Requirements specification and verification for safety-critical systems: a train set example.

R. de Lemos, A. Saeed and T. Anderson

Abstract
Requirements analysis plays a vital role in the development of safety-critical systems since any faults in the requirements specification will corrupt the subsequent stages of system development. Experience in safety-critical systems has shown that faults in the requirements can and do cause accidents. This paper presents a general framework for the requirements analysis of safety-critical systems, which incorporates the verification and validation of the safety specifications produced during the analysis, and propose feasibility analysis to assess whether the development of the system (as described by the requirements specification) should continue. To illustrate the proposed approach an example based on a train set crossing is presented.
Bibliographical details

DE LEMOS, Rogerio

Requirements specification and verification for safety-critical systems: a train set example.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Laboratory, 1991.

(University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, no. 341)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE.
Computing Laboratory. Technical Report Series. 341
SAEED, Amer
ANDERSON, Thomas

Abstract

Requirements analysis plays a vital role in the development of safety-critical systems since any faults in the requirements specification will corrupt the subsequent stages of system development. Experience in safety-critical systems has shown that faults in the requirements can and do cause accidents. This paper presents a general framework for the requirements analysis of safety-critical systems, which incorporates the verification and validation of the safety specifications produced during the analysis, and propose feasibility analysis to assess whether the development of the system (as described by the requirements specification) should continue. To illustrate the proposed approach an example based on a train set crossing is presented.

About the author

Mr. R. de Lemos is a Ph.D. student in the Computing Laboratory.
Dr. A. Saeed is a Research Associate in the Computing Laboratory.
Professor T. Anderson is a Professor in the Computing Laboratory.

Suggested keywords

FORMAL MODELS
REQUIREMENTS ANALYSIS
SAFETY-CRITICAL SYSTEMS
VALIDATION
VERIFICATION

Suggested classmarks (primary classmark underlined)
Dewey (18th): 001.6425 658.47
U.D.C. 681.322.06 519.718
Requirements Specification and Verification for Safety–Critical Systems: a Train Set Example

Rogério de Lemos, Amer Saeed and Tom Anderson
Computing Laboratory
University of Newcastle upon Tyne, NE1 7RU, UK

Abstract

Requirements analysis plays a vital role in the development of safety–critical systems since any faults in the requirements specification will corrupt the subsequent stages of system development. Experience in safety–critical systems has shown that faults in the requirements can and do cause accidents. This paper presents a general framework for the requirements analysis of safety–critical systems, which incorporates the verification and validation of the safety specifications produced during the analysis, and propose feasibility analysis to assess whether the development of the system (as described by the requirements specification) should continue. To illustrate the proposed approach an example based on a train set crossing is presented.

Keywords: safety–critical systems, requirements analysis, formal models, verification, validation.

1. Introduction

A major motivation for the work presented in this paper is the belief that a substantial improvement in the dependability of safety–critical systems can be achieved by performing a formal assessment on the results of the requirements analysis before proceeding to any subsequent phases. The aim of this approach is to locate and remove faults introduced during the requirements analysis in the development of software for safety–critical systems. (Although “safety” is an attribute of the system rather than just software, in this paper the study is restricted to problems related to “software safety”.)

Faults introduced during requirements analysis have been identified, by several researchers, as one of the primary causes of software safety problems /Ericsson 81/. Furthermore, the longer (in terms of system development) a fault goes undetected the more costly will be its consequences. Early requirements faults have typically cost a hundred to a thousand times as much to rectify as those introduced during software implementation (coding) /Boehm 81/. During requirements analysis two sources of faults can be identified: those arising from a fundamental misunderstanding of the user needs, and those created when expressing these needs.

Our concern is with the analysis of the requirements and not with the elicitation of the requirements (i.e. the process of acquisition of the relevant information from the user). We deal with techniques that can be used to reduce (or eliminate) the possibility of the occurrence of a hazard due to faults introduced during the requirements analysis, we propose a role for the assessment of the effectiveness of the techniques used. In this paper we concentrate on how knowledge of the process used to produce a requirements specification can provide a
basis for a structured approach to both qualitative and quantitative assessments of the requirements specification. We also investigate some means which would enable us to assess the feasibility of the system based on the requirements specification. Such an assessment would be invaluable for safety-critical systems since they are costly to develop and are at the limits of what can currently be achieved in terms of ultra-high dependability.

The approach to be followed for the analysis of the safety requirements is based on a clear separation of the mission and the safety requirements. The mission requirements focus on what the system is supposed to achieve in terms of function, timeliness and some dependability requirements – namely the attributes of reliability, availability and security. On the other hand, the safety requirements focus on the elimination and control of hazards, and the limitation of damage in the case of an accident; thus they are related to the safety attribute of dependability /Laprie 90/. Some of the benefits of making this distinction during requirements analysis are: the resolution of potential conflicts, detection of omissions and inconsistencies between the mission and safety issues, the ability to focus on the safety-critical issues, and the simplification of safety certification.

