ERTL: An Extension to RTL for Requirements Analysis for Hybrid Systems

Rogério de Lemos[1] and Jon Hall[2]

[1]Centre for Software Reliability
Department of Computing Science
University of Newcastle upon Tyne, NE1 7RU, UK

[2]High Integrity Systems Engineering Group
Department of Computer Science
University of York, YO1 5DD, UK

December 11, 1996

1 Introduction

Real Time Logic, RTL, was introduced by Jahanian and Mok in [5] as a formalism for reasoning about the absolute timing properties of real-time systems. The need for RTL came from a perceived inextensibility of research at that time from relative timing problems of systems to absolute timings. RTL is a first order logic with an uninterpreted predicate which relates events of a system to the time of their occurrence. RTL assumes, as its model of time, a total order isomorphic to the natural numbers. In this paper we provide extensions to RTL which allow reasoning about absolute timings for both continuous and discrete time, and reasoning about system behaviour in the value and time domains by parametrising predicates in terms of system variables. Incorporating these features into RTL we obtain Extended Real-Time Logic (ERTL) which is suitable for the modelling and analysis of hybrid systems.

The paper is organized as follows. In the next section we present the concepts of an event-action model, and discuss their formalisation in terms of RTL and ERTL. Section 3 defines the syntax and semantics of ERTL. Section 4 presents two case studies, the first is the traditional cat and mouse problem, and the second concerns temperature control for a nuclear reactor by moving the reactor's rods. Finally, section 5 presents some concluding remarks.

2 Event-Action Model

The event-action model proposed by Jahanian and Mok captures the temporal ordering of computational actions, and is based on events, actions, state predicates and timing constraints. Some of the concepts on which the Jahanian and Mok event-action model is based were first introduced by Heninger [4] as a collection of concepts for specifying the software requirements of the operational flight program for the A-7 aircraft.

The motivation in our work for selecting the concepts of the event-action model for conducting the requirements analysis for hybrid systems is that they can be used to model system behaviour ranging from the activities of the physical entities (which are part of the environment of the computing system) to the temporal ordering of computational tasks [1, 2, 3]. However, to fully exploit the event-action model in the continuous time domain, it is necessary to extend RTL (originally proposed to specify only discrete aspects of system behaviour) to enable modelling and analysis of both discrete and continuous aspects of system behaviour to be performed. An advantage of such an approach is that it allows the modelling and analysis of the system and its environment using the same formal technique.

2.1 RTL as a Formaliser of the Event-Action Model

RTL was introduced in order to represent the event-action model concepts in a way which would allow the formal analysis to be performed by mechanical manipulation. RTL is a first order logic which captures the notion of time through the occurrence function "@" by relating an event with its time of occurrence. (In a subsequent publication [6], the occurrence function "@", which had time as a dependent variable, was replaced by (Boolean) occurrence relation "Θ" which has time as parameter; in the rest of this paper we will be referring only to the later publication.) The properties that characterise the behaviour of a particular system, are established from the subset of occurrence relations which provide models for that system. Occurrence relations are characterised by the *monotonicity axioms* which state that:

i. the same occurrence of an event cannot happen at two distinct times;

ii. two distinct occurrences of the same event must happen ate different times.

2.2 Extended RTL as a Formaliser of the Event-Action Model

As an extension to RTL, the original motivation for the introduction of that formalism will carry over to ERTL. This includes:

i. the ability to express the causal independence of subsystems which do not interact;

ii. the ability to express the interaction of systems through synchronous occurrence of events;

iii. the ability to express safety and liveness assumptions;

iv. the ability to express hard real time deadlines;

v the provision of notions for proof of satisfaction.

To these we will add the ability to describe the interactions of the whole system, which includes the computing system and its environemnt, rather than just the computing systems.

The axiom system for ERTL is an expanded form of that for RTL. The formal demonstration of the containment of RTL within ERTL is not difficult. The complexity of RTL is already suffiently high for there to exist undecideable subsets. This result will, of course, transfer to ERTL.

This section presents an informal view of ERTL by comparing and contrasting its basic concepts with those of RTL.

2.2.1 System Variables and System Predicates

A *system variable* holds the value of an attribute of a system. A system variable will, in general, vary over time and so may be considered as time dependent. Unlike RTL, ERTL allows the representation of system variables, supporting sophisticated reasoning techniques over the *value domain*.

In ERTL, a *system predicate* is a formulae over relational operators parameterised, in particular, by system variables. As such, the evaluation of a system predicate may vary with the values of the system variables which are its parameters, hence is also time dependent.

2.2.2 Events

An *event* defines and is defined by a point of the timeline. Intuitively, an event does not consume system resources. The notion of an event is common to both RTL and ERTL.

The transition of a system predicate from $false$ to $true$ or from $true$ to $false$ at a particular time point defines a *transition event*: given the system predicate P, the transitions events related to P are given by $\nearrow P$ and $\searrow P$, respectively.

2.2.3 Actions

An *action* defines and is defined by a closed interval (a contiguous subset of the timeline) which is also a non-singleton set. In particular, an action defines two events which are the first and last points of the defined interval which we call, respectively, the *start* and *end events* of the action. The *duration* of an (occurrence of an) action is the length of the interval it defines. Intuitively, an action consumes a bounded quantity of system resources. To be considered as consuming resources an action must be of non-zero duration (hence it is a non-singleton interval).

