Risk Bands – A Novel Feature of Safecharts

Nimal Nissanke
Centre for Applied Formal Methods
School of Computing, Information
Systems and Mathematics
South Bank University,
103 Borough Road, London SE1 0AA,
United Kingdom
nissanke@sbu.ac.uk

Hamdan Z. Dammag
Department of Computer Science
The University of Reading
Whiteknights, PO.Box 255,
Reading RG6 6AY,
United Kingdom
H.Z.Dammag@reading.ac.uk

Abstract

Safecharts [2] are a safety oriented variant of Statecharts [3] and have been developed especially for the use in specification and design of safety critical systems. One of the fundamental aspects of Safecharts is the explicit ordering of system states according to their risk levels. Based on this ordering, transitions are classified according to the nature of their risk and are given a priority scheme favouring the execution of safer transitions in the event of any non-determinism. As a precaution, transitions between states with unknown relative risk levels are not permitted. As a result, many transitions, including those which might be functionally desirable, may be potentially excluded between states which are located in sparsely populated areas of risk graphs. This is an inadequacy which may be attributed to factors such as incomplete hazard analysis, the lack of information about relative risk levels of different states of the system, etc. In order to extend the permitted transition space in such circumstances and to enhance the risk ordering relation, this paper introduces the concept of risk band. Risk bands enable an unambiguous interpretation of the relevant risk level of states, thus allowing a well understood enhancement of risk graphs and an extension of the concept of safe non-determinism introduced in [2]. An example drawn from the nuclear industry demonstrates the application of Safecharts.

1 Introduction

Statecharts [3] are a visual formalism which extends conventional state diagrams with structuring and communication mechanisms for describing large and complex systems. Despite its success in modelling, the syntax and semantics of Statecharts restrict its usage. In order to overcome these problems, several variants of Statecharts have been proposed. A detailed discussion may be found in [8].

Another variant of Statecharts is Safecharts, proposed in [2] for use especially in the specification and design of safety critical systems. In order to highlight safety issues, Safecharts make a clear separation between functional and safety requirements in the representation of any system. The separate representations take the form of two separate layers, one dealing with the functional behaviour under normal operational conditions, and the other dealing with safety features required under both normal operational conditions and adverse conditions. The two layers are superimposed to form a master diagram, which can be used in the study of interactions between safety and function and the behaviour of the overall system. The different features of Safecharts are explained in Section 3.

Ordering of system states according to their risk level is one of the fundamental aspects of Safecharts. Based on a risk ordering relation $\sqsubseteq$ on the state space, transitions are classified into the following categories: *safe*, *unsafe* and *neutral*. One of the unique features distinguishing Safecharts from Statecharts is that Safecharts permit transitions only between states with known risk levels. However, this approach can be too restrictive under certain circumstances as it might exclude many transitions, including those which are useful from a functional point of view. As a result, the risk graph constructed according to $\sqsubseteq$ might not be an adequate representation of relative risks level of the system states. This could be due to an inconsistency in the degree of risk assessment devoted to certain areas of the state space, or due to a lack of information about the relative risk levels of some states. This inadequacy of the risk graph may lead to ambiguities in the interpretation of risk levels of system states and narrows down the range of permitted tran-

0-7695-0807-3/00 $10.00 © 2000 IEEE

sitions. As a means of overcoming the above problem, this paper introduces the concept of *risk band* and thus extends the range of permitted transitions but without jeopardising safety. Risk bands provide a default and an unambiguous interpretation of risk levels of states. This leaves the designer with the responsibility either to accept the default interpretation or to refine the relative risk levels of states concerned. The introduction of risk bands also allows a further reduction in non-deterministic behaviour in the event of any simultaneously enabled transitions. This is achieved through the concept of *risk distance*.

This paper is structured as follows. Section 2 presents a brief introduction of Statecharts. Section 3 outlines the basic ideas of Safecharts, including the ordering of states according to risk and a safety–oriented classification transitions. Section 4 introduces the concept of risk band as a way of enhancing the risk ordering relation. Section 5 illustrates the application of Safecharts using a case study drawn from the nuclear industry. Section 6 points to an issue to be addressed by further research, namely, the need for dynamic reconfiguration of the risk ordering relation. Section 7 concludes the paper with a statement of achievements. Appendix A provides an informal definition of some of the mathematical notation being used.

2 Statecharts

Statecharts are a visual specification formalism introduced originally in [3]. They enrich state-transition diagrams with a hierarchical structuring of states and an explicit representation of parallelism and communication among parallel components. States in Statecharts are of three main types: AND, OR and BASIC. Both OR states and AND states consist of a number of substates such that, being in an OR state means being in exactly one of its substates, whereas being in an AND state means being in all of its substates simultaneously. The substates of an AND state are depicted with dashed lines and sometimes are called *parallel* states. On the other hand, the substates of an OR state are shown by solid lines. Each OR state has an immediate substate known as its *default* state, pointed in diagrams by an arrow. For example, in Figure 1 both states A and D are default states for the OR states S_1 and S_2 respectively. A BASIC state is a state with no substates.

