11-13-2013

A Modal Specification Theory for Timing Variability

Andrew King
University of Pennsylvania, kingand@cis.upenn.edu

Oleg Sokolsky
sokolsky@cis.upenn.edu

Insup Lee
University of Pennsylvania, lee@cis.upenn.edu

Recommended Citation

Andrew King, Oleg Sokolsky, and Insup Lee, "A Modal Specification Theory for Timing Variability", . November 2013.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-13-11

A Modal Specification Theory for Timing Variability

Abstract

Modal specifications are classical formalisms that can be used to express the functional variability of systems; it is particularly useful for capturing the stepwise refinement of component-based design. However, the extension of such formalisms to real-time systems has not received adequate attention. In this paper, we propose a novel notion of time-parametric modal specifications to describe the timing as well as functional variability of real-time systems.We present a specification theory on modal refinement, property preservation and compositional reasoning. We also develop zone-graph based symbolic methods for the reachability analysis and modal refinement checking. We demonstrate the practical application of our proposed theory and algorithms via a case study of medical device cyber-physical systems.

Disciplines

Computer Engineering | Computer Sciences

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-13-11

A Modal Specification Theory for Timing Variability [⋆]

Andrew L. King, Lu Feng [⋆⋆], Oleg Sokolsky, and Insup Lee

Department of Computer & Information Science, University of Pennsylvania
{kingand,lufeng,sokolsky,lee}@cis.upenn.edu

Abstract. Modal specifications are classical formalisms that can be used to express the *functional* variability of systems; it is particularly useful for capturing the stepwise refinement of component-based design. However, the extension of such formalisms to real-time systems has not received adequate attention. In this paper, we propose a novel notion of *time-parametric modal specifications* to describe the *timing* as well as functional variability of real-time systems. We present a specification theory on modal refinement, property preservation and compositional reasoning. We also develop zone-graph based symbolic methods for the reachability analysis and modal refinement checking. We demonstrate the practical application of our proposed theory and algorithms via a case study of medical device cyber-physical systems.

1 Introduction

Modal specifications introduced by Larsen and Thomsen in 1980s [2] allow the stepwise refinement design of component-based systems. The classical formalism is *modal transition systems* (MTSs), which are essentially labelled transition systems equipped with two types of transitions: *must* transitions which are required in all implementations, and *may* transitions which are allowed (but not required). Refining a MTS is an iterative process that chooses to implement or omit each of the may transitions, until a fully specified implementation is achieved. Thus, a MTS permits a set of different implementations, making it possible to express the *functional* variability of systems. Indeed, modal specifications have proven useful in many applications, for example, software product lines [16] where a specification defines a family of various products. More recently, we applied modal specifications to on-demand medical device systems [14], allowing the design of complex medical device interfaces and the verification of safety properties at the system level.

One limitation of our previous work is that the focus was on the time-abstract behavior of systems. However, safety critical systems such as medical devices typically have real-time elements, and often the safety and/or effectiveness of such systems depends on doing the right thing at the right time. For example, we might require that an infusion pump starts delivering drugs within t time units after it receives an *on* signal. This would be an example of a *timeliness* constraint; while we require that something

[⋆] Research is supported in part by NSF grants CNS-1035715 and IIS-1231547.

[⋆⋆] Lu Feng is supported by James S. McDonnell Foundation 21st Century Science Initiative - Postdoctoral Program in Complexity Science/Complex Systems - Fellowship Award.

happens by time t it would be ok if it happened earlier. Another example could be a watchdog timer that puts an actuator into a fail-safe mode if it hasn't received a recent control signal. In this situation we probably don't want the watchdog to activate prior to time t, because it might disrupt the normal function of the system.

There are a few extensions on timed modal specifications [9, 8, 7], but none of them considers the *timing* variability of systems. For example, suppose we want to design a medical system that satisfies the following requirement: "the infusion pump should start infusing no later than 3 time units, once it receives an *on* signal"; we may express this requirement with a clock constraint $x \leq 3$ in the specification, meaning that the clock x can progress within 3 time units. If we follow the theory of existing literature, then implementations must preserve the exact timing behavior of specification, *i.e.,* $x \leq 3$ for *all* implementations. Nevertheless, from the system designer's perspective, we think that an infusion pump requiring $x \leq 2$ is also a good implementation, since it does not violate the deadline. To capture this kind of timing variability, we propose the notion of *time-parametric modal specifications* which handles the above example by using a clock constraint $x \leq \alpha$, where α is a parameter satisfying $\alpha \leq 3$ and $\alpha \in \mathbb{N}$.

In our specification theory, an implementation is modeled as a timed I/O automaton (TIOA). Time-parametric modal specifications augment the notion of TIOAs, by distinguishing *must* and *may* transitions (for functional variability) and allowing parametric clock constraints (for timing variability). The relation between a specification and an implementation is captured by *modal refinement*. We prove that the satisfaction of *safety* and *liveness* properties on specifications is preserved on its implementations. To reason about the behavior of systems with multiple components, we also define the notion of *composition* and prove the compositionality of modal refinement. We develop a symbolic reachability analysis algorithm for time-parametric modal specifications, based on *parametric zone-graphs*. We also present a decision algorithm for the specification-implementation relation. Finally, we illustrate the applicability of our theory through a case study of medical device plug and play systems.

To the best of our knowledge, this is the first modal specification theory that can handle both *timing* and *functional* variability of real-time systems. In summary, there are three main contributions of this paper:

(1) We propose a novel notion of time-parametric modal specifications, which can be used to describe the timing and functional variability of real-time systems; based on this notion, we develop a specification theory on modal refinement, property preservation and compositional reasoning.
(2) We present a new symbolic semantical model as parametric zone-graphs, and develop two symbolic algorithms for the reachability analysis and modal refinement checking.
(3) We demonstrate the practical application of our theory and algorithms with a case study of medical device systems.

The rest of the paper is organized as follows: we describe contributions (1), (2) and (3) in Sections 2, 3 and 4, respectively; we then make a conclusion and point out directions for future work in Section 5.

Related work. The seminal paper on *modal* transition systems (MTSs) was by Larsen and Thomsen [17]. Many extensions (*e.g.,* modal specifications with data, MTSs with structured labels and disjunctive MTSs) have been developed over the years, for which an overview could be found in [2]. Note that the so-called *parametric* MTS [4], which uses boolean-valued parameters for expressing the functional variability of systems, should be distinguished from our notion where integer-valued parameters are used to capture timing variability.

There are a few extensions on timed modal specifications: a formalism in the CCS style was proposed in [9], but no logical characterization or model satisfaction relation is defined; Bertrand *et al.* [7] developed a complete specification theory for event-clock automata, a subclass of timed automata. However, none of these work considers the timing variability. There is also a formalism called MTS with durations [5], where controllable or uncontrollable intervals are used to express the duration of time; but this notion is not compositional, thus not suitable for component-based design.

Our work is also closely related to the timed automata theory [1, 12, 6]. The classical notion of *parametric timed automata* [13] is similar to our notion of time-parametric modal specifications in the sense that it also allows (real-valued) parameters in clock constraints, but its lack of the *modal* ingredient prevents the modeling of functional variability. Timed I/O automata are used as specifications in the theory of [10], but this work cannot handle timing variability.

2 Modal Specifications for Timing Variability

In this section, we propose a new notion of *time-parametric modal specifications* which can be used to describe the variability of timing and functional behavior in corresponding implementations. We first define *parametric clock constraints* in Section 2.1, and present the *syntax* and *operational semantics* of time-parametric modal specifications in Section 2.2. We develop the specification theory on *modal refinement*, *property preservation* and *compositional reasoning* in Sections 2.3, 2.4 and 2.5, respectively.

2.1 Clocks

As in the classical theory of timed automata [1, 12], we use a finite set of real-valued clock variables, called *clocks* for short, to describe the progress of time. Given a finite set of clocks Clk, we refer to a function $v : \mathit{Clk} \to \mathbb{R}_{\geq 0}$ as a *clock valuation*. Given $d \in \mathbb{R}_{\geq 0}$, let $v + d$ denote the clock valuation that assigns all clocks $x \in \mathit{Clk}$ to $v(x) + d$. The set of all clock valuations is denoted by $\mathbb{R}_{\geq 0}^{\mathit{Clk}}$. We use **0** to denote the clock valuation that assigns 0 to all clocks in Clk. Clocks can be reset to zero: for $r \subseteq \mathit{Clk}$, let $v[r \mapsto 0]$ denotes the clock valuation that resets all clocks in r to 0 and keeps the value $v(x)$ for clocks $x \in \mathit{Clk} \setminus r$. A *clock constraint* is a conjunctive formula of atomic predicates $x \sim c$, where $x \in \mathit{Clk}$ is a clock, $\sim \in \{\leq, <, =, >, \geq\}$ is an equality/inequality relation operator and $c \in \mathbb{N}$ is a constant. A clock valuation v satisfies a clock constraint g, denoted by $v \models g$, iff the proposition formula of g resolves to `true` when substituting all occurrence of clock x with value $v(x)$.