The adoption of a general structure for a safety-critical system will be a useful guide to the requirements analysis, since it allows the analysis to be split into different phases. For applications referred to as process control systems, which are the primary focus of this paper, a commonly accepted structure is to partition the system into three distinct components: the operator, the controller and the physical process (or plant). The environment is that part of the rest of the world which may affect, or be affected by, the system, and may be harmed by an accident cause by it. This structure is further decomposed to reflect the decision to separate the mission and safety requirements into mission and safety operators, and mission and safety controllers. The safety operator and the safety controller are the components which must ensure that the system does not enter a hazardous state (i.e. a state in which a hazard is satisfied).

The rest of the paper is organized as follows. The next section describes an approach for the requirements analysis which is based on a framework and a hierarchical structure of the safety specifications. Section 3 describes how to assess, qualitatively and quantitatively, the quality of the requirements specification. In section 4, a case study based on a train set crossing is discussed in accordance with the framework and the hierarchical structure previously presented. Finally, section 5 presents some concluding remarks.

2. Requirements Analysis of Safety-Critical Systems

In this section we propose an approach for the requirements analysis phase in the development of safety-critical systems. This approach is based on a framework for the analysis of the safety requirements which produces a hierarchical structure of safety specifications. The notions defined below, when established for a system are known as safety specifications.

- **accident** – an unintended event or sequence of events that cause death, injury, environmental or material damage.
• **hazard** – a physical situation, expressed as a system condition, that can lead to an accident.

• **safety constraint** – a condition imposed on the system that is the negation of a hazard modified to incorporate safety margins;

• **safety strategy** – a means defined as a set of conditions over the physical process to maintain a safety constraint;

• **safety controller strategy** – is a refinement of a safety strategy incorporating the sensors and actuators, and their relationship with the real world.

The first two notions, accidents and hazards, are specified informally whereas the others are specified formally.

In the following, we present a framework for requirements analysis which provides a systematic approach to the production of the requirements specification of the safety controller. We also define how the hierarchical structure of the safety specifications are obtained from the requirements analysis.

### 2.1. Framework for Requirements Analysis

A general framework for the requirements analysis of safety-critical systems, as presented in [Saeed 91], is shown in figure 1. The basic aim behind our framework is to subdivide the whole problem into smaller domains where the analysis of the requirements can be simplified, thereby leading to more accurate requirements specifications. This is obtained by, first, splitting the mission from safety requirements, and second, subdividing the analysis of the safety requirements into a sequence of phases.

- **Conceptual Analysis** – the objective of this phase is to produce an initial, informal statement of the aim and purpose of the system and to determine what is meant by safety for the system. As a product of the Conceptual Analysis we obtain the Safety Requirements, containing the potential accidents, and the hazards related to these accidents. It is the identification of the hazards of a system, by performing a Preliminary Hazard Analysis, that allows us to make a distinction between the mission and safety issues of the system. For each of the identified hazards an estimate of the associated risk must also be determined.

- **Safety Requirements Analysis** – the objective of this phase is the identification of the real world properties relevant to the Safety Requirements: physical laws and rules of operation. As a product of this real world analysis, we obtain the Safety Requirements Specification, containing the safety constraints and the safety strategies.

- **Safety System Analysis** – the objective of this phase is the identification of the interface between the safety controller and the physical process, and the specification of the system behaviour that must be observed at the identified interface. Also in this phase a top level organization of the system is realized in terms of the properties of the sensors and actuators of the system, and the effects of the possible failures of these sensors and
Figure 1. A Framework for Requirements Analysis.

- **Rapid Prototype** – the objective of this phase is to exercise the Safety System Specification, by using this specification to construct a low-cost prototype.

The main product of the requirements analysis is the Safety System Specification which provides the basis for subsequent development of the system. In this paper, we are specifically concerned with the phases of Safety Requirements Analysis and Safety System Analysis.

Because of the differing characteristics of each phase, instead of seeking a single formalism, our approach is to use different formalisms for each phase. This has the advantage of allowing us to select formalisms in accordance with the properties that should be expressed at each phase of development. In /Saeed 91/ we proposed the utilization of a logical formalism, such as Timed History Logic – THL /Saeed 90/, for the Safety Requirements Analysis, and a net formalism, such as Predicate–Transition Nets – PrT nets /Genrich 87/, for the Safety System Analysis. The Conceptual Analysis is by nature informally recorded and the formal notation for the Rapid Prototype phase is determined by the formalism employed in the Safety System Analysis phase.

Since we propose the use of different formalisms, means to relate the specifications produced at each phase of the requirements analysis are presented in the following. At the end of the first phase the safety requirements are expressed as a set of the safety strategies which are predicates over the THL model of the system. In the second phase, for each safety strategy a
PrT net model that specifies a safety controller strategy is then constructed. To verify a safety controller strategy against its safety strategy, an essential step is to express the predicates that describe the safety strategy in terms of the predicates of the PrT net model. The latter ones are the logical invariants of the net which will be used to verify if the PrT net model captures the safety strategies.