2.2.4 Occurrence Relation

As time progresses we may reach a time point which an event defines. At this point the event is said to *occur*. For an event e, the proposition that event e occurs at t is denoted $\Theta(e,t)$. The proposition that the i-th instance of an event e occurs at time t is denoted $\Theta(e,i,t)$. Θ is called an *occurrence relation.*

An action is said to be *in progress* if its start event has occurred, but its end event has not yet occurred. *In progress* may be translated into a proposition relating the time of start and end events through Θ.

2.2.5 Holding Relation

A system predicate is said to *hold* at a time point when the values of the system variables evaluated at that point in time and substituted for the corresponding system variables in the system predicate satisfy the resulting formula. For a system predicate, f, the proposition that formula f holds at t is denoted $\Phi(f,t)$. The proposition that f holds for the i-th time at t is denoted by $\Phi(f,i,t)$. Φ is called a *holding relation.*

Notice that within any one occurrence of a holding relation all system variables are evaluated at the same time point; for instance, it is not possible, within single propositions $\Phi(f,t)$ or $\Phi(f,i,t)$ to compare directly the value of a system variable at two distinct time points: if $Temp$ is a system variable representing the temperature of some part of a real-time system, to ask whether $Temp(5) = Temp(6)$ requires us to write $\exists u @ \Phi(Temp = u, 5) \wedge \Phi(Temp = u, 6)$.

3 ERTL: Syntax and Semantics

3.1 Syntax of ERTL

ERTL is a first order logic, $\varXi$, in which propositional variables are interpreted. The interpretation of a propositional variable takes one of two forms:

- an instance of the occurrence relation, $\varTheta$, relating the events to the time points of their occurrence, and
- an instance of the holding relation, $\varPhi$, relating actions and system predicates to the time intervals over which they hold.

As mentioned above, system predicates are predicates whose truth value varies over time. They form a separate first order logic, $\varUpsilon$, in which system variables may appear as ordinary variables. Formulae of $\varUpsilon$ may be embedded into $\varXi$ through their appearence as the first operand of the holding relation. There is a strong relationship between formulae of $\varUpsilon$ and those of $\varXi$ which is characterised below.

The enclosing predicate calculus we call the *outer calculus*, the embedded predicate calculus we call the *inner calculus*.

3.1.1 Syntax of the Inner Calculus

The inner calculus consists of the following symbols:

i. constants, such as 0, 100, Yes, δ;

ii. time independent system variables, such as i, j;

iii. time dependent system variables, such as $CiTemp(t)$, $PTemp(t)$;

iv. operations, such as $+$;

v. relations, including $=$, $<$, $\leq$, $>$, $\geq$;

vi. event constants, such as $BUTTON1$, $\nearrow(Temp > 5000)$, $\uparrow CoDMovRods$;

vii. action constants, such as $CoDMovRods$.

The inner calculus consists of the following terms:

i. a constant is an inner term;

ii. a time independent system variable is an inner term;

iii. a time dependent system variable is an inner term;

iv. if $tr_1, \ldots, tr_n$ are inner terms and o is an operator of arity n, then $o(tr_1, \ldots, tr_n)$ is an inner term, such as $i + 1$, $x + \delta$.

The inner calculus consists of the following formulae:

i. if $tr_1, \ldots, tr_n$ are inner terms and r is a relation of arity n, then $r(tr_1, \ldots, tr_n)$ is an inner formula, such as $Temp > 5$;

ii. if p is an inner formula then so is $\neg p$, such as $\neg Temp > 5$;

iii. if p, q are inner formulae then so is $p \wedge q$, such as $Temp > j \wedge j > 7$;

iv. if p is an inner formula v a logical variable then $\forall v \bullet p$ is an inner formula, such as $\forall j \bullet Temp > 5 \Rightarrow j > k$.

An inner formula in which only time independent system variables appear is called a *time independent* formula, and otherwise a *time dependent* formula.

The inner calculus has the following syntactic restrictions:

i. for an action constant a there are events $\uparrow a, \downarrow a$: the start and stop events of an action, and

ii. for an inner formula f there are transitions events $\nearrow f, \searrow f$: the start and stop transition events of a formula.

3.1.2 Syntax of the Outer Calculus

The outer calculus consists of the following symbols:

i. constants,

ii. time independent system variables,

iii. time variables,

iv. operations,

v. relations,

vi. Θ and Φ.

The outer calculus consists of the following sub-propositions:

i. for e an event constant, i an integer and t a time variable, there are subpropositions $\Theta(e,t)$ and $\Theta(e,i,t)$;

ii. for a an action constant, i an integer and t a time variable there are subpropositions $\Theta(\uparrow a,i,t)$ and $\Theta(\downarrow a,i,t)$;

iii. for f an inner formula, i an integer and t a time variable there are subpropositions $\Phi(f,t)$, $\Phi(f,i,t)$, $\Theta(\nearrow f,i,t)$ and $\Theta(\searrow f,i,t)$.