To reason about timing variability, we extend the notion of clock constraints to *parametric clock constraints*, which are conjunctive formulae of predicates $x \sim (c \pm \alpha)$ where α is a non-negative integer parameter bounded by a set of linear constraints C. Let Θ be a set of parameters, we call a function $f : \Theta \to \mathbb{N}$ a *parameter assignment*. A parameter value $f(\alpha)$ is *valid* if it satisfies the linear constraints C and $c \pm f(\alpha) \in \mathbb{N}$. Let g be a parametric clock constraint; by assigning a set of valid values f to its parameters Θ, we obtain an *instance* of g as a clock constraint, denoted by $g[f(\Theta)]$. For example, a parametric clock constraint $x \leq 1 + \alpha$ bounded by $1 \leq \alpha \leq 3, \alpha \in \mathbb{N}$ has three instances: $x \leq 2$, $x \leq 3$ and $x \leq 4$. A clock valuation v satisfies a parametric clock constraint g, denoted by $v \models g$, iff v satisfies *all* instances of g; and we say that v *partially* satisfies g, denoted by $v \models^{\mathsf{p}} g$, iff v satisfies *some* instances of g. Let g_i for $i = 1, 2$ be two parametric clock constraints, each of which is bounded by a set of linear constraints C_i over parameters Θ_i; their conjunction $g = g_1 \wedge g_2$ is then bounded by the linear constraints $C = C_1 \wedge C_2$ over parameters $\Theta = \Theta_1 \cup \Theta_2$.

For the rest of the paper, we use $\mathcal{B}(\mathit{Clk})$ (*resp.* $\mathcal{P}(\mathit{Clk})$) to denote the set of clock constraints (*resp.* parametric clock constraints) over clocks Clk. We have $\mathcal{B}(\mathit{Clk}) \subseteq \mathcal{P}(\mathit{Clk})$, since a clock constraint can be considered as a special case of parametric clock constraints whose instance is itself.

2.2 Syntax and Operational Semantics

Now we propose the notion of *time-parametric modal specifications*, sometimes called *specifications* for short in this paper, to capture the timing and functional variability of different *implementaions* modeled as *timed I/O automata*. A timed I/O automaton is a timed automaton [6] that distinguishes *input*, *output* and *internal* actions. Formally,

Definition 1 (Timed I/O Automaton). *A* timed I/O automaton *(TIOA) $\mathcal{A}$ is a tuple $(\mathit{Loc}, \bar{l}, \mathit{Clk}, \mathit{Act}, \mathit{Inv}, \hookrightarrow)$ where*

- *Loc is a finite set of* locations*, and $\bar{l} \in \mathit{Loc}$ is an* initial location
- *Clk is a finite set of* clocks
- *$\mathit{Act} = \mathit{Act}^I \uplus \mathit{Act}^O \uplus \mathit{Act}^\tau$ is a finite set of* actions *partitioned into* input *Act^I,* output *Act^O and* internal *Act^τ actions*
- *$\mathit{Inv} : \mathit{Loc} \to \mathcal{B}(\mathit{Clk})$ assigns* invariants *to locations* [1]
- *$\hookrightarrow \subseteq \mathit{Loc} \times \mathcal{B}(\mathit{Clk}) \times \mathit{Act} \times 2^{\mathit{Clk}} \times \mathit{Loc}$ is the* transition relation

A transition $l \overset{g,a,r}{\hookrightarrow} l'$ is enabled at location l when guard $g \in \mathcal{B}(\mathit{Clk})$ holds and action $a \in \mathit{Act}$ occurs; any clock in $r \subseteq \mathit{Clk}$ will be reset to 0 once the transition moves to location l'.

Syntactically, a time-parametric modal specification looks similar to a TIOA; it distinguishes *must* and *may* transition relations for functional variability, and allows parametric clock constraints for timing variability.

Definition 2 (Time-Parametric Modal Specification). *A* time-parametric modal specification *$\mathcal{M}$ is a tuple $(\mathit{Loc}, \bar{l}, \mathit{Clk}, \mathit{Act}, \mathit{Inv}, \hookrightarrow_\Box, \hookrightarrow_\Diamond, C(\Theta))$ where*

[1] We restrict invariants to be downwards closed as in the UPPAAL verification tool [3].

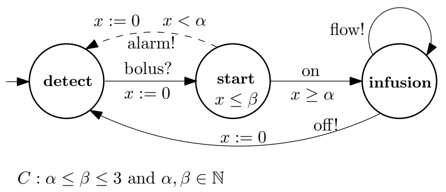


Fig. 1: An example of time-parametric modal specification.

- *$Loc, \bar{l}$, Clk and Act are the same as in Definition 1*
- *$Inv : Loc \rightarrow \mathcal{P}(Clk)$ assigns downwards closed parametric clock constraints in the form of $x \leq c \pm \alpha$ or $x < c \pm \alpha$ to locations*
- *$\hookrightarrow_\Box \subseteq Loc \times \mathcal{P}(Clk) \times Act \times 2^{Clk} \times Loc$ is the* must transition relation *describing required behavior*
- *$\hookrightarrow_\Diamond \subseteq Loc \times \mathcal{P}(Clk) \times Act \times 2^{Clk} \times Loc$ is the* may transition relation *describing allowed behavior*
- *$C(\Theta)$ is a set of linear constraints on a finite set of non-negative integer parameters Θ that are used in $\mathcal{P}(Clk)$*

We consider only *syntactically consistent* specifications where $\hookrightarrow_\Box \subseteq \hookrightarrow_\Diamond$, *i.e.,* a required transition should also be allowed. Definition 2 coincides with Definition 1 if $\hookrightarrow_\Box = \hookrightarrow_\Diamond = \hookrightarrow$, $\mathcal{P}(Clk) = \mathcal{B}(Clk)$ and $\Theta = \emptyset$.

A time-parametric modal specification abstracts the behavior of various TIOA implementations, which are required to contain all transitions from the set $\hookrightarrow_\Box$ and allowed to optionally include transitions from the set $\hookrightarrow_\Diamond$ (*i.e.,* functional variability); transitions may occur in different timing intervals, depending on the values of specification parameters (*i.e.,* timing variability).

Example 1. Fig. 1 shows a time-parametric modal specification for an infusion pump, which has 3 locations: **detect**, **start** and **infusion**. The initial location **detect** is indicated with an incoming arrow. There is only one clock variable x. The actions include input ("bolus?"), output ("alarm!", "flow!", "off!") and internal ("on"). There are two parameters $\Theta = \{\alpha, \beta\}$, bounded by the linear constraints $\alpha \leq \beta \leq 3$ and $\alpha, \beta \in \mathbb{N}$. The invariants on **detect** and **infusion** are both true (omitted in the figure), while the invariant on **start** is a parametric clock constraint $x \leq \beta$. Must (*resp.* may) transitions are indicated by solid (*resp.* dashed) lines in the figure. For example, once a pump detects a "bolus?" request, it *must* move to the **start** location; then, when the guard $x < \alpha$ is true, the pump *may* issue an "alarm!", reset clock x and move back to location **detect**.

We follow the classical interpretation that defines the operational semantics of timed automata as *timed transition systems* [6].

Definition 3 (TIOA's Operational Semantics). *The operational semantics of a timed I/O automaton $\mathcal{A} = (Loc, \bar{l}, Clk, Act, Inv, \hookrightarrow)$, denoted by $[\![\mathcal{A}]\!]$, is a* timed transition system (TTS) *represented as a tuple $(S, \bar{s}, \Sigma, \rightarrow)$ where*

- *$S = \{\langle l, v\rangle \in Loc \times \mathbb{R}_{\geq 0}^{Clk} \mid v \models Inv(l)\}$ is an infinite set of states*
- *$\bar{s} = \langle \bar{l}, \mathbf{0}\rangle$ is an initial state*

- $\Sigma = Act \cup \mathbb{R}_{\geq 0}$ *is the alphabet*
- $\rightarrow \subseteq S \times \Sigma \times S$ *is the transition relation defined by the following two rules:*
  - action transition*:* $\langle l, v\rangle \xrightarrow{a} \langle l', v'\rangle$ *for* $a \in Act$ *if there is transition* $l \overset{g,a,r}{\hookrightarrow} l'$ *in* $\mathcal{A}$ *such that* $v \models g$*,* $v' \models Inv(l')$ *and* $v' = v[r \mapsto 0]$
  - delay transition*:* $\langle l, v\rangle \xrightarrow{d} \langle l, v + d\rangle$ *for* $d \in \mathbb{R}_{\geq 0}$ *if* $v + d \models Inv(l)$

To define the operational semantics of time-parametric modal specifications, we first extend the notion of TTSs to *modal timed transition systems*, by partitioning the transition relation into *must action*, *may action* and *delay* transitions.

