Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

Automata-Based Reasoning for Short Circuit Diagnosis in Power Transmission Networks

P. Baroni[†], G. Lamperti[†], P. Pogliano[‡], G. Tornielli[‡] & M. Zanella[†]

[†] *Dipartimento di Elettronica per l'Automazione, Università degli Studi di Brescia, Italy*

Email: {baroni, lamperti, zanella}@bsing.ing.unibs.it

[‡] *CISE SpA, Milano, Italy*

EMail: {paolo, gt}@sia.cise.it

Abstract

This document introduces an original technique for short circuit diagnosis in power transmission networks. The basic idea is that the diagnostic task can be conveniently viewed as a multi-layer interpretation process of available observations, at the end of which, as a sort of harvest, relevant diagnostic information can be gathered to generate the required diagnosis. Three different interpretation layers are considered: *local interpretation*, *global interpretation*, and *heuristic interpretation*. This paper presents the first (application-independent) two layers. The local interpretation focuses on the behaviour of single protection components that are distributed in the power network and operate when a short circuit occur. The global interpretation provides a global behaviour of the protection apparatus by combining consistent local behaviours so that given *interface constraints* among components are met. Finally, the heuristic interpretation is meant both to shrink the cardinality of the global interpretation, by eliminating a number of spurious global behaviours on the basis of application-dependent *heuristic constraints*, and to eventually localize the short circuit and possibly faulty protection components within the transmission network. The implementation of the proposed technique is under way. The resulting system is going to be tested by ENEL, the Italian electricity board, using the transmission network of part of northern Italy.

1 Introduction

To avoid service interruptions caused by short circuits in a power transmission network, it is necessary to isolate faults as soon as possible. For this reason, there are a number of protection mechanisms distributed over the network. The protection system is in charge of detecting dangerous conditions, of disconnecting a component (such as a line, a bus, a transformer or a generation group) as soon as it begins to operate in a dangerous condition, and of keeping in operation non-faulty components as much as possible, in order to avoid a black-out. This is achieved by tripping the circuit breaker associated with each protection. Each protection has to protect mainly one component, but must also operate as a backup to other protections nearby.

When intervening, all the protections and the breakers send logical signals reporting their operation: these are recorded by an event recorder and transmitted to a *Regional Control Center* (RCC) where they are used for fault localization. Records, called *messages*, consist of a unique address of the event source, an event code, and possibly a timestamp. Operators of the RCC have to decide as soon as possible (possibly within one minute) where the fault is located and what recovery actions have to be applied. In the current practice, due both to time constraint reasons and to the large amount of messages received, only the current status of the breakers is considered by human operators for fault localization, but this is not generally sufficient. Consequently, the need of an automated support tool for such task is strongly recognized.

The basic idea for short circuit diagnosis is that the diagnostic task can be conveniently viewed as a multi-layer interpretation process of available observations, at the end of which relevant diagnostic information can be gathered to generate the required diagnosis. Three different interpretation layers are considered: *local interpretation*, *global interpretation*, and *heuristic interpretation*. However, in the context of this paper the heuristic layer will not be presented. The implementation of the proposed technique is under way. The resulting system is going to be tested by ENEL, the Italian electricity board, using the transmission network of part of northern Italy.

2 Reactive models

The basic idea underlying the proposed diagnostic technique is to build a model of the protection apparatus in terms of interconnected protection components. Each protection component is modeled by means of a finite state machine (FSM), called *reactive model*, which is driven by a series of *input* events and generates some other *output* events when transitions between states (called *reactive states*) occur. For example, a distance protection is started when the impedance on the relevant line goes under a certain threshold (input event). After that, the protection is expected to change state at fixed time intervals (clock input events) and, possibly, to

trip the associated breaker (tripping output event). Note that the tripping output event for the protection generates an input (tripping) event for the breaker: we say that the output event is *exported* by the protection and is *imported* by the breaker. Reactive states can be either *steady* or *unsteady*. This classification improves the semantics of reactive models and indirectly poses a number of constraints on the behaviour of protection components which can be conveniently exploited by the interpretation algorithm. A *null event* ϵ is a formal artifice to denote the absence of events.

Formally, a reactive model M is a record of four elements: $M = (\Sigma, I, O, \delta)$, where:

- Σ is the set of the reactive states,
- I is the set of input events,
- O is the set of output events, and
- δ is the transition function, $\delta : \Sigma \times (I \cup \epsilon) \times (2^O \cup \epsilon) \rightarrow \Sigma$

where 2^O is the power-set of O. This means that a transition T from state S_1 to state S_2 is in general triggered by an input event i and generates, before changing state, the list of output events $< o_1, o_2, \ldots, o_n >$. This is denoted by $T = S_1 \overset{i \,|\, o_1, o_2, \ldots, o_n}{\longrightarrow} S_2$. Furthermore, $\Sigma = \Sigma_s \cup \Sigma_u$, $\Sigma_s \cap \Sigma_u = \emptyset$, where Σ_s and Σ_u denote the set of steady and unsteady states respectively.

Example 1 $M = (\Sigma, I, O, \delta)$, $\Sigma = [S_0, S_1, S_2]$, $I = [i_1, i_2, i_3]$, $O = [m_1, m_2, m_3]$, $\delta = [S_0 \overset{i_1 \,|\, m_1}{\longrightarrow} S_1, S_1 \overset{i_1 \,|\, m_2}{\longrightarrow} S_1, S_1 \overset{i_2 \,|\, m_2}{\longrightarrow} S_2, S_2 \overset{i_3 \,|\, m_3}{\longrightarrow} S_0]$, $\Sigma_s = [S_0]$, and $\Sigma_u = [S_1, S_2]$.

The specification of the reactive model of each type of protection component and the instantiation of such classes of components into a given network topology yields the whole model of the diagnosed system.

Each reactive model describes both the correct and faulty behaviour of a class of protection components. Furthermore, the model allows for uncertainty due to possible loss of messages. Specifically, if during a transition $T_1 = S_1 \overset{i \,|\, m}{\longrightarrow} S_2$, message m is possibly lost, then the δ function will include an additional transition $T_2 = S_1 \overset{i \,|\, \epsilon}{\longrightarrow} S_2$. Note that this yields non-determinism, as from state S_1 either T_1 or T_2 can be triggered by the same input event i.

In the sequel, we assume the completeness of each reactive model, that is to say, the reactive model is assumed to describe each possible reaction.

3 Misbehaviours, observations, and histories

When a short circuit occurs, we say there is a *misbehaviour* μ of the transmission network. The part of the network which reacts to a misbehaviour

μ is called the *misbehaviour extent* of μ, and is denoted by $extent(\mu)$. After the specification of each reactive model, it is possible to interpret a given sequence of observations, the messages, in order to eventually find out the sequence of transitions covered by the involved protection components during μ. When a short circuit occurs, each actual protection component is expected to react in a way that corresponds to an instantiation of the model, called the *local history* of the component. In practice, a local history is a sequence of transitions within the reactive model. The initial and final states of the history are required to be steady. The local history is derived on the basis of the *local observation* $obs(P)$ of the component P, namely a list of messages, $obs(P) =< m_1, m_2, \ldots, m_n >$. A local observation which is empty is called a null observation and is denoted by ϵ.

Definition 1 (*local history*) Let M be a reactive model of a protective component. A *local history* of M is a (possibly empty) sequence $h =< T_1, T_2, \ldots, T_n >$ of transitions in M so that h conforms to the following *morphology constraints*:

1. *Determinism.* Each transition $T_i, i = 1..n$, is adorned with at most one allowed input event,

2. *Contiguity.* For each pair of contiguous transitions T_i, T_{i+1} in h, the final state of T_i coincides with the initial state of T_{i+1}, and

3. *Stability.* Both the initial state of T_1 and the final state of T_n are in Σ_s.

If the sequence of transitions is empty, h is called a *null history*, and is denoted by ϵ.

Definition 2 (*local observation*) Let M be a reactive model of a protection component P. A *local observation obs* of P, $obs(P) =< o_1, o_2, \ldots, o_n >$, is a sequence of temporally ordered observable output events, generated by a local history of M.

Definition 3 (*global history*) Let $P_1, P_2, \ldots, P_m$ be a set of protective components involved in a misbehaviour μ. A *global history* H of μ, $H(\mu) = (h_1, h_2, \ldots, h_m)$, is the aggregation of the local histories $h_1, h_2, \ldots, h_m$ relevant to $P_1, P_2, \ldots, P_m$ respectively.

