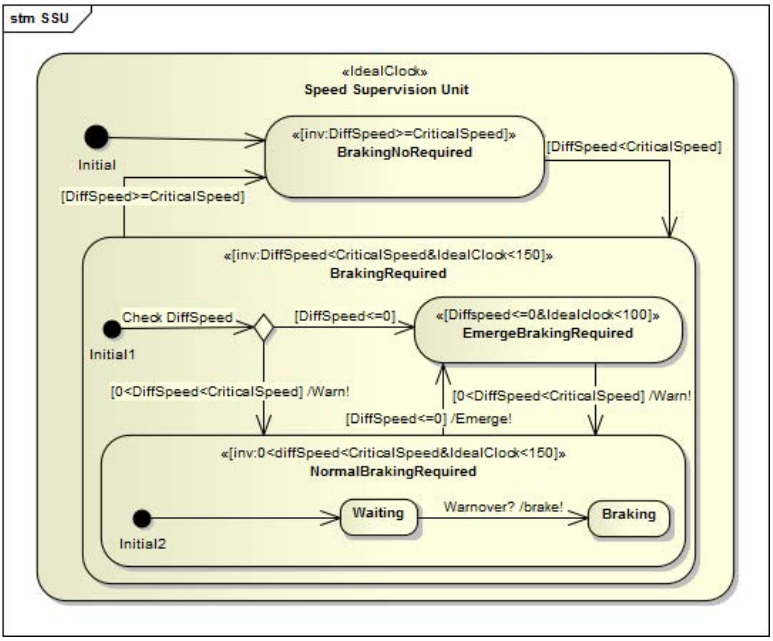

Fig. 2. Behavior of Speed Supervision Unit

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.

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

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

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

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

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

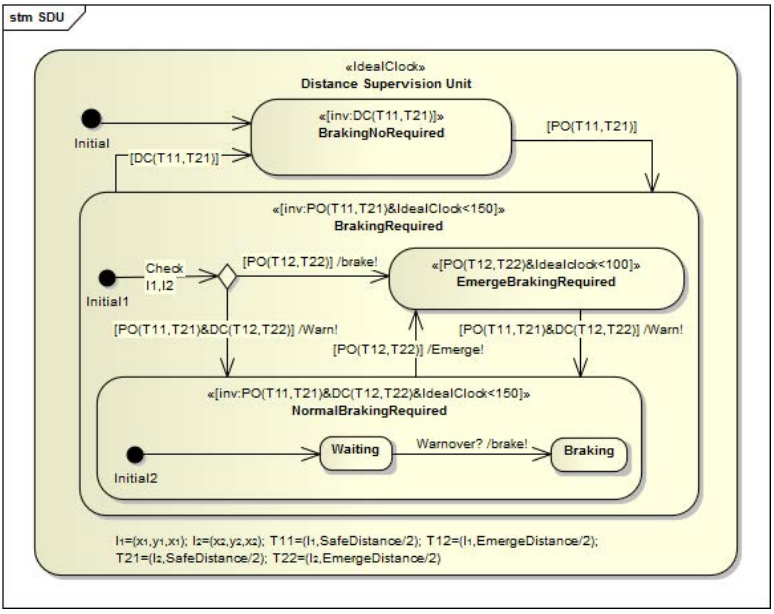

Fig. 3. Behavior of Distance Supervision Unit

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 spatio-temporal hybrid automata model.

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.

$evt(tr) \wedge grd(tr) = Emerge? \wedge TRUE,$

$jmp(tr) = IdealClock := 0,$

When $v, c, \iota \models grd(tr)$ and $(v, c, \iota, v, 0, \iota') \models jmp(tr)$,
$tr \triangleq (initial_1, v, 0, \iota) \rightarrow (EmergeBraking, v, 0, \iota')$
Edge tr is fired immediately after entering submode $Emergency$.

$$inv(EmergeBraking) = (IdealClock < 100) \wedge (v \geqslant 0)$$

$$act(EmergeBraking) = \begin{cases} -fW = \frac{W}{g} \cdot \frac{\mathrm{d}v}{\mathrm{d}t} \\ v \mid_{t=0} = v_0 \end{cases}.$$