2.2. Safety Specification Hierarchy

The safety specification hierarchy consists of the following notions defined earlier: accidents, hazards, safety constraints, safety strategies, and safety controller strategies. Our basic concern is to ensure that accidents do not occur, but we encounter the problem that the safety controller cannot directly influence all factors that could lead to an accident. To deal with this problem the notion of hazard is typically used to describe a circumstance from which an accident might ensue, but can be prevented by the safety controller. Thus, to ensure safety, we strive to identify all the hazards and then provide techniques to reduce or eliminate these hazards. The effort that should be employed to handle a hazard depends on the risk associated with that hazard. Risk is defined as a function of the likelihood of a hazard occurring, the likelihood that the hazard will lead to an accident, and the worst possible potential loss associated with such an accident /Leveson 91/.

An example of a possible structure for a safety specification hierarchy is shown in figure 2. The following predicates on a system state are used in the safety specification hierarchy:

\[ AC_i, \ i = 1, \ldots, m \text{ are the accidents;} \]
\[ HZ_{i,j}, \ j = 1, \ldots, n(i) \text{ are the hazards;} \]
\[ SC_{i,j}, \ j = 1, \ldots, n(i) \text{ are the safety constraints;} \]
\[ SS_{i,j,k}, \ k = 1, \ldots, r(j) \text{ are the safety strategies;} \]
\[ SCS_{i,j,k,l}, \ l = 1, \ldots, q(k) \text{ are the safety controller strategies.} \]

Consistency Conditions

The consistency conditions are used to confirm that the specifications at each level of the hierarchy can be satisfied by the system. To perform the consistency checks the specifications at each level should be partitioned into classes. Each class consists of those specifications which must be satisfied during a particular mode of operation (since only certain accidents may be possible in a given mode of operation) and when specifications are exclusive the value of the state predicates which determine the specification that must be satisfied. Hence, to identify these classes the application characteristics must be investigated.

Safety Conditions

The safety conditions are those that relate safety specifications at different levels of the hierarchy. These conditions can be used to check that the safety (absence of accidents/hazards) is maintained down the hierarchy (vertical checks). The hazards of the accidents must capture all the system conditions that can lead to the occurrence of the accident, in other words, if the system is not in a hazardous state then an
accident cannot occur. This can be stated formally as:

\[ \forall i: \land_{j=1}^{m_i} \neg HZ_{i,j} \Rightarrow \neg AC_i. \]  

(1)

The safety constraint of a hazard must imply the negation of the hazard. This can be stated formally as:

\[ \forall i: \exists j: SC_{i,j} \Rightarrow \neg HZ_{i,j}. \]  

(2)

For each safety constraint there must exist at least one safety strategy that will maintain the safety constraint. This can be stated formally as:

\[ \forall i: \exists j: \lor_{k=1}^{n_{i,j}} SS_{i,j,k,l} \Rightarrow SC_{i,j}. \]  

(3)

For each safety strategy there must exist at least one safety controller strategy that will implement it. This can be stated formally as:

\[ \forall i: \exists k: \exists l: \lor_{k=1}^{n_{i,j,k,l}} SCS_{i,j,k,l} \Rightarrow SS_{i,j,k,l}. \]  

(4)

**Lemma 2.1**

If safety conditions 1, 2, 3 and 4 hold for an accident AC\(_i\) then implementing all the safety controller strategies SCS\(_i\) will ensure that accident AC\(_i\) cannot occur.

**Proof.** Immediate.

**Fault Tolerance in the Safety Specification Hierarchy**

Safety condition 3 imposes the relationship that every safety constraint must be maintained by at least one safety strategy. During the requirements analysis the number of safety strategies that are constructed for a safety constraint of a hazard will be based on an analysis of the risk associated with that hazard. An interesting situation from the perspective of fault tolerance arises when for some safety constraints more than one safety strategy is required to ensure
that the risk associated with a hazard is acceptable. For example, in the hierarchy for accident \( AC_i \) illustrated by figure 2, to reduce the risk associated with hazard \( HZ_{i,2} \) two safety strategies \( SS_{i,1,2} \) and \( SS_{i,2,2} \) are devised.

On the other hand, the number of safety controller strategies that must be constructed for a safety strategy depends not only on the risk associated with the hazard controlled by the safety strategy, but also the confidence that can be placed in the sensors and actuators of the safety controller. For example, in the hierarchy for accident \( AC_i \) illustrated by figure 2, for safety strategies \( SS_{in(1),1} \) two safety controller strategies \( SCS_{in(1),1,1} \) and \( SCS_{in(1),1,2} \) are formulated.