Definition 4 (*global observation*) Let $H(\mu) = (h_1, h_2, \ldots, h_m)$ be the global history of μ. A *global observation* OBS of μ, $OBS(\mu) = < o_1, o_2, \ldots, o_g >$, is a sequence of observable output events generated by $H(\mu)$.

4 Local Interpretation

The first step towards the diagnosis of the fault circuit is represented by the local interpretation of every local observation. A local interpretation algorithm (*lia*) is provided. The algorithm takes as input a local observation $obs(P)$ and generates a set h^* of consistent local histories on the basis of the relevant reactive model. In general, a local interpretation gives rise to several consistent local histories, namely: $h^* = lia(obs(P)) = [h_1, h_2, \ldots, h_n]$. In what follows we provide a list concepts relevant to the local interpretation algorithm.

Definition 5 (*abduced transition*) Given an observable output event o associated in the reactive model M with a transition T. T is called the *abduced transition* of o. The set of abduced transitions, $abd(o)$, associated with o is called the *abduced transition set*.

Definition 6 (*abduced chain*) Let $obs(P) =< m_1, m_2, \ldots, m_n >$ be an observation of a protection component P. The list of abduced transitions isomorphic to $obs(P)$ is called the *abduced chain* of the observation of P, and is denoted by $abd(obs(P))$.

Example 2 Considering Example 1 and the following observation for protective component P: $obs(P) =< m_1, m_2, m_2, m_2, m_3 >$, we will have:

$$\begin{aligned} abd(obs(P)) &= < abd(m_1), abd(m_2), abd(m_2), abd(m_2), abd(m_3) > \\ &= < \{T_1\}, \{T_2, T_3\}, \{T_2, T_3\}, \{T_2, T_3\}, \{T_4\} > . \end{aligned}$$

Definition 7 (*abduced local history*) A sequence of abduced transitions relevant to an observation $obs(P)$ is called an *abduced local history* if and only if it conforms to the morphology constraints. The set of abduced local histories relevant to $obs(P)$ is called the *abduced local history set*.

Therefore, given a protective component P and a local observation $obs(P)$, the main problem for the interpretation algorithm is the computation of the abduced local history set starting from $obs(P)$.

Example 3 To show this, consider the reactive model M given in Example 1 and the local observation of Example 2. Notice that two abduced transitions (T_2 and T_3) are inferred from the same observation m_2. Thus, in principle, several local histories might be abduced from $obs(P_2)$, but only those sequence of transitions which are actually consistent with the morphology constraints are local histories. In practice the local interpretation algorithm is required to conceptually make a series of simplifications aimed at pruning the number of local histories by applying the morphology constraints on the abduced chain of transitions (the first morphology constraint,

determinism, is automatically met by the abductive mechanism). Therefore, applying the *contiguity* morphology constraint to the above abduced chain yields the following new simplified chain of transitions:

$$S_0 \xrightarrow{i_1 \mid m_1} S_1 \xrightarrow{i_1 \mid m_2} S_1 \xrightarrow{i_1 \mid m_2} S_1 \xrightarrow{i_2 \mid m_2} S_2 \xrightarrow{i_3 \mid m_3} S_0$$

Informally, this is achieved by comparing each pair of contiguous sets of abduced transitions and removing those transitions which do not match, namely those transitions in the first abduced transition set for which the final state does not correspond to any of the initial states in the second abduced transition set. More formally, for each contiguous pair $< abd(o_i), abd(o_{i+1}) >$ of abduced transition sets, let $To(abd(o_i))$ denote the set of final states of $abd(o_i)$, whereas $From(abd(o_{i+1}))$ denote the set of initial states of $abd(o_{i+1})$. Let T_{from} and T_{to} stand for the initial and final state of transition T respectively. Then, a transition T is removed if and only if:

$$(T \in abd(o_i) \wedge T_{to} \notin From(abd(o_{i+1}))) \vee (T \in abd(o_{i+1}) \wedge T_{from} \notin To(abd(o_i))).$$

In general it is necessary to apply the simplification step more than once, namely until no other simplifications are possible. Finally we observe that the application of the *stability* morphology constraint does not make any change on the found history, as both the initial state and the final state coincide with the steady state S_0. Thus, the set of abduced local histories will be: $h^* = lia(obs(P)) = [h] = [< T_1, T_2, T_2, T_3, T_4 >]$.

Definition 8 (*silent vs observable transition*) Let M be a reactive model. A transition T in M for which no observable output event is defined is called a *silent transition*. A transition which is not silent is called an *observable transition*.

Intuitively, models involving silent transitions should entail additional complexity to the interpretation task, as silent transitions introduce uncertainty in the abductive mechanism. In practice we cannot assume any more that the chain of transitions composing the abduced local history is isomorphic to the observable output events in an observation.

Definition 9 (*macrotransition*) Let M be a reactive model. A chain $T^+ = S_1 \rightarrow S_2 \rightarrow \cdots \rightarrow S_n$ of contiguous transitions in M is called a *macrotransition*, and is denoted by $S_1 \hookrightarrow S_n$. If all the transitions involved in T^+ are silent, $S_1 \hookrightarrow S_n$ is called a *silent macrotransition*.

Definition 10 (*bound macrotransition*) Let $T^+ = S_1 \hookrightarrow S_n$ be a macrotransition of a reactive model M. T^+ is called a *bound macrotransition* if and only if the following three conditions hold:

1. either S_1 is the final state of an observable transition or $S_1 \in \Sigma_s$,
2. either S_n is the initial state of an observable transition or $S_n \in \Sigma_s$, and
3. $\forall T_i \in T^+$, T_i is a silent transition.

The basic problem in the local interpretation algorithm when dealing with silent transitions is represented by the isomorphism between the sequence of observable output events (observation) and the chain of abduced transition sets. Although we correctly associate an abduced transition set with an observed action, we cannot in principle chain the list of abduced transition sets. This is due to the fact that, given a pair of contiguous abduced transition sets $< abd(o_i), abd(o_{i+1}) >$, a bound macrotransition might occur in between. Therefore, when considering the pair of contiguous observable output events $< o_i, o_{i+1} >$, we have to infer the following sequence of abduced transition sets: $< abd(o_i), abd(\emptyset), abd(o_{i+1}) >$ where $abd(\emptyset)$ denotes the set of silent bound macrotransitions, called the *silent set*. The new chain which incorporates the silent sets is called the *bridged abduced chain*.

The fundamental difference when dealing with silent transitions is that the consistency of the silent set is to be checked against the neighboring abduced transition sets. This follows the principle that the occurrence of a silent macrotransition is only a conjecture to be verified. Therefore, in a naive approach, if n is the cardinality of an observation of P, $obs(P)$, we should analyze 2^{n+1} combinations (called *instances* of the bridged abduced chain), whereby each silent set in a given position in the abduced chain is considered or not. However, this number may be drastically reduced by applying the morphology constraints firstly. In particular, the following simplifications may be carried out:

- a silent set $abd(\emptyset)$ bridging two observable events o_i and o_{i+1} in the subsequence $< abd(o_i), abd(\emptyset), abd(o_{i+1}) >$ can be simplified by looking at its adjacent transition sets $abd(o_i)$ and $abd(o_{i+1})$, so that a transition $T \in abd(\emptyset)$ is definitively removed if and only if:

$$(T_{from} \notin To(abd(o_i))) \vee (T_{to} \notin From(abd(o_{i+1})));$$

- an abduced transition set $abd(o_i)$ in a subsequence

$$< abd(o_{i-1}), abd(\emptyset), abd(o_i), abd(\emptyset), abd(o_{i+1}) >$$

  can be simplified by looking at its two adjacent silent sets $abd(\emptyset)$ and its two neighboring abduced transition sets $abd(o_{i-1})$ and $abd(o_{i+1})$, so that a transition $T \in abd(o_i)$ is definitively removed if and only if:

$$(T_{from} \notin (To(abd(o_{i-1})) \cup To(abd(\emptyset)))) \vee$$

$$(T_{to} \notin ((From(abd(\emptyset)) \cup From(abd(o_{i+1}))))).$$

Definition 11 (*hypertransition*) Let M be a reactive model of a protective component P. Let S_1 and S_2 be two (possibly identical) states in M. Then the whole set of macrotransitions connecting S_1 to S_2, denoted by $S_1 \Rightarrow S_2$, is called an *hypertransition* from S_1 to S_2.