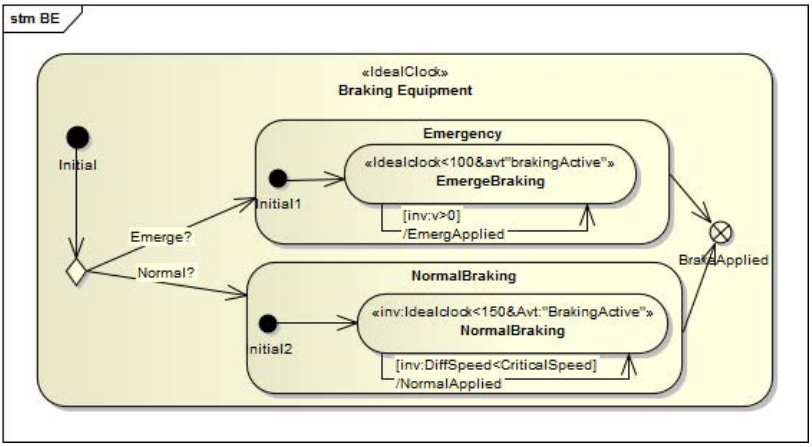

Fig. 4. Behavior of component Braking Equipment

If f is a solution of equation $act(EmergeBraking)$, f satisfies
$inv(EmergeBraking)[f/v] = TRUE \ \wedge$
$act(EmergeBraking)[f/v, \dot{f}/\dot{v}] = TRUE.$
In the submode $EmergeBraking$, the continuous variable v is changing when time passes, following the $act(EmergeBraking)$. 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.

5 Related Work

Hybrid automata are formal models for mixed discrete-continuous hybrid system which constitute the foundations of CPS [21]. It can be viewed as a generalization of time automata, in which the behavior of variables is governed in each state by a set of differential equations [22]. Compared to hybrid automata, spatio-temporal hybrid automata extend the expression on spatial terms and spatial relations. Since, CPS events not only include time and observe variables but also locations. We also classify the type of variables and expressions for clear description in spatio-temporal hybrid automata. The semantics of spatio-temporal hybrid automata is based on labeled transition system. We define the states and transitions of the labeled transition system, and discuss the trace semantics and parallel composition of spatio-temporal hybrid automata.

In [23], CPSs are modeled as co-inductive coroutined constraint logic programs, physical quantities are faithfully represented as continuous quantities (i.e., not discretized) and the constraints imposed on them by CPS physical interactions are modeled with constraint logic programming over reals. Therefore, CPSs are modeled as coroutined, co-inductive CLP(R) programs which can be used for verifying interesting properties of the system such as safety, utility, and

liveness. Compare to their work, our approach support a spatio-temporal logic to describe logical connections between Algebraic Expressions, Differential Expressions, Clock Expressions, Spatial Expressions and Action Expressions with time flows. The spatio-temporal logic is constructed by extends temporal logic with spatial characteristics based on spatial logic. Whatever, the logic can be used to describe the interesting properties of cyber-physic systems for verification.

In formal verification of software, transition systems serve as a formal model of systems, meanwhile temporal logic serves as a formal language for behavioral properties. PTL is one of the well known temporal logics and used to specify requirements of real-time systems [1, 9]. Spatial logic is a number of logics used for representation and reasoning space, e.g. 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 in [10, 11, 13]. Nevertheless, spatial logic can not express changes in time of the truth-values of purely spatial propositions. Spatio-temporal logic is used for describing the change of spatial proposition over time. There have been attempts to construct Spatio-temporal hybrids. For example, in [12], 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. The method can be looked upon as a guide to combine an arbitrary logic and PTL for our work. In [14], 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 [15]. The syntax of RCC-8 only contains eight binary predicates. Nor can RCC-8 represent complex relations between more than two regions. Therefore, we chose the spatial logic $S4_u$, which has the more expressive power as one of the basic logic in our work. Furthermore, those spatial-temporal logic only focus on the change of space over time. Our work concentrates on construct a spatio-temporal logic which can express both spatial proposition and normal proposition for CPSs.

6 Conclusion and Future Work

In Cyber-Physical System design and modeling, a big problem is how to capture time and location information into CPS models, together with specifying logistical properties or constrains. For modeling the relations between spatial terms and time, we have proposed a spatio-temporal logic based on PTL, $S4_u$ and the method to enrich a logic. The logic is used to express the logical connections between all kinds of expressions (including spatial expressions) in CPS. Thus, time, spatial terms and other type of variables can calculate together. We have constructed a spatio-temporal hybrid automaton for Cyber-Physical Systems, which is an extension of hybrid automata with spatial variables, spatial expressions based on spatio-temporal logic. Then, we give formal semantics (including states, transition, trace and parallel composition) of the automaton based on

labeled transition systems. Finally, a Train Control System is employed as a case study to show the usage of spatio-temporal hybrid automata.