The outer calculus consists of the following terms:

i. a constant is an outer term;

ii. a time independent system variable is an outer term;

iii. a time variable is an outer term, such as t_1;

iv. if $tr_1,\ldots,tr_n$ are outer terms and o is an operator of arity n, then $o(tr_1,\ldots,tr_n)$ is an outer term such as $t_1+\delta$.

The outer calculus consists of the following formulae:

i. if $tr_1,\ldots,tr_n$ are outer terms and r is a relation of arity n, then $r(tr_1,\ldots,tr_n)$ is an outer formula, such as $t_1+\delta > 100$;

ii. an outer subproposition is an outer formula, such as $\Phi(PTemp > 5000, 5, t)$;

iii. if p is an outer formula then so is $\neg p$, such as $\neg t_1+\delta > 100$;

iv. if p,q are outer formulae then so is $p \wedge q$, such as $\forall t \bullet \Phi(PTemp > 5000, 5, t) \wedge \neg t_1+\delta > 100$;

v. if p is an outer formula, and v is a time independent system variable then $\forall v \bullet p$ is an outer formula such as $\forall i,t \bullet \Phi(PTemp > 5000, i, t) \wedge \neg t_1+\delta > 100$;

vi. if p is an outer formula, and t a time variable then $\forall t \bullet p$ is an outer formula such as $\forall i,t \bullet \Phi(PTemp > 5000, i, t) \wedge \neg t_1+\delta > 100$.

3.2 Semantics of ERTL

We now partially characterise the models of the systems through the interrelationship of the symbols and strings of the syntax.

3.2.1 Type Correctness

If there are types associated with logical or system variables, we will expect a model to be type-consistent.

3.2.2 Minimal Models

Any model must contain the following objects:

i. $\mathcal{N}_\infty$, the positive integers.

ii. $\mathcal{T} \subseteq \mathcal{R}_+$, the timeline. The timeline should be a total order, isomorphic to the usual total order on (a subset of) the positive real numbers.

To aid the description of the semantics we will use the following notation: F_I are inner formulae, F_O are outer formulae with similar notation for inner and outer terms: Tr_I, Tr_O, and inner and outer constants: C_I, C_O. Tr_I further partitions into Tr_I^i and Tr_I^d, these being time independent and time dependent formulae, respectively. E is the collection of event constants; A is the collection of action constants. Unless otherwise necessary we will assume that all sets are pairwise disjoint.

3.2.3 Occurrence and Holding Relations and their Interrelation

i. An *occurrence relation* is a subset of $(E \times \mathcal{T}) \cup (\mathcal{E} \times \mathcal{N}_\infty \times \mathcal{T})$ such that the following *indexing conditions* hold:

$$\begin{aligned}\forall t \bullet \Theta(e, 1, t) &\Leftrightarrow \Theta(e, t) \wedge \forall t_1 \bullet 0 \le t_1 < t \Rightarrow \neg\Theta(e, t_1)\\ \forall i, t \bullet \Theta(e, i+1, t) &\Leftrightarrow \Theta(e, t) \wedge \exists t_1 \bullet t_1 < t\\ &\qquad \wedge\ \Theta(e, i, t_1) \wedge \forall t_2 \bullet t_1 < t_2 < t \Rightarrow \neg\Theta(e, t_2)\end{aligned}$$

and such that the following *monotonicity conditions* hold:

$$\begin{aligned}\forall i, t, t' \bullet [\Theta(e, i, t) \wedge \Theta(e, i, t')] &\Rightarrow t = t'\\ \forall i, t \bullet \Theta(e, i+1, t) &\Rightarrow [\exists t' \bullet \Theta(e, i, t') \wedge t' < t]\end{aligned}$$

Each action a must have non-zero duration but be bounded above:

$$\begin{aligned}\forall i, t, t' \bullet \Theta(\uparrow a, i, t) \wedge \Theta(\downarrow a, i, t') &\Rightarrow t < t'\\ \forall i, t \bullet \Theta(\uparrow a, i, t) &\Rightarrow \exists t' \bullet \Theta(\downarrow a, i, t')\end{aligned}$$

ii. A *holding relation* is a subset of $((F_I \cup A) \times \mathcal{T}) \cup ((\mathcal{F_I} \cup \mathcal{A}) \times \mathcal{N}_\infty \times \mathcal{T})$ such that the following *monotonicity conditions* hold:

$$\begin{aligned}\forall i, t_1, t_2 \bullet \Phi(f, i, t_1) \wedge \Phi(f, i, t_2) &\Rightarrow \forall t \bullet t_1 \le t \le t_2 \Rightarrow \Phi(f, i, t)\\ \forall i, i', t \bullet \Phi(f, i, t) \wedge \Phi(f, i', t) &\Rightarrow i = i'\end{aligned}$$

and such that, if f, g are inner formulae, then

$$\begin{aligned}\forall t \bullet \Phi(\neg f, t) &\Leftrightarrow \neg\Phi(f, t)\\ \forall t \bullet \Phi(f \wedge g, t) &\Leftrightarrow \Phi(f, t) \wedge \Phi(g, t)\\ \forall t \bullet \Phi(\forall v \bullet f, t) &\Leftrightarrow \forall v \bullet \Phi(f, t)\end{aligned}$$

and if f is time independent then

$$f \Leftrightarrow \forall t \bullet \Phi(f, t)$$

The latter collection of four properties of a holding relation are the ‘strong relationship’ between the outer and inner calculus mentioned above; they allow the migration of logical connectives appearing in a formula, and time independent formulae, between calculi.