Definition 4 (Modal Timed Transition System). *A* modal timed transition system *(MTTS)* M *is a tuple* $(S, \overline{s}, Act, \rightarrow_{\Box}, \rightarrow_{\Diamond}, \rightarrow_{\mathsf{d}})$ *where*

- S *is an (infinite) set of states,* $\overline{s} \in S$ *is an initial state, and* Act *is a set of actions*
- $\rightarrow_{\Box} \subseteq S \times Act \times S$ *is the* must action *transition relation,* $\rightarrow_{\Diamond} \subseteq S \times Act \times S$ *is the* may action *transition relation, and* $\rightarrow_{\mathsf{d}} \subseteq S \times \mathbb{R}_{\geq 0} \times S$ *is the* delay *transition relation*

Definition 5 (Specification's Operational Semantics). *For a time-parametric modal specification* $\mathcal{M} = (Loc, \bar{l}, Clk, Act, Inv, \hookrightarrow_{\Box}, \hookrightarrow_{\Diamond}, C(\Theta))$*, its operational semantics, denoted by* $[\![\mathcal{M}]\!]$*, yields a finite set of MTTSs such that there is a one-to-one mapping between a valid parameter assignment* $f(\Theta)$ *and a MTTS* $M = (S, \overline{s}, Act, \rightarrow_{\Box}, \rightarrow_{\Diamond}, \rightarrow_{\mathsf{d}})$ *where*

- $S = \{\langle l, v\rangle \in Loc \times \mathbb{R}_{\geq 0}^{Clk} \mid v \models Inv(l)[f(\Theta)]\}$*, and* $\overline{s} = (\bar{l}, \mathbf{0}) \in S$
- $\langle l, v\rangle \xrightarrow{a}_{\Box} \langle l', v'\rangle$ *(*resp. $\langle l, v\rangle \xrightarrow{a}_{\Diamond} \langle l', v'\rangle$*) if there is a must (*resp. *may) action transitions* $l \overset{g,a,r}{\hookrightarrow}_{\Box} l'$ *(*resp. $l \overset{g,a,r}{\hookrightarrow}_{\Diamond} l'$*) in* $\mathcal{M}$ *such that* $v \models g[f(\Theta)]$*,* $v' = v[r \mapsto 0]$ *and* $v' \models Inv(l')[f(\Theta)]$
- $\langle l, v\rangle \xrightarrow{d}_{\mathsf{d}} \langle l, v + d\rangle$ *for* $d \in \mathbb{R}_{\geq 0}$ *if* $v + d \models Inv(l)[f(\Theta)]$

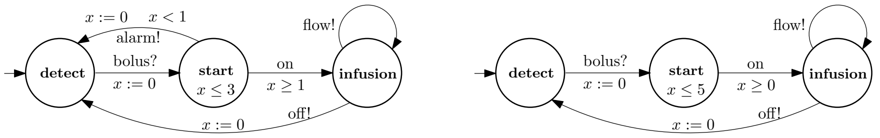
Note that $Inv(l)[f(\Theta)]$ and $g[f(\Theta)]$ are clock constraints obtained by substituting the occurrences of parameters Θ with values $f(\Theta)$. Computing the set of valid parameter assignments $f(\Theta)$ reduces to solving the linear constraints $C(\Theta)$ of $\mathcal{M}$, which can be efficiently solved by using a SMT solver such as Z3 [11].

2.3 Modal Refinement

We check whether a TIOA is an implementation of a time-parametric modal specification by checking the *modal refinement* relation.

Definition 6 (Modal Refinement of MTTSs). *Let* $M_i = (S_i, \overline{s}_i, Act, \rightarrow_{\Box,i}, \rightarrow_{\Diamond,i}, \rightarrow_{\mathsf{d},i})$ *for* $i = 1, 2$ *be two MTTSs. We say that* M_1 modally refines M_2*, denoted by* $M_1 \preceq M_2$*, iff there exists a binary relation* $R \subseteq S_1 \times S_2$ *containing* $(\overline{s}_1, \overline{s}_2)$ *such that for each* $(s, t) \in R$ *we have*

- *for all* $(t, a, t') \in \rightarrow_{\Box,2}$ *there is some* $(s, a, s') \in \rightarrow_{\Box,1}$ *with* $(s', t') \in R$
- *for all* $(s, a, s') \in \rightarrow_{\Diamond,1}$ *there is some* $(t, a, t') \in \rightarrow_{\Diamond,2}$ *with* $(s', t') \in R$



(a) **(b)**

Fig. 2: Two example TIOAs.

- *for all* $(s, d, s') \in \rightarrow_{\mathsf{d},1}$ *there is some* $(t, d, t') \in \rightarrow_{\mathsf{d},2}$ *with* $(s', t') \in R$*, and for all* $(t, d, t') \in \rightarrow_{\mathsf{d},2}$ *there is some* $(s, d, s') \in \rightarrow_{\mathsf{d},1}$ *with* $(s', t') \in R$

Since a TTS (Definition 3) can be considered as a special case of MTTSs where $\rightarrow_{\Box} = \rightarrow_{\Diamond}$, the above definition is also applicable for $A \preceq M$ where A is a TTS and M is a MTTS.

Definition 7 (Implementation). *Let $\mathcal{A}$ be a TIOA and $\mathcal{M}$ be a time-parametric modal specification over the same actions. We say that $\mathcal{A}$ is an* implementation *of $\mathcal{M}$, denoted by $\mathcal{A} \sqsubseteq \mathcal{M}$, if there exist a MTTS $M \in [\![\mathcal{M}]\!]$ such that $[\![\mathcal{A}]\!] \preceq M$.*

A time-parametric modal specification $\mathcal{M}$ may admit a set of implementations with timing and functional variability. By fixing a parameter assignment for the specification, a MTTS $M \in [\![\mathcal{M}]\!]$ representing certain timing behavior is chosen; and by checking whether the operational semantics of $\mathcal{A}$ modally refines M, we compare their functional behavior. $\mathcal{A}$ is an implementation of $\mathcal{M}$ when both the timing and functional requirements are met. Later in Section 3.3, we present a symbolic method to check the specification-implementation relation.

Example 2. Consider the time-parametric modal specification shown in Fig. 1. The TIOA in Fig. 2a is an implementation of the specification, where the may transition labelled by "alarm!" is implemented and the parameters are fixed for $\alpha = 1, \beta = 3$. The TIOA in Fig. 2b is not a valid implementation, because the invariant in location **start** is $x \leq 5$, violating the specification's constraints of $x \leq \beta$ and $\beta \leq 3$.

2.4 Model Checking

Property preservation (*i.e.,* the satisfaction of certain property on a specification implies that all implementations also satisfy the property) is a crucial for component-based design. In this section, we prove the property preservation for time-parametric modal specifications. We consider two most commonly used types of properties: *safety* and *liveness*.

Let ϕ be state formulae about locations and time. A safety property claims "something bad will never happen"; formally, it can be written in temporal logic $\texttt{A[ ]}\phi$ (*i.e.,* invariantly ϕ), or expressed as the negation of reachability property $\neg\texttt{E}\langle\rangle\neg\phi$ (*i.e.,* never possibly $\neg\phi$). A liveness property says "something will eventually happen", expressed with path formulae $\texttt{A}\langle\rangle\phi$ (*i.e.,* always eventually ϕ). We refer the readers to [6] for the details of temporal logic.

Let $\mathcal{A}$ be a TIOA implementation and ψ be a safety or liveness property; we say that $\mathcal{A}$ satisfies ψ, denoted by $\mathcal{A} \models \psi$, iff its operational semantics $[\![\mathcal{A}]\!] \models \psi$. Let M

be a MTTS, we define M must (*resp.* may) satisfy ψ, denoted by $M \models_{\Box} \psi$ (*resp.* $M \models_{\Diamond} \psi$), iff the formula ψ is $\texttt{true}$ on paths that only contain must (*resp.* may) and delay transitions. $M \models_{\Box} \neg\phi$ iff $M \models_{\Diamond} \phi$ is false. We say that a time-parametric modal specification $\mathcal{M}$ *must* satisfy a property ψ, denoted by $\mathcal{M} \models_{\Box} \psi$, iff $M \models_{\Box} \psi$ for *all* MTTSs $M \in [\![\mathcal{M}]\!]$; and $\mathcal{M}$ *may* satisfy ψ, denoted by $\mathcal{M} \models_{\Diamond} \psi$, iff $M \models_{\Diamond} \psi$ for *some* $M \in [\![\mathcal{M}]\!]$.

Lemma 1. *Let A be a TTS and M be a MTTS such that $A \preceq M$. Let ψ be a safety or liveness property. Suppose $M \models_{\Box} \psi$, then $A \models \psi$.*

Proof. See the appendix.

