Qualitative formal method for requirements specification and validation of hybrid real-time safety systems

J.-S.Lee and S.-D.Cha

Abstract: The difficulties that engineers have in understanding and applying the quantitative methods in an abstract requirements phase are major obstructions in using formal methods for hybrid real-time safety systems. While formal methods technology in safety-critical systems can help increase confidence of software, the difficulty and complexity in using them can cause another hazard. The authors have proposed a framework for requirements engineering, called the qualitative formal method (QFM) for the specification and validation of hybrid real-time safety systems. The QFM emphasises the idea of a causal and qualitative reasoning in formal methods to reduce the difficulty of specifying and validating the software requirements of hybrid safety systems. They use the qualitative formal languages, Compositional Modelling Language, and Causal Functional Representation Language in particular, to specify hybrid system dynamics and the required behaviour, respectively. The system behaviour has been simulated by the Device Modelling Environment, and validated against the required behaviour. Using the Shutdown System 2 of Wolsong nuclear power plants as a realistic example, they demonstrate the effectiveness of their approach.

1 Introduction

Hybrid systems must deal with continuous as well as discrete and instantaneous state changes. Examples of hybrid systems include process control systems installed in nuclear power plants; satellites; and automobiles. Since a hybrid system is often safety-critical in nature, and must fulfill real-time requirements, we refer to such a system as a hybrid real-time safety system (HRTSS). When specifying requirements (S_p) for HRTSS or proving its safety properties, one must consider the behaviour of both plant (P) and controller (C) [1]. That is, the truth of the following proposition

$$P \text{ and } C \rightarrow S_p \tag{1}$$

must be demonstrated. Finite state automata, can accurately describe the discrete behaviour of a digital controller, but plant behaviour, analogue in nature, is usually modelled in differential equations. It is the interaction of continuous and discrete state changes that make requirements specification and analysis of HRTSS challenging.

Several formal methods have been proposed for specifying requirements for HRTSS, and they vary significantly in their expressiveness, analysability, and levels of rigour. Unfortunately, there are a couple of limitations common to them all. First, requirements for HRTSS are often available only in qualitative and abstract forms during early phases of requirements engineering, and there is a need for formal yet abstract formalism to allow effective communication and review among engineers who possess different technical backgrounds. There are several reported cases of mishaps [2] caused by misunderstandings and misinterpretations. Second, current approaches do not provide a formal basis for conducting systematic safety analysis. Fault tree analysis is the most commonly employed safety analysis technique, and its industrial practice depends heavily on the technical expertise of human analysts and is often ad hoc.

In this paper, we argue that the qualitative formal methods (QFM) is effective in reducing the difficulties when developing requirements for HRTSS. The QFM is based on causal reasoning and qualitative approximation theories of qualitative physics. In particular, Compositional Modelling Language (CML) [3] and Causal Functional Representation Language (CFRL) [4] are used to specify system dynamics and required behaviour, respectively. Device Modelling Environment (DME) [5], used for behavioral simulation, provides a basis for conducting formal safety analysis. We illustrate an application of our approach using partial requirements for Wolsong SDS2, which is an emergency shutdown system, currently in service, for a Korean nuclear power plant.

2 Related works

Formal methods proposed for HRTSS attempt to describe nonlinear physical phenomena using quantitative differential equations and time functions [1, 6–10]. While these are

IEE Proceedings online no. 20000460

DOI: 10.1049/ip-sen:20000460

Paper received 22nd October 1999

J.-S. Lee is with the MMIS Team, Korea Atomic Energy Research Institute, 150 Duckjin-dong, Yusong-ku, Taejon 305-600, Korea
E-mail: jslee@nanum.kaeri.re.kr

S.-D. Cha is with the CS Divison, EECS Department, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-ku, Taejon 305-701, Korea
E-mail: cha@salmosa.kaist.ac.kr

effective in formal verification of the specification in part, they make the elicitation and specification difficult. The rigorous and quantitative formal languages are not appropriate for understanding the problem of HRTSS in a conceptual requirements engineering phase.

Recently, to overcome the difficulty and the computational complexity problems many researchers have been trying to approximate the quantitative formal methods. For example, Henzinger and Ho proposed the model checking and abstract interpretation strategies for the hybrid automata [11, 12]; and Puri and Varaiya suggested the verification method for the hybrid system using abstraction [13]. The duration calculus that requires the quantitative specification and analysis [2, 8] is approximated into the probabilistic duration calculus [14].

The discrete approximation was the most successful technique in the practical industry. The four-variable approach [15], one of the software cost reduction (SCR) methods [16–19], have been used to state requirements for hybrid systems such as the Wolsong SDS2 and the Darlington shutdown system. Requirements are stated as mathematical relations involving the monitored (M), controlled (C), input (I), and output (O) variables as shown in Fig. 1.

While NAT describes any constraints on behaviour, such as those governed by physical laws, REQ defines the additional black-box constraints to be enforced by the controller. There are three additional relations, IN, OUT, and SOF, relevant to the controller specification [15], and proposition 1 can be rewritten as follows:

$$\text{NAT(M) and OUT(SOF(IN(M))} \rightarrow \text{REQ(M)} \quad (2)$$

Parnas [18] and Heitmeyer [16] successfully used a modified Four-variable approach in which discrete approximation is employed where tabular and formal notations are used to document the REQ relations. The same approach has been used to document the Wolsong SDS2 requirements. Recently, Heitmeyer and Bharadwaj [20] tried to perform model checking of hybrid systems using timed automata and PVS [21]. They, however, need human guidance in model checking for reasoning about nondeterministic automata.

These approaches are primarily intended to reduce a computational complexity by approximating the continuous physical phenomena mathematically. The QFM, however, tries to approximate the physical phenomena of HRTSS qualitatively and physically, by using the idea of qualitative reasoning based on the physical knowledge of HRTSS, in order to reduce not only the computational complexity but also the difficulty.

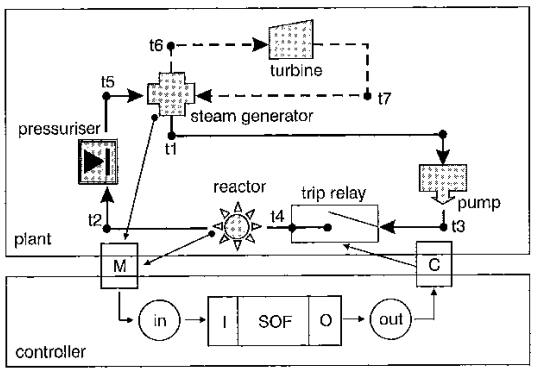

Fig. 1 *Shutdown system 2 (SDS2)*
—— primary thermal connection
---- secondary thermal connection
—— signal connection

Our work is motivated by Moffet's causal logic-based approach [22]. Causal reasoning is an effective and natural approach when documenting and analysing behaviour for complex systems. A cognitive approach to system engineering tasks has been the subject of extensive research, as documented in [23]. Software engineering researchers have recently begun to investigate cognitive aspects of requirements engineering, and the intent specification approach, proposed by Leveson [24], is such an example. The QFM is an approach where we try to combine advantages offered by formal specification and qualitative reasoning techniques in order to allow systematic software safety analysis in the early phases of requirements engineering for HRTSS.