iii. Holding and occurrence relations are related thus:

$$\begin{aligned}&\forall t \bullet \Theta(\nearrow f, t)\\ &\quad\Leftrightarrow\ \Phi(f, t) \wedge (t = 0 \vee \exists t_1 \bullet t_1 < t \wedge \forall t_2 \bullet t_1 \le t_2 < t \Rightarrow \neg\Phi(f, t_2))\\ &\qquad\quad \vee \neg\Phi(f, t) \wedge \exists t_1 \bullet t_1 > t \wedge \forall t_2 \bullet t < t_2 \le t_1 \Rightarrow \Phi(f, t_2)\\ &\forall t \bullet \Theta(\searrow f, t)\\ &\quad\Leftrightarrow\ \neg\Phi(f, t) \wedge (t = 0 \vee \exists t_1 \bullet t_1 < t \wedge \forall t_2 \bullet t_1 \le t_2 < t \Rightarrow \Phi(f, t_2))\\ &\qquad\quad \vee\ \Phi(f, t) \wedge \exists t_1 \bullet t_1 > t \wedge \forall t_2 \bullet t < t_2 \le t_1 \Rightarrow \neg\Phi(f, t_2)\end{aligned}$$

iv. There are indexing conditions for Φ as well:

$$\begin{aligned}&\forall i, t \bullet \Phi(f, i, t)\\ &\quad \Leftrightarrow \quad \Phi(f, t) \wedge \exists t_1 \leq t \bullet \Theta(\nearrow f, i, t_1) \wedge \forall t_2 \bullet t_2 < t \Rightarrow \neg \Theta(\searrow f, i + \delta_f, t_2)\end{aligned}$$

where

$$\delta_f = \begin{cases} 0 & \Phi(f, 0) \\ 1 & \neg \Phi(f, 0) \end{cases}$$

this being shorthand for considering the initial value of the formula.

In order for ERTL to be compatible with RTL, in the axioms of items *iii* and *iv* above, we postulate the occurrence of a transition event at time 0. This is captured by the existance of a disjunction with $(t = 0)$ at the righ hand side of both axioms of item *iii*. Whether it is a start $\Theta(\nearrow f, t)$ or a stop $\Theta(\searrow f, t)$ transition event, depends on whether the initial condition is assumed to be *true* or *false*, respectively.

4 Case Studies

To clarify the ERTL concepts introduced so far, in the following we present two hybrid systems case studies: the cat and mouse problem, and temperature control of a nuclear reactor.

4.1 Cat and Mouse

A sleeping cat is awakened just in time to catch a fleeing mouse [7]. The mouse and the cat both move along a straight line towards the mouse hole, where the mouse will be safe. The mouse starts running from a certain position mp_0 (left of the origin) with the velocity mv, and at the same time the cat starts running from a certain position cp_0 with velocity cv. The problem is, will the cat catch the mouse? The cat and mouse are regarded as being point objects (assumed to be capable of instantaneous acceleration/deceleration).

4.1.1 System Model

In the following we model the cat and mouse system in terms of its variables, constants, physical laws, assumptions and initial conditions. The system variables and constants are presented in the tab:1.

Name	Type	Comments
cp	≤ 0	cat position
mp	≤ 0	mouse position
cp_0	> 0	initial cat position
mp_0	> 0	initial mouse position
cv	> 0	constant cat velocity, moving to right
mv	> 0	constant mouse velocity, moving to right

Table 1: System variables and constants of the cat and mouse system

The only *physical law* of the system states that, if the velocity $\dot{p}$ of an arbitrary object remains constant then the distance travelled by the object from time t_1 to time t_2 is equal to the product of $\dot{p}$ and the difference between t_2 and t_1,

$$\forall t_1, t_2, p_1, p_2 \bullet \Phi(p = p_1, t_1) \wedge \Phi(p = p_2, t_2) \Rightarrow p_2 - p_1 = \dot{p}(t_2 - t_1).$$

To simplify the modelling of the cat and mouse system, the following *assumptions* establish the permitted range of velocities for the cat and mouse.

i. The cat is either at rest or moving with a constant velocity in the positive x direction:

$$\forall t \bullet \Phi(\dot{cp} = 0 \vee \dot{cp} = cv, t).$$

ii. The mouse is either at rest or moving with a constant velocity in the positive x direction:

$$\forall t \bullet \Phi(\dot{mp} = 0 \vee \dot{mp} = mv, t).$$

The *initial conditions* establish the initial positions of the cat and mouse, and their relative initial positions.

i. The initial position of the cat is cp_0 units to the left of the origin:

$$\Phi(cp = -cp_0, 0).$$

ii. The initial position of the mouse is mp_0 units to the left of the origin:

$$\Phi(mp = -mp_0, 0).$$

iii. Initially, the cat is farther away from the origin than the mouse:

$$(-cp_0 < -mp_0).$$

4.1.2 Finish Conditions

To establish the finish conditions we introduce the following two events which characterise whether the cat or the mouse wins:

i. The cat wins if it catches the mouse before the cat reaches the origin:

$$\forall t \bullet [\Theta(CatWon, t) \Leftrightarrow \Phi(cp = mp, t) \wedge \neg\Phi(cp = 0, t)].$$

ii. The mouse is safe if it reaches the origin before the cat catches it up:

$$\forall t \bullet [\Theta(MouseSafe, t) \Leftrightarrow \Phi(mp = 0, t) \wedge \forall t_1 \bullet t_1 < t \Rightarrow \neg\Phi(mp = cp, t_1)].$$

4.1.3 Analysis

In the following we identify the conditions under which the two finishing situations may occur. To conduct the analysis we introduce a constant which relates the positions of the cat and the mouse, at particular time points.