Theorem 1. *Let $\mathcal{A}$ be a TIOA implementation of a time-parametric modal specification $\mathcal{M}$,* i.e., *$\mathcal{A} \sqsubseteq \mathcal{M}$. Let ψ be a safety or liveness property. Suppose $\mathcal{M} \models_{\Box} \psi$, then $\mathcal{A} \models \psi$.*

Proof. Since $\mathcal{A} \sqsubseteq \mathcal{M}$, based on Definition 7, there must exist a MTTS $M \in [\![\mathcal{M}]\!]$ such that $[\![\mathcal{A}]\!] \preceq M$. Given that $\mathcal{M} \models_{\Box} \psi$, we know that $M \models_{\Box} \psi$ for every $M \in [\![\mathcal{M}]\!]$. Thus, based on Lemma 1, we have $[\![\mathcal{A}]\!] \models \psi$, meaning that $\mathcal{A} \models \psi$.

2.5 Compositional Reasoning

We now introduce the notion of *composition*, which is important for component-based design, and prove the property preservation compositionally. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two time-parametric modal specifications. They are *composeable* iff they have disjoint sets of clocks and parameters, *i.e.,* $\mathit{Clk}_1 \cap \mathit{Clk}_2 = \emptyset$ and $\Theta_1 \cap \Theta_2 = \emptyset$, and their actions only overlap on complementary types: $(\mathit{Act}_1^I \cup \mathit{Act}_1^\tau) \cap (\mathit{Act}_2^I \cup \mathit{Act}_2^\tau) = \emptyset$ and $(\mathit{Act}_1^O \cup \mathit{Act}_1^\tau) \cap (\mathit{Act}_2^O \cup \mathit{Act}_2^\tau) = \emptyset$.

Definition 8 (Composition). *Given two* composeable *time-parametric modal specifications $\mathcal{M}_i = (\mathit{Loc}_i, \bar{l}_i, \mathit{Clk}_i, \mathit{Act}_i, \mathit{Inv}_i, \hookrightarrow_{\Box,i}, \hookrightarrow_{\Diamond,i}, C_i(\Theta_i))$ for $i = 1, 2$. Their* composition *product, denoted by $\mathcal{M}_1 \| \mathcal{M}_2$, yields a specification $(\mathit{Loc}_1 \times \mathit{Loc}_2, (\bar{l}_1, \bar{l}_2), \mathit{Clk}_1 \cup \mathit{Clk}_2, \mathit{Act}, \mathit{Inv}, \hookrightarrow_{\Box}, \hookrightarrow_{\Diamond}, C(\Theta))$ such that*

- *$\mathit{Act} = \mathit{Act}^I \uplus \mathit{Act}^O \uplus \mathit{Act}^\tau$ where $\mathit{Act}^I = (\mathit{Act}_1^I \setminus \mathit{Act}_2^O) \cup (\mathit{Act}_2^I \setminus \mathit{Act}_1^O)$, $\mathit{Act}^O = (\mathit{Act}_1^O \setminus \mathit{Act}_2^I) \cup (\mathit{Act}_2^O \setminus \mathit{Act}_1^I)$, and $\mathit{Act}^\tau = \mathit{Act}_1^\tau \cup \mathit{Act}_2^\tau \cup (\mathit{Act}_1^I \cap \mathit{Act}_2^O) \cup (\mathit{Act}_1^O \cap \mathit{Act}_2^I)$*
- *$\mathit{Inv}(l_1, l_2) = \mathit{Inv}_1(l_1) \wedge \mathit{Inv}_2(l_2)$*
- *$\hookrightarrow_{\Box}$ and $\hookrightarrow_{\Diamond}$ are defined by the following rules (interchangeable for $\mathcal{M}_1$ and $\mathcal{M}_2$):*

$$\frac{(l_1, g_1, a!, r_1, l_1') \in \hookrightarrow_{\gamma,1} \quad (l_2, g_2, a?, r_2, l_2') \in \hookrightarrow_{\gamma,2}}{\big((l_1, l_2), g_1 \wedge g_2, a, r_1 \cup r_2, (l_1', l_2')\big) \in \hookrightarrow_{\gamma}} \ \text{(synchronizing)}$$

$$\frac{(l_1, g_1, a, r_1, l_1') \in \hookrightarrow_{\gamma,1} \quad a \notin \mathit{Act}_2}{\big((l_1, l_2), g_1, a, r_1, (l_1', l_2)\big) \in \hookrightarrow_{\gamma}} \ \text{(interleaving)}$$

where $\gamma \in \{\Box, \Diamond\}$: if $\hookrightarrow_{\gamma,1} = \hookrightarrow_{\Box,1}$ and $\hookrightarrow_{\gamma,2} = \hookrightarrow_{\Box,2}$ in the synchronizing rule, or $\hookrightarrow_{\gamma,1} = \hookrightarrow_{\Box,1}$ in the interleaving rule, then $\hookrightarrow_{\gamma} = \hookrightarrow_{\Box}$; otherwise, $\hookrightarrow_{\gamma} = \hookrightarrow_{\Diamond}$.

- $\Theta = \Theta_1 \cup \Theta_2$ *and* $C(\Theta) = C_1(\Theta_1) \wedge C_2(\Theta_2)$.

Since a TIOA can be considered as a special case of time-parametric modal specifications, the above definition is also applicable for the composition of two TIOAs (*resp.* a TIOA and a specification), which yields a product TIOA (*resp.* specification).

Lemma 2. *Let $\mathcal{A}$ be a TIOA implementation of a specification $\mathcal{M}$,* i.e., $\mathcal{A} \sqsubseteq \mathcal{M}$. *Let $\mathcal{A}'$ be a composeable TIOA with $\mathcal{A}$. Then $\mathcal{A} \| \mathcal{A}' \sqsubseteq \mathcal{M} \| \mathcal{A}'$.*

Proof. See the appendix.

Theorem 2. *Let $\mathcal{A}$ be a TIOA implementation of a specification $\mathcal{M}$, and $\mathcal{A}'$ be a composeable TIOA with $\mathcal{A}$. Let ψ be a safety or liveness property. Suppose $\mathcal{M} \| \mathcal{A}' \models_{\Box} \psi$, then $\mathcal{A} \| \mathcal{A}' \models \psi$.*

Proof. It's straightforward from Theorem 1 and Lemma 2.

3 Symbolic Semantics and Verification

Recall that the operational semantics of a time-parametric modal specification consists of a (finite) set of MTTSs, each of which contains an infinite set of states. It's not feasible to verify such infinite state-space models directly. To tackle this challenge, we define a new symbolic semantics as *parametric zone-graphs* in Section 3.1, and propose practical methods for symbolic reachability analysis and modal refinement checking in Sections 3.2 and 3.3, respectively.

3.1 Symbolic Semantics: Parametric Zone-Graphs

Firstly, we extend the classical notion of *zones* and *zone-graphs* for timed automata [6] to parametric zones and zone-graphs. A *parametric zone* is in the form of a parametric clock constraint (defined in Section 2.1), representing the maximal set of clock valuations satisfying any *instance* of the parametric clock constraint. Let v be a clock valuation and D be a parametric zone, we define $v \in D$ iff v partially satisfies the parametric clock constraint of D, *i.e.,* $v \models^{\mathsf{p}} D$. Given two parametric zones D_1 and D_2, if $v \in D_1$ implies $v \in D_2$, then zone D_1 is included in D_2, denoted by $D_1 \subseteq D_2$. We define $D^{\uparrow} = \{v + d \mid v \in D, d \in \mathbb{R}_{\geq 0}\}$ for the zone progression, and $r(D) = \{v[r \mapsto 0] \mid v \in D\}$ for the clock reset of zones. A *parametric zone-graph* is a graph where each node consists of a location and a parametric zone. We define the symbolic semantics of time-parametric modal specifications based on parametric zone-graph as follows.

Definition 9 (Specification's Symbolic Semantics). *The symbolic semantics of a time-parametric modal specification $\mathcal{M} = (Loc, \bar{l}, Clk, Act, Inv, \hookrightarrow_{\Box}, \hookrightarrow_{\Diamond}, C(\Theta))$, denoted by $[\![\mathcal{M}]\!]_{\mathsf{z}}$, is a* parametric zone-graph $(S, \overline{s}, Act, \rightsquigarrow_{\Box}, \rightsquigarrow_{\Diamond}, \rightsquigarrow_{\mathsf{d}}, C(\Theta))$ *where*

- $S = \{\langle l, D\rangle \in Loc \times \mathcal{P}(Clk) \mid D \subseteq Inv(l)\}$ *is a finite set of* symbolic states*, and* $\overline{s} = \langle \bar{l}, D_{\mathbf{0}}\rangle$ *is the* initial state
- *symbolic* must action *transition:* $\langle l, D\rangle \overset{a}{\rightsquigarrow}_{\Box} \langle l', r(D \wedge g) \wedge Inv(l')\rangle$ *if there is a must transition* $l \overset{g,a,r}{\hookrightarrow}_{\Box} l'$ *in* $\mathcal{M}$

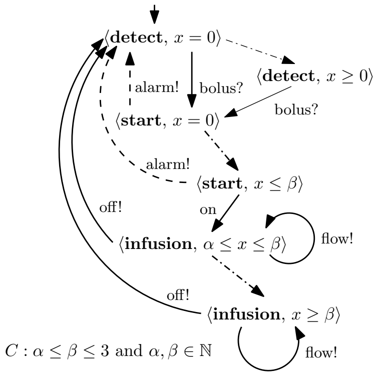


Fig. 3: Parametric zone-graph of the specification in Fig. 1.