Notice that that the set of macrotransitions relevant to hypertransition $S_1 \Rightarrow S_2$ might be either:

- *empty*, whereby S_1 and S_2 are not connected by any path in M,
- *finite*, whereby S_2 is reachable from S_1 by means of a limited number of different paths,
- *infinite*, whereby S_2 is reachable from S_1 by means of an unlimited number of different paths.

Definition 12 (*silent hypertransition*) Let $T^* = S_1 \Rightarrow S_2 = [{T_1}^+, {T_2}^+, \ldots]$ be a hypertransition in reactive model M. If $\forall$ macrotransition ${T_i}^+ \in T^*$, ${T_i}^+$ is silent, then T^* is called a *silent hypertransition* .

Definition 13 (*bound subgraph*) Let M be a reactive model involving a number of silent transitions. A subgraph of M, denoted by $Sub(M)$, is called a *bound subgraph* if and only if the following conditions hold:

1. $Sub(M)$ is connected, whereby $\forall$ pair of states (S_1, S_2), $S_1 \in Sub(M)$, $S_2 \in Sub(M), \exists(S_1 \hookrightarrow S_2) \vee \exists(S_2 \hookrightarrow S_1)$,
2. $\forall T \in Sub(M)$, T is silent, and
3. $\forall$ silent transition $T = S_1 \rightarrow S_2, T \in M, T \notin Sub(M)$, we have $S_1 \notin M, S_2 \notin M$.

The definition of bound subgraph is useful when dealing with both acyclic and cyclic silent transitions. It somehow represents an extension of the concept of *bound macrotransition*, as the frontier of a bound subgraph (namely the set of states involved in transitions not belonging to the subgraph) is composed of states which are in compliance with the definition of bound macrotransition.

Definition 14 (*cyclic macrotransition*) Let M be a reactive model. A macrotransition $T^+ \in M$, in which the initial state S coincides with the final state, $T^+ = S \hookrightarrow S$, is called a *cyclic macrotransition*.

Definition 15 (*cycle*) Let M be a reactive model. Let $T^+ = S \hookrightarrow S$ be a cyclic macrotransition in M. The set of transitions encompassed by T^+ is called a *cycle* and is denoted by Ω. A cycle Ω composed of silent transitions only is called a *silent cycle*.

Definition 16 (*cyclic bound subgraph*) Let $Sub(M)$ be a bound subgraph of reactive model M. If $Sub(M)$ includes a silent cycle Ω, then $Sub(M)$ is called a *cyclic bound subgraph*.

Definition 17 (*connection*) Let M be a reactive model. A *connection* in M is either a transition $T \in M$, or a macrotransition $T^+ \in M$, or a hypertransition $T^* \in M$.

Definition 18 (*hyperchain*) Let M be a reactive model. Let $\mapsto$ denote a generic connection, that is $\mapsto \in [\rightarrow, \hookrightarrow, \Rightarrow]$. Let $S_1, S_2, \ldots, S_n$ be reactive states in M. Then the expression $S_1 \mapsto S_2 \mapsto \cdots \mapsto S_n$ is called a *hyperchain*.

Definition 19 (*hyperhistory*) Let M be a reactive model of a protective component P. Let $H_c = S_1 \mapsto S_2 \mapsto \cdots \mapsto S_n$ be a hyperchain relevant to M. The set of local histories entailed by H_c is called an *hyperhistory* of P.

As for hypertransitions, depending upon its cardinality, a hyperhistory can be either empty, finite, or even infinite. Of course, hyperhistories involving cycles are infinite, while those including only macrotransitions are finite.

Algorithm 1 (*silent set algorithm*) The silent set algorithm, *ssa*, is a function having in input a reactive model M and generating the silent set $abd(\emptyset)$ for M as follows:

1. $abs(\emptyset) \leftarrow \emptyset$;
2. find the set of bound subgraphs $Sub^*(M) = [Sub_1(M), \ldots, Sub_n(M)]$;
3. $\forall Sub_i(M) \in Sub^*(M)$ do
   (a) if $Sub_i(M)$ is a silent transition $S_1 \rightarrow S_2$ then $abs(\emptyset) \leftarrow abs(\emptyset) \cup [S_1 \rightarrow S_2]$
   (b) else if $Sub_i(M)$ is a silent macrotransition $S_1 \hookrightarrow S_2$ then $abs(\emptyset) \leftarrow abs(\emptyset) \cup [S_1 \hookrightarrow S_2]$
   (c) else $\forall$ transition $S_1 \rightarrow S_2$, $S_1 \notin Sub_i(M)$, $S_2 \in Sub_i(M)$, $\forall$ transition $S_3 \rightarrow S_4$, $S_3 \in Sub_i(M)$, $S_4 \notin Sub_i(M)$, if $\exists$ a macrotransition $S_2 \hookrightarrow S_3 \in Sub_i(M)$ then $abs(\emptyset) \leftarrow abs(\emptyset) \cup [S_2 \Rightarrow S_3]$.

Definition 20 (*local history domain*) Let M be a reactive model of a protective component P. The whole set of local histories consistent with M is called the *local history domain* of P and is denoted by $h_d(P)$.

Definition 21 (*local interpretation*) Let M be a reactive model of a protective component P, $obs(P)$ a local observation of P. The set of local histories $h^* = lia(obs(P))$ generated by the local interpretation algorithm is called *local interpretation.*

Algorithm 2 (*local interpretation algorithm*) The local interpretation algorithm, *lia*, is a function having in input an observation $obs(P) =< m_1, m_2, \ldots, m_n >$, and generating the local interpretation $h^* = lia(obs(P))$ as follows:

1. $h^* \leftarrow \emptyset$;
2. $abd(\emptyset) \leftarrow ssa(M)$;
3. create the bridged abduced chain $Chain(obs(P))$ using $abd(\emptyset)$;
4. $\forall$ instance C of $Chain(obs(P))$ do
   (a) simplify C by applying the morphology constraints;
   (b) extract the consistent hyperchains $[\xi_1, \xi_2, \ldots, \xi_q]$ from C;
   (c) $h^* \leftarrow h^* \cup [\xi_1, \xi_2, \ldots, \xi_q]$.

4.1 Safe models

From the above definitions it comes out that by applying the extended abductive mechanism we might obtain hyperhistories of infinite cardinality. This situation clearly poses some difficulties as we should deal with an unlimited number of cases, all of them being in principle consistent with observations. On the other hand we might try to apply additional constraints to the result in order to make the hyperhistory limited. Actually, up to now we did not consider an important additional constraint related to reactive models, namely the *interface* constraint. The interface constraint is concerned with the interconnections of protective components. Two protective components P_1 and P_2 are connected when an output event e of P_1 is an input event of P_2 or vice versa. In this case we say that P_1 and P_2 *share* event e (e is called an *interface event*). More precisely we say that event e is *exported* by P_1 and *imported* by P_2.

Definition 22 (*safe vs unsafe model*) Let M be a reactive model of a protective component P. M is called a *safe model* if and only if $\forall obs(P)$ it is possible, using the local interpretation algorithm, to abduce a finite number of local histories. A reactive model M which is not safe is called *unsafe.*

Definition 23 (*strong safe model*) Let M be a reactive model of a protective component P. M is called a *strong safe model* if and only if it does not comprise any cyclic silent macrotransition.

Definition 24 (*weak safe model*) Let M be a reactive model of a protective component P. M is called a *weak safe model* if and only if it is an unsafe model in which each silent cycle is adorned at least with an interface event.

5 Global Interpretation

Once observations of protective components reacting to a misbehaviour μ are interpreted by means of the local interpretation algorithm, the set of consistent global histories must be generated. Remember that a global history H is an aggregation of local histories $h_1, h_2, \ldots, h_m$, each of which is relevant to a *different* protective component of $extent(\mu)$. Generally speaking, however, not all the combinations of local histories are consistent.

Definition 25 (*global history domain*) Let $[obs(P_1), obs(P_2), \ldots, obs(P_n)]$ be the set of local observations relevant to $extent(\mu)$. The *global history domain* of μ, $H_d(\mu)$, is the cartesian product of the local interpretations of $P_1, P_2, \ldots, P_n$, namely: $H_d(\mu) = lia(obs(P_1)) \times lia(obs(P_2)) \times \cdots \times lia(obs(P_n))$. If $H \in H_d(\mu)$, H is called a *candidate global history*.