### 3. Evaluation of the Safety Requirements

In this section, we describe a general approach which will enable us to assess, qualitatively and quantitatively, the quality of the requirements specification – qualitatively, by applying verification and validation techniques to the framework previously introduced, and quantitatively, by assessing the methods utilized and the results obtained from the requirements analysis. The quality of the requirements specification is related to the probability of the occurrence of undesirable events in the final system attributed to faults in the requirements specification. Product quality assessment is the measurement of quality or the provision of evidence, that a desired level has been achieved /Miller 89/.

#### 3.1. Qualitative Assessment

The qualitative assessment of the requirements specification is based on formal and informal analysis, described below. We are only concerned here with verification and validation of the specifications of the safety requirements, and not of the entire system.

- **Verification** – the process of checking that the safety specifications produced at a given phase are internally consistent, that is the requirements captured by the specifications are non-contradictory, and that they comply with the specifications produced during the previous phase. Verification is performed by formal proofs of consistency, and aims to remove faults caused by inconsistencies or ambiguities, and to detect incompleteness of specifications.

- **Validation** – the process of checking that a safety specification accurately reflects the requirements and constraints imposed on the system by the user, or some other authority. Validation is performed by inspecting the steps used to produce the safety specification, and also by exercising it. The aim is to remove faults caused by a misunderstanding of the user needs, and those generated by mistakes in expressing these needs.

Within the framework for the analysis of the safety requirements, previously presented, these two forms of qualitative assessment are performed as follows.

The formal representation of the hazards are the safety constraints and these must be validated against the results of the Preliminary Hazard Analysis, to ensure safety condition 2 and check for inconsistencies (as discussed in 2.2). On the other hand, each safety strategy
must be verified, against the relevant safety constraints to confirm safety condition 3, and these strategies must be checked for inconsistencies. The safety strategies are then validated against the mission requirements to ensure that these do not conflict with the safety strategies while the system is in a safe state of operation.

During the Safety System Analysis, after refining a safety strategy with respect to the system components (such as the sensors and actuators), a safety controller strategy is obtained; its verification ensures safety condition 4. As far as the validation of the safety controller strategies is concerned, it will include executing the formal specifications obtained from the Safety System Analysis phase. Within the proposed framework this would be made through the simulation of the PrT net model of the physical process and the safety controller. This type of validation checks whether the Safety System Specification does indeed capture the intentions of the user, and if it is consistent with the mission requirements. There are other ways in which errors can be detected in the Safety System Specification, such as general correctness criteria which must be satisfied by all systems of a certain type /Jaffe 91/.

3.2. Quantitative Assessment

In this section, we discuss how quantitative assessment complements the qualitative assessment of the safety requirements; specifically, we aim to obtain levels of confidence for the qualitative assessment. Although the qualitative assessment aims to ensure the total correctness of the requirements specifications, there are two basic reasons which indicate that this aim may not be achieved. The first stems from the limitations that exist in capturing user requirements: faults introduced at the requirements stage may not all be removed during the verification process, nor can it be guaranteed that all such faults will be removed during validation. The second reason arises from observing past experience in the utilization of formal methods, which shows that a formal verification may itself contain faults. Thus, we are still faced with uncertainties concerning the correctness of the requirements specifications. An approach to deal with these uncertainties is to quantify them in terms of levels of confidence, thereby providing evidence for the quality of the product.

A possible way to obtain levels of confidence for the quantitative assessment is by statistical means (by estimation or prediction). In terms of the proposed framework, the probabilistic assessment by estimation is based on dynamic analysis of the requirements specification. We construct a prototype from the Safety System Specification and establish an operational environment for the prototype; execute this prototype and collect failure data which will be used to estimate the reliability of the prototype. Two basic assumptions for this approach are: a suitable operational environment can be constructed, and an appropriate fault model for the prototype can be selected. Probabilistic assessment by prediction is based on static analysis of the requirements specification. The approach involves the use of metrics to assess complexity and completeness of the product. The type of metrics to be used at each phase of the proposed framework depends on the properties to be expressed, and the formal notation employed /Ramamorthy 85/.

The results of the quantitative assessment may also be used to assess the feasibility of further development of the system. A system development can be regarded as feasible if the cost of
developing the system will be recuperated by the services it provides, and the benefits outweigh the risk. By checking the feasibility of a system after requirements analysis we wish to reduce the possibility of difficulties arising in the later phases, from over ambitious requirements specification. For example, despite adherence to a strict fault avoidance approach which was successfully applied in a specific nuclear reactor shutdown system /Pilaud 90/, when this approach was used for a similar system, but with more functions, the certification became so difficult that the new project may have to be abandoned /Nucleonic 91/.

Feasibility analysis of the requirements specification may be defined as follows.

- **Feasibility Analysis** – a study performed to determine whether the development of the system should proceed, or not.