- *symbolic* may action *transition:* $\langle l, D\rangle \overset{a}{\rightsquigarrow}_{\Diamond} \langle l', r(D \wedge g) \wedge \mathit{Inv}(l')\rangle$ *if there is a may transition* $l \overset{g,a,r}{\hookrightarrow}_{\Diamond} l'$ *in* $\mathcal{M}$
- *symbolic* delay *transition:* $\langle l, D\rangle \rightsquigarrow_{\mathsf{d}} \langle l, D^{\uparrow} \wedge \mathit{Inv}(l)\rangle$

A symbolic state $\langle l, D\rangle$ in $[\![\mathcal{M}]\!]_{\mathsf{z}}$ corresponds to a set of states in the operational semantics of $\mathcal{M}$. A symbolic transition $\langle l, D\rangle \rightsquigarrow_{\gamma} \langle l', D'\rangle$ with $\gamma \in \{\Box, \Diamond, \mathsf{d}\}$ implies that, for every $v' \in D'$, there must exist at least one MTTS $M \in [\![\mathcal{M}]\!]$ which has a transition $\langle l, v\rangle \rightarrow_{\gamma} \langle l', v'\rangle$ for some $v \in D$. Indeed, we show that the symbolic semantics given in Definition 9 is a correct and full characterization of the operation semantics given in Definition 5 as follows.

Theorem 3. *Let* $\mathcal{M}$ *be a time-parametric modal specification,* $[\![\mathcal{M}]\!]_{\mathsf{z}}$ *be its symbolic semantics and* $[\![\mathcal{M}]\!]$ *be its operational semantics.*

- (Soundness) *if the initial symbolic state* $\langle \bar{l}, D_{\mathbf{0}}\rangle$ *in* $[\![\mathcal{M}]\!]_{\mathsf{z}}$ *must (*resp. *may) lead to a target state* $\langle l_f, D_f\rangle$*, then for all* $v_f \in D_f$*, state* $\langle l_f, v_f\rangle$ *must (*resp. *may) be reachable from the initial state* $\langle \bar{l}, \mathbf{0}\rangle$ *in some* $M \in [\![\mathcal{M}]\!]$
- (Completeness) *if, in any* $M \in [\![\mathcal{M}]\!]$*, a target state* $\langle l_f, v_f\rangle$ *must (*resp. *may) be reachable from the initial state* $\langle \bar{l}, \mathbf{0}\rangle$*, then state* $\langle \bar{l}, D_{\mathbf{0}}\rangle$ *in* $[\![\mathcal{M}]\!]_{\mathsf{z}}$ *must (*resp. *may) lead to* $\langle l_f, D_f\rangle$ *for some* D_f *such that* $v_f \in D_f$

Proof. See the appendix.

The clock valuations in a (parametric) zone-graph may drift unboundedly, inducing infinite symbolic transition relations and thus an infinite zone-graph. We can apply techniques such as the *k-normalization* [6] to normalize zones and guarantee the finiteness of transition relations. Note that the clock ceiling k of a parametric clock constraint $x \sim c \pm \alpha$ is given by the maximum clock constant c plus the maximum value of parameter α.

Example 3. Fig. 3 illustrates the parametric zone-graph induced from the time-parametric modal specification shown in Fig. 1. There are 6 symbolic states in the zone-graph; for example, $\langle$**infusion**$, \alpha \le x \le \beta\rangle$ is a state associated with a location **infusion** and a

Algorithm 1 Must (*resp.* may) reachability analysis on parametric zone-graphs

```
Input: a parametric zone-graph $G$ whose initial state is $\langle \bar{l}, D_0 \rangle$, a target $\langle l_f, \phi_f \rangle$
Output: "YES", if $\langle l_f, \phi_f \rangle$ is reachable from $\langle \bar{l}, D_0 \rangle$; "NO", otherwise
 1: PASSED=$\emptyset$, WAIT=$\{\langle \bar{l}, D_0 \rangle\}$
 2: while WAIT$\neq \emptyset$ do
 3:    take $\langle l, D \rangle$ from WAIT
 4:    if $D = \emptyset$ for any valid parameter assignments $f(\Theta)$ then
 5:       prune state $\langle l, D \rangle$ and its incoming/outgoing transitions from $G$
 6:    else
 7:       if $l = l_f$ and $D \cap \phi_f \neq \emptyset$ for all (resp. some) valid $f(\Theta)$ then
 8:          return "YES"
 9:       end if
10:       if $D \not\subseteq D'$ for all $\langle l, D' \rangle \in$ PASSED then
11:          add $\langle l, D \rangle$ to PASSED
12:          for all $\langle l', D' \rangle$ such that $(\langle l, D \rangle, \langle l', D' \rangle) \in \rightsquigarrow_{\mathsf{d}}$ and $\rightsquigarrow_{\Box}$ (resp. $\rightsquigarrow_{\Diamond}$) do
13:             add $\langle l', D' \rangle$ to WAIT
14:          end for
15:       end if
16:    end if
17: end while
18: return "NO"
```

parametric zone $\alpha \leq x \leq \beta$, which is bounded by the linear constraints: $\alpha \leq \beta \leq 3$ and $\alpha, \beta \in \mathbb{N}$.

In Fig. 3, *solid* lines represent symbolic must action transitions, *e.g.,* there is a must transition labelled with action "bolus?" from state $\langle$**detect**$, x = 0\rangle$ to $\langle$**start**$, x = 0\rangle$; *dashed* lines are for the symbolic may action transitions, *e.g.,* state $\langle$**start**$, x \leq \beta\rangle$ may loop back to $\langle$**detect**$, x = 0\rangle$ with action "alarm!"; and *dash dotted* lines are for symbolic delay transitions, *e.g.,* state $\langle$**detect**$, x = 0\rangle$ evolves to $\langle$**detect**$, x \geq 0\rangle$ via zone progression.

3.2 Symbolic Reachability Analysis

Reachability analysis lies at the core of many verification problems, *e.g.,* we can verify safety properties by checking whether some *bad* states are reachable. Inspired by the zone-graph based reachability algorithm in [6], we propose a symbolic algorithm for the reachability analysis of parametric zone-graphs, which is useful for the verification of time-parametric modal specifications.

As illustrated in Algorithm 1, we check whether a target state $\langle l_f, \phi_f \rangle$ is reachable from the initial state $\langle \bar{l}, D_0 \rangle$ by exploring the state-space of parametric zone-graph [2] on-the-fly. Note that we twist Algorithm 1 for both the *must* and *may* reachability analysis (the differences are in Lines 7 and 12). The algorithm maintains two sets of states: PASSED for those having been traversed and WAIT for those to be considered. Starting with the initial state $\langle \bar{l}, D_0 \rangle$, the algorithm processes each element $\langle l, D \rangle$ of WAIT till

[2] We assume a normalized zone-graph with finite transition relations.

Algorithm 2 Checking the specification-implementation relation

```
Input: a TIOA $\mathcal{A}$ and a time-parametric modal specification $\mathcal{M}$
Output: "YES", if $\mathcal{A}$ is an implementation of $\mathcal{M}$; "NO", otherwise
obtain a finite set $F$ of valid parameter assignments by solving $C(\Theta)$ of $\mathcal{M}$
while $F \neq \emptyset$ do
    take a parameter assignment $f(\Theta)$ from $F$
    obtain a pair of TIOAs $\mathcal{M}_f^{\Box}$, $\mathcal{M}_f^{\Diamond}$ by substituting all occurrences of parameters $\Theta$ with
    values $f(\Theta)$, and keeping only must and may transitions, respectively
    if $\mathcal{M}_f^{\Box}$ simulates $\mathcal{A}$, and $\mathcal{A}$ simulates $\mathcal{M}_f^{\Diamond}$ then
        return "YES"
    end if
end while
return "NO"
```

the set becomes empty. If there is no clock valuation $v \in D$ for any parameter assignment $f(\Theta)$, *i.e.,* $D = \emptyset$, then state $\langle l, D\rangle$ and all transitions from/to it are pruned (see Line 5). If $l = l_f$ and $D \cap \phi_f \neq \emptyset$ for all valid parameter assignment $f(\Theta)$, then the target *must* be reachable; on the other hand, we say that $\langle l_f, \phi_f\rangle$ *may* be reachable if $D \cap \phi_f \neq \emptyset$ is only true for some $f(\Theta)$, in the sense that the target may (not) be reachable for some implementation of the time-parametric modal specification. If $\langle l, D\rangle$ does not hit the target and has not been traversed (see Line 10), then the algorithm adds $\langle l, D\rangle$ to PASSED and all its successor states to WAIT. The algorithm terminates when WAIT is empty, and outputs that the target state $\langle l_f, \phi_f\rangle$ is not reachable.

The termination of Algorithm 1 is guaranteed, because the parametric zone-graph G has a finite set of symbolic states and finite transition relations, *i.e.,* the size of WAIT is finite. The correctness of Algorithm 1 is given by Theorem 3.