3 HRTSS and qualitative physics

3.1 Wolsong SDS2 in nuclear power plants

Wolsong SDS2 is designed to rapidly terminate the nuclear chain reaction and keep the reactor in a safe state upon the detection of various unsafe operating conditions. In order to provide fault-tolerance capability, the trip decision for SDS2 is made using general coincidence two-out-of-three channel voting logic [19]. Although there are several trip parameters defined for SDS2, we use a steam generator low level (SGLL) trip as an example whose behaviour is informally described as follows:

SDS2 operates by de-energising relays, after a trip condition is detected by the trip logic to initiate the opening of quick opening valves. This causes high pressure helium to push neutron absorbing liquid poison from the tank into the core via injection lines in the primary coolant. Steam generators function by transferring energy from the primary coolant to the turbines where it is converted to electricity. As such, they serve as a heat sink for the reactor core. For proper performance, the steam generator water level must be held within predetermined operating bounds. Too low a level may be an indication of inadequate heat removal from the core, while too high a level may degrade the steam quality.

In this paper, we consider a trip controller for a SGLL trip of only one channel, shown in Fig. 2. This figure shows a part of the Four-variable model specifications of Wolsong SDS2 [19], and the generation of a trip signal upon detection of a steam generator level that is too low.

The monitored variables /SGLevel-1/ to /SGLevel-4/ in Fig. 2 are defined as a real number between −2.67 to +3.5 m. There are physically four level measurements of steam generator levels per channel in Wolsong nuclear power plants. The monitored variable /LogPower/, Ion chamber log power, is defined as a real number between 0.00001% and 150%FP. Here, %FP represents the current level of operation in terms of percentage of the full reactor power. The input variable <CAvgPower>, compensated average power, is calculated from three neutron flux detector signals in another module. There are two controlled variables, //SGLevelTrip// of trip signal and //SGLCondOut// of trip condition out signal.

In order to meet the system design requirements, safety engineers calculate the SGLL trip setpoints for Wolsong nuclear power plants from the understanding of steam generator water level dynamics. The SGLL trip setpoint, LSP, is provided in Fig. 3, as a function of reactor power, CAvgPower.

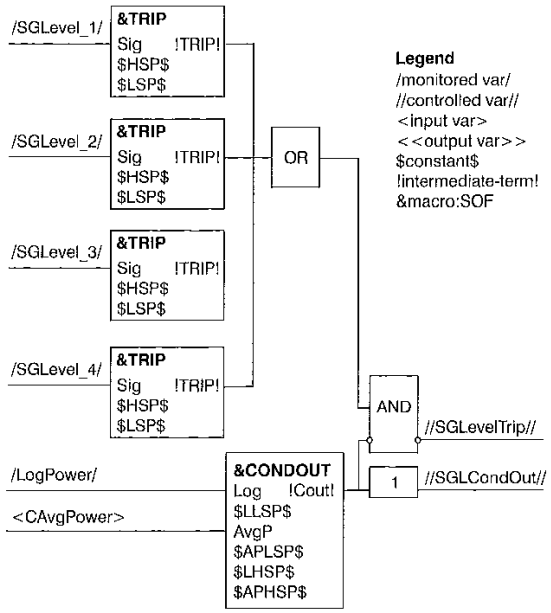

Fig. 2 *Logic of steam generator low level trip controller*

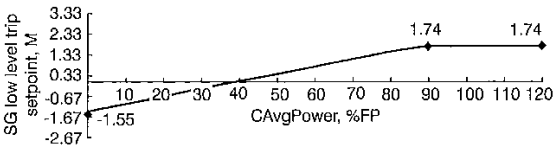

Fig. 3 *SGLL trip setpoints*

3.2 Qualitative physics

Qualitative physics is a research topic in artificial intelligence, and provides theories on how to conduct qualitative reasoning on the behaviour of natural or man-made systems whose behaviour is governed by the laws of physics. Qualitative reasoning uses information such as relative magnitudes, and the directions of change in variable values, as opposed to precise values. Since data used in this reasoning process is imprecise, the specification is numerically imprecise as well.

However, as illustrated in our paper, one need not have, especially if such information is unavailable, numerically precise and detailed information to perform meaningful analysis. While periodic polling and discrete approximation are the most commonly employed approaches, the QFM approach is based on the observation that semantically significant and correct information, though imprecise in value, facilitates effective behavioural and safety analysis when combined with domain-specific theories on the plant. For example, if we were to use temperature changes as an example, the symbolic landmark states (e.g. increasing, decreasing, steady, freezing, or boiling points) can be used to specify how the discrete controller must react to such environmental changes (see Fig. 4). The key advantage of the QFM approach is the abstract but formal semantics the model provides.

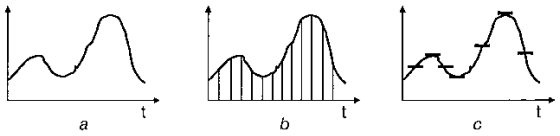

Fig. 4 *QFM model versus discrete model of continuous plant*
a Plant behaviour
b Discrete model
c QFM model

In Fig. 4, the plant behaviour of HRTSS is governed by physical laws, and represented by nonlinear and continuous changes of states. When modelling the behaviour by a discrete method, the number line is divided mathematically. The physical values are sampled at each point that might be divided by one second, fuzzy values, or probabilistic values. The QFM, however, divides the number line by a physical landmark. The reason for QFM being able to divide the nonlinear continuous behaviour by a physical meaning is that we can utilise the domain knowledge of the target plant. Therefore, a qualitative approximation by landmarks is more abstract than other approximation approaches, and yet preserves all the important information of the continuous physical environment.

The scale of landmark can be optimised according to the required level of abstraction. There must be an engineering judgement to decide the optimal scale in a trade-off between computational complexity and precision. In a requirements engineering phase for HRTSS, because the detail values for plant model, controller model, and requirements are not available, the conceptual QFM can provide a solution for catching a bird's-eye view of the models in an abstract yet formal manner.