The feasibility analysis of the system will indicate whether the development costs will be recuperated from the service provided, and if the benefits outweigh the risks of the system, and is based on:

a. the level of confidence, or the evaluation of effectiveness, of the formal and informal analysis of the safety requirements /Leveson 91/;

b. a prediction of the achievable and certifiable system reliability.

The evaluation of effectiveness should be based on the results obtained from the quantitative assessment of the requirements specification, and also on historical information from previous products when there has been a gradual evolution. However, as mentioned in /Laprie 91/, there are no means presently available for assessing the “degree of novelty” in a system that has evolved from an earlier version. Prediction of the achievable and certifiable system reliability must be tackled in two parts. Firstly, we must determine if it is possible to produce a system that satisfies the Safety System Specification; secondly we must predict the level of reliability which can be claimed for such a product.

4. The Train Set Crossing

With the aim of exemplifying the proposed framework and the safety specification hierarchy, a train set crossing was selected as a case of study. An obvious advantage of using a train set is that safety strategies can be implemented without endangering the travelling public. The train set crossing described below raises safety–critical issues that are similar to those found at the traditional level crossing (i.e. road–rail). In /Saeed 91/ we performed the analysis of the safety requirements; here we demonstrate how the safety specification hierarchy can be used to structure the produced specifications. The logical formalism employed for the Safety Requirements Analysis was Timed History Logic – THL /Saeed 90/, and the net formalism employed for the Safety System Analysis was Predicate–Transition nets /Genrich 87/. Here we consider only the qualitative assessment of the safety specification hierarchy. Although full proofs have been constructed for the lemmas presented in this section, only sketches are provided here due to space limitations.

**Conceptual Analysis**
The physical process consists of two track circuits \( C_p \) and \( C_s \), and two types of trains – primary (Tp) and secondary (Ts). The circuits are divided into sections and there are two separate crossing sections \( CC_a \) and \( CC_b \) at which they intersect. Trains of type Tp travel around circuit \( C_p \) and trains of type Ts travel around circuit \( C_s \), both types of train travel in one direction (clockwise) only. The longest train is shorter than the smallest section. A crossing section is that part of the track which consists of the sections (one from each circuit) at which the two circuits intersect. The circuits \( C_p, C_s \) and the crossing sections \( CC_a \) and \( CC_b \) are illustrated in figure 3.

Figure 3. The train set circuits and the crossing section

**General Model**

The type of circuit is denoted by \( c \in L, L = \{ p, s \} \), the crossing section by \( r \in R, R = \{ a, b \} \), the trains which run on \( C_c \) are denoted by \( x, y \in T_{rc} = \{ 1, \ldots, N_{tc} \} \) and the sections of \( C_c \) are denoted by \( i, j, k, m, n \in S_c, S_c = \{ 0, \ldots, N_{sc} \} \).

Addition \( \oplus \) and subtraction \( \ominus \) on circuit section numbers are performed modulo the number of sections of the circuit. A crossing section is that part of the track which consists of the sections (one from each circuit) at which the two circuits intersect; we consider only one crossing section \( CC \). The danger zone on circuit \( C_c \) for \( CC(r) \) is defined as: \( DZc(r) = \{ CCc(r), CCc(r) \oplus 1 \} \), where \( CCc(r) \) is the number of the section of \( C_c \) that is part of \( CC(r) \). The danger zones \( DZp \) and \( DZs \) for a crossing section \( CC \) are illustrated in figure 3. The behaviour of the system is captured by two state variables \( P_{train}, R_{train} \), described below.

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>( P_{train} )</td>
<td>( Sp^{N_{ip}} \times Ss^{N_{is}} )</td>
<td>The position of each train expressed as a section number, that is, the section containing the front of a train.</td>
</tr>
<tr>
<td>( p_2 )</td>
<td>( R_{train} )</td>
<td>( Rp^{N_{ip}} \times Rs^{N_{is}} )</td>
<td>The reservation sets of the trains, where ( R_c = \mathcal{P}(S_c) ).</td>
</tr>
</tbody>
</table>

\( P_{train}(c, x) \) denotes the state variable for the position of train \( x \) on circuit \( C_c \), and \( R_{train}(c, x) \) the reservation set of train \( x \) on circuit \( C_c \). State variables are taken to be functions over time \( T \) represented by the non-negative reals to their respective ranges. The universal history set \( \Gamma H \) is the set for all functions \( H: T \to Sp^{N_{ip}} \times Ss^{N_{is}} \times Rp^{N_{ip}} \times Rs^{N_{is}} \). A history satisfies a safety
constraint or safety strategy if and only if the system predicates that describe the notion are invariant relations (i.e. they are satisfied at all time points in T) for that history.

4.1. Safety Specification Hierarchy

Here we present the accidents, hazards, safety constraints, and safety strategies, associated with the train set case study. From the various accidents possible on the train set, we only consider the following two cases.

Accidents

$AC_1$ - trains of the same type collide;

$AC_2$ - trains of different type collide.