Statecharts are a kind of directed graph, with nodes denoting states and arrows denoting transitions. Each transition is labelled in the form: $e[c]/a$, e being an event that triggers the transition, c a condition that guards the transition when e occurs, and a an action which is generated when the transition takes place. Upon its generation, the action a is broadcast to the whole Statechart, triggering, if applicable, other transitions in the system.

Referring to Figure 1, the transition from A to B (in substate S_1) takes place only if A is active, the event e has occurred and the condition c is satisfied. Consequently, the action part a is generated and, hence, the transition from D to E in S_2 takes place provided that the state D is active at the time. Certain parts of the transition label can be left out. An example is the transition from B to A in S_1, where the realisation of the condition c is sufficient to trigger the transition concerned.

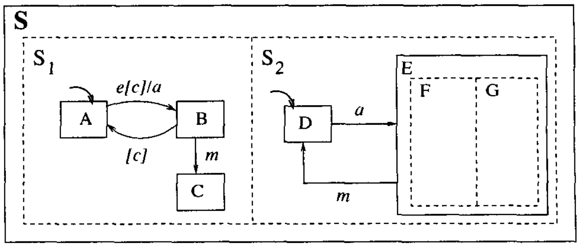


Figure 1. Example of AND/OR decomposition of states

Orthogonality is the term used to describe the AND decomposition of states, where no transition is allowed between the substates of an AND-state. It is a distinctive feature of Statecharts that captures *concurrency* (entering an AND-state means entering its every orthogonal component) and *synchronisation* (a single event can cause simultaneous occurrences of several other events). For example, the occurrence of event m in Figure 1 causes the transformation from B to C in S_1 and the transformation from E to D in S_2 to take place simultaneously. For further reading on the semantics of Statecharts, the reader is referred to [3, 4, 7, 8].

3 Safecharts

Safecharts are an extended version of Statecharts with some unique features for use exclusively in the specification and design of safety critical systems. In focusing on safety concerns, the main strategy of Safecharts is to make a clear separation between the functional and safety requirements so that they can be reviewed independently. In order to achieve this, Safecharts maintain two separate layers, one dealing with functional behaviour of the system, which conforms with the conventional use of Statecharts, and the other dealing with safety behaviour, designed to facilitate the study of safety requirements without being distracted by functional ones. The two layers are superimposed to form a master diagram, which is used in the study of interactions between safety and function and the behaviour of the overall system.

In order to represent failures of the system or any of its components, Safecharts adopt an abstraction of failures

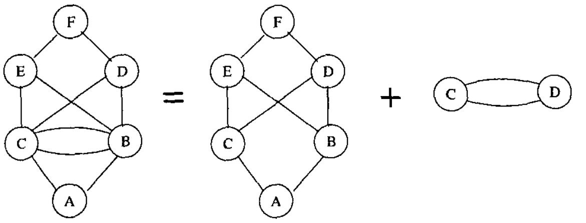


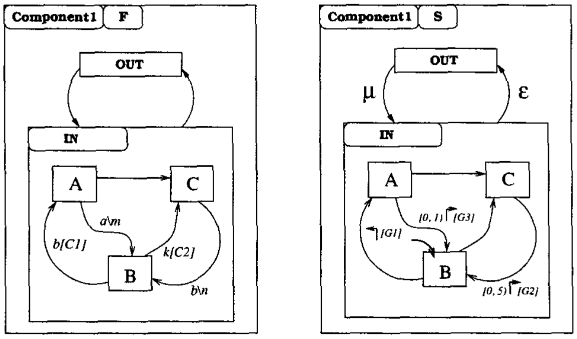
Figure 2. The Decomposition of risk graph of $\sqsubseteq$ relation

based on two generic types of distinguished state: IN, representing a state of normal functioning, and OUT, representing a state of malfunctioning. Two generic types of event, referred to as ε and μ events, are associated with them. An event of type ε from IN to OUT denotes a failure that occurs non-deterministically and can be generated internally or by an external agent. An event of type μ from OUT to IN denotes a maintenance or repair action. It is an external event signifying the return of a component back to its normal functioning state. Different failure modes can be represented by having a set of labelled OR substates in the OUT state of each component and a corresponding set of generic events. These generic events are shown only in the safety layer but, analogous to other events, are broadcast throughout the system and are observable in both layers, potentially generating actions elsewhere in the system.