Example 4. Consider the parametric zone-graph shown in Fig. 3. A target $\langle \textbf{infusion}, x \leq 5\rangle$ *must* be reachable, because there is a path

$$\langle \textbf{detect}, x = 0\rangle \to \langle \textbf{start}, x = 0\rangle \to \langle \textbf{start}, x \leq \beta\rangle \to \langle \textbf{infusion}, \alpha \leq x \leq \beta\rangle$$

and, for any valid parameter values satisfying the constraint: $\alpha \leq \beta \leq 3$, we have $x \leq \beta \leq 5$. Suppose the target is $\langle \textbf{infusion}, x \leq 1\rangle$, then it *may* be reachable, because $(\alpha \leq x \leq \beta) \wedge (x \leq 1) \neq \emptyset$ is true for some parameter assignments, *e.g.,* $\alpha = \beta = 1$, but false for others, *e.g.,* $\alpha = \beta = 3$.

3.3 Symbolic Modal Refinement Checking

Recall from Section 2.3 that we verify whether a TIOA is an implementation of a time-parametric modal specification via modal refinement check. We now propose a symbolic method to check the specification-implementation relation, by reducing the problem to the timed simulation check of timed automata, where standard zone-graph based algorithms [18] are available.

Algorithm 2 illustrates our method. Given a time-parametric modal specification $\mathcal{M}$, we first solve the linear constraints $C(\Theta)$ and obtain a finite set of parameter as-

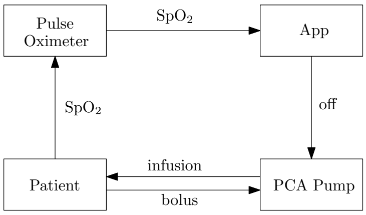


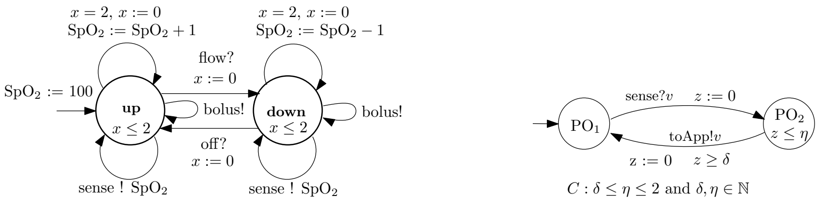
Fig. 4: A closed-loop PCA system.

signments. For each parameter assignment f, we substitute all the occurrences of parameters Θ with values $f(\Theta)$, resolving the timing variability of the specification. Then, by keeping only the must (*resp.* may) transition relations of $\mathcal{M}$, we obtain a TIOA $\mathcal{M}_f^{\square}$ (*resp.* $\mathcal{M}_f^{\lozenge}$). We check the timed simulation relation between TIOAs $\mathcal{M}_f^{\square}$, $\mathcal{A}$ and $\mathcal{M}_f^{\lozenge}$ (see Line 5), by applying the standard zone-graph based algorithm [18]. If both simulation checks are successful, we claim that $\mathcal{A}$ is an implementation of $\mathcal{M}$; otherwise, we continue with another parameter assignment. The termination of Algorithm 2 is guaranteed, because there are only finite parameter assignments; if we cannot find any assignment f that makes the condition in Line 5 true, then $\mathcal{A}$ is not an implementation of $\mathcal{M}$.

We argue the correctness of Algorithm 2 as follows. According to the definition of timed simulation [18], $\mathcal{M}_f^{\square}$ simulates $\mathcal{A}$ iff every delay/action transition in $[\![\mathcal{M}_f^{\square}]\!]$ is matched in $[\![\mathcal{A}]\!]$; similarly, $\mathcal{A}$ simulates $\mathcal{M}_f^{\lozenge}$ yields that every transition in $[\![\mathcal{A}]\!]$ has an exact match in $[\![\mathcal{M}_f^{\lozenge}]\!]$. Based on Definition 6, we can claim that $[\![\mathcal{A}]\!]$ modally refines MTTS $[\![\mathcal{M}_f]\!]$, whose must (*resp.* may) action transitions are given by the action transitions of $[\![\mathcal{M}_f^{\square}]\!]$ (*resp.* $[\![\mathcal{M}_f^{\lozenge}]\!]$) and delay transitions coincide with that of $[\![\mathcal{M}_f^{\square}]\!]$ or $[\![\mathcal{M}_f^{\lozenge}]\!]$. Moreover, $[\![\mathcal{M}_f]\!]$ belongs to the set $[\![\mathcal{M}]\!]$. Therefore, based on Definition 7, $\mathcal{A}$ is an implementation of $\mathcal{M}$.

4 Case Study

In this section we describe how time parametric modal specifications can be used to specify and analyze on-demand medical systems. In an on-demand medical system clinicians assemble a composite medical device by coupling interoperable medical devices with applications that implement clinical algorithms. The devices carry a capabilities specification, while the applications carry requirements specifications. An underlying platform checks that only devices with capabilities compatible with an application are used; If a clinician attempts to couple an incompatible device they are notified and the coupling process is aborted [14]. As the specific device (*i.e.,*, make, brand, or model) to be coupled to the application is not known apriori by the application developer the safety assesment of an application must be done with respect to the application's device requirements. Because of this, these specifications play an important role. Not only must the formalism be able to capture enough details about device behavior to enable a realistic safety assesment, they must also allow the developer to provide specifications that are as general as possible in order to ensure that a given application is compatible with as many devices as is safe.



(a) A patient model. **(b) Pulse-oximeter specification**

Fig. 5: Other components of the closed-loop system.

For the rest of this section we will describe how such a system providing closed loop control of PCA therapy can be specified using time-parametric modal specifications, how refinement is used to check compatibility, and how the behavior of the overall application can be analyzed against its requirements. In this section we adopt a syntactic sugar by allowing each specification to contain state variables. Locations at the semantic level are then related to the cross-product of the state variable values in the standard way.

In PCA therapy a patient is attached to an infusion pump that provides a bolus of painkiller at the request of the patient. If the patient requests too much painkiller they run the risk of overdose. In the closed loop system (Fig. 4) a pulse oximeter is used to measure the blood oxygenation of the patient (SpO_2). The SpO_2 reading is forwarded over the network to the application which then determines if infusion should be canceled. If so, an ‘off’ signal is sent to the pump. First we adopt a simple patient model (Figure 5a) that relates the activity of the infusion pump to the $\mathrm{Sp}O_2$ of the patient. When infusion is off, the patient’s $\mathrm{Sp}O_2$ rises by 1 every two time units. When infusion is on, the $\mathrm{Sp}O_2$ decreases by 1 every time unit (the $\mathrm{Sp}O_2$ can never go above 100 or below 0). In each location, the patient model can emit the current $\mathrm{Sp}O_2$ value. Next we model the pulse oximeter which is a device that senses the $\mathrm{Sp}O_2$ of the patient then forwards its reading onto the application (Fig. 5b). Valid pulse oximeter implementations are allowed to sense and forward $\mathrm{Sp}O_2$ values as fast as they want, but they are required to read and forward a value at least once every 2 time units.

For the sake of space we omit a visualization of the safety interlock application. However, its functionality is simple. On every update from the pulse oximeter it determines if it should cancel any ongoing infusion. If the value is < 95 infusion is canceled. The application will also cancel infusion if it does not receive a $\mathrm{Sp}O_2$ update for more than 3 time units. Finally, the pump is as modeled previously in Fig. 1. If we wanted to verify a simple safety property, *e.g.,* that the patient’s SpO_2 never goes below 85, we would compose the application, patient model, and the requirements specification on the pump and pulse oximeter and then apply the state reachability algorithm from Section 3.1. Compatibility is simply a refinement check between the device’s capabilities specification and the applications requirements specification.

5 Conclusion

In this paper we introduce *time-parametric modal specifications* which are timed I/O automata extended with must/may transitions and parametric clock constraints. This formalism enables system designers to specify a family of implementations who differ not only functionally but also in terms of timing behavior. We build a specification theory based on his formalism, including modal refinement, safety / liveness property preservation, and compositional reasoning. We also develop two symbolic algorithms for the reachability analysis and specification-implementation relation checking. The usefulness of our theory is demonstrated via a case study of medical device systems.