3.3 Compositional modelling language (CML)

Compositional Modelling Language (CML) [3] is a declarative model-building language for logically specifying the symbolic and mathematical properties of the structure and behaviour of physical systems qualitatively. CML can specify an abstract behaviour of the plant by using qualitative differential equations and causality, instead of differential equations and time function. The indicative physical situation is modelled in a general purpose domain theory as a collection of model fragments, each of which represents a physical object or a conceptually distinct physical phenomenon, such as a particular aspect of component behaviour or a physical process. A model fragment representing a phenomenon specifies a set of conditions under which the phenomenon occurs, and a set of consequences of the phenomenon. The conditions specify a set of instances of object classes that must exist (called participants) and a set of relations (called conditions) that must hold among those objects and their attributes for the phenomenon to occur. The consequences hold when an instance of the model fragment is active. Model fragment consequences are typically qualitative differential equations that describe the behaviour of the entities that satisfy the conditions.

Fig. 5 shows a model fragment example, shown in Hoare logic style, for a physical phenomenon of a reactor in

Reactor – thermal – energy – generating (RT)
Participant: (Reactor ?r)
Condition: (In-closed-thermal-loop ?r)
Consequence: (Thermal-energy-thru-hot-leg ?r) = M_4 (d(CAvgPower ?r)/dt)

Fig. 5 *Model fragment for a physical phenomenon in SDS*

Wolsong nuclear power plants. The precondition (In-closed-thermal-loop) describes a structural condition where the reactor is under a normal operation state in the primary thermal connection shown in Fig. 1. CML was designed to model time-varying physical systems, such as the movement of a mechanical device or the process of a chemical reaction. In engineering models, the properties and state of such systems are described by variables, parameters, coefficients, and constants. In CML the term 'quantity' encompasses these notions. Thermal-energy-thru-hot-leg is a quantity type of the variable?r, and defined as a separate quantity function. When the precondition is satisfied, as a consequence, the model fragment indicates that the thermal energy through the hot leg of the reactor will monotonically increase the function of the derivative of reactor power as denoted by M. (Readers are referred to [4] for detailed discussion on the syntax and semantics of CML.) In qualitative physics, a piece of physical phenomena occurring in the reactor can be specified as a model fragment of CML qualitatively. In an analytic approach, however, the same phenomena must be represented quantitatively by a set of differential equations and time functions that are based on the thermodynamic and neutron dynamic theories.

3.4 Causal functional representation language (CFRL)

Causal Functional Representation Language (CFRL) [4] is a language for specifying a required function of a device. The language allows engineers to explicitly describe the physical context in which the requirements are to be achieved, the structural characteristics of the system that are assumed in the functional requirements, and the causal sequence of events that must occur for the function to be achieved.

CFRL has the same behaviour-based semantics as the behaviour-of the plant, because the required behaviour for a hybrid controller is stated in CFRL through a qualitative approximation. Therefore, one can formally verify the required behaviour of the controller against that of the plant. The required function F for HRTSS can be specified with four-tuple Ef, Df, Cf, and Gf, where:

- Ef is the name of a required function that F elaborates.
- Df describes the indicative system of which F is a required function.
- Cf describes the context in which the function is performed.
- Gf describes the functional goal to be achieved.

The indicative system description of Df consists of three parts: device, components, and conditions. Device and its component is described as a pair (var, type) shown further on is the paper, where the var is the symbol to be used in the description of the function. The type is the class of the var. The Conditions typically specify aspects of the system structure that are assumed in the description of the required function. The notion of a system function assumes some physical context in which the system is placed, and Cf is a specification of such context. The description of the goal for the required safety properties-Gf-is represented as an expression consisting of a causal process description (CPD), quantifiers, and Boolean connectives. There are two quantifiers, ALWAYS and SOMETIMES. Connectives are AND, OR, IMPLIES, and NOT. CPD, a directed graph shown in Fig. 6, is essentially a qualitative state machine in which each node describes a condition on a state and each arc describes temporal and causal relations between states.

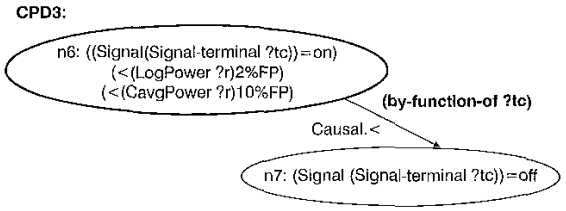

Fig. 6 *Causal process description (CPD)*

The condition in the node is a logical sentence about the state of the world at some time using the variables defined in the Df and Cf portions of the function description. One or more nodes in each CPD are distinguished as the initial node(s). The initial nodes are indicated with a thick oval. In Fig. 6, CPD3 means that when the signal of the trip controller is 'on' if the reactor is in low power range, the signal should be conditioned out. There is a causal relation between n6 and n7, and n6 precedes n7 in time.

3.5 Device modelling environment (DME)

Device modelling environment (DME) is a software tool, implemented at Stanford [5], that produces state trajectory of the system through behavioural simulation. DME is capable of modelling both continuous and discontinuous physical phenomena. The state trajectory can be considered as the behaviour of a qualitative hybrid automata (QHA) rather than the quantitative hybrid automata. The main difference between a hybrid automata and a QHA is that while the state in a hybrid automata is represented by an invariant and differential equations, the state in a QHA is described by the conditions of landmarks and the qualitative differential equations. Because the CPD describes the desired partial QHA, and the state trajectory is the indicative total QHA for the plant and controller, we can validate proposition-1-by comparing the QHAs.

State trajectory is generated as follows (see Fig. 7). Given a description of the functional structure of a device in terms of components and their structural relations, the system generates a model of how the device would work. To perform this first step, the system uses its engineering knowledge base to identify applicable physical laws and processes. The resulting process model contains descriptions of what physical processes could possibly occur and how components could function, along with sets of constraints among quantities.

This process model is then transformed into a set of governing equations that can be used by a simulator to predict device behaviour. From this collection of identified laws, processes, objects, component behaviour, constraints, material properties, and other associated knowledge, the

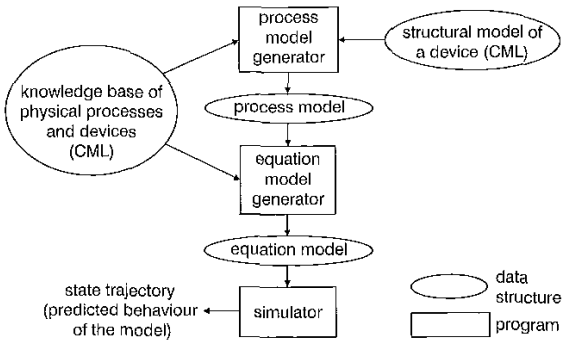

Fig. 7 *DME simulation scheme*

system generates mathematical equations. The simulation phase uses the equation model to analyse and predict the behaviour of HRTSS, and produces the state trajectory of the behaviour.