Hazards

$HZ_{1,1}$ - some part of any two trains are in the same section.

$HZ_{2,1}$ - some part of a primary train and a secondary train are in the same crossing section.

Safety Constraints

$SC_{1,1}$ - for any two trains there must be at least one section between the sections containing the fronts of the trains, formalized as a system predicate:

$$\forall c \in L: \forall x, y \in Trc: x \neq y \Rightarrow |Ptrain(c, x) \oplus Ptrain(c, y)| > 1.$$  

$SC_{2,1}$ - either the front of no primary train is in a danger zone $DZp(r)$ or the front of no secondary train is in the danger zone $DZs(r)$, formalized as a system predicate:

$$\forall r \in R: \exists c \in L: \forall x \in Trc: Ptrain(c, x) \not\subseteq DZc(r).$$

Safety Strategies.

$SS_{1,1,1}$ - the basic rules for the strategy for $SC_{1,1}$ are:

a. for any train the current section (i.e. the position of the front of the train) and the section behind the current section must always be reserved;

b. no section can be reserved by more than one train.

These rules are formalized as two system predicates:

a. $\forall c \in L: \forall x \in Trc: \{Ptrain(c, x) \oplus 1, Ptrain(c, x)\} \subseteq Rtrain(c, x);$  
b. $\forall c \in L: \forall x, y \in Trc: x \neq y \Rightarrow Rtrain(c, x) \cap Rtrain(c, y) = \emptyset.$

$SS_{2,1,1}$ - the basic rules for the strategy for $SC_{2,1}$ are:

a. if any train $x$ on circuit $Cc$ is in a danger zone then the crossing section contained within that danger zone is reserved for circuit $Cc$;

b. sections $CCp(r)$ and section $CCs(r)$ cannot both be reserved.

These rules are formalized as two system predicates:
\[ a. \forall r \in R: \forall c \in L: \forall x \in Trc: P_{train}(c, x) \in DZc(r) = CCc(r) \in R_{train}(c, x)(r); \]
\[ b. \forall r \in R: \exists c \in L: (\forall x \in Trc: CCc(r) \not\in R_{train}(c, x)); \]

**Safety Controller Strategies**

The safety controller strategies of the safety strategies SS_{1,1,1} and SS_{2,1,1} will be represented in terms of PrT nets. To illustrate the modelling of the behaviour of trains in a circuit, we assume that each circuit contains seven sections (Nsc = 6) and two trains (Trc = 2).

SCS_{1,1,1} – the PrT net model of the safety controller strategy of SS_{1,1,1} is shown in figure 4. The predicates of the PrT net model are:

\[ Sx_i \text{ – train x occupies section i of the circuit;} \]
\[ ICPx_j \text{ – train x is allowed to enter section j;} \]
\[ IPCx_j \text{ – train x has entered section j;} \]
\[ FS_n \text{ – section n is not reserved by the safety controller;} \]
\[ RS1xm \text{ – train x has reserved section m;} \]
\[ RS2xk + x_j \text{ – train x has (temporarily) reserved sections k and j.} \]

![Diagram](image)

**Physical Process Model**

**Safety Controller Model**

**Figure 4. The PrT net Model of the Train Set Circuit**

SCS_{2,1,1} – the PrT net model of the safety controller strategy of SS_{2,1,1} is shown in figure 5. The predicates of the PrT net model are:

\[ S0cx, S1cx, S2cx \text{ – a section of the circuit c is occupied by train x;} \]
\[ CCcx \text{ – the crossing section of circuit c is occupied by train x;} \]
\[ ICPcx_j \text{ – train x in circuit c is allowed to enter section j;} \]
\[ IPCcx_j \text{ – train x in circuit c has entered section j;} \]
\[ APcx_j \text{ – train x in circuit c is in section j = 2, (i.e. x is to enter in DZ);} \]
ZDcxj - train x in circuit c has access to section j (= 3, the CC); ME - either primary or secondary trains allowed to enter the CC.

![Diagram of Physical Process Model and Safety Controller Model]

A = \{(p,1); (s,1)\}

Figure 5. The PrT net Model of the Crossing Section

In the next section we present an qualitative assessment of the specifications, presented above, in the context of the framework.

4.2. Qualitative Assessment of the Safety Requirements

Consistency Conditions

Safety Constraints
We must confirm that the following system predicate can be satisfied: SC_{1,1} \land SC_{2,1}. Firstly, we consider SC_{1,1} which can be satisfied if \forall c \in L: |Sc| \geq 2|Trc|, for example, consider the state in which the trains are positioned as follows \forall c \in L: \forall x \in Trc: Ptrain(c, x) = 2x \in 2. Secondly, we consider SC_{2,1} which can be satisfied if \forall c \in L: |Sc| \geq 3. Both safety constraints can be satisfied if \forall c \in L: |Sc| \geq \max(2|Trc|, 3).