Definition 26 (*global interpretation*) Let $[P_1, P_2, \ldots, P_n]$ be the set of protective components relevant to $extent(\mu)$. Let $[obs(P_1), obs(P_2), \ldots, obs(P_n)]$ be the set of relevant observations. The global interpretation of μ, $H^*(\mu)$, is a relation among the local history domains $h^*(P_1), h^*(P_2), \ldots, h^*(P_n)$, namely a subset of the global history domain $H_d(\mu)$: $H^*(\mu) \subseteq H_d(\mu) = h^*(P_1) \times h^*(P2) \times \cdots \times h^*(P_n)$ so that the following conditions hold:

1. $\forall\, H = (h_1, h_2, \ldots, h_n) \in H^*(\mu)$, H is globally consistent with respect to the interface constraints, and
2. $\neg\exists$ a globally consistent $H' \in H_d(\mu)$ such that $H' \notin H^*(\mu)$.

Therefore, the global interpretation can ultimately be seen as a selection of a relation on local interpretations, which in turn can be viewed as selections of the relevant local history domains. Consequently, we can provide an operational definition of global interpretation as follows:

$$H^*(\mu) = \sigma_g((\sigma_{obs(P_1)}(h_d(P_1))) \times (\sigma_{obs(P_2)}(h_d(P_2))) \times \cdots \times (\sigma_{obs(P_n)}(h_d(P_n))))$$

where σ_g denotes the selection relevant to the global consistency required by the interface constraints, while $\sigma_{obs(P_i)}$ indicates the selection pertinent to the local consistency, namely the compliance of the local history with the local observation.

Therefore, each expression $\sigma_{obs(P_i)}(h_d(P_i))$ denotes a selection on the local history domain, operationally performed by the *lia* algorithm. Once all these selections are performed, the resulting local interpretations are associated in a relation by means of the cartesian product: this corresponds to

the global history domain. Finally the resulting relation is filtered using a selection predicate relevant to the global consistency of the single candidate global history.

Interface events are expected to go through *communication channels*, these being physical connections among components. However, when specifying a reactive model we are not supposed to know the actual topology of the relevant protective components, and specifically the way in which protective components are linked to one another. We can only specify a set of *import terminals* and a set of *export terminals* for the component. In compliance with this specification, each interface event of the model has to be associated with the relevant terminal. Specifically every input event which is supposed to be generated by another protective component is associated with an import terminal, while every output event to be given as input to another component is associated with an export terminal.

Three system-defined *virtual* terminals are always associated with protective components: the *standard input* (*In*), the *standard output* (*Out*), and the *standard error* (*Err*) which are intended for events from the outside of the system (e.g. clock events), for messages, and for diagnostic events respectively. The latter are analyzed by the heuristic interpretation.

The aim of the local interpretation is to find out local histories which are consistent with observations of protective components considered singularly. Thus, as we saw in our analysis, we made local interpretation of a protective component P without any regard to the local history of another component P' possibly linked to P. As such, a local interpretation $h^* = lia(obs(P))$ is expected to include all the possible local histories which are in compliance with $obs(P)$. Formally, we say that the *lia* algorithm is *complete*. On the other hand, due to links between protective components and, consequently, due to exchanges of interface events, the local interpretation might incorporate a number of *spurious histories* which do not conform to the interface constraints. To this end, we have to compare each of the local histories of P with each of the local histories of P'. Thus the main problem is to determine whether or not two local histories h and h', belonging respectively to the local interpretation of P and P', meet the interface constraints holding for P and P'.

Definition 27 (*history interface*) Let h be a local history of a protective component P having reactive model M. The *history interface* of h, denoted by $\Upsilon(h)$, is the sequence of the interface events included in h.

Definition 28 (*balance*) Let h_1 and h_2 be two local histories of protective components P_1 and P_2 respectively; $balance(h_1, h_2)$ is a boolean function checking the consistency of local histories h_1 and h_2 with respect to the interface constraints relevant to h_1 and h_2 only.

Definition 29 (*history interface restriction*) Let h_1 and h_2 be two local histories of protective components P_1 and P_2 respectively. Let c denote the

set of communication channels linking P_1 with P_2. The *history interface restriction* of h_1 on c, denoted by $\pi_c(h_1)$ is the subsequence of the history interface of h_1 obtained by removing all the events which do not pass through any channel in c, and where terminal identifiers are replaced by the corresponding identifiers of channels.

Considering the global history generation, we can hide the concept of communication channel in the *balance* function, so that we will be only concerned with a more general concept of *link*. A link between two protective components P_1 and P_2 is defined whenever at least a communication channel connects P_1 with P_2.

A systematic approach for selecting non spurious global histories is to apply the *balance* function to every pair of local histories (h, h') so that $h \in h^*(P)$, $h' \in h^*(P')$, and $\exists$ a link joining P and P'. To do so, we make use of a so called *consistency matrix*, namely a structure in which we record the result of each application of the balance function. A consistency matrix is relevant to a misbehaviour μ and is composed of rows and columns associated with the protective components belonging to $extent(\mu)$. Each row and column is in turn composed of a set of sub-rows and sub-columns corresponding to the local interpretation of the relevant protective component.

Definition 30 (*inconsistency set*) Let $M(\mu)$ denote the consistency matrix for a misbehaviour μ having extent $[P_1, P_2, \ldots, P_n]$. The *inconsistency set* of $M(\mu)$, denoted by $\Im(M(\mu))$ is the set composed of those local history pairs (h, h') for which $balance(h, h') = false$.

On the basis of the inconsistency set we are allowed to remove the inconsistent global histories from the global history domain $H_d(\mu)$. This operation is trivial as it corresponds to the deletion of the global histories which include a pair of the inconsistency set. More formally, $H \in H_d(\mu)$ is a consistent global history if and only if $\neg\exists(h, h') \in \Im(M(\mu))$ so that $(h, h') \subseteq H$. Note that the cardinality of the global interpretation $H^*(\mu)$ is less or equal to the cardinality of the global history domain $H_d(\mu)$. The complete taxonomy for the balance function includes the following three cases:

1. $balance(h_1, h_2)$ whereby both h_1 and h_2 are local histories,
2. $balance(h_1^*, h_2)$ whereby one argument is a local history and the other is a hyperhistory, and
3. $balance(h_1^*, h_2^*)$ in which both the arguments are hyperhistories.

In our above analysis we implicitly considered only the first case, so that *balance* was a boolean function returning the consistency of h_1 and h_2 with respect to the interface constraints. For the second case, the semantics of $balance(h_1^*, h_2)$ is slightly more complicated since, in principle, we ought to

consider every history in h_1^* and verify the consistency with h_2, so that the final result is no more a boolean value but a (possibly infinite) set of boolean values. In other terms, the set of boolean values $[\beta_1, \beta_2, \ldots, \beta_n, \beta_{n+1}, \ldots]$ is isomorphic to h_1^*, whereby $\forall\, h_i \in h_1^*, \beta_i = balance(h_i, h_2)$. Thus, conceptually, $balance(h_1^*, h_2)$ is in turn a (generally different) hyperhistory $h_s^* \subseteq h_1^*$ which is composed of those local histories $h_j \in h_1^*$ for which $balance(h_j, h_2)$ evaluates true. Finally, the semantics of the third case, $balance(h_1^*, h_2^*)$, is an extension of the second case, whereby the resulting set $\Im$ of inconsistent pairs of local histories is obtained as a selection of the cartesian product of h_1^* and h_2^*, namely $\Im \subseteq h_1^* \times h_2^*$. To understand this, consider the fact that in principle we ought to check the consistency between every local history $h_i \in h_1^*$ and every local history $h_j \in h_2^*$, so that: $((h_i, h_j) \in \Im) \equiv (balance(h_i h_j) = false)$.

The graph representation of a hyperhistory, called the *hyperhistory graph*, is isomorphic to its textual counterpart. The additional information provided by the graph is the actual topology of hypertransitions, as those correspond graphically to bound subgraphs. Like a reactive model, the graphic hyperhistory represents implicitly a (possibly infinite) number of local histories, namely all the local histories which conforms to the graph. In this perspective, the hyperhistory can be viewed as a *restricted* model of the component.