4 Qualitative requirements engineering: Framework and application

Requirements engineering can be decomposed into the interactive activities of requirements elicitation, specification, and validation. A framework for requirements engineering processes to provide easily understandable and verifiable formalism and to provide early protection against hazards, is of great importance. An engineer who specifies and validates HRTSS must well understand the properties of the controlled physical processes. The challenge, however, is how to define the requirements engineering framework that allows designers to specify and validate such understanding without cognitive burdens. The QFM provides a streamlined approach for this understanding, which relies on the ideas from the causal qualitative reasoning of artificial intelligence.

The QFM-based requirements engineering framework, shown in Fig. 8, consists of the following six steps:

QFM 1: Elicit a goal-based specification from system specification by a natural language, using causal reasoning [22] and the means-ends abstraction from the cognitive systems engineering [23, 24].
QFM 2: Specify qualitatively the representational models describing the indicative plant, controller, and their interface, using CML.
QFM 3: Specify the requirements describing the desired behaviour by using CFRL.
QFM 4: Conduct a qualitative simulation of the hybrid system to capture the physical causality using DME [5].
QFM 5: Validate the total indicative specifications of the plant and controller, against the requirements.
QFM 6: Conduct a fault tree analysis on the safety requirements using causality information produced in QFM 4.

QFM 1 is primarily concerned with requirements elicitation, and it is outside the scope of this paper. We assume the results of QFM1 could be the Wolsong SDS2 specifications[19], partially shown in Fig. 2. The requirements had been informally elicited from a system specification, and specified by the formal method of the Four-variable model. A systematic elicitation method is one of the important research areas in requirements engineering [22–24]. We are studying a goal-based causal elicitation method for requirements engineering when developing HRTSS software. This paper discusses the QFM 2 through 5 in detail through an application on the SGLL trip requirements for Wolsong SDS2. The goal of QFM 6 is to conduct formal safety analysis, such as fault tree analysis, based on the state trajectory information produced through qualitative behavioural simulation; and it remains an important research area.

4.1 QFM 2: Indicative specification of HRTSS by CML

Plants (P) are usually a nonlinear continuous system with disturbances, and modelled by differential equations. The software-based controllers (C) are discrete and so usually modelled by state machines. In the QFM 2 step, however, the plant and controller behaviour, P and C, are specified by a qualitative formal language, CML. CML can qualitatively specify the physical facts of the continuous plant behaviour and the discrete controller behaviour. The structural information of SDS2 as shown in Fig. 1 is described as a collection of model fragments representing components and their connections. These model fragments represents the static aspect of the situation, and they are always active. In addition, there are model fragments representing various behavioural aspects of the components. Figs. 9 and 10 show the set of model fragments specifying the indicative plant and controller models of the shutdown system for Wolsong nuclear power plants. Their conditions are indicated as Pm and include type restrictions on the participants. Their consequences are indicated as Cm for model fragments representing continuous phenomena and as Dm for model fragments representing discrete events.

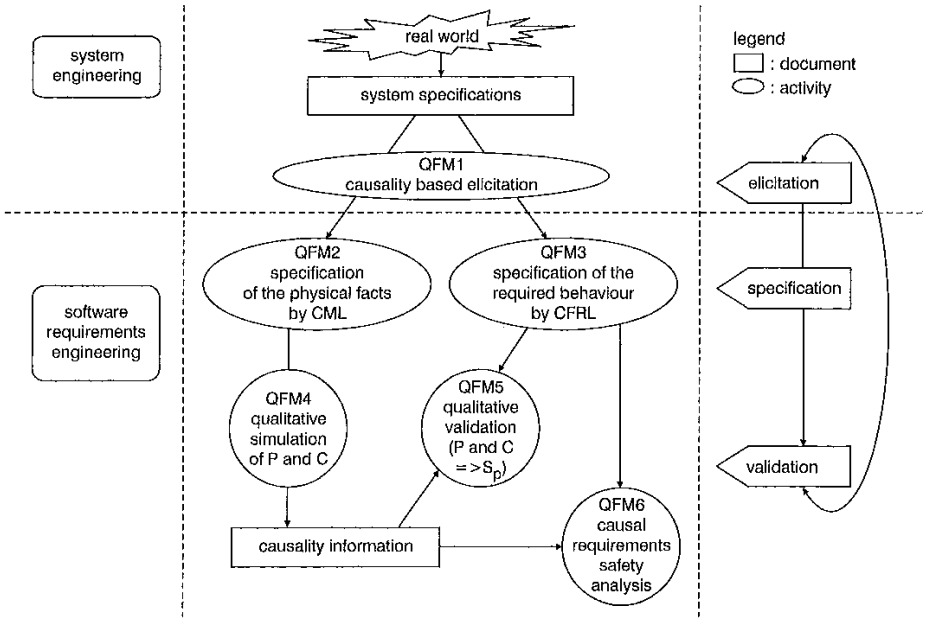

Fig. 8 *QFM process for HRTSS*

Behaviour of Steam Generator
Steam Generator (SG)
Pm : (Steam-generator ?s)
Cm : (Thermal-energy-steam-out ?s) = M_{+}(Thermal-energy-primary-in ?s)
∧ (Heat-transfer-rate ?s) = M_{+}(Level ?s)
∧ (Level ?s) = IL − SO + FI + M_{+} (Thermal-energy-primary-in ?s)
SG-operating-in-low-reactor-power (SL)
Pm : (Steam-generator) ?s) ∧ (Reactor ?r) ∧ (Level ?s) > LSP ∧ (CAvgPower ?r) < 10%FP
Cm : (Thermal-energy-feed-in ?s) = 0/ * no recirculation * /
SG-operating-in-normal-reactor-power (SN)
Pm : (Steam-generator ?s) ∧ (Reactor ? r) ∧ (Level ?s) > LSP
∧ 10%FP =< (CAvgPower ?r) < 90%FP
Cm : (Heat-transfer-rate ?s) = 35%
SG-operating-in-high-reactor-power (SH)
Pm : (Steam-generator ?s) ∧ (Reactor ? r) ∧ (Level ?s) > LSP ∧ (CAvgPower ?r) >= 90%FP
Cm : (Thermal-energy-steam-out ?s) = $Max-S$

Behaviour of secondary loop
Thermal-load (TL)
Pm : (Steam-generator ?s)
Cm : (Thermal-energy-feed-in ?s) = (Thermal-energy-steam-out ?s) − $Work$ − $Tloss2$

Behaviour of reactor
Reactor-thermal-energy-generating (RT)
Pm : (Reactor? r) ∧ (In-closed-thermal-loop ?r)
Cm : (Thermal-energy-hot-leg ?r) = M_{+}(d(CAvgPower ?r)/dt) − $Tloss1$ + (Thermal-energy-cold-leg ?r)
Reactor-in-open-loop (RO)
Pm : (Reactor ?r) ∧ ¬ (In-closed-thermal-loop ?r)
Cm : (Thermal-energy-hot-leg ?r) = 0