The behaviour of the cat and mouse can be described by the physical law in combination with initial conditions to give

$$\begin{array}{lll} \forall t, b & \bullet & \Phi(cp = -cp_0, 0) \wedge \Phi(cp = b, t) \Rightarrow b + cp_0 = cv.t; \\ \forall t, b & \bullet & \Phi(mp = -mp_0, 0) \wedge \Phi(mp = b, t) \Rightarrow b + mp_0 = mv.t. \end{array}$$

Considering the finish condition for when the mouse is safe, we infer from the second of the above physical laws the following condition for the mouse to escape

$$\forall t \quad \bullet \quad \Phi(mp = 0, t) \Rightarrow t = mp_0/mv.$$

However, for the mouse to escape, the positions of the cat and the mouse should never be equal

$$\forall t_1 \quad \bullet \quad \Phi(mp = cp, t_1) \Rightarrow t_1 = (mp_0 - cp_0)/(mv - cv).$$

For t_1 to be positive, we require that $mv - cv < 0$. Equating the original finish condition, for the mouse to be safe, in terms of the above two conditions, we obtain

$$\forall t \quad \bullet \quad \Theta(MouseSafe, t) \Leftrightarrow t = mp_0/mv \wedge \forall t_1 \bullet t_1 < t \Rightarrow t_1 \neq (mp_0 - cp_0)/(mv - cv).$$

So that the mouse is safe given

$$mp_0/mv \leq (mp_0 - cp_0)/(mv - cv)$$

or, as $mv - cv < 0$ than

$$mp_0/mv \leq cp_0/cv.$$

4.2 Controlling the Temperature of a Nuclear Reactor

The case study involves a system used to control the temperature of the coolant of a nuclear reactor by moving a control rod. The following case study is a modified version of the case study presented in [6]. The differences are as follows:

- Whereas in [6] two subsystems control the position of the rods, and a manager coordinates the actions of both subsystems, in this paper we consider only one of the subsystems, and specify its activities in terms of the requirements and constraints imposed by physics of the real-world [3].
- Instead of illustrating the ability of RTL to deal with the modelling of distributed systems, we emphasize the ability of ERTL of modelling the environment in which the computer system is to be embedded. For that the temperature of the coolant is explicitly modelled, and we assume that the rate of change of temperature is dependent on the position of the control rods.

The approach adopted for the modelling and analysis of the system, broadly matches the approach adopted in [1, 2], which aims to obtain a formal model of the interface between the computing system and its environment, thus establishing the requirements specification of the computing system to be designed.

For the presentation of this case study, we initially define the model of the system which provides the basis for conducting the analysis, and then we specify the safety requirement to be associated with the system and the respective control strategy necessary to maintain the safe behaviour. Finally, we conduct a formal analysis of the circumstances under which the safety requirement is maintained by the control strategy. Three different approaches to the formal analysis are presented, these vary on the extent to which they exploit the information available in the system model.

4.3 System Model

The model of the system is described next, in terms of the system variables and constants, physical laws and rules of operation, and the initial conditions of the system.

4.3.1 System Variables

The system variables and the constants are described in tab:2.

4.3.2 Abbreviations

Abbreviations simplify certain, more complex, syntactic entities to allow the expression of more simple formal models of the system. To be able to formally manipulate the position of the rod more easily, we define predicates $MovRod(d)$, for $d \in \{up, down\}$, such that:

Constants

Constants	Range	Comments	Unit
$KRate$	R_+	The proportionality constant between the position of a rod and the rate of change of the temperature.	$^oK/cm\ s$
$UpRate$	R_+	The rate at which the rod moves up.	cm/s
$DownRate$	R_-	The rate at which the rod moves down.	cm/s
$StablePos$	R_+	The position at which the temperature is stable, $minPos < StablePos < maxPos$.	cm
Δ	R_+	The distance moved by the rod in one step.	cm
$minPos$	R_+	The minimum position of the control rod, which is an integer multiple of Δ.	cm
$maxPos$	R_+	The maximum position of the control rod, which is an integer multiple of Δ.	cm
$minTemp$	R_+	The minimum temperature of the coolant.	^oK
$maxTemp$	R_+	The maximum temperature of the coolant.	^oK
$maxDecrease$	R_-	The maximum rate at which the temperature of the coolant can decrease.	$^oK/s$
$maxIncrease$	R_+	The maximum rate at which the temperature of the coolant can increase.	$^oK/s$
$minReq$	R_+	The minimum operating temperature at which the rod can remain stationary indefinitely.	^oK
$maxReq$	R_+	The maximum operating temperature at which the rod can remain stationary indefinitely.	^oK
$maxSafe$	R_+	The maximum safe operating temperature.	^oK
IT	R_+	Initial temperature, between $minReq$ and $maxReq$.	^oK
IP	R_+	Initial position of the rod, between $minPos$ and $maxPos$, is an integer multiple of Δ.	cm