Definition 31 (*history subsumption*) Let h_1^* and h_2^* be two hyperhistories relevant to the same reactive model *M*. We say that h_1^* *subsumes* h_2^*, denoted by $h_1^* \supseteq h_2^*$, if and only if all the local histories relevant to h_2^* are included in h_1^*. Formally: $(h_1^* \supseteq h_2^*) \equiv ((\forall\, h \in h_2^*)\ h \in h_1^*)$. Furthermore, if $h_1^* \supseteq h_2^*$ and $\exists h \in h_1^*, h \notin h_2^*$ (non-trivial case), we say that h_1^* *strictly subsumes* h_2^*, and write $h_1^* \supset h_2^*$.

Therefore the problem of shrinking an hyperhistory h^* is concerned with the possibility to derive, by enforcing the interface constraints, a new hyperhistory h'^* so that $h^* \supset h'^*$. If this is the case, we are allowed to consider the result of $balance(h, h^*)$ as a set of local histories $h'^* = [h_1, h_2, \ldots, h_n, \ldots]$, $h^* \supset h'^*$, all of them being consistent with h. This shrinking process is also called *hyperhistory reduction*. When the simplified hyperhistory h'^* has a finite cardinality, the process is called *hyperhistory resolution*.

Definition 32 (*interface sequence*) Let h be a local history for a protective component *P*. Let *P'* be a protective component connected to *P*. The *interface sequence* of h *for P'*, $\Upsilon_{P'}(h)$, is the subsequence of $\Upsilon(h)$ corresponding to the interface events exchanged with *P'*, whereby each terminal identifier is replaced by the relevant connected terminal identifier of *P'*.

Similarly to the *lia* local interpretation algorithm, we now introduce a *silent interpretation algorithm*, called *sia*, having as input an interface sequence

$\Upsilon_{P'}(h)$ and a hyperhistory h^* for P', and providing as output a set of local histories h'^* which are consistent with $\Upsilon_{P'}(h)$ and h^*, namely:

$$h'^* = sia(h^*, \Upsilon_{P'}(h)) \subseteq h^*.$$

Definition 33 (*associated transition*) Let $\Upsilon_P(h)$ be an interface sequence for the protective component P. Let h^* be a hyperhistory for P. Let $e(t)$ be an interface event flowing through the communication channel bound by terminal t, and belonging to $\Upsilon_P(h)$. A transition T in the hyperhistory graph which is adorned with $e(t)$ is called an *associated transition*. The set $ass(e(t)) = [T_1, T_2, \ldots, T_n]$ of associated transitions, is called the *associated transition set*.

The above definition is introduced to deal with the interpretation of the interface sequence $\Upsilon_P(h)$ for a protective component P having a hyperhistory h^* as (part of) the result of the local interpretation task. This new task differs from the local interpretation in that it is not based on any real observation $obs(P)$, but rather on a sequence of interface events, $\Upsilon_P(h)$, exchanged with another component with respect to a local history h. Moreover, the interface sequence is not interpreted on the basis of the reactive model of P, but, more strictly, on the basis of the hyperhistory graph relevant to h^*. This is due to the fact that the hyperhistory graph includes all the constraints enforced by the local observation of P, $obs(P)$. Finally, the silent set $abd(\emptyset)$ is denoted in the context of the silent interpretation by $ass(\emptyset)$, and is obtained by considering all the transitions in the hyperhistory graph but those adorned with interface events belonging to $\Upsilon_P(h)$.

An important question to be answered is how to extend the approach based on the consistency matrix whenever an element of the matrix corresponds to a history h and a hyperhistory h^*. We have seen that three diverse cases are possible:

1. $balance(h, h^*) = \emptyset$. This is the easiest case since it establishes that every candidate global history involving both h and h^* is to be discarded. The corresponding element of the matrix is marked with the empty set symbol.

2. $balance(h, h^*) = [h_1, h_2, \ldots, h_n]$. In that case the hyperhistory is resolved and a finite number of local histories determined (possibly a singleton). The corresponding element of the matrix is marked with a reference to this set.

3. $balance(h, h^*) = h'^* = [h_1, h_2, \ldots, h_n, h_{n+1}, \ldots], h'^* \subseteq h^*$. The hyperhistory is reduced but not resolved. A reference to the hyperhistory graph of h'^* is put in the corresponding element of the matrix.

However, the real problem is how to extend the global history generation process once the consistency matrix has been filled with these information. Recall that the global interpretation of a misbehaviour μ, $H^*(\mu)$, is a subset of the global history domain $H_d(\mu)$, namely:

$$H^*(\mu) \subseteq H_d(\mu) = h^*(P_1) \times h^*(P_2) \times \cdots \times h^*(P_n)$$

whereby $h^*(P_i) = lia(obs(P_i))$ denotes the local interpretation of P_i. When all the local interpretations are composed of a limited number of local histories, we are able to enumerate all the candidate global histories $H \in H_d(\mu)$. In that case we say that we have an *extensional* representation of $h^*(P_i)$. By contrast, the possible inclusion of hyperhistories of infinite cardinality in the local interpretation forces us to maintain a so called *intensional* representation of $h^*(P_i)$, by means of the hyperhistory graph.

Definition 34 (*hyper global history*) Let $H \in H_d(\mu)$ be a candidate global history. If there exists a hyperhistory $h^* \in H$, then H is called an *hyper global history*.

Example 4 Consider a hyper global history $H = (h_1, h_2, {h_3}^*)$ in which both h_1 and h_2 are local histories, while ${h_3}^*$ is an infinite hyperhistory. In addition we assume that $balance(h_1, h_2) = true$, $balance(h_1, {h_3}^*) = [h_4, h_5, h_6]$, and $balance(h_2, {h_3}^*) = [h_5, h_6, h_7]$. In that case, only the intersection of the resolutions of hyperhistory ${h_3}^*$ can be retained, namely:

$$balance(h_1, {h_3}^*) \cap balance(h_2, {h_3}^*) = [h_4, h_5, h_6] \cap [h_5, h_6, h_7] = [h_5, h_6].$$

This is due to the fact that, to be globally consistent, the global history H has to include those histories which are consistent with all the local histories in H. In our example, only h_5 and h_6 are consistent with both h_1 and h_2. In general, if the intensional global history comprehends a number of resolved hyperhistories $h^*_{1'}, h^*_{2'}, \ldots, h^*_{n'}$ then the globally consistent subset is represented by the intersection of all of them, namely: $h^*_{1'} \cap h^*_{2'} \cap \cdots \cap h^*_{n'}$.

A question to be answered is how the above intersection should be performed when some of the involved hyperhistories $h^*_{i'}$ are not resolved, but simply reduced, namely when the intersection involves infinite hyperhistories. This requires a *symbolic* intersection approach, so as to operate directly on the hyperhistory graphs, instead of reasoning on the single hyperhistories. Before doing that, let us consider how the more complicate case of balancing two hyperhistories is expected to be dealt with. Remember that the approach we used for the less complicated case of $balance(h, h^*)$ was to interpret the interface sequence of h by means of the hyperhistory relevant to h^*. This method is in general useless for computing $balance(h^*_1, h^*_2)$ since the two hyperhistories are supposed to provide an unlimited number of interface sequences, one for each encompassed local history.

Another important difference is concerned with the symmetry of the *balance* function, whereby $balance(h_1^*, h_2^*) = balance(h_2^*, h_1^*)$. This implies that the reduction (or possibly the resolution) of the hyperhistories is a matter for both h_1^* and h_2^*. In other words, we require $balance(h_1^*, h_2^*)$ to be a two attributes structure $(h_{1'}^*, h_{2'}^*)$, called a *complex reduction*, representing the reduction of h_1^* and h_2^* respectively, so that $h_{1'}^* \subseteq h_1^*$ and $h_{2'}^* \subseteq h_2^*$.

Definition 35 (*hyperhistory interface*) Let h^* be a hyperhistory. The union of the imported and exported events in the hyperhistory graph relevant to h^* is called the *hyperhistory interface* of h^* and is denoted by $\Upsilon^*(h^*)$.

Definition 36 (*hyperhistory interface restriction*) Let h^* be a hyperhistory of a protective component P connected to another protective component P'. Let C denote the set of communication channels linking P and P'. The *hyperhistory interface restriction* of h^* on C, denoted by $\Upsilon_C^*(h^*)$ is the subset of the hyperhistory interface of h^* obtained by removing all the events which do not pass through any channel in C, and where terminal identifiers are replaced by identifiers of corresponding channels.