Safety Strategies
We must confirm that the following condition can be satisfied: SS_{1,1,1} \land SS_{2,1,1}. Both safety strategies can be satisfied if \forall c \in L: |Sc| \geq 2|Trc| + 1.

Safety conditions

Safety Constraints
Firstly we consider the condition: SC_{1,1} = -HZ_{1,1}.
From SC_{1,1} it follows that there must be at least one section between the front of any two
trains, and \( -HZ_{1,1} \) follows from the fact that any train is smaller than a section. Hence, the assumption that the largest train is smaller than the smallest section is a safety assumption. If sufficient confidence cannot be placed in this assumption, a safety margin of an additional section could be introduced into the safety constraint.

Secondly we consider the condition: \( SC_{2,1} \Rightarrow -HZ_{2,1} \).
From \( SC_{2,1} \) it follows that at least one danger zone does not contain the front of a train, and \( -HZ_{2,1} \) follows from the fact that any train is smaller than a section.

**Safety Strategies**
The proofs for the safety strategies are given as lemmas 4.1 and 4.2.

**Lemma 4.1.**
A history that satisfies \( SS_{1,1,1} \) must satisfy \( SC_{1,1} \).

**Proof:** (By contradiction.) Assume \( \exists c \in L: \exists x, y \in Trc: x \neq y \land |Ptrain(c, x) \oplus Ptrain(c, y)| \leq 1 \). Then by rule \( a \), \( \{Ptrain(c, x) \oplus 1, Ptrain(c, x)\} \subseteq Rtrain(c, x) \land \{Ptrain(c, y) \oplus 1, Ptrain(c, y)\} \subseteq Rtrain(c, y) \), hence \( Rtrain(c, x) \cap Rtrain(c, y) \neq \emptyset \). This contradicts rule \( b \).

**Lemma 4.2.**
A history that satisfies \( SS_{2,1,1} \) must satisfy \( SC_{2,1} \).

**Proof:** (By contradiction.) Assume \( \exists r \in R: \forall c \in L: (\exists x \in Trc: Ptrain(c, x) \in DZc(r)) \). Then by rule \( a \) and the definitions of \( DZp \) and \( DZs \), \( \exists r \in R: \forall c \in L: CCc(r) \in Rtrain(c, x) \). This contradicts rule \( b \).

**Circuit Safety Controller Strategy**
The safety controller strategy \( SCS_{1,1,1,1} \) is verified by proving that the rules of safety strategy \( SS_{1,1,1} \) are a properties of the PrT net model of figure 4.

To verify the PrT net model a link must be identified between the THL and PrT net models of the train set system. This link can be established in terms of system predicates of the THL model and the predicates of the PrT net model, as follows:

\[
Ptrain(c, x) = i \Rightarrow Sx_i;
\]
\[
i \in Rtrain(c, x) \Rightarrow RS1xi \lor RS2xi.
\]

The two rules of \( SS_{1,1,1} \) can be expressed in terms of the PrT net model as logical formulae (If1 and If2) over the predicates of the PrT net model, by substituting the equivalent predicates of the PrT net model for the THL predicates.

**If1.** If any train \( x \) is in section \( i \) then sections \( i \) and \( i \oplus 1 \) are reserved by train \( x \).

\[
(\forall x)(\forall i) [Sx_i \Rightarrow (RS1xi \lor RS2xi) \land (RS1xi \oplus 1 \lor RS2xi \oplus 1)].
\]

**If2.** Any section \( i \) is reserved by at most one train.

\[
(\forall x)(\forall y)(\forall i) [x \neq y \Rightarrow (-RS1xi \land -RS2xi) \lor (-RS1yi \land -RS2yi)].
\]

**Lemma 4.3**
The formulae If1 and If2 are logical invariants (i.e. they hold on all reachable markings) of the PrT net model.

**Proof:** a sketch of a proof is given below by analysing the transitions of the PrT net model. In the initial marking If1 holds, both trains have reserved the current and previous sections.
\( (1, 0) \in S = \{(1, 0), (1, 6)\} \subseteq RS1 \cup RS2; \)
\( (2, 3) \in S = \{(2, 3), (2, 2)\} \subseteq RS1 \cup RS2. \)

We show that the firing of the three transitions cannot violate \(lf1\).
Transition \(t1\) can fire with \((x, i)\) only when \((x, j)\) is in ICP \((j = i \oplus 1)\), therefore only after \(t2\) fires producing \((x, j)\) in ICP and RS2, prior to the firing of \(t1\), \(\{(x, i), (x, i \oplus 1)\} \subseteq RS1 \cup RS2\), after \(t1\), \(\{(x, j), (x, i)\} \subseteq RS1 \cup RS2\) (since \((x, j)\) must remain in RS2 until IPC fires producing \((x, j)\)).
Firing \(t2\) adds \((x, j)\) to RS1 \(\cup\) RS2, therefore \(t2\) cannot violate \(lf1\).
Firing \(t3\) removes \((x, k)\) from RS1 \(\cup\) RS2 only when \((x, j)\) is in S and from the transition selector of \(t3\) \((k = j \oplus 2\)\), therefore \(t3\) cannot violate \(lf1\).