System Variables

Variables	Range	Comments	Unit
$Temp$	R_+	Temperature of the coolant, $minTemp \leq Temp \leq maxTemp$	^oK
$\dot{Temp}$	R	The rate of change of the temperature of the coolant, $minDecrease \leq \dot{Temp} \leq maxIncrease$	$^oK/s$
$PosRod$	R_+	The vertical position of the control rod, $minPos \leq PosRod \leq maxPos$	cm
$\dot{PosRod}$	R	The rate of change of the position of the rod, $\dot{PosRod} \in \{DownRate, 0, UpRate\}$	cm/s

Table 2: System variables and constants of the nuclear reactor system

i. $MovRod(up) \Leftrightarrow \dot{PosRod} > 0$.

ii. $MovRod(down) \Leftrightarrow \dot{PosRod} < 0$.

$MovRod(up)$, $MovRod(down)$ and $\dot{PosRod} = 0$ partition *true*.

4.3.3 Physical Laws and Rules of Operation

The *physical laws* describe the continuous behaviour of the system.

i. A quantity whose rate is constant varies linearly over time:

$$\forall t \bullet \dot{s}(t) = K \Rightarrow (\forall t_1, t_2, u_1, u_2 \bullet \Phi(s = u_1, t_1) \wedge \Phi(s = u_2, t_2) \Rightarrow u_2 - u_1 = K(t_2 - t_1)).$$

ii. A quantity whose rate is linear varies quadratically over time:

$$\forall t \bullet \dot{s}(t) = K_1.t + K_2 \Rightarrow \\ (\forall t_1, t_2, u_1, u_2 \bullet \Phi(s = u_1, t_1) \wedge \Phi(s = u_2, t_2) \Rightarrow u_2 - u_1 = K_1(t_2^2 - t_1^2)/2 + K_2(t_2 - t_1)).$$

iii. A quantity whose rate is bounded in an interval is bounded by the length of the interval times the maximum rate:

$$\int_a^b f(x)\, dx \leq [b - a] \max_{x \in [a,b]} f(x).$$

The *rules of operation* describe the discrete behaviour of the system.

i. The rate of temperature increase depends (linearly) on the height of the control rod:

$$\forall t \bullet \Phi(\dot{Temp} = KRate.(PosRod - StablePos), t).$$

ii. The rod moves at a constant velocity between fixed positions at rate *UpRate* or *DownRate*:

$$\forall t \bullet \Phi(\exists n \bullet PosRod = n.\Delta \vee \dot{PosRod} = DownRate \vee \dot{PosRod} = UpRate, t).$$

iii. The minimum position of the rod is an integer multiple of Δ:

$$\exists n_{min} \bullet minPos = n_{min}.\Delta.$$

iv. The maximum position of the rod is an integer multiple of Δ:

$$\exists n_{max} \bullet maxPos = n_{max}.\Delta.$$

v. The rod takes precisely *20* units of time to move between fixed positions:

$$\forall t_1, i, d \bullet \Theta(\nearrow MovRod(d), i, t_1) \Rightarrow \Theta(\searrow MovRod(d), i + 1, t_1 + 20).$$

vi. The rod remains stationary for at least 10 units of time between consecutive movements:

$$\forall t_1, i \bullet \Theta(\searrow \exists d \bullet MovRod(d), i, t_1) \wedge \Theta(\nearrow \exists d \bullet MovRod(d), i, t_2) \Rightarrow t_2 \geq t_1 + 10.$$

vii. When the rod is at its minimum position it may only move up:

$$\forall t, d \bullet \Phi(PosRod = minPos, t) \wedge \Theta(\nearrow MovRod(d), t) \Rightarrow d = up.$$

viii. When the rod is at its maximum position it may only move down:

$$\forall t, d \bullet \Phi(PosRod = maxPos, t) \wedge \Theta(\nearrow MovRod(d), t) \Rightarrow d = down.$$

4.3.4 Initial Conditions

The *initial conditions* define the state of the system at time *0*.

i. The value of the temperature is between *minReq* and *maxReq*:

$$\Phi(Temp = IT, 0) \wedge minReq \leq IT \leq maxReq.$$

ii. The rod is stationary:

$$\Phi(Pos\dot{R}od = 0, 0).$$

iii. The rod has a position which is an integer multiple of Δ:

$$\Phi(PosRod = IP, 0) \wedge \exists n \bullet IP = n.\Delta \wedge minPos \leq IP \leq maxPos.$$

4.4 Safety Requirement

A *safety requirement* is a condition imposed on the system, if violated might breach the safe behaviour of the system. The safety requirement of the system that we consider for this case study is that the temperature of the coolant should never breach the maximum safe operating temperature. This may be expressed as:

$$\forall t \bullet \Phi(Temp \leq maxSafe, t)$$

4.5 Control Strategy

A *control strategy* is a way of maintaining safe behaviour, and is defined as a set of conditions, in terms of controllable factors, over the system.