Extending the scope of the balance function to cope with hyperhistories, requires the consistency matrix to record a more complex set of information. Specifically, the codomain of the *balance* function is now represented by a set ∇ including the following elements:

$$\nabla = [true, false, \emptyset, h^*, (\emptyset, \emptyset), (h^*, \emptyset), (\emptyset, h^*), (h_1^*, h_2^*)]$$

which in turn can be seen as the union of three parts: $\nabla = \nabla_{h,h'} \cup \nabla_{h,h^*} \cup \nabla_{h^*,h'^*}$, whereby: $\nabla_{h,h'} = [true, false]$, $\nabla_{h,h^*} = [\emptyset, h^*]$, and $\nabla_{h^*,h'^*} = [(\emptyset, \emptyset), (h^*, \emptyset), (\emptyset, h^*), (h_1^*, h_2^*)]$, corresponding respectively to the domain of two local histories, a local history and a hyperhistory, and two hyperhistories.

Definition 37 (*resolvable hyper global history*) Let H^* be a candidate hyper global history involving a number of infinite hyperhistories: $H^* = (h_1, h_2, \ldots, h_k, h_{k+1}^*, h_{k+2}^*, \ldots, h_n^*) = (h_1, h_2, \ldots, h_k) \times h_{k+1}^* \times h_{k+2}^* \times \cdots \times h_n^* = [H_1, H_2, \ldots, H_i, H_{i+1}, \ldots]$. If it is possible to enforce the interface constraints so that H^* is reduced to a finite set $H^+ = [H_1', H_2', \ldots, H_p'] \subset H^*$, then H^* is called a *resolvable hyper global history*.

Definition 38 (*symbolic intersection*) Let h_1^* and h_2^* be two hyperhistories relevant to the same protective component P. The hyperhistory graph corresponding to $h^* = h_1^* \cap h_2^*$ is called the *intersection graph* of h_1^* and h_2^*. The process of deriving the intersection graph is called *symbolic intersection*.

Definition 39 (*residue*) Let H^* be a candidate hyper global history. Let $h^* \in H^*$ be a hyperhistory embraced by H^*. Let $h_1^*, h_2^*, \ldots, h_n^*$ be the list of reduced hyperhistories relevant to $balance(h_i^*, h^*)$, $h_i^* \in H^*$, $h_i^* \neq h^*$. The *residue* of h^* in the context of H^*, denoted by $\Re(h^*, H^*)$, is the symbolic intersection of $h_1^*, h_2^*, \ldots, h_n^*$, namely $\Re(h^*, H^*) = h_1^* \cap h_2^* \cap \ldots \cap h_n^*$.

Example 5 Given $H^* = (h_1, h_2, h_3^*, h_4^*)$, whereby $balance(h_1, h_3^*) = h_{3'}^*$, $balance(h_2, h_3^*) = h_{3''}^*$, and $balance(h_3^*, h_4^*) = (h_{3'''}^*, h_{4'}^*)$, the residue of h_3^* in the context of H^* is $\Re(h_3^*, H^*) = h_{3'}^* \cap h_{3''}^* \cap h_{3'''}^*$.

Using the concept of residue it is natural to define the process of hyper global history reduction as a new hyper global history H'^* whereby each hyperhistory $h^* \in H^*$ is replaced by $\Re(h^*, H^*)$. Specifically, if H'^* does not include any infinite hyperhistory, it means that H^* is resolved.

Algorithm 3 (*global interpretation algorithm*) The *gia* global interpretation algorithm can be concisely described as a function having as input a global history domain $H_d(\mu)$ and returning a global interpretation $H^*(\mu) \subseteq H_d(\mu)$ by means of the following steps:

1. $H^*(\mu) \leftarrow H_d(\mu)$;
2. create the consistency matrix $M(\mu)$;
3. by applying *balance*, associate to each element of M the relevant symbol in ∇;
4. build the inconsistency set $\Im(M(\mu))$ as composed of the pairs of (possibly hyper) histories (h_i, h_j) for which either $balance(h_i, h_j) = false$ or $balance(h_i, h_j) = \emptyset$ or $balance(h_i, h_j) = (\emptyset, h_{j'})$ or $balance(h_i, h_j) = (h_{j'}, \emptyset)$ or $balance(h_i, h_j) = (\emptyset, \emptyset)$;
5. remove from the global interpretation those global histories H including a pair $(h_i, h_j) \in \Im(M(\mu))$, namely:
$$H^*(\mu) \leftarrow H^*(\mu) - [H_i \mid \exists (h_i, h_j) \subseteq H(\mu)(h_i, h_j) \in \Im];$$
6. for each hyperhistory h^* of every hyper global history $H^* \in H^*(\mu)$, replace h^* with the corresponding residue, namely:
$$\forall H^* \in H^*(\mu), \forall h^* \in H^*, h^* \leftarrow \Re(h^*, H^*).$$

6 A simple application

Let us consider a simple system consisting of three connected components from the power transmission network domain: the *starting device* of a distance protection in charge of detecting a short circuit exploiting voltage and

current measurements, the *reclosure component* of a distance protection, in charge to actually operate the breaker and to perform the automatic reclosure, when needed, and the *breaker* itself. The scenario is depicted in Figure 1, where *Trip* and *Cmd* are communication channels, *TripEx* and *CmdEx* are export terminals, and *TripIm* and *CmdIm* are import terminals.

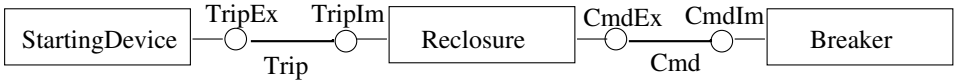

Figure 1: The connection of three protective components

The model of *StartingDevice* is the following (for the sake of brevity, we specify the model in terms of Σ_s, Σ_u, and δ):

$$\Sigma_s = [StdBy], \qquad \Sigma_u = [Start, Sw_2, Sw_3, Trip]$$
$$\delta = [StdBy \xrightarrow{z_{start}|start(Out)} Start, Start \xrightarrow{t_1(In)|sw_2(Out)} Sw_2,$$
$$Sw_2 \xrightarrow{t_2(In)|sw_3(Out)} Sw_3, Sw_3 \xrightarrow{z_3|openOff(TripEx)} Trip, Trip \xrightarrow{z_{ok}|\epsilon} StdBy,$$
$$Start \xrightarrow{z_1|openOn(TripEx)} Trip, Start \xrightarrow{z_{ok}|\epsilon} StdBy, Sw_2 \xrightarrow{z_2|openOff(TripEx)} Trip,$$
$$Sw_2 \xrightarrow{z_{ok}|\epsilon} Stdby, Sw_3 \xrightarrow{z_{ok}|\epsilon} StdBy].$$

The starting device is in charge to detect the occurrence of short circuits in the power system and to estimate the distance from the same. This is achieved by continuously monitoring the impedance value obtained by the ratio between the voltage and the current flowing through the protection. The impedance is compared against five thresholds named respectively z_{start}, z_1, z_2, z_3, z_{ok}. z_{start} is the threshold used by the protection to detect a short circuit; namely, when the measured impedance goes below z_{start}, there is the evidence for a short circuit. The other thresholds are used to define areas of the transmission network, named *zones*, in which the short circuit can be located. z_1 defines the first zone, which corresponds to the primary element the protection has to protect. Therefore, when a short circuit is detected in the first zone the protection has to immediately de-energize the faulted component by opening the associated breaker.

The second and the third zones, defined by z_2 and z_3 respectively, correspond to components of the transmission network more faraway from the protection, and they are used to define the backup behaviour of the protection. For instance, when a protection detects a short circuit in the second, it has to intervene only if the short circuit is still present after a given and fixed amount of time: this only happens when the protection that had the short circuit located its first zone failed to intervene.

The above described behaviour is modeled by means of a reactive model

as follows. When a short circuit is detected, the protection starts, generates an observable event $start(Out)$ and goes to the *Start* state. Then, it has to determine where the short circuit is located. Threshold z_1 is used to determine whether or not the short circuit is in the first zone; when this is true, the *openOn* event is generated. When the short circuit is not in the first zone, later in time the protection will compare the measured impedance against other thresholds, namely z_2 and successively z_3, to determine whether the fault is either in the second or in the third zone. When one of the two cases happens, the *openOff* command is generated. Observable messages sw_2 and sw_3 are used to inform that the protection has changed the impedance thresholds or, equivalently, that time is passed since the starting. At any time instant the short circuit may disappear, possibly because other protections have de- energized the fault, and hence, the impedance may return to a nominal value(z_{ok}), which in turn for the protection means to return to a *StdBy* steady state. However, the short circuit might not disappear even after that a tripping command has been issued, possibly because the breaker is blocked. In this case, the protection is not deactivated and it will generate other messages reporting that new thresholds are used for comparisons and new attempts to open the breaker are performed.