A fundamental aspect of Safecharts is the explicit ordering of system states according to their risk levels. This is achieved by the use of *risk ordering* relation, denoted by $\sqsubseteq$, on system states. For any two given states s_1 and s_2 belonging to the set of states S, $s_1 \sqsubseteq s_2$ is true if and only if the risk level of s_1 is known to be less than, or equal to, the risk level of s_2. We assume here that domain experts are able to assess, and compare, the risk levels of states either qualitatively or quantitatively. On the basis of the relation $\sqsubseteq$, a risk graph can be constructed over the states in S. Figure 2 illustrates a risk graph of a set of states, along with a decomposition of the relation $\sqsubseteq$ to be described below. Risk graphs are directed graphs. The direction of arcs in Figure 2 is implicit and runs upwards, unless the pair of states concerned lie at the same level, in which case the arcs are bidirectional and are shown with two parallel arcs. Two states s_1 and s_2 are said to be *comparable* in terms of risk if and only if $s_1 \sqsubseteq s_2$ or $s_2 \sqsubseteq s_1$. In other words, risk levels of s_1 and s_2 are assumed to be identical or the risk level of one of the states is lower than that of the other, an implication being that their relative risk levels are known. Otherwise s_1 and s_2 are said to be *non-comparable*. For example, the states comparable by the relation $\sqsubseteq$ in Figure 2 include the pairs of states (A, B), (A, C) and (C, F), but not (D, E).

The relation $\sqsubseteq$ can be decomposed into two relations: a partial order relation $\preccurlyeq$ and an equivalence relation $\approx$ such that $s_1 \preccurlyeq s_2$ is true if and only if the risk level of s_1 is *strictly* lower than that of s_2, unless s_1 and s_2 happen to be the same state, and $s_1 \approx s_2$ is true if and only if the risk levels of s_1 and s_2 are known to be, or are assumed to be, identical. The partial order relation $\preccurlyeq$ is acyclic and can be defined as the reflexive closure of an immediate precedence relation $\prec$.

In diagrams of risk graphs the higher risk states are placed higher in the graph than the lower risk states. As a convention, the relative positions of states in the functional and safety layers are maintained at the same level as in the risk graph. This is to maintain the consistency of both layers and to facilitate their integration from a presentation point of view. Referring to Figure 3, according to the positions of substates A, B and C of the state IN, it is the case that B $\preccurlyeq$ A, A $\approx$ C and B $\preccurlyeq$ C.



a) Functional layer

b) Safety layer

Figure 3. An example of a Safechart

Based on the risk ordering relation, every transition in Safecharts is classified according to the nature of risk posed by it. Representing a given transition in the form of $x \rightsquigarrow y$ (with the arrow $\rightsquigarrow$ leading from the source state x of the transition to its target state y), $x \rightsquigarrow y$ is said to be a *safe* transition if $y \preccurlyeq x$, an *unsafe* transition if $x \preccurlyeq y$, and a *neutral* transition if $x \approx y$. Safecharts do not permit the introduction of transitions between non-comparable states. This is because of the difficulty in determining the nature of risk of such transitions without knowing the relative risks involved with their source and target states. The above restriction is intended as a discipline so that the designers are compelled to resolve the risk level of any non-comparable states prior to introducing transitions between them.

Transitions in Safecharts have an extended labelling scheme: $e[c]/a[l,u)\Psi[G]$, where e, c and a are as in conventional Statecharts discussed in Section 2, $[l,u)$ is a right–open time interval from a lower bound l to an upper bound u, Ψ stands for one of the alternative symbols: $\Lsh$

and ↱, and $[G]$ is a safety clause. The time interval $[l, u)$ imposed on a transition t appears only with ↱ and indicates that t does not execute until at least l time units have elapsed since its most recent enabling and must execute strictly within u time units. The additional components of the labelling together constitute a safety enforcement pattern: a label of the form $e[c]/a$ ↰ $[G]$ on a transition t signifying a *prohibition* enforcement on an unsafe transition t and a label of the form $e[c]/a[l, u)$ ↱ $[G]$ a *mandatory* enforcement on a safe transition t. The former indicates that t is not allowed to execute if the safety clause G happens to be true and the latter indicates that t is enabled and forced to execute within $[l, u)$ whenever G holds, even in the absence of a triggering event.

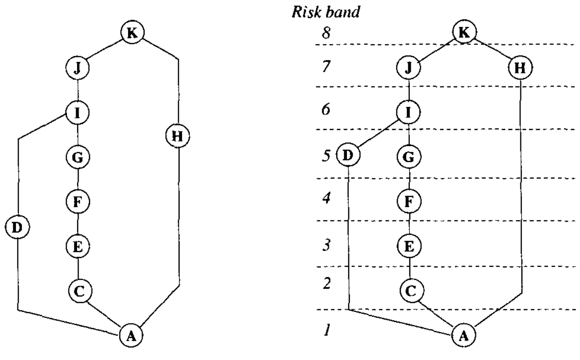
Safecharts propose a new priority scheme for the execution of conflicting transitions in the event of any non-determinism. Non-determinism is resolved in favour of safe transitions that lead the system to a relatively safer state. The term *safe non-determinism* applies to situations where any inevitable non-determinism has no adverse repercussion on safety.