Behaviour of trip relay
Relay-closed (TC)
Pm : (Steam-generator ?s) ∧ (Reactor ? r) ∧ (Trip-relay ? r) ∧ (closed-p ?t)
Cm : (Thermal-energy-hot-leg ?r) + (Thermal-energy-primary-in ?s) = 0
Relay-opened (TO)
Pm : (Reactor ? r) ∧ (Trip-relay ?t) ∧ (Open-p ?t)
Cm : (Thermal-energy-hot-leg ?r) = 0
Relay-closed (CL)
Pm : (Trip-relay ?t) ∧ (Open-p ?t) ∧ ¬ (Signal-on (Sig-terminal ?t))
Dm : (closed-p ?t)
Relay-opening (OP)
Pm : (Trip-relay ?t) ∧ (closed-p ?t) ∧ (Signal-on (Sig-terminal ?t))
Dm : (Open-p ?t)

Fig. 9 *Model fragments for behavior of SDS2*

Behaviour of Trip Controller
Trip-signal-on (TN)
Pm : (Reactor ? r) ∧ (Trip-controler ?tc) ∧ (Signal (Signal-terminal ?tc) = off
∧ (LogPower ?r) >= 2%FP ∧ (CAvgPower ?r) >= 10%FP
∧ (Level ?s) =< LSP
Dm : (Signal (Signal-terminal ?tc)) = on
Trip-condition-out (TT)
Pm : (Reactor ? r) ∧ (Trip-controler ?tc) ∧ (Signal (Signal-teminal ?tc)) = on
∧ (LogPower ?r) < 2%FP ∧ (CAvgPower ?r) < 10%FP
Dm : (Signal (Signal-terminal ?tc)) = off

Fig. 10 *Model fragments of trip controller*

The first model fragment definition (SG) in Fig. 9 defines the behaviour of a steam generator. The thermal energy of the generated steam increases monotonically according to the thermal energy of the primary coolant. The heat transfer rate of the steam generator is proportional to the level, and the level is changed by the thermal energy of the primary coolant and other constants (IL, SO, FI) whose definitions are given in a later figure. There is no recirculation until the reactor power goes up to 10 %FP (SL). The steam generator has a normal heat transfer rate between 10 and 90 %FP (SN), and the heat transfer rate is saturated when the power becomes greater than 90 %FP (SH). The quantity space of steam generator operating region can be represented as {0,10,90,120}. Here, the 10 and 90 are the landmarks that represent the key changing points of the physical states.

For each participant, the name and its type must be given. ?s is the name by which the participant will be referred to in the definition. Model fragments can have quantitative attributes. The range of values the attribute can take, and the time-derivative of the attribute, may be specified (see Fig. 10). Model fragments can also have nonquantitative attributes, such as Colour-p, Relay-open-p, Sun-shining-p, etc.

It should be noted that CML specification can be developed incrementally as relevant information becomes available.

4.2 QFM 3: Specification of required behaviour by CFRL

CFRL can be used as a language for specifying a required HRTSS, behaviour, and defines its semantics in terms of

the type of behaviour representation used in model-based, qualitative simulation. To be able to determine whether the required behaviour is actually accomplished by an observed behaviour of HRTSS, the representation of the requirements must specify conditions that can be evaluated against the behaviour. Without clear semantics given to a representation of requirements in terms of behaviour, it would be impossible to evaluate proposition 1 based on its predicted behaviour (P and C) and the required behaviour (S_p).

Figs. 11 and 12 show the required behaviour for the SGLL trip controller of SDS2. The goal-based specification of the requirements shown Fig. 11 can be refined into Fig. 12 by using causal process description (CPD) of CFRL, which can represent the causal and temporal relationship.

The goal Gf of F1 means that when the reactor generates power as a heat-source and the primary loop is closed, the steam generator must act as an appropriate heat-sink. Df of F11 describes the physical connections and conditions of the plant. Cf of F11 describes the context in which the hybrid system is to function, and inherit the Cf of F1.

When the reactor starts to generate power, the reactivity of the reactor increases, the refined goal Gf of F11 specifies the desired behaviour of the trip controller, i.e. shutdown and condition out. A CPD can be considered as an abstract specification of a desired trajectory. Unlike an indicative trajectory, it does not specify every state or everything known about each state. It only specifies some of the facts that must be true during the course of the trajectory and partial temporal/causal orderings among those facts. In other words, ordering is total for states in an indicative trajectory because the trajectory is a linear sequence of states, while the order is partial for states in a CPD, shown in Fig. 13.

4.3 QFM 4: Simulation of indicative models by DME

Given the CML representations of the indicative plant and controller models, the DME of Fig. 7 automatically simulates the behaviour of the indicative models as follows: to generate the qualitative equation model, give a causal ordering between variables from equations, and produce the state trajectory of the models.

The qualitative equations can be produced from CML model fragments using knowledge about the basic physical laws, e.g., the energy conservation law. Fig. 14 shows the physical variables and their qualitative equations. The right hand column indicates the name of model fragments in Fig. 9, which directly contribute to produce the qualitative equations. The 'junction' means that the equations are generated from the model fragments for the thermal connection structure of SDS2. These equations are just a form of the engineer's understanding about a physical situation in a requirements engineering phase.

DME simulator identifies the causal ordering between the physical variables from the equations according to the causal ordering theory [25], as shown in Fig. 15. The theory of causal ordering provides a technique for inferring the causal relations among variables in a set of functional relations. Causal ordering reflects people's intuitive notion of causal dependency relations among variables in a system. Establishing a causal ordering involves finding

```
If: F1
Df: Device :(?nsss Nuclear-steam-supply-system)
Cf: Objects :
        (?pzr Pressuriser)
        (?pump Charging-pump)
        (?t-load Thermal-load-of-secondary-loop-with-Turbine)
    Conditions :
        (Thermal-connected (Thermal-out ?nsss) (Thermal-in ?t-load))
        (Thermal-connected (Thermal-in ?nsss) (Thermal-out ?t-load))
Gf: (ALWAYS (AND
        (Generating-power ?r) (Closed-p ?t) (Heat-sinked ?t-load))
```