The starting device informs the reclosure that a fault has been detected either in the first zone or in the higher zones by means of the *TripEx* terminal. In the case of a first zone, the breaker has to be opened and the automatic reclosure has to operate (*openOn*). In the other cases, the breaker has still to be opened but the reclosure has not to intervene (*openOff*).

The model of *Breaker* is the following:

$$\Sigma_s = [Closed, Open], \qquad \Sigma_u = [GoToOpen, GoToClosed],$$
$$\delta = [Closed \xrightarrow{open(CmdIm)|\epsilon} GoToOpen, GoToOpen \xrightarrow{t_1(In)|open(Out)} Open,$$
$$Open \xrightarrow{closed(CmdIm)|\epsilon} GoToClosed, GoToClosed \xrightarrow{t_2(In)|closed(Out)} Closed,$$
$$Closed \xrightarrow{open(CmdIm)|stuckClosed(Err)} Closed,$$
$$Open \xrightarrow{closed(CmdIm)|stuckOpen(Err)} Open].$$

The circuit breaker's reactive model consists of four states: two steady states representing the open and the closed states and two unsteady states. The latter are used to represent the delay necessary for a breaker to actually open and close after that the relevant command has been received. Thus, for instance, when the circuit breaker is closed and the opening command is received, assuming that the breaker works correctly, it goes to a temporary state and after a fixed interval of time (t_1) it actually becomes open. The latter transition is associated with the generation of an observable message $open(Out)$. It is, however, possible for the breaker to be faulty and hence to remain stuck in the closed position even if it received the opening command.

The transitions departing from the *open* state have a similar meaning.

The model of *Reclosure* is the following:

$$\Sigma_s = [Enabled, Disabled], \Sigma_u = [S_1],$$
$$\delta = [Enabled \overset{openOff(TripIm)|\overline{open}(CmdEx),trip(Out)}{\longrightarrow} Enabled,$$
$$Enabled \overset{openOn(TripIm)|\overline{open}(CmdEx),trip(Out)}{\longrightarrow} S_1,$$
$$S_1 \overset{t_1(In)|closed(CmdEx),reclOn(Out)}{\longrightarrow} Disabled,$$
$$S_1 \overset{\epsilon|missingOperation(Err)}{\longrightarrow} Enabled,$$
$$Disabled \overset{openOn(TripIm)|\overline{open}(CmdEx),trip(Out)}{\longrightarrow} Disabled,$$
$$Disabled \overset{openOff(TripIm)|\overline{open}(CmdEx),trip(Out)}{\longrightarrow} Disabled,$$
$$Disabled \overset{t_2(In)|enabled(Out)}{\longrightarrow} Enabled].$$

The automatic reclosure is meant to automatically close the breaker after that it was opened by the protection to isolate a short circuit. This is useful because, since most of the faults in the power system are temporary, they disappear after that the breaker is opened; thus, the automatic reclosure brings the power transmission network in the configuration it was before the occurrence of the fault. In the case the fault is permanent, after the reclosure the starting device detects the fault again and the breaker is now opened definitively, so that no more automatic reclosure is performed. The automatic reclosure only operates when the fault has been detected in the first zone.

The reclosure may be in one of two steady states: *Enabled* or *Disabled*. When it is enabled and an *openOff* event is received through the *Trip* channel, the *open* event is sent to the associated breaker via the *Cmd* channel and an observable message is generated stating that the tripping command has been sent. The reclosure remains enabled. When the *openOn* event is received, the same output event and the same observable message as before are generated, but now the reclosure goes to a state S_1 which means that it is ready to automatically reclose the breaker. If the device works correctly, the reclosure is performed after a fixed amount of time t_1 and the functionality becomes now disabled. The reclosure consists of sending the *closed* event to the breaker and of generating an observable message informing that the reclosure has intervened $reclOn(Out)$. In the case the reclosure does not intervene because of a fault, it remains enabled. When the reclosure is disabled both the *openOn* and the $openOff$ open the breaker and generate an observable message. The transition between states *Disabled* and *Enabled* is time driven and associated with messages.

Consider now the following global observation: $OBS(\mu) =< start, sw_2, trip, sw_3 >$. Intuitively, the protection has detected a short circuit followed by a switching to the second zone. The short circuit is recognized as belonging to the area of the second zone, and hence the breaker is tripped.

After that, there is still a switching to the third zone due to the persistence of the short circuit.

We expect that the result of the interpretation is that both the starting device and the automatic reclosure worked properly, while the breaker did not open. The diagnostic process starts with the execution of the *lia* algorithm on the following subsets of observations: $obs(StartingDevice) =< start, sw_2, sw_3 >$, $obs(ReclosureDevice) =< trip >$, $obs(Breaker) = \epsilon$. The expected result of *lia* is the following (for the sake of brevity, transitions are not labeled):

$$lia(obs(StartingDevice)) = [h_{11}, h_{12}, h_{13}, h_{14}]$$
$$lia(obs(ReclosureDevice)) = [h_{21}, h_{22}]$$
$$lia(obs(Breaker)) = [h_{31}, h_{32}]$$
$$h_{11} = StdBy \rightarrow Start \rightarrow Sw_2 \rightarrow Sw_3 \rightarrow StdBy$$
$$h_{12} = StdBy \rightarrow Start \rightarrow Sw_2 \rightarrow Sw_3 \rightarrow Trip \rightarrow StdBy$$
$$h_{13} = StdBy \rightarrow Start \rightarrow Sw_2 \rightarrow Trip \rightarrow Sw_3 \rightarrow StdBy$$
$$h_{14} = StdBy \rightarrow Start \rightarrow Sw_2 \rightarrow Trip \rightarrow Sw_3 \rightarrow Trip \rightarrow StdBy$$
$$h_{21} = Enabled \rightarrow S_1 \rightarrow Enabled, \quad h_{22} = Enabled \rightarrow Enabled$$
$$h_{31} = \epsilon, \quad h_{32} = Closed \rightarrow Closed$$

The *gia* algorithm is able to shrink the set of admissible local histories to a single history for each model. In particular, the balance function determines the following inconsistency set: $\Im = [(h_{11}, h_{21}), (h_{12}, h_{21}), (h_{13}, h_{21}), (h_{14}, h_{21}), (h_{11}, h_{22}), (h_{12}, h_{22}), (h_{11}, h_{21}), (h_{14}, h_{22})]$. Note that the balancing between the starting device and the breaker is not performed as the two components are not directly linked to each other. The global history is therefore the following: $gia(OBS(\mu)) = [h_{13}, h_{22}, h_{32}]$. The resulting global history contains a transition labeled with diagnostic information, namely the transition of the circuit breaker stating that the breaker is blocked in the closed position. Similarly, transition $Sw_2 \xrightarrow{Z_2 \,|\, openOff(Trip)} Trip$ states that the short circuit is located in the area corresponding to the second zone. The latter pieces of information are used by the heuristic interpretation, together with similar pieces of information coming from other protective devices in the power system, to eventually locate the short circuit on the basis of specific application constraints.

7 Comparison with related works

Several attempts to develop automated support tools for fault diagnosis in power transmission networks are reported in the literature. Somewhat surprisingly, they are based on a variety of different technological approaches including neural networks, Kezunovic[11], fuzzy expert systems, Cho[3], Petri nets, Wang[19], model-based diagnosis, Tornielli[18], Beschta[2], and temporal reasoning techniques, Baroni[1].

In Kezunovic[11] a neural network is trained to recognize typical fault patterns, by processing in input the values of voltage and current measured in a given substation. Therefore this system aims mainly to provide a detailed characterization of a local fault, within the context of a specific substation and of its electrical and structural configuration, rather than to perform a diagnosis at the level of the overall network, considering the behaviour of the protection system. Moreover, even within this limited context, it seems that a major difficulty of this approach is related to the huge amount of preliminary work required, since the neural networks must be trained on a wide number of different cases each time it has to be applied to a different substation configuration.