4 Risk Bands

4.1 Motivation for Risk Bands

Exclusion of transitions between non-comparable states can be too restrictive and may result in prohibiting many functionally desirable transitions. This might be the case if the risk graph happens to be an inappropriate description of the risks involved. Such a graph may contain states that are non-comparable with a large number of other, but mutually comparable, states. A reason for this can be an inconsistency in the degree of the risk assessment process as applied to the states in different parts of the risk graph. An example is given in Figure 4(a) where, relative to other states, the state H may have received less attention in the risk assessment, resulting in it becoming non-comparable with many other states in the graph, namely, the states C, D, E, F, G, I and J. Similarly, the state D is non-comparable with the states C, E, F, G and H. Consequently, Safecharts as described in [2] do not allow any transition between them, for instance, a transition such as H $\rightsquigarrow$E.

In order to overcome the above problem, we introduce here the concept of *risk band.* Our definition of risk band is such that each state in the risk graph belongs to a unique risk band and that two arbitrary states s_1 and s_2 belong to the same risk band, if and only if they are known to be, or are assumed to be, identical in risk level, that is $s_1 \approx s_2$, or they are non-comparable according to $\sqsubseteq$. Alternatively, two arbitrary but distinct states s_1 and s_2 belong to different risk bands if and only if they are related by the relation $\precsim$. The idea behind risk bands is to allow only transitions between states that belong to different risk bands or states that are comparable by the relation $\approx$. The nature of risk of such transitions depends exclusively on the risk bands of their source and target states. For example, an arbitrary transition $s_2 \rightsquigarrow s_1$ is said to be *safe* if s_1 resides in a lower risk band than that of s_2.



a) without the notion of risk band

b) with the notion of risk band

Figure 4. A risk graph of states

Introduction of risk bands overcomes the problem of ambiguity in the interpretation of the risk levels of non-comparable states by providing a safety oriented default interpretation. The designers are therefore left with the responsibility of accepting the default interpretation or, in the event of any disagreement, refining the relative risk levels of states concerned. The introduction of risk bands is viewed as a means of enhancing the overall safety analysis process as applied to system states.

4.2 A Definition of Risk Bands

Risk bands are numerically indexed consecutively from 1 to some n. The construction of a risk graph of a set of states S incorporating risk bands (referred to later as 'banded risk graph') is based on the risk ordering relation $\sqsubseteq$ but obeys additionally the following rules:

(i) States in the highest risk band n consists of exactly

   (a) maximal elements (states) in the partial order relation $\precsim$ but excluding those elements, if any, which are comparable by $\approx$ with the rest of elements in $\precsim$, and

   (b) elements which are comparable by $\approx$ with those elements defined in (a) above.

(ii) Any state s with just a single immediate successor state in i th risk band according to $\prec$ is in $(i-1)$th risk band. If the state s, however, has more than one immediate successor state, it has a risk band index one less than the lowest of the risk band indices of its immediate successor states.

(iii) For any states s_1 and s_2, if $s_1 \approx s_2$ then both states s_1 and s_2 are in the same risk band.

The position of every state in the risk graph is determined by the above rules. The application of the above rules on the set of states shown in Figure 4(a) results in a banded risk graph shown in Figure 4(b). As can be seen from this figure, the states H and D now have risk bands 7 and 5 respectively. In addition, the resulting banded risk graph extends the scope of permitted transitions. For example, transitions between the state H and other states belonging to different risk bands can now be permitted. Furthermore, the prohibited transitions are now reduced to only four transitions, namely, $H \rightsquigarrow J$, $J \rightsquigarrow H$, $D \rightsquigarrow G$ and $G \rightsquigarrow D$.

The risk bands of states can be defined formally as follows. Letting the set of terminal states (states which do not precede in other state in the immediate precedence relation $\prec$) appearing in the highest risk band be $\mathcal{T}$, we have

$$\begin{aligned} \mathcal{T} &= (\operatorname{ran} \prec) - (\operatorname{dom} \prec) - \\ &\quad \{s_1 \mid \exists\, s_2 \bullet s_1 \approx s_2 \land s_2 \in \operatorname{dom} \prec\} \qquad (1) \end{aligned}$$

[See Appendix A for the mathematical notation.] For example, referring to Figure 4(b), $\mathcal{T} = \{K\}$. Let us also define a function $\mathcal{B} : N \rightarrow \mathbb{P}\, S$, where N is a subset of natural numbers forming a contiguous segment of numbers from 1 to some n, n as mentioned above being the highest risk band index, and S is the set of system states. $\mathcal{B}$ is defined such that for a given risk band $b \in N$, $\mathcal{B}(b)$ returns a set of all states in the risk band b. The function $\mathcal{B}$ may be defined recursively as

$$\mathcal{B}(n) = \mathcal{T} \cup \{s_1 \mid \exists\, s_2 \bullet s_1 \approx s_2 \land s_2 \in \mathcal{T}\} \qquad (2)$$

$$\mathcal{B}(m) = \mathcal{NC}(m) \cup \{s_1 \mid \exists\, s_2 \bullet s_1 \approx s_2 \land s_2 \in \mathcal{NC}(m)\} \qquad (3)$$

where $1 \leq m < n$ and $\mathcal{NC} : N \rightarrow \mathbb{P}\, S$ is a function that takes a risk band index m and returns a set of all non-comparable states in the mth risk band. For $1 \leq m < n$, $\mathcal{NC}$ can be defined as

$$\begin{aligned} \mathcal{NC}(m) &= \operatorname{dom}(\prec \rhd \mathcal{B}(m+1)) - \\ &\quad \operatorname{dom}((\prec^+ - \prec) \rhd \mathcal{B}(m+1)) \qquad (4) \end{aligned}$$