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

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, Shanghai Knowledge Service Platform ZF 1213, NSFC Creative Team 61021004 and 863 Project 2011AA010101.

References

1. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems Specification. Springer (1992)
2. Gabelaia, D., Kontchakov, R., Kurucz, A., Wolter, F., Zakharyaschev, M.: Combining Spatial and Temporal Logics. Journal of Artificial Intelligence Research, 167–243 (2005)
3. Lee, E.A.: Cyber Physical Systems: Design Challenges. In: 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), pp. 363–369 (2008)
4. Shao, Z., Liu, J., Ding, Z., Chen, M., Jiang, N.: Spatio-Temporal Properties Analysis for Cyber-Physical Systems. In: Proceedings of the 18th International Conference on Engineering of Complex Computer Systems, ICECCS 2013 (2013)
5. IEEE, IEEE Recommended Practice for Communications-Based Train Control (CBTC) System Design and Functional Allocations. IEEE Std 1474.3-2008 (2008)
6. Fouquet, F., Morin, B., Fleurey, F., Barais, O., Plouzeau, N., Jezequel, J.: A dynamic component model for cyber physical systems. In: Proceedings of the 15th ACM SIGSOFT Symposium on Component Based Software Engineering, pp. 135–144 (2012)
7. Chomicki, J.: Temporal query language: a survey. In: Gabbay, D.M., Ohlbach, H.J. (eds.) ICTL 1994. LNCS, vol. 827, pp. 506–534. Springer, Heidelberg (1994)
8. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press (1995)
9. Manna, Z., Pnueli, A.: Temporal Verification of Reactive Systems: Safety. Springer (1995)
10. Stone, M.H.: Application of the theory of Boolean rings to general topology. Transactions of the AMS 41, 321–364 (1937)
11. Chen, T.: Algebraic postulates and a geometric interpretation of the Lewis calculus of strict implication. Bulletin of the AMS 44, 737–744 (1938)
12. Finger, M., Gabbay, D.M.: Adding a temporal dimension to a logic system. Journal of Logic, Language and Information 1(3), 203–233 (1992)
13. McKinsey, J.C.C.: A solution of the decision problem for the Lewis systems S2 and S4, with an application to topology. Journal of Symbolic Logic 6(4), 117–134 (1941)
14. Wolter, F., Zakharyaschev, M.: Spatio-temporal representation and reasoning based on RCC-8. In: Proceedings of the 7th Conference on Principles of Knowledge Representation and Reasoning (KR 2000), pp. 3–14 (2000)

15. Egenhofer, M.J., Herring, J.R.: Categorizing topological relationships between regions, lines and points in geographic databases. Tech. rep., University of Maine (1991)
16. Wolper, P.: Expressing interesting properties of programs in propositional temporal logic. In: Proceedings of the 13th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, pp. 184–192 (1986)
17. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. ACM Transactions on Programming Languages and Systems, 244–263 (1986)
18. Reynolds, M.: The complexity of the temporal logic with until over general linear time. Journal of Computer and System Sciences 66(2), 393–426 (2003)
19. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. Journal of the ACM 32(3), 733–749 (1985)
20. Alexander, C., Zakharyaschev, M.: Modal Logic. Oxford Logic Guides, vol. 35. Clarendon Press, Oxford (1997)
21. Henzinger, T.A.: The theory of hybrid automata. In: Logic in Computer Science, LICS 1996, pp. 278–292 (1996)
22. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: Grossman, R.L., Ravn, A.P., Rischel, H., Nerode, A. (eds.) HS 1993. LNCS, vol. 736, pp. 209–229. Springer, Heidelberg (1993)
23. Saeedloei, N., Gupta, G.: A logic-based modeling and verification of CPS. ACM SIGBED Review 8(2), 31–34 (2011)
24. Gupta, R.: Programming models and methods for spatiotemporal actions and reasoning in cyber-physical systems. In: NSF Workshop on CPS (2006)
25. Miller, J.S.: Decidability and Complexity Results for Timed Automata and Semilinear Hybrid Automata. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 296–310. Springer, Heidelberg (2000)