i. If the temperature has reached the maximum required temperature *maxReq* and the rod is allowed to move, then the rod must start to move down:

$$\begin{aligned} &\forall t \bullet \Phi(Temp \geq maxReq, t) \\ &\quad \wedge \quad (\forall t' \bullet t - 10 \leq t' \leq t \Rightarrow \Phi(\neg \exists d \bullet MovRod(d), t')) \\ &\qquad\qquad \Rightarrow \Theta(\nearrow MovRod(down), t) \end{aligned}$$

ii. If the temperature has reached the minimum required temperature *minReq* and the rod is allowed to move, then the rod must start to move up: afterwards:

$$\begin{aligned} &\forall t \bullet \Phi(Temp \leq minReq, t) \\ &\quad \wedge \quad (\forall t' \bullet t - 10 \leq t' \leq t \Rightarrow \Phi(\neg \exists d \bullet MovRod(d), t')) \\ &\qquad\qquad \Rightarrow \Theta(\nearrow MovRod(up), t) \end{aligned}$$

From the system model and control strategies, we should be able to prove formally that the safety requirement is always maintained.

4.6 Formal Analysis

In the following we present the main steps of the formal analysis, which has shown that the safety requirement is always maintained for the specified system model and control strategy. After establishing the condition for maintaining the safety requirement, we have developed three distinct system models, from which we are able to determine different values for the maximum required temperature *maxReq* depending on the accuracy of the adopted system model.

The main steps of the formal analysis are aas follows:

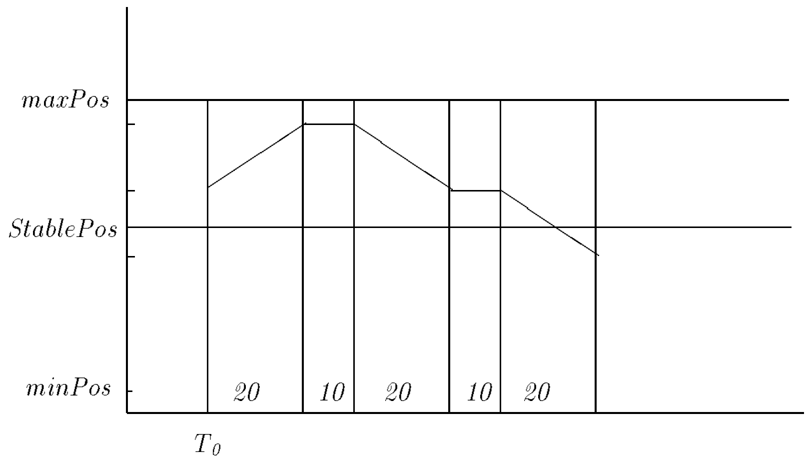

Figure 1: An instance of the control strategy in the worst case for N (the maximum number of steps required to reach *StablePos*) is 3

i. The initial step is to demonstrate that, if the supremum of the temperatures at which the rate of change of the temperature is non-positive is greater than or equal to *maxReq*, then the temperature can be bounded. This can be shown by applying the control strategy i. In fact, by analysis we may produce a bound on the number of steps it takes for $\dot{Temp}$ to become negative.

ii. In the second step, we demonstrate that the maximum number of whole steps that is required for the rate of change of temperature to become non-positive is $N = \lceil (maxPos - StablePos)/\Delta \rceil + 1$.

iii. In the third step we identify the maximum time that is required for the rate of change of temperature to become non-positive is $DownTime = 30 + 20.(maxPos - StablePos)/\Delta + 10.\lfloor (maxPos - StablePos)/\Delta \rfloor$.

iv. In the last step, we show that If the initial temperature of the coolant is *maxReq* or below, then the temperature is bounded above by $\int_{T_0}^{T_0+DownTime} \dot{Temp}\, dt + maxReq$, where $\Phi(Temp = maxReq, T_0)$.

Hence, we infer that the safety requirement follows from the system when for all T_0:

$$\int_{T_0}^{T_0+DownTime} \dot{Temp}\, dt < maxSafe - maxReq.$$

In the following, we now give the analysis of the integral in increasing detail in terms of three distinct system models. Models 1 and 2 give pessimistic estimates of the integral, being Model 2 better than that of Model 1. Model 3 gives a more accurate value. From these we may assign values to the constants parametrising the system for which the formal system is satisfiable.

4.6.1 Model 1

Our first estimate of the maximum value of the integrand, i.e., $\dot{Temp}$, over a range will be the maximum value it attains over that interval. Hence, from range of *PosRod* and rule of operation i:

$$\forall t \in [T_0, T_0 + DownTime] \bullet \Phi(\dot{Temp} \leq KRate.(maxPos - StablePos), t).$$

Therefore, from the estimate given by Model 1, the system is safe if

$$DownTime.KRate.(maxPos - StablePos) < maxSafe - maxReq.$$

4.6.2 Model 2

Our second estimate is more accurate: by subdividing the interval and estimating the value within the interval as the maximum value attained during that interval. Hence, from the rules of operation i, v and vi:

$$\forall t \in [T_0, T_0 + 20] \bullet \Phi(\dot{Temp} \leq KRate.(maxPos - StablePos), t).$$

and, given that our intervals begin with the period when the rod is stationary

$$\begin{array}{l} \forall i \in \{1, \ldots, N-1\} \bullet \\ \quad \forall t \in [T_0 + i.30 - 10, T_0 + i.30 + 20] \bullet \\ \quad\quad \Phi(\dot{Temp} \leq KRate.(maxPos - (i-1).\Delta - StablePos), t). \end{array}$$