In other words, $\mathcal{NC}(m)$ is the set of all states such that, according to $\prec$, each one of its state has at least one immediate successor in $(m+1)$th band but none below it. In order to illustrate the above definition of risk bands, consider the following example. Let us assume that the assessment of relative risk levels of a set of states $\{A, B, C, D, E, F, G, H, I\}$ results in the following information: $C \prec A$, $B \approx A$, $D \prec A$, $E \prec D$, $F \prec E$, $F \prec C$, $D \approx I$, $G \prec E$, $G \prec I$ and $H \prec G$. By applying the above rules we then obtain a banded risk graph shown in Figure 5. It can be seen from the risk graph that, although the relative nature of risk of states C and E was not covered by the outcome of the given risk assessment, they are now located in different risk bands. Hence, if necessary, the safe transition $C \rightsquigarrow E$ and the unsafe transition $E \rightsquigarrow C$ can now be allowed.

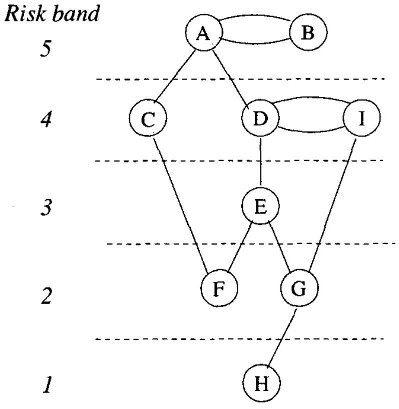


Figure 5. An example of a banded risk graph

4.3 Risk Distance and Safe Non–determinism

Compared to other variants of Statecharts, conflicting transitions in Safecharts are given priorities according to the risk level of their target states. The approach given in [2] states that lower the risk level of the target state of a transition, higher the priority it enjoys over other conflicting transitions. Figure 6 shows a possible collection of transitions defined over a subset of the states (excluding I) in the risk graph shown in Figure 5. Let us assume that the transitions t_1 and t_8 are in conflict, that is, they are enabled simultaneously, then according to the approach in [2], t_8 is given a higher priority. However, the above approach fails to resolve the non-determinism between t_3 and t_4 if they were to be in conflict. This is because both transitions are safe and their target states C and E are non-comparable. The above situation is regarded as a safe non-determinism in [2]. However, it can be seen from Figure 5 that the state E resides in a lower risk band than that of the state C and, therefore, from a safety point of view, selecting t_4 for execution is safer than selecting t_3.

The advantage of risk bands is that they provide us with a way to measure the relative position of states in the risk graph and, hence, differentiate between transitions of the same risk nature as resulting from $\sqsubseteq$. This is achieved through the concept of *risk distance* of transitions. Risk distance of a transition t is the number of band boundaries

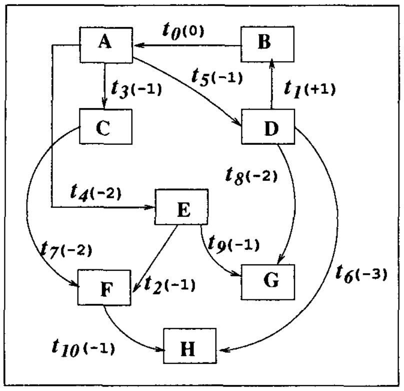


Figure 6. A collection transitions using a subset of states in Figure 5

between the source and target states of t. The risk distance of a given transition is calculated by subtracting the risk band index of the source state from that of the target state. It follows from the above that safe transitions have negative risk distances, unsafe transitions positive risk distances and neutral transitions zero risk distances. In Figure 6, risk distances are shown in parentheses beside the transition labels. It can be seen from the figure that both t_3 and t_4 are safe transitions but, according to their risk distances, t_4 is *safer* than t_3. Risk distances can now be used as a new priority scheme for transitions so that smaller the risk distance of a transition higher is the priority given to it. Thus, referring to Figure 6, the transition t_4 enjoys a higher priority over t_3.