For the future work, there are several potential directions. Firstly, we aim to improve the modal refinement decision algorithm in 3.3; instead of checking the refinement for every possible parameter assignment, we would like to find a way to narrow down the choices of parameters when performing the refinement check on-the-fly. Secondly, we will consider *weak modal refinement* which concerns only the observable behavior of systems. Moreover, we plan on building a tool that implements all proposed algorithms. This tool would be used to both verify properties and check refinement of time-parametric specifications. Additionally, we plan to incorporate the refinement check into a prototype on-demand medical system platform.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126, 183–235 (1994)
2. Antonik, A., Huth, M., Larsen, K.G., Nyman, U., Wasowski, A.: 20 years of modal and mixed specifications. Bulletin of the EATCS 95, 94–129 (2008)
3. Behrmann, G., David, A., Larsen, K.G., Hakansson, J., Petterson, P., Yi, W., Hendriks, M.: Uppaal 4.0. In: Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems. pp. 125–126. QEST '06, IEEE Computer Society, Washington, DC, USA (2006)
4. Benes, N., Kretínský, J., Larsen, K.G., Møller, M.H., Srba, J.: Parametric modal transition systems. In: ATVA. pp. 275–289 (2011)
5. Benes, N., Kretínský, J., Larsen, K.G., Møller, M.H., Srba, J.: Dual-priced modal transition systems with time durations. In: LPAR. pp. 122–137 (2012)
6. Bengtsson, J., Yi, W.: Timed automata: Semantics, algorithms and tools. In: Lectures on Concurrency and Petri Nets. pp. 87–124 (2003)
7. Bertrand, N., Legay, A., Pinchinat, S., Raclet, J.B.: Modal event-clock specifications for timed component-based design. Sci. Comput. Program. 77(12), 1212–1234 (2012)
8. Bertrand, N., Pinchinat, S., Raclet, J.B.: Refinement and consistency of timed modal specifications. In: LATA. pp. 152–163 (2009)
9. Cerans, K., Godskesen, J.C., Larsen, K.G.: Timed modal specification - theory and tools. In: CAV. pp. 253–267 (1993)
10. David, A., Larsen, K.G., Legay, A., Nyman, U., Wasowski, A.: Timed i/o automata: a complete specification theory for real-time systems. In: HSCC. pp. 91–100 (2010)
11. De Moura, L., Bjørner, N.: Z3: an efficient smt solver. In: TACAS'08. pp. 337–340. Springer-Verlag, Berlin, Heidelberg (2008)
12. Henzinger, T.A., Nicollin, X., Sifakis, J., Yovine, S.: Symbolic model checking for real-time systems. Inf. Comput. 111(2), 193–244 (1994)

13. Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.W.: Linear parametric model checking of timed automata. In: TACAS. pp. 189–203 (2001)
14. King, A.L., Feng, L., Sokolsky, O., Lee, I.: A modal specification approach for on-demand medical systems. In: Third International Symposium on Foundations of Health Information Engineering and Systems (2013)
15. King, A.L., Feng, L., Sokolsky, O., Lee, I.: A modal specification theory for timing variability (full version). Tech. Rep. MS-CIS-13-11, University of Pennsylvania (2013)
16. Larsen, K.G., Nyman, U., Wasowski, A.: Modal i/o automata for interface and product line theories. In: Programming Languages and Systems, pp. 64–79. Springer (2007)
17. Larsen, K.G., Thomsen, B.: A modal process logic. In: LICS. pp. 203–210 (1988)
18. Weise, C., Lenzkes, D.: Efficient scaling-invariant checking of timed bisimulation. In: STACS. pp. 177–188 (1997)

Appendix

Lemma 1. *Let A be a TTS and M be a MTTS such that $A \preceq M$. Let ψ be a safety or liveness property. Suppose $M \models_{\Box} \psi$, then $A \models \psi$.*

Proof. Case 1: $\psi = \neg \mathtt{E}\langle\rangle\neg\phi$ is a safety property where ϕ is a state formula for *good* behavior. For the sake of contradiction, we assume $A \not\models \psi$, *i.e.,* $A \models \mathtt{E}\langle\rangle\neg\phi$, meaning that there exist a path π in A reaching some bad state $s_f \models \neg\phi$ from the initial state $\overline{s}$. Since $A \preceq M$, every transition in A is also allowed in M; that is, every action transition $s \xrightarrow{a} s'$ along π corresponds to a may transition $t \xrightarrow{a}_{\Diamond} t'$ in M, and every delay transition $s \xrightarrow{d} s'$ for $d \in \mathbb{R}_{\geq 0}$ along π has an exact match $t \xrightarrow{d}_{\mathtt{d}} t'$ in M. Thus, from the initial state $\overline{t}$ of M, there is path π' corresponding to π and reaching some state $t_f \models \neg\phi$; since path π' consists of only may action and delay transitions, we have $M \models_{\Diamond} \mathtt{E}\langle\rangle\neg\phi$. This is a contradiction to $M \models_{\Box} \neg\mathtt{E}\langle\rangle\neg\phi$, which suggests that $M \models_{\Diamond} \mathtt{E}\langle\rangle\neg\phi$ is false. Therefore, we shall have $A \models \psi$.

Case 2: $\psi = \mathtt{A}\langle\rangle\phi$ is a liveness property where ϕ is a state formula. Given that $M \models_{\Box} \psi$, every path π_i leaving the initial state $\overline{t}$ of M would eventually reach some state $t_f \models \phi$ via a sequence of must action and delay transitions. Since $A \preceq M$, for every must action transition $t \xrightarrow{a}_{\Box} t'$, there is a corresponding transition $s \xrightarrow{a} s'$ in A; and for every delay transition $t \xrightarrow{d}_{\mathtt{d}} t'$ with $d \in \mathbb{R}_{\geq 0}$, there is a counterpart $s \xrightarrow{d} s'$ in A. Thus, every path initiated from $\overline{s}$ in A corresponds to a path π_i in M and would eventually reach some state $s_f \models \phi$; that is, $A \models \mathtt{A}\langle\rangle\phi$, *i.e.,* $A \models \psi$.

Lemma 2. *Let $\mathcal{A}$ be a TIOA implementation of a specification $\mathcal{M}$,* i.e., $\mathcal{A} \sqsubseteq \mathcal{M}$. *Let $\mathcal{A}'$ be a composeable TIOA with $\mathcal{A}$. Then $\mathcal{A}\|\mathcal{A}' \sqsubseteq \mathcal{M}\|\mathcal{A}'$.*

Proof. Since $\mathcal{A} \sqsubseteq \mathcal{M}$, based on Definition 7, there must exist a MTTS $M \in [\![\mathcal{M}]\!]$ such that $[\![\mathcal{A}]\!] \preceq M$. Let f be the parameter assignment corresponding to M. By substituting all occurrences of parameters in $\mathcal{M}$ with values f, we obtain $\mathcal{M}_f$ such that $[\![\mathcal{M}_f]\!] = M$ and $[\![\mathcal{M}_f\|A']\!]$ yields a single MTTS belonging to the set $[\![\mathcal{M}\|A']\!]$. In order to ensure $\mathcal{A}\|\mathcal{A}' \sqsubseteq \mathcal{M}\|\mathcal{A}'$, we only need to prove that $[\![\mathcal{A}\|\mathcal{A}']\!] \preceq [\![\mathcal{M}_f\|\mathcal{A}']\!]$ is true, by considering the following three cases.

1. For any *must action* transition $\langle l, v\rangle \xrightarrow{a}_{\Box} \langle l', v'\rangle$ in $[\![\mathcal{M}_f\|\mathcal{A}']\!]$, we shall prove that there is a corresponding transition in $[\![\mathcal{A}\|\mathcal{A}']\!]$.
   - Suppose $a \in Act_{\mathcal{M}}$ and the transition is a product of the synchronizing rule in Definition 8. Let $l = (l_s, l_q)$ and $l' = (l'_s, l'_q)$, where l_s, l'_s (*resp.* l_q, l'_q) are locations of $\mathcal{M}_f$ and are locations of $\mathcal{A}'$. There is a transition $\langle l_s, v_s\rangle \xrightarrow{a}_{\Box} \langle l'_s, v'_s\rangle$ in M. Since $[\![\mathcal{A}]\!] \preceq M$, there must exist some $\langle l_t, v_t\rangle \xrightarrow{a} \langle l'_t, v'_t\rangle$ in $[\![\mathcal{A}]\!]$ and hence a corresponding transition $\langle (l_t, l_q), \mu\rangle \xrightarrow{a} \langle (l'_t, l'_q), \mu'\rangle$ in $[\![\mathcal{A}\|\mathcal{A}']\!]$.
   - If $a \in Act_{\mathcal{M}}$, $l = (l_s, l_q)$ and $l' = (l'_s, l_q)$, then the transition is obtained from a interleaving step on $\mathcal{M}_f$, so that there is a transition $\langle l_s, v_s\rangle \xrightarrow{a}_{\Box} \langle l'_s, v'_s\rangle$ in M. Since $[\![\mathcal{A}]\!] \preceq M$, there must exist a corresponding transition $\langle l_t, v_t\rangle \xrightarrow{a} \langle l'_t, v'_t\rangle$ in $[\![\mathcal{A}]\!]$ and hence $\langle (l_t, l_q), \mu\rangle \xrightarrow{a} \langle (l'_t, l_q), \mu'\rangle$ in $[\![\mathcal{A}\|\mathcal{A}']\!]$.
   - If $a \notin Act_{\mathcal{M}}$, then the transition is a result of the interleaving on $\mathcal{A}'$, *i.e.,* $l = (l_s, l_q)$ and $l' = (l_s, l'_q)$. Thus, a corresponding interleaving transition $\langle (l_t, l_q), \mu\rangle \xrightarrow{a} \langle (l_t, l'_q), \mu'\rangle$ shall take place in $[\![\mathcal{A}\|\mathcal{A}']\!]$.