Fig. 11 *Requirements of SGLL trip controller*

```
Ef: F11
Df:
    Components :
      (?t Trip-relay)
      (?r Reactor)
      (?s Steam-generator)
      (?tc Trip-controller)
    Conditions :
      (Thermally-connected (Hot-leg ?r) (In-terminal ?pzr))
      (Electrically-connected (Flux-terminal ?r) (Flux-sensing-terminal ?tc))
      (Thermally-connected (Out-terminal ?pzr) (Coolant-in ?s))
      (Electrically-connected (Level-terminal ?s) (Level-sensing-terminal ?tc))
      (Thermally-connected (Inlet ?pump) (Coolant-out ?s))
      (Thermally-connected (Outlet ?pump) (Cold-leg ?r))
      (Electrically-connected (Signal-terminal ?t) (Signal-terminal ?tc))
Cf: Objects : nil
    Conditions : nil
Gf: (ALWAYS  (AND
      (AND (Start-up ?r) CPD1)
      (IMPLIES (AND (≥ (LogPower ?r) 2%FP)(≥ (CAvgPower ?r) 10%FP)
                (≥ (Level ?s) $LSP$) (Closed-p ?t)) CPD2)
            (IMPLIES (AND (< (LogPower ?r) 2%FP)(< (CAvgPower ?r) 10%FP)
                        (Closed-p ?t) CPD3)))
```

Fig. 12 *Refined requirements of SGLL trip controller*

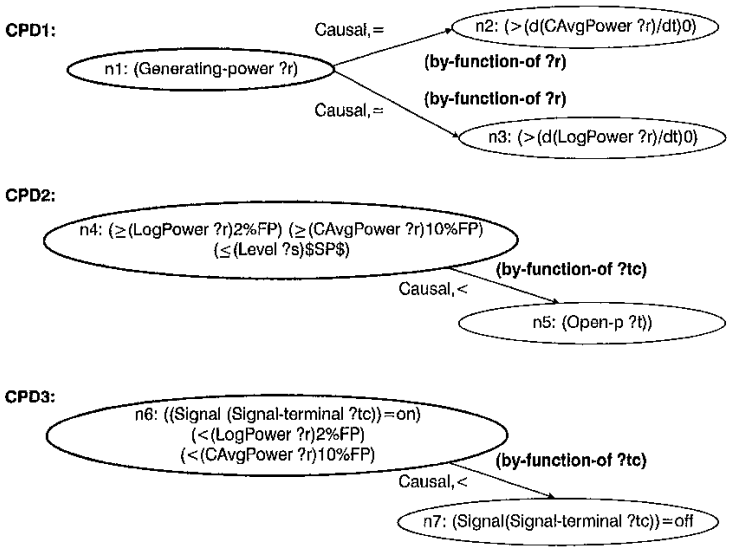

Fig. 13 *CPDs of required behaviour F11*

E1:	$T2 = M_{+}(d(CAP)/dt) + T1 - \$Tloss1\$$	RT
E2:	$T2 + T3 = 0$	junction
E2:	$T3 + T4 = 0$	TC
E4:	$T4 + T5 = 0$	junction
E5:	$T1 + T5 + T6 + T7 = 0$	SN
E5':	$T7 = 0$	SL
E5'':	$T6 = \$Max - S\$$	SH
E6:	$T7 = T6 - \$Work\$ - \$Tloss2\$$	TL
E7:	$T6 = M_{+}(T5)$	SG
E8:	$HTR = M_{+}(Level)$	SG
E9:	$HTR = 35$	SN
E10:	$Level = \$IL\$ - \$SO\$ + \$FI\$ + M_{+}(T5)$	SG

Ti: thermal energy at terminal t1 in Fig. 1
CAP: compensated average reactor power
Level: steam generator water level
HTR: heat transfer rate of steam generator
$Tloss1$ thermal loss into atmosphere in primary connection
$Tloss 2$ thermal loss into atmosphere in secondary connection
$Max-S$ thermal energy of maximum steam
$Work$: kinetic energy by turbine
IL : initial steam generator water level
SO : quantity of steam out
FI : quantity of feed water in

Fig. 14 *Qualitative equations and physical variables*

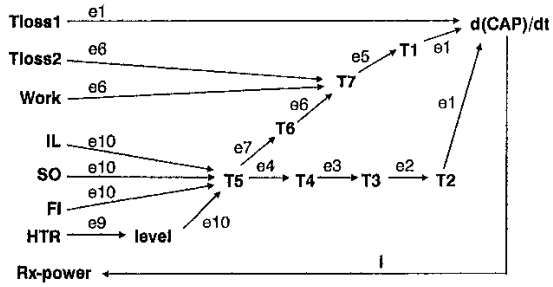

Fig. 15 *Causal ordering of variables*

subsets of variables whose values can be computed independently of the remaining variables, and then using those values to reduce the structure to a smaller set of equations containing only the remaining variables. In Fig. 15, the arrows indicate the causal order between variables, the equation numbers on the arrows indicate the equations that contribute to make the order, and the 'i' indicates a causal relationship by an integral function.

We used DME to produce the state trajectory, shown in Fig. 16, of the indicative models for SDS2 by a qualitative simulation based on the causal ordering. The variable values are shown with their magnitude and the sign of their derivative. In a state where their derivative is undefined, the sign is shown as 'x'. The left hand column of the table shows the set of active models in each state. The set of all active model fragments in each state is actually much larger, but since most of them represent components, terminals, junctions, etc. and are active throughout the simulation, we show only the ones that change their activation status at some point. A model fragment that becomes activated in a given state is shown in bold. A model fragment with an underbar indicates that it becomes deactivated in the given state.

In this simulation, we assume that the thermal energy in the primary thermal connection (T1 and T5) is measured by temperature, the steam out thermal energy (T6) is calculated by the electrical energy produced by the steam, the feed water thermal energy (T7) by the flow rate and temperature, and the unit of compensated average power (CAP) of the reactor is a percentage of full power. In order to simplify the simulation we also assume that the low water level setpoint of the steam generator is constant (30%).

As the initial state of the simulation, we assumed that the water level of the steam generator is 50%, the reactor is in heat-up mode, the trip relay is closed, and that the reactor power in between 0 and 120% FP. In the initial state S0, the thermal energy T1 and T2 increase at t1 and t2, respectively, in Fig. 1. When the primary coolant temperature reaches 292 (S1), the reactor enters the start-up mode, and then the secondary thermal connection starts to produce steam (S2). At this point, the feed water flow increases, and the water level of the steam generator goes under the set point because of a fluctuation by the swell and shrink effects, and the trip signal occurs (S4). However, because the reactor power (CAP) at this state is less than 10%, the trip signal is conditioned out (S5). The reactor power increases continuously during the power operation mode. When CAP reaches 10% FP, the steam generator enters the normal operating state (S6), and when it reaches 90% FP, the steam generator enters the high operating state (S8). From the high operating state, the thermal energy of the