However, the above approach is unable to resolve the non-determinism with respect to transitions t_3 and t_5 in Figure 6. This is because both transitions have the same risk distance. Nevertheless, the above non-determinism can be resolved by considering cumulative risk distances of *future transitions* of conflicting transitions. Informally, future transitions of a transition t are the transitions *outgoing* from the target state of t. The set *outgoing*(s), defined as

$$outgoing(s) = \{t \mid source(t) = s\} \tag{5}$$

where, for $t = x \rightsquigarrow y$, $source(t)$ being x, is the set of all transitions from a given state s. All possible future transitions of a transition t which can be taken in the next step can be defined as the set $\mathcal{FT}(t)$

$$\mathcal{FT}(t) = \{t' \mid t' \in outgoing(target(t))\} \tag{6}$$

where, for $t = x \rightsquigarrow y$, $target(t)$ being y. For example, in Figure 6, t_1 has one future transition, namely t_0, whereas t_0 has three future transitions, namely t_3, t_4 and t_5. Note that, as defined above, $\mathcal{FT}(t)$ is the set of transitions that can be taken in the next step after t. However, an alternative would be to consider instead an appropriate longer path of future transitions.

It is worth noting that the resolution of non-determinism by considering future transitions is a decision about which transition to be executed among the conflicting ones, and not about which sequence of transitions involving the possible future transitions to be executed. If one of the conflicting transitions has more than one future transition then we only consider the future transition with the smallest risk distance. Returning to the non-determinism between the transitions t_3 and t_5 in Figure 6, t_3 has only one immediate future transition, namely t_7 whereas, t_5 has two, namely t_6 and t_8. Among the future transitions of t_5, t_6 is the one to be considered, since it has a smaller risk distance. Thus, the cumulative risk distances of conflicting transitions t_3 and t_5 over their future transitions are -3 and -4 respectively and, therefore, t_5 will be given higher priority over t_3.

Nevertheless, in some situations, non-determinism will continue to persist even when considering paths of future transitions. For example, if we were to introduce a new transition t_{11} from the state C to the state H in Figure 6, then the system will non-deterministically select one of the conflicting transitions t_3 and t_5 to be executed. We consider this kind of non-determinism as a *safe non-determinism* because all possible outcomes are identical in relation to safety or risks involved, relative risk being determined in the approach proposed here on the basis of risk bands.

5 A Case Study

In order to illustrate Safecharts, this section considers as a case study a reactor of a nuclear power plant. This choice, and the aspects covered, are inspired by the accident at the Three Mile Island (TMI) nuclear power plant in Pennsylvania on 28th March, 1979.

The reactor, which is a thick steel vessel, is situated at the heart of the nuclear plants and houses a bundle of fuel rods called the *core*. The intensity of the nuclear reaction taking place in the core can be controlled by lowering/raising *control rods* into/out of the core. In order to protect the environment from possible radioactive contamination, pressurised water reactors consist of two independent closed loops of circulating water called *primary* and *secondary* circuits. The former is situated completely within the containment building while the latter completely outside the reactor. Primary circuit has a dual role. On the one hand, as its water circulates through the core the primary circuit extracts heat from the nuclear fission and, on the other hand, it plays the role of a coolant preventing the fuel rods from overheating. The purpose of the secondary circuit is to extract heat from the water flowing through the primary circuit but without coming into direct contact with the latter. This process of interaction between the two circuits is known as *heat exchange*. The boiling of water in the secondary circuit gives rise to steam, the source of energy that turns the turbines and thus creates electricity. A simplified sketch of a plant is given in Figure 7.

The loss of coolant accident (LOCA) is one of the possible accidents that may take place during the operation of a nuclear plant, leading to extreme temperatures and pressures inside the reactor that may progressively lead to cracks and eventual explosion. In the case of such an event, operation of a pressure relief valve (PRV), which is located at the top of the pressuriser, is designed to relieve the pressure, by letting the coolant to flow out into a safe drainage system.