Therefore, from the estimate given by Model 2, the system is safe if

$$\begin{array}{l} 20.KRate.(maxPos - StablePos) \\ \quad + \sum_{i=1}^{N-1} 30.KRate.(maxPos - (i-1).\Delta - StablePos) < maxSafe - maxReq. \end{array}$$

4.6.3 Model 3

The precise value of the temperature at the end of the interval is given by evaluating the integral. Hence, from the rules of operation i, v and vi:

$$\forall t \in [T_0, T_0 + 20] \bullet \Phi(\dot{Temp} = KRate.(maxPos - \Delta - StablePos + UpRate.(t - T_0)), t)$$

$$\begin{array}{l} \forall i \in \{1, \ldots, N-1\} \bullet \\ \quad \forall t \in [T_0 + 30.i - 10, T_0 + 30.i] \bullet \Phi(\dot{Temp} = KRate.(maxPos - (i-1).\Delta - StablePos), t) \end{array}$$

$$\begin{array}{l} \forall i \in \{1, \ldots, N-2\} \bullet \\ \quad \forall t \in [T_0 + 30.i, T_0 + 30.i + 20] \bullet \\ \quad\quad \Phi(\dot{Temp} = KRate.(maxPos - (i-1).\Delta - StablePos + DownRate.(t - T_0 - 30.i)), t) \end{array}$$

$$\begin{array}{l} \forall t \in [T_0 + 30.(N-1), T_0 + 30.StablePos/\Delta] \bullet \\ \quad\quad \Phi(\dot{Temp} = KRate.(maxPos - (N-2).\Delta - StablePos + DownRate.(t - T_0 - 30.(N-1))), t) \end{array}$$

so that the system is safe if

$$\begin{array}{l} 20.KRate.(maxPos - \Delta - StablePos + 10.UpRate) \\ \quad + \sum_{i=1}^{N-1} 10.KRate.(maxPos - (i-1).\Delta - StablePos) \\ \quad\quad + \sum_{i=1}^{N-2} 20.KRate.(maxPos - (i-1).\Delta - StablePos + 10.UpRate) \\ \quad\quad\quad + 20.KRate.(maxPos - (N-2).\Delta - StablePos + UpRate.(StablePos/\Delta)^2/2) \\ \quad\quad\quad\quad < maxSafe - maxReq \end{array}$$

5 Conclusions

Instead of proposing a novel formal technique for hybrid systems, in this paper we have presented an approach based on an extension of RTL which is an established formal technique. The proposed extensions are relatively straightfoward, allowing ERTL to be used in the modelling and analysis of hybrid systems, particularly for the stage of requirements analysis.

ERTL is in its early stage of development, and a richer proof theory than that provided by the axioms is being sought. This will, in the usual way, be built upon the identification of useful predicates within the logic, and their interrelation. Initial experimentation with JAPE (Just Another Proof Editor) has shown the difficulty of proof from first principles, so such extensions are to be considered necessary for a usable formalism.

Acknowledgments

The authors would like to thank Amer Saeed for his comments and criticism, and would like to acknowledge the financial support of the EPSRC (UK) SCHEMA Project.

References

[1] T. Anderson, R. de Lemos, J. Fitzgerald and A. Saeed. "On Formal Support for Industrial-Scale Requirements Analysis". *Hybrid Systems*. Eds. R. L. Grossman et al. Lectures Notes in Computer Science 736. Springer-Verlag. 1993. pp. 426-451.

[2] R. de Lemos, A. Saeed, T. Anderson. *"Requirements Analysis for Safety-Critical Systems: A Chemical Batch Processing Example"*. Department of Computing Science TR 469. University of Newcastle upon Tyne, UK. December 1993.

[3] R. de Lemos, A. Saeed, T. Anderson. "Formal Techniques for Requirements Analysis for Safety-Critical Systems". *Mathematics of Dependable Systems*. C. Mitchell and V. Stavridou (Eds.). Oxford University Press. 1995. pp. 63-95. (to appear)

[4] K. L. Heninger. "Specifying Software Requirements for Complex Systems: New Techniques and their Applications". *IEEE Transactions on Software Engineering Vol. SE-6 (1)*. January 1980. pp 2-13.

[5] F. Jahanian, A. Mok. "Safety Analysis of Timing Properties in Real-Time Systems". *IEEE Transactions on Software Engineering Vol. SE-12 (9)*. September 1986. pp 890-904.

[6] F. Jahanian, A. K. Mok, D. A. Stuart. *"Formal Specifications of Real-Time Systems"*. Department of Computer Science TR-88-25. University of Texas at Austin, TX. June 1988.

[7] O. Maler, Z. Manna, A Pnuelli. "From Timed to Hybrid Systems". *Real-Time: Theory in Practice*. Eds. J. W. de Bakker, C. Huizing, and G. Rozenberg. Lectures Notes in Computer Science 600. Springer-Verlag. 1992