Active models	state	Reactor	Relay	Signal	T1	T5	T6	T7	Level	CAP
SL, TC	**S0**	heat-up	close	off	60-292 inc	60-326 inc	0-1000 std	0-840 std	50 std	0-120 std
SL, TC, **RT**	**S1**				292 inc	292 inc	0 inc	0 inc		
SL, TC, RT, **TL**	**S2**	start-up					30 inc		52 dec	
SL, TC, RT, TL, **TN**	**S3**						50 inc	15 inc		2 inc
SL, TC, TL, TL, RT TN, **TT**	**S4**			on	292 std	294 inc				
SL, TC, RT, TL, TT	**S5**			off					30 inc	10 inc
SN, TC, RT, TL	**S6**									
SN, TC, RT, TL	**S7**									
SH, TC, RT, TL	**S8**									90 inc
SH, TC, RT, TL	**S9**				292 std	326 std	1000 std	840 std	50 std	100 std
SH, TC, RT, TL	**S10**									
SH, TC, RT, TL	**S11**									
SH, TC, RT	**S12**								30 dec	100 std
SH, TC, RT, **TN**	**S13**									
SH, TC, RT, TN, **OP**	**S14**			on						
SH, RT, TN, OP, **TC**	**S15**		open							
RO, TO	**S16**	shut down								
RO, TO	**S17**	heat remove			292 inc	327 inc	x dec	x x	20 dec	0

Fig. 16 *State trajectory of SDS2*

primary and secondary connections and the water level become steady state (S9). At this point, we assumed that the water level of the steam generator could be under the set point (30%) because of some accidents like loss of coolant accident (LOCA) or loss of main feed water (LOFW). When the water level becomes under 30%, the trip signal On occurs (S14), and the trip relay opens (S15), and the reactor enters the shut down state (S16). After shut down, the thermal energy of the primary connection increases because of the residual heat of the reactor (S17).

4.4 QFM 5: Validation of indicative specifications against the requirements

Requirements engineering is not a process of simple derivation of a controller specification (C) from a given set of requirements (S_p) and plant (P) properties. Rather, the three models, P, C, and S_p, are developed until an acceptable combination is reached, with the formal justification of the proposition (1). A consequence of this iterative aspect of the requirements engineering process is that the three models may be progressively developed and modified, leading to a need for the formal methods to be not only rigorous, but also amenable to proof and re-proof with a minimum of effort.

When the Four-variable model was used to specify and verify the software requirements specification (SRS) of the shutdown system of Wolsong nuclear power plant, there was no method to verify the required behaviour modelled of REQ(M) against the plant and controller models (P and C). While the controller part by OUT(SOF(IN(M))) in proposition 2, was formally verified with respect to the software requirements specification, modelled as REQ(M), the satisfaction of proposition 2, however, was informally reviewed by the system and software engineers. That is, it was not possible to verify NAT(M) and OUT(SOF(IN (M))) $\Rightarrow$ REQ(M).

In order to validate the indicative specification of HRTSS against the requirements, we have used the algorithm [4] developed for verifying a device behaviour in the qualitative reasoning domain of artificial intelligence. Given a required function F and an indicative trajectory T, the validation algorithm performs the following tests:

- Determines whether trajectory T of indicative models achieves the required function F
- If trajectory T achieves the required function F and function F elaborates the required function F', determines whether trajectory T also achieves function F'.

For the first test, we can show that the indicative trajectory Tr achieves the goal Gf, because it is easily seen from Table 1, where T6 (thermal load of the secondary connection) is positive during the power generation mode between S2 and S16 in the trajectory. This means that the steam generator acts as an appropriate heat-sink and so the trajectory satisfies the goal Gf.

The intuitive meaning of the second test is that there is a state in the trajectory, which satisfies the conditions in each

nodes of CPD, and the trajectory satisfies the causal and temporal relations specified in the ares of CPD. The details of the validation by the second test can be explained as follows:

- A trajectory T achieves a function F when the conditions specified in Df and Cf hold throughout T and Gf matches T. That is, the Boolean equations of Gf must be true in the trajectory T.
- The initial nodes of CPDs match the initial state in the trajectory T and for every arc in CPD, the temporal and causal constraints specified in the arc are satisfied by the states in T.
- The temporal and causal relations among the variables in CPD match with the temporal and causal relations among the variables in states of the trajectory T. The causal ordering among variables shown in Fig. 15 can be produced by the causal ordering theory [25].

The validation steps for the second test must be applied to the refined function, F11. The refined goal Gf of F11 means that three Boolean equations under AND must always be true. The three Boolean equations are explained as follows:

- (AND (Start-up ?r) CPD1) means that when the reactor starts to generate power, the reactivity of reactor increases.
- (IMPLIES (AND ($\geq$ (LogPower ?r) 2 %FP) ($\geq$ (CAvg-Power ?r) 10 %FP) ($\leq$ (Level ?s) LSP) (Closed-p ?t)) CPD2) means that if the measured values of the generated power are greater than the pre-determined values, and the water level of steam generator is less than or equal to the setpoint, the system must be shutdowned.
- (IMPLIES (AND (< (LogPower ?r) 2 %FP) (< (CAvg-Power ?r) 10 %FP) (Closed-p ?t)) CPD3))) means that if the measured values of the generated power are less than the predetermined values, the trip signal must be conditioned out.

For the first Boolean equation in Gf of F11, Table 1 shows that the condition, (Start-up ?r), holds between S2 and S16 in Tr, and also CPD1 matches the sequence of states in Tr because the reactivity, CAP, increases between S2 and S16. Similarly, we can easily perform the validation of the second and third Boolean equations of the goal Gf. We can conclude that the indicative plant and controller models of the HRTSS satisfy the desired goal, Gf. So, we have qualitatively demonstrated the truth of proposition 1, P and $C \rightarrow S_p$.

5 Conclusion

We have presented a requirements engineering process for HRTSS, QFM, which provides a formal framework using the causal qualitative reasoning languages and simulation method, CML, CFRL, and DME. In this paper, we demonstrated that the QFM is effective in specifying and validating the behaviour of a real-time process control system by presenting a case study of the shutdown system 2 (SDS 2) in Wolsong nuclear power plants. The causal qualitative reasoning ideas from artificial intelligence were selected to help the designers specify their insights on the properties of the physical plant, to eliminate unnecessary numerical details at a requirements phase. While discrete approximation methods on SCR-style specification used in several safety-critical industries could not provide an adequate means of verifying proposition 2, we were able to overcome such limitations using QFM.

As seen in the Wolsong example, of course it is not easy to build a full-scale nuclear power plant model using CML model fragments. However, there are many research efforts in the qualitative reasoning community to construct a sharable knowledge base for many engineering domains, such as nuclear power plant, satellite, and the electronic circuit. When those knowledge bases become mature and realistic, specification and validation methods using QFM could be practical.