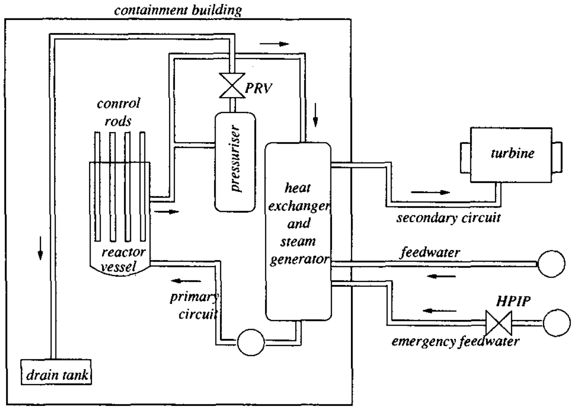


Figure 7. A sketch of a nuclear plant

In the TMI nuclear plant, the accident originated when the main feed-water pumps stopped supplying water from the secondary circuit to the steam generator. As a result of no steam being produced, the turbine was first shut down and, consequently, the process of heat exchange was effectively prevented from taking place. The temperature of the coolant increased and the pressure inside the pressuriser rose to an abnormal level. In response, as was intended, PRV was opened and steam and water began to flow out and water through drain pipes into a tank on the floor of the containment building. In addition, the control rods dropped (*scrammed*) into the core halting the nuclear fission. With the PRV being left open and the control rods being scrammed, the pressure inside the pressuriser fell to normal level. At this time, PRV should have closed. However, PRV remained stuck open instead, although the light on the control panel indicated that the electrical solenoid operating the valve had functioned. It has been established later that the solenoid had actually withdrawn alone, leaving the PRV stuck open and, at the same time, misleading the operators into thinking that the PRV has actually closed. The stuck-open valve caused the pressure in the system to continue to drop. If the operators were able to establish in time that the PRV was open at the time, they could have closed another backup valve and left on the high-pressure-injection pump (HPIP) which started working after the main feed-water pumps stopped working. The accident involved other failures and is too complex to be modelled fully within the scope of this paper. For more details about the accident and the different failures that occurred, the reader is referred to [1, 6]. Thus, we focus our discussion on one particular scenario, namely the failure of the PRV.

In our model, we assume that the PRV and its solenoid are two components of the pressuriser and, therefore, modelling the above failure can be achieved by modelling the behaviour of the pressuriser. Figure 8 presents only an integrated view of the functional and safety layers of the pressuriser. As can be seen form this figure, the pressuriser is an AND state, consisting of five OR substates, namely Pressure, PRV, Sol-sensor, Solenoid and Coolant. The increase of the pressure inside the pressuriser is modelled by means of internal events *up*, which are not directly controllable. If the pressure inside the pressuriser rises beyond a certain level, then the state Excessive is entered causing the generation of a so–called static reaction *drawR, sc!(scram, t_0)*. A static reaction of a state is a conventional Statecharts action [4] that is carried out when the state it resides in is active. The event *drawR* is intended to cause the solenoid to move its position from Left to Right, whereas the event *scram* is broadcast through the system causing the control rods to drop into the core. The event *scram* belongs to the model of control rods (not shown here) and is scheduled to occur t_0 time units after the most recent time when the Excessive state was entered.

In normal situations, the solenoid is expected to function correctly and, therefore, Solenoid would be in Sol-IN state. Upon the generation of *drawR*, the transition Left⇝Right in the Solenoid state takes place and, at the same time, generating event $open_v$ in the PRV. The event $open_v$ triggers the transition Close⇝Open in PRV, which has to execute within a specified interval of t time units after becoming enabled. However, if the solenoid happens to be out of order, that is, Solenoid in Sol-OUT substate, then the occurrence of the event *drawR* will cause the solenoid to move to Right state, as before, but without actually generating the event $open_v$. Instead, upon entering the state Right in Sol-OUT the generic event ε_v would be triggered, causing the PRV to move to either Stuck-open or Stuck-close, depending on its current position. This is an instance of failure propagation.

The substates Sol-moved and Sol-still of Sol-sensor represent what is shown on the control panel of the plant and indicate, respectively, whether the solenoid has just moved or remains, or has just become, stationary. Since every functional transition in substates within Solenoid generates the event a and since the event a triggers the transition from

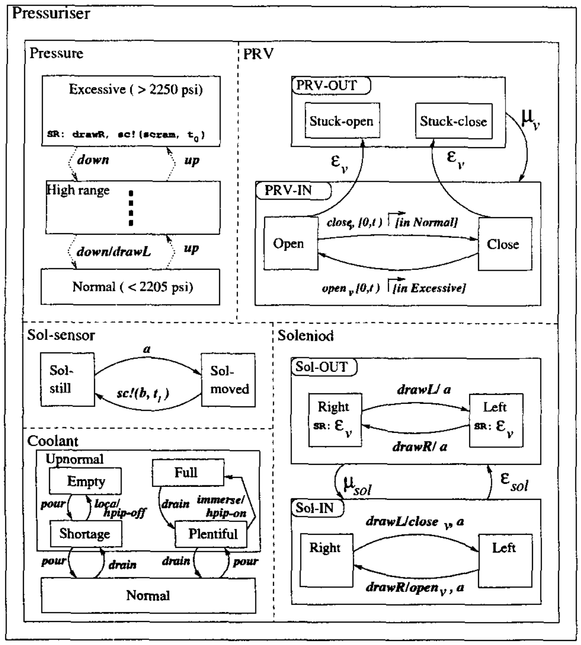


Figure 8. Safechart modelling the pressuriser

Sol-still to Sol-moved in Sol-sensor, the sensor attached to the solenoid will always indicate whether the solenoid has actually moved, whether from Left to Right or in the opposite direction. Once in Sol-moved, Sol-sensor reverts back to Sol-still after some t_1 time units. As a result of not attaching the sensor directly to the actual PRV but to the solenoid instead, and because of the non–deterministic nature of the failure event ε_{sol} in Solenoid, indicators in the control panel may not necessarily show the actual position of the valve but its intended position. In the event of such a failure, the operators are therefore exposed to a potential misrepresentation of the actual valve position.