In the initial marking \(lf2\) holds, no section is reserved by more than one train:
RS1 \(\cup\) RS2 = \{(1, 0), (1, 6), (2, 3), (2, 2)\}.
Firing \(t1\) does not change the markings on RS1 or RS2.
Firing \(t2\) removes \((x, m)\) from RS1 and adds \((x, k) + (x, j)\) to RS2, where \(k = m\) and \(j = m \oplus 2\).
The only change in RS1 \(\cup\) RS2 is the addition of \((x, j)\). From the transition selector of \(t2\), \(j (= n)\) is a free section, hence prior to \(t2\) being fired \(-(\exists y) (y, j) \in (RS1 \cup RS2)\). Therefore \(t2\) cannot violate \(lf2\).
Firing \(t3\) removes \((x, k)\) from RS1 \(\cup\) RS2, therefore \(t3\) cannot violate \(lf2\).

**Crossing Section Safety Controller Strategy**

The safety controller strategy SCS_{2,1,1} is verified by proving that the rules of safety strategy SS_{2,1,1} are properties of the PrT net model of figure 5.

The THL predicates and the predicates of the PrT net model, are as follows:

\[ P_{\text{train}}(c, x) \in DZc(r) \Rightarrow CCx \vee S0cx; \]
\[ CCc(r) \in R_{\text{train}}(c, x) \Rightarrow ZDcxj. \]

The two rules of SS_{2,1,1} can be expressed in terms of the PrT net model as logical formulae (lf1 and lf2) over the predicates of the PrT net model, by substituting the equivalent predicates of the PrT net model for the THL predicates.

\(lf1\). If any train is in a danger zone then the crossing section is reserved by that train.

\((\forall c)(\forall x) [CCx \vee S0cx \Rightarrow ZDcxj].\)

\(lf2\). A crossing section cannot be reserved for both the primary circuit and the secondary circuit.

\((\exists c)(\forall x)[-ZDcxj].\)

**Lemma 4.4**
The formulae \(lf1\) and \(lf2\) are logical invariants (i.e. they hold on all reachable markings) of the PrT net model.

**Proof:** a sketch of a proof is given by analysing the transitions of the PrT net model.

In the initial marking \(lf1\) holds, since the sections CC and S0 are empty: (CC \(\cup\) S0) = \(\emptyset\).
Firstly, we make the observation that tuples can be added to the set CC \(\cup\) S0 only by transition \(t4\) and removed only by transition \(t2\). In the following we argue that \((c, x, j) \in ZD\) before \(t4\) adds \((c, x)\) to CC \(\cup\) S0 and until \(t2\) removes \((c, x)\) from CC \(\cup\) S0.
Transition $t_4$ can fire with $(c, x)$ only if $(c, x, j)$ is in ICP, therefore after $t_6$ fires with $(c, x, j)$. Hence before $t_4$ fires $(c, x, j) \in ZD$. Now $(c, x, j)$ can be removed from ZD only when $t_7$ fires with $(c, x, j)$, this transition can fire only when $(c, x, j) \in ICP$. Therefore $t_7$ can fire with $(c, x, j)$ only after $t_2$ has fired with $(c, x, j)$. Hence $(c, x, j) \in ZD$, at least until $t_2$ fires.

In the initial marking $I_f$ holds since the crossing section is not reserved: $ZD = \emptyset$. We make the observation that tuples can be added to ZD only by firing transition $t_6$ and are removed by transition $t_7$. The transition $t_6$ can fire only by removing a token from ME, a token is added to ME only by $t_7$. Hence, since there is only one token in ME in the initial marking, after $t_6$ fires $t_7$ must fire before $t_6$ can fire again. Therefore there can be at most one tuple in ZD this implies that $I_f$ is indeed a logical invariant of the PrT net.

5. Conclusions

Requirements analysis plays a crucial role in the development of safety-critical systems, hence before proceeding to the subsequent phases the results of the requirements analysis must be thoroughly assessed. This involves a qualitative assessment to verify and validate the requirements specifications, and a quantitative assessment to check if the specifications are adequate and if production of the system should proceed. The adoption of a framework for the requirements analysis and a hierarchical structure for the safety specifications facilitates this assessment, by enabling the assessment to be performed in clearly defined stages.

The approach for the analysis of the safety requirements presented in this paper either provides evidence that the requirements specification does not achieve what was required, or increases confidence in the fact that the required quality was attained. This general approach was shown to be useful by applying it to the train set example.

Acknowledgements

The authors would like to acknowledge the financial support of BAe (DCSC) and CAPES/Brazil.

References