We are studying a requirements safety analysis method utilising the causality information resulting from QFM. Because safety analysis of a software system is to seek the logical contribution of the software elements to the physical hazard, early hazard analysis provides many benefits, for example, to meet the regulatory requirements. The most important benefit is that earlier hazard analysis makes it easier to find hazards because the semantic distance between a software element and the physical hazard in the early phase of the life cycle is shorter than in late phases. Since the simulation system, such as DME, knows the exact condition necessary at each branching point to follow a desired path, there is also a possibility of using the conditions as a guide for safety analysis, like fault tree analysis in the requirement phase. The causal ordering which resulted from QFM on the dynamic behaviour of the physical systems will carry out the role of guide to find the root cause of a fault.

6 Acknowledgments

The work described here was supported in part by the Ministry of Science and Technology under the Korea Nuclear Long Term Program, specifically KALIMER project, and in part by AITrc of KAIST. The authors acknowledge the contributions of colleagues at KAERI and KAIST. They express our gratitude to anonymous referees for their critical and constructive comments and suggestions on the revision of this paper and in part by AITrc of KAIST.

7 References

1 OSTROFF, J.S.: 'Temporal logic for real-time systems' (Research Studies Press, 1989)
2 HANSEN, K.M., RAVN, A.P., and STAVRIDOU, V.: 'From safety analysis to software requirements', *IEEE Trans. Softw. Eng.*, 1998, **24**, (7), pp. 573–584
3 FALKENHAINER, B., FARQUHAR, A., BOBROW, D., FIKES, R., FORBUS, K., GRUBER, T., IWASAKI, Y., and KUIPERS, B.: 'CML: A Compositional Modelling Language'. KSL Report no. KSL-94-16, , KSL in SRI, Stanford University, 1994
4 IWASAKI, Y., VESCOVI, M., FIKES, R., and CHANDRASEKARAN, B.: 'A causal functional representation language with behavior-based semantics', *Appl. Arti. Intell.*, 1995, **9**, (1), pp. 5–31
5 IWASAKI, Y., and LOW, C.M.: 'Model generation and simulation of device behavior with continuous and discrete change'. KSL Report no. KSL-91-69, KSL in SRI, Stanford University, Nov. 1991
6 ALUR, R., COURCOUBETIS, C., HENZINGER, T.A., and HO, P.H.: 'Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems', Hybrid Systems Workshop, Lecture notes in computer science, **736**, (Springer-Verlag, 1993), pp. 209–229
7 MALER, O., MANNA, Z., and PNUELI, A.: 'From timed to hybrid systems', Real time: Theory in practice, REX Workshop, Lecture notes in computer science, DE BAKKER, J.W., HUIZING, C., DE ROEVER, W.P., and ROZENBERG, G. (Eds.), (Springer-Verlag, 1992), **600**, pp. 447–484
8 RAVN, A.P., RISCHEL, H., and HANSEN, K.M.: 'Specifying and verifying requirements of real-time systems', *IEEE Trans. Softw. Eng.*, 1993, **19**, (1), pp. 41–55
9 HENZINGER, T.A., KOPKE, P.W., PURI, A., and VARAIYA, P.: 'What's decidable about hybrid automata'. 27th Annual ACM Symposium on Theory of Computing, Las Vegas, Nevada, USA, 1995, pp. 373–382
10 DE LEMOS, R., and HALL, J.G.: 'Extended RTL in the specification and verification of an industrial press', Hybrid Systems Workshop III, Lecture notes in computer science, ALUR, R., HENZINGER, T., and SONTAG, E. (Eds.), (Springer-Verlag, 1996), **1066**, pp. 114–125

11 HENZINGER, T.A., and HO, P.H.: 'A note on abstract interpretation strategies for hybrid automata', Hybrid Systems Workshop II, Lecture notes in computer science' ANTSAKLIS, P., KOHN, W., NERODE, A., SASTRY, S. (Eds.), (Springer-Verlag, 1995), **999**, pp. 252–264
12 HENZINGER, T.A., and HO, P.H.: 'Model checking strategies for linear hybrid systems'. no. CSD-TR-94-1437, Cornell Technical Report, 1994
13 PURI, A., and VARAIYA, P.: 'Verification of hybrid systems using abstractions', Hybrid Systems Workshop II, Lecture notes in computer science' ANTSAKLIS, P., KOHN, W., NERODE, A., SASTRY, S., (Eds.), (Springer-Verlag, 1995), **999**, pp. 359–369
14 LIU, Z., RAVN, A.P., SORENSEN, E.V., and ZHOU, C.: 'A probabilistic duration calculus, Responsive Computer Systems, Dependable Computing and Fault-Tolerant Systems' (Springer-Verlag, 1993), **7**, pp. 29–52
15 PARNAS, D.L., and MADEY, J.: 'Functional documentation for computer systems engineering'. CRL Report no. 237, TRIO McMaster University, Sep. 1990
16 HEITMEYER, C.L., JEFFORDS, R.D., and LABAW, B.G.: 'Automated consistency checking of requirements specification', *ACM Trans. Softw. Eng. Methodol.*, 1996, **5**, (3), pp. 231–261
17 HEITMEYER, C.L.: 'Requirements specification for hybrid systems', Hybrid Systems Workshop III, Lecture notes in computer science' ALUR, R., HENZINGER, T., and SONTAG, E., (Eds.), (Springer-Verlag, 1996), **1066**, pp. 304–314
18 COURTOIS, P.J., and PARNAS, D.L.: 'Documentation for safety critical software'. 15th International Conference on Software Engineering, ICSE-15, Baltimore, Maryland, USA, 1993, pp. 315–323
19 Wolsong NPP 2/3/4, 'Software requirements specification for shutdown system 2 PDC'. Design Document no. 86-68350-SRS-001, Rev. 0, June 1993
20 BHARADWAJ, R., and HEITMEYER, C.L.: 'Model checking complete requirements specifications using abstraction'. Memorandum Report, no. NRL/MR/5540-97-7999, Naval Research Laboratory, Washington, 1997
21 ARCHER, M., and HEITMEYER, C.L.: 'Verifying hybrid systems modelled as timed automata: A case study', HART'97, Grenoble, France, Lecture notes in computer science' (Springer-Verlag, 1997), **1201**, pp. 171–185
22 MOFFETT, J., HALL, J., COOMBES, A., and MCDERMID, J.: 'A model for a causal logic for requirements engineering', *J. Requirements Eng.*, 1996, **1**, (1), pp. 27–46
23 RASMUSSEN, J., PEJTERSEN, A.M., and GOODSTEIN, L.P.: 'Cognitive systems engineering' (John Wiley & Sons, New York, 1994)
24 LEVESON, N.G.: 'Intent specifications: An approach to building human-centered specifications'. 3rd International Conference on Requirements Engineering, 1998, Colorado, USA
25 IWASAKI, Y., and SIMON, H.A.: 'Causality and model abstraction', *Artif. Intell.*, 1994, **67**, (1), pp. 143–194