In Cho[3], explicit relations between the location of a fault in the networks and the operation of protections and breakers are represented through sagittal diagrams. Temporal behaviour of the components is not considered, however in order to encompass the fact that some protections should operate before others when a fault is located in a given position, a numerical label is associated with each relation, representing the possibility that each protection operates when the fault is in the considered position. Different diagnostic hypotheses can be generated starting from observations and are ranked according to their possibility value. This approach, not including an explicit model of the diagnosed system, suffers from all the well-known limitations of first- generation, rule-based, diagnostic systems (see for instance a discussion in Beschta[2]). In particular, it can be observed that within this approach, sagittal diagrams must be able to cover all different fault cases and that only the list of faulty components is provided, without any characterization of the malfunctions showed by each of them.

An alternative approach Wang[19] resorts to Petri nets in order to build a model of the protection system behaviour. However this model is very simplified: temporal aspects are not considered and the overall protection system behaviour is modeled as a simple two steps activity. More detailed models are used in Tornielli[18] and Beschta[2], where a classical model-based diagnosis approach is adopted, resorting to the GDE+ diagnostic engine, Struss[17] . The approaches differ in the nature of the adopted model. In Tornielli[18], the behaviour of network components is defined in terms of equations between the admittance values seen from different points, whereas the behaviour of the components of the protection system is related to the impedance value seen by the component itself. The approach of Beschta[2] is quite similar but a qualitative evaluation of the fault distance is used instead of the admittance value in the component models. A major problem with these approaches is the inherent inefficiency of the truth maintenance system used by the diagnostic engine. Moreover the component models adopted, being quite abstract and not very detailed, do not take into account the rather complex behaviour of the protection system and are not able to produce detailed diagnoses about its malfunctions.

Detailed models of component behaviours, including temporal features, are used in Baroni[1], where an original diagnostic algorithm for time-varying systems is proposed and applied to the case of power networks. This algorithm is able to produce a detailed temporal reconstruction of the events occurred in the network, but it relies on the assumption that the timestamp of each message received is available, that is not always the case in practice.

The approach presented in this paper, while retaining the advantages of using detailed component models, does not rely on the availability of temporal information. It uses an original approach, based on local, global, and heuristic interpretation and is able to produce quite detailed diagnoses.

An approach whose rationale has some significant similarities with the work presented here has been developed in the frame of diagnosis of communication protocols, Riese[16]. In this context the problem consist of verifying, starting from a set of observations, if an implementation of a communication protocol is compliant with its formal definition. The protocol is modeled through a FSM and observations are matched with the possible evolution of the FSM, simulated on a discrete time scale. If none of the possible evolutions matches with the observations, the algorithm searches for modifications of the FSM that can explain the observations: each modification corresponds to a lack of compliance with respect to the standard. A detailed comparison with this work is beyond the scope of the present paper, however it can be noted that, even though the nature of the diagnostic problem is rather different, we share the successful use of FSM-based techniques, involving observation matching, propagation, and interpretation, in order to face complex diagnostic problems of time-varying systems.

This choice differs with respect to other approaches proposed in literature for diagnosis of dynamic systems, a topic that has been addressed by many researchers in recent years Dvorak[5], Gluckenbieh[7], Hamscher[8], Ng[13], Lackinger[12], Dressler[4]. As a matter of fact, even though the importance of dynamic system diagnosis and the need of specific techniques for this problem have been early recognized, Pan[14], Hamscher[9], the *classical* theory of diagnosis, Reiter[15], de Kleer[10], has been conceived for static systems only. Subsequent attempts to overcome this limitation have been, however, strongly influenced by the approach initially adopted for static systems. In fact, most of these approaches, Dvorak[5], Hamscher[8], Ng[13], Lackinger[12], Friedrich[6], Dressler[4], share a common rationale: Reiter's algorithm is simply applied iteratively to subsequent instantaneous snapshots of the behaviour of the system, generated through a suitable dynamic model. This method presents however some significant drawbacks, since it is unable to deal with faults whose manifestation involves several time instants and does not exploit efficiently knowledge about system dynamic behaviour. An approach closer to our ideas is presented in Gluckenbiehl[7] where diagnostic activity for a dynamic system is seen as the task of recon-

structing the history of the system starting from some temporally located information about system attributes and resorting to a model of system behaviour. However the behavioural representation adopted is made up of IF- THEN rules including temporal information and suffers therefore from limited expressiveness.

Acknowledgments

The present work is partially supported by the Commission of the European Union under the ongoing Esprit III Project 8491 *Timely* (Time-Constrained Integrated Management of Large- Scale Systems).

Keywords

model-based reasoning, fault diagnosis, knowledge representation, short circuit localization, power transmission networks.

Bibliography

1. Baroni, P., Canzi, U., & Guida, G. SHORT: a knowledge- based system for fault diagnosis in power transmission networks, *Proc. of the 8th Int. Symp. on Artificial Intelligence ISAI 95*, Monterrey, MX, pp. 199-208, 1995.

2. Beschta, A., Dressler, O., Freitag, H., Montag, M. & Struss, P. A model-based approach to fault localisation in power transmission networks, *Intelligent Systems Engineering*, pp. 3-14, 1993.

3. Cho, H.-J., Park, J.-K. & Lee, H.-J. A fuzzy expert system for fault diagnosis of power systems, *Proc. of the Int. Conf. on Intelligent System Application to Power Systems ISAP 9)*, Montpellier, F, pp. 217-226, 1994.

4. Dressler, O. & Freitag, H. Prediction sharing across time and contexts, *Proc. of the 12th Nat. Conf. on Artificial Intelligence AAAI-94*, Seattle, WA, pp. 1136-1141, 1994.

5. Dvorak D. & Kuipers, B. Model-based monitoring of dynamic systems, *Proc. of the 11th Int. Joint Conf. on Artificial Intelligence IJCAI-89*, Detroit, MI, pp. 1238-1243, 1989.

6. Friedrich, G. & Lackinger, F. Diagnosing temporal misbehaviour, *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*, Sydney, Australia, pp. 1116-1122, 1991.

7. Gluckenbiehl, T. & Schäfer-Richter, G. SIDIA: Extending prediction based diagnosis to dynamic models, *Working Notes of the 1st Int. Workshop on Principles of Diagnosis*, Stanford, CA, pp. 74-82, 1990.

8. Hamscher, W. Modeling digital circuits for troubleshooting, *Artificial Intelligence*, **51 (1-3)**, pp. 223-271, 1991.

9. Hamscher, W. & Davis, R. Diagnosing circuits with state an inherently underconstrained problem, *Proc. of the 4th Nat. Conf. on Artificial Intelligence AAAI-84*, Austin, TX, pp. 142-147, 1984.

10. de Kleer, J. & Williams, B.C. Diagnosing multiple faults, Artificial Intelligence, **32(1)** , pp. 97-130, 1987.

11. Kezunovic, M., Rikalo, I. & Sobajic, D.J. Neural network applications to real-time and off- line fault analysis, *Proc. of the Int. Conf. on Intelligent System Application to Power Systems ISAP 94* , Montpellier, F, pp. 29-36, 1994.

12. Lackinger, F. & Nejdl, W. Integrating model-based monitoring and diagnosis of complex dynamic systems, *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*, Sydney, Australia, pp. 1123-1128, 1991.

13. Ng, H.T. Model-based, multiple-fault diagnosis of dynamic, continuous physical devices, *IEEE Expert* **6(6)**, pp. 38-43, 1991.

14. Pan, J. Y.-C. Qualitative reasoning with deep-level mechanism models for diagnosis of mechanism failures, *Proc. of the 1st IEEE Conf. on AI Applications*, Denver, CO, pp. 295-301, 1984.

15. Reiter, R. A theory of diagnosis from first principles, *Artificial Intelligence*, **32(1)**, pp. 57-95, 1987.

16. Riese, M. Diagnosis of communicating systems: dealing with incompleteness and uncertainty, *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence IJCAI-93*, Chambery, F, pp. 1480-1485, 1993.

17. Struss, P. & Dressler, O. Physical negation-integrating fault models into the general diagnostic engine, *Proc. of the 11th Int. Joint Conf. on Artificial Intelligence IJCAI-89*, Detroit, Michigan, 1989, 1318-1323.

18. Tornielli, G., Capetta, L., Cermignani, S., Fabiano, A.S. & Schinco R. An interval-based approach to fault diagnosis of power systems, *Proc. of the Int. Conf. on Intelligent System Application to Power Systems ISAP 94* , Montpellier, F, pp. 809-816, 1994.

19. Wang, J.P. & Trecat J. A parallel fault diagnosis expert system for one dispatch center, *Proc. of the Int. Conf. on Intelligent System Application to Power Systems ISAP 94* , Montpellier, F, pp. 201-208, 1994.