Let us now consider a revised representation of PRV designed to overcome the above problem. It is shown in Figure 9. According to the revised representation, the sensor is attached directly to the PRV and not to the solenoid. The other substates of Pressuriser in Figure 9 remain basically unchanged. When the generic event ε_v occurs, the PRV moves to either Stuck-open or Stuck-close depending on its current position, generating the event $fail_v$ and, thus, moving the sensor to the PORV-OUT state, indicating a faulty valve. An appropriate action, such as opening of another backup valve or leaving on the HPIPs, can then be taken. Furthermore, the failure of the sensor is also modelled by two special states, namely P-Sen-OUT and P-Sen-IN. Should the sensor fail, that is, should the generic event ε_{sen} occur, an alarm could be triggered, indicating that no reliance is to be placed on any sensor output.

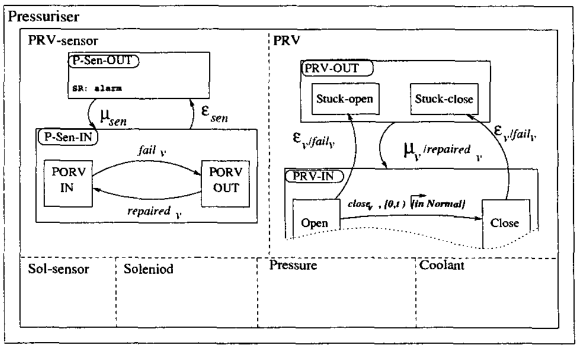


Figure 9. An alternative model of PRV

6 Future Work

As the environment around the system changes, the system responds to such changes by modifying its behaviour accordingly. As a result, a state which is considered a safe state in a particular situation may not be a safe state in another. In order to capture these changes, and the extent of their effect on system states, the current research is considering the construction of *dynamic* risk graphs. The relative risk levels of states in the dynamic risk graph are not fixed, but may vary from situation to situation depending on the system state at the given point in time. The concept of *situational event* is intended to address such dynamic aspects and to facilitate the representation of such changes and their consequences in the behavioural semantics of Safecharts.

7 Conclusions

Exclusion of transitions between non-comparable states in [2] is eliminated, and thus leaving many functionally desirable transitions at the disposal of the designer. An inconsistency in the degree of the risk assessment process, or the lack of information about the relative nature of risk of the system states, may result in an inadequate definition of risk graphs. In this paper, we have introduced the concept of risk band as a way of enhancing the risk ordering relation $\sqsubseteq$. We have shown how risk bands provide an unambiguous interpretation of the risk level of system states and, hence, extended the scope of permitted transitions between them. Risk bands offer a further reduction in the extent of non-deterministic behaviour when the system is forced to select between simultaneously enabled transitions. This is achieved through the concept of risk distance of conflicting transitions and, if non-determinism continues to persist despite this, through the concept of cumulative risk distance of their future transitions. The effectiveness of Safecharts as a design tool has been illustrated using a realistic scenario drawn from a documented actual accident, namely, that at the Three Mile Island in 1979.

A Mathematical Notation

Below are informal definitions of certain mathematical notations used in this paper.

$\{\ldots\}$	Set definition by enumeration
$\{s \mid pred(s)\}$	Implicit set definition (the notation given defines a set of elements, each defined by the term s satisfying a predicate $pred(s)$)
$\in$	Set membership
$\exists\, x \bullet pred(x)$	Existential quantification of a variable x satisfying a predicate $pred(x)$
$\cup$	Set union
$-$	Set difference
$\mathbb{P}$	Power set ($\mathbb{P}\, S$ denotes the set of all subsets of the operand set S)
dom	Domain of a relation
ran	Range of a relation
$\rhd$	Restriction of the range of a relation R to a set S. It is written as $R \rhd S$
R^+	Irreflexive transitive closure of the relation R on a set.

References

[1] V. Bignell and J. Fortune. *Understanding Systems Failures.* Manchester University Press, 1984.

[2] H. Dammag and N. Nissanke. Safecharts for specifying and designing safety critical systems. In *18th IEEE Symposium on Reliable Distributed Systems, Lausanne*. IEEE, Oct. 1999.

[3] D. Harel. *Statecharts:* a visual formalism for complex systems. volume 8 of *Science of Computer Programming*, pages 231–274. North-Holland, 1987.

[4] D. Harel and A. Naamad. The STATEMATE semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology*, pages 293–333, 1996.

[5] D. Harel, J. P. Schmidt, and R. Sherman. On the formal semantics of Statecharts. In *2nd IEEE Symposium on Logic in Computer Science*, pages 54–64, 1985.

[6] N. G. Leveson. *Safeware – System Safety and Computers.* Addison-Wesley Publishing Company, 1995.

[7] A. Pnueli and A. Shalev. What is in a step: On the semantics of Statecharts. In *Symposium on Theoretical Aspects of Computer Software*, volume 526 of *LNCS*. Springer-Verlag, 1991.

[8] M. von der Beeck. A comparison of Statecharts variants. volume 863 of *LNCS*, pages 128–148. Springer–Verlag, 1996.