2. For any transition $\langle l, v\rangle \xrightarrow{a} \langle l', v'\rangle$ in $[\![\mathcal{A} \| \mathcal{A}']\!]$, we can prove that there is *may action* transition in $[\![\mathcal{M}_f \| \mathcal{A}']\!]$, by reasoning about the correspondence between their projected transitions on $[\![\mathcal{A}]\!]$ and $[\![\mathcal{M}_f]\!]$ in a similar way as for the *must action* case.
3. For any *delay* transition $\langle l, v\rangle \xrightarrow{d}_{\mathsf{d}} \langle l, v + d\rangle$ in $[\![\mathcal{M}_f \| \mathcal{A}']\!]$, let its projection on $[\![\mathcal{M}_f]\!]$ (or M) be $\langle l_s, v_s\rangle \xrightarrow{d}_{\mathsf{d}} \langle l_s, v_s + d\rangle$. Since $[\![\mathcal{A}]\!] \preceq M$, based on Definition 6, there must exist a transition $\langle l_t, v_t\rangle \xrightarrow{d} \langle l_t, v_t + d\rangle$ in $[\![\mathcal{A}]\!]$ and hence a corresponding delay transition in $[\![\mathcal{A} \| \mathcal{A}']\!]$. Similarly, we can prove that there is a matching transition in $[\![\mathcal{M}_f \| \mathcal{A}']\!]$ for any delay transition in $[\![\mathcal{A} \| \mathcal{A}']\!]$.

According to Definition 6, the above three cases yield $[\![\mathcal{A} \| \mathcal{A}']\!] \preceq [\![\mathcal{M}_f \| A']\!]$. Thus, we have proved that $\mathcal{A} \| \mathcal{A}' \sqsubseteq \mathcal{M} \| \mathcal{A}'$.

Theorem 3. *Let $\mathcal{M}$ be a time-parametric modal specification, $[\![\mathcal{M}]\!]_{\mathsf{z}}$ be its symbolic semantics and $[\![\mathcal{M}]\!]$ be its operational semantics.*

- (Soundness) *if the initial symbolic state $\langle \bar{l}, D_{\mathbf{0}}\rangle$ in $[\![\mathcal{M}]\!]_{\mathsf{z}}$ must (*resp. *may) lead to a target state $\langle l_f, D_f\rangle$, then for all $v_f \in D_f$, state $\langle l_f, v_f\rangle$ must (*resp. *may) be reachable from the initial state $\langle \bar{l}, \mathbf{0}\rangle$ in some $M \in [\![\mathcal{M}]\!]$*
- (Completeness) *if, in any $M \in [\![\mathcal{M}]\!]$, a target state $\langle l_f, v_f\rangle$ must (*resp. *may) be reachable from the initial state $\langle \bar{l}, \mathbf{0}\rangle$, then state $\langle \bar{l}, D_{\mathbf{0}}\rangle$ in $[\![\mathcal{M}]\!]_{\mathsf{z}}$ must (*resp. *may) lead to $\langle l_f, D_f\rangle$ for some D_f such that $v_f \in D_f$*

Proof. We will prove by induction on the length of paths. Without loss of generality, we assume that all paths are expressed in the form of alternating (must or may) action transitions and delay transitions, *i.e.*, $\cdots \langle l_{i-1}, v_{i-1}\rangle \xrightarrow{a} \langle l_i, v_i\rangle \xrightarrow{d} \langle l_{i+1}, v_{i+1}\rangle \cdots$ for $a \in \mathit{Act}$ and $d \in \mathbb{R}_{\geq 0}$.

(Soundness) Assume $\langle \bar{l}, D_{\mathbf{0}}\rangle \rightsquigarrow^* \langle l_n, D_n\rangle \overset{\sigma}{\rightsquigarrow} \langle l_{n+1}, D_{n+1}\rangle$, where $\rightsquigarrow^*$ represents a succession of transitions. By induction, we have $\langle \bar{l}, \mathbf{0}\rangle \rightarrow^* \langle l_n, v_n\rangle$ for all $v_n \in D_n$. We need to prove for all $v_{n+1} \in D_{n+1}$, there is a transition $\langle l_n, v_n\rangle \xrightarrow{\sigma} \langle l_{n+1}, v_{n+1}\rangle$. There are two cases, since σ can be an action or a delay.

- Suppose $\langle l_n, D_n\rangle \overset{a}{\rightsquigarrow}_{\gamma} \langle l_{n+1}, D_{n+1}\rangle$ for $a \in \mathit{Act}$ and $\gamma \in \{\Box, \Diamond\}$. Based on Definition 9, we have $l_n \overset{g,a,r}{\hookrightarrow}_{\gamma} l_{n+1}$ and $D_{n+1} = r(D_n \wedge g) \wedge \mathit{Inv}(l_{n+1})$. By Definition 5, there is a transition $\langle l_n, v_n\rangle \xrightarrow{a}_{\gamma} \langle l_{n+1}, v_{n+1}\rangle$ in some $M \in [\![\mathcal{M}]\!]$ such that $v_n \in g$. Thus, for all $v_{n+1} \in D_{n+1}$, there is a $v_n \in D_n$ such that $v_n \in g$, $v_{n+1} \in \mathit{Inv}(l_{n+1})$ and $v_{n+1} = v_n[r \mapsto 0]$.
- Suppose $\langle l_n, D_n\rangle \overset{d}{\rightsquigarrow}_{\mathsf{d}} \langle l_{n+1}, D_{n+1}\rangle$ for $d \in \mathbb{R}_{\geq 0}$. From Definition 9, we have $l_n = l_{n+1}$ and $D_{n+1} = D_n^{\uparrow} \wedge \mathit{Inv}(l_n)$. Due to the definition of zone progression, we have $D_{n+1} = \{v_n + d \mid v_n \in D_n, d \in \mathbb{R}_{\geq 0}$ and $v_n + d \in \mathit{Inv}(l_n)\}$. Based on Definition 5, we have $\langle l_n, v_n\rangle \xrightarrow{d}_{\mathsf{d}} \langle l_n, v_n + d\rangle$ if $v_n + d \in \mathit{Inv}(l_n)$. Thus, for all $v_{n+1} \in D_{n+1}$, there is a $v_n \in D_n$ such that $v_{n+1} = v_n + d$ and $v_n, v_{n+1} \in \mathit{Inv}(l_n)$.

(Completeness) Assume $\langle \bar{l}, \mathbf{0}\rangle \rightarrow^* \langle l_n, v_n\rangle \xrightarrow{\sigma} \langle l_{n+1}, v_{n+1}\rangle$. The induction step gives $\langle \bar{l}, D_{\mathbf{0}}\rangle \rightsquigarrow^* \langle l_n, D_n\rangle$ and $v_n \in D_n$. We need to prove that $\langle l_n, D_n\rangle \overset{\sigma}{\rightsquigarrow} \langle l_{n+1}, D_{n+1}\rangle$ for some D_{n+1} and $v_{n+1} \in D_{n+1}$. There are two cases:

- Suppose $\langle l_n, v_n \rangle \xrightarrow{a}_\gamma \langle l_{n+1}, v_{n+1} \rangle$ for $a \in Act$ and $\gamma \in \{\Box, \Diamond\}$ in any $M \in [\![\mathcal{M}]\!]$. Based on Definition 5, there is a transition $l_n \overset{g,a,r}{\hookrightarrow}_\gamma l_{n+1}$ in $\mathcal{M}$ and $v_n \in g$, $v_{n+1} = v_n[r \mapsto 0]$ and $v_{n+1} \in Inv(l_{n+1})$. By Definition 9, we have $\langle l_n, D_n \rangle \overset{a}{\rightsquigarrow}_\gamma \langle l_{n+1}, D_{n+1} \rangle$ and $D_{n+1} = r(D_n \wedge g) \wedge Inv(l_{n+1})$. Thus, $v_{n+1} \in D_{n+1}$.
- Suppose $\langle l_n, v_n \rangle \xrightarrow{d}_{\mathsf{d}} \langle l_{n+1}, v_{n+1} \rangle$ for $d \in \mathbb{R}_{\geq 0}$. Then we have $l_n = l_{n+1}$, $v_{n+1} = v_n + d$ and $v_n, v_{n+1} \in Inv(l_n)$. From Definition 9, we get $\langle l_n, D_n \rangle \overset{d}{\rightsquigarrow}_{\mathsf{d}} \langle l_{n+1}, D_{n+1} \rangle$ and $D_{n+1} = D_n^{\uparrow} \wedge Inv(l_n) = \{v_n + d \mid v_n \in D_n, d \in \mathbb{R}_{\geq 0}$ and $v_n + d \in Inv(l_n)\}$. Thus, $v_{n+1} \in D_{n+1}$.