Alarm management via temporal pattern learning

John William Vásquez Capacho, Audine Subias, Louise Travé-Massuyès, Fernando Jimenez

To cite this version:

John William Vásquez Capacho, Audine Subias, Louise Travé-Massuyès, Fernando Jimenez. Alarm management via temporal pattern learning. Engineering Applications of Artificial Intelligence, 2017, 65, pp.506 - 516. 10.1016/j.engappai.2017.07.008 . hal-01611635

HAL Id: hal-01611635
https://laas.hal.science/hal-01611635

Submitted on 6 Oct 2017

Alarm management via temporal pattern learning

J.W. Vásquez Capacho,[a,b,1], A. Subias[a,2], L. Travé-Massuyès[a,3], F. Jimenez[b,4]

[a] *LAAS-CNRS, Université de Toulouse, CNRS,INSA, Toulouse, France.*
[b] *GIAP, Engineering department, Andes University. Bogotá, Colombia.*

Abstract

Industrial plant safety involves integrated management of all the factors that may cause accidents. Process alarm management can be formulated as a pattern recognition problem in which temporal patterns are used to characterize different typical situations, particularly at startup and shutdown stages. In this paper we propose a new approach of alarm management based on a diagnosis process. Assuming the alarms and the actions of the standard operating procedure as discrete events, the diagnosis step relies on situation recognition to provide the operators with relevant information on the failures inducing the alarm flows. The situation recognition is based on chronicle recognition where we propose to use the hybrid causal model of the system and simulations to generate the representative event sequences from which the chronicles are learned using the Heuristic Chronicle Discovery Algorithm Modified ($HCDAM$). An extension of this algorithm is presented in this article where the expertise knowledge is included as temporal restrictions which are a new input to $HCDAM$. An illustrative example in the field of petrochemical plants is presented.

Keywords: Alarm management; Pattern recognition; Chronicles; Transitional stages; Hybrid models

[1] jwvasque@laas.fr
[2] subias@laas.fr
[3] louise@laas.fr
[4] fjimenez@uniandes.edu.co

1. Introduction

The operation of many industrial processes, especially in the petrochemical sector, involves inherent risks due to the presence of dangerous material such as gases and chemicals which in specific conditions can cause emergencies [1]. Safety in industrial processes is supplied by layers of protection as illustrated by Figure 1. These layers initiate with a safe design and an effective process control (Layers 1 and 2), followed by an "*alarm*" display to the operators (Layer 3) that may trigger manual operator actions. The next layer corresponds to the automatic (Safety Instrumented System) prevention layer (Layer 4), continuing with the layers (Layers 5, 6, and 7) to mitigate the consequences of an event (in safety theory an "event" corresponds to a dangerous situation that happens, for example an explosion). Our work focuses on Layer 3. In the process state transitions such as startup and shutdown stages, the alarm flood increases and generates critical conditions in which the operator does not respond efficiently then, a dynamic alarm management is required [2]. The dynamics of a process can be represented by an approach that depicts the process behavior using the events that occur. In this context, the chronicle approach has been applied in many diagnosis applications. Applications such as diagnosis of network telecommunication [3], cardiac arrhythmia detection [4] and intrusion detection systems [5] can be mentioned. Another application of the chronicles is the recognition in the setting of unmanned aircraft systems and unmanned aerial vehicles operating over road and traffic networks [6]. Chronicles are designed to provide temporal patterns of total and partial order. While chronicles consider temporal constraints between event type occurrences, one of the main difficulties of chronicle discovery is to guarantee robustness to variations. Another difficulty is to obtain automatically a base of chronicles that represents each situation. To obtain relevant chronicles from a set of event sequences representing a given situation, it is often necessary to incorporate expert knowledge. This paper enhances the results of the chronicle learning algorithm proposed in [7] by incorporating expert knowledge in the form of temporal restrictions, as well as

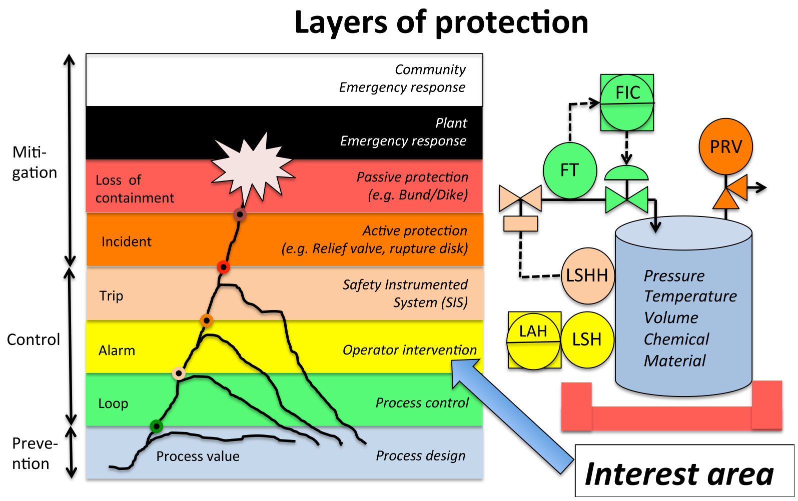


Figure 1: Safety layers of protection from [1]

additional information that allows us to limit the conservatism of chronicles.

The paper is divided into 6 sections. Section 2 gives an overview on the relevant literature of alarm management. Section 3 presents the problem statement and overviews the new method Chronicle Based Alarm Management (CBAM). Section 4 provides a background on chronicles including the $HCDAM$ description. Section 5 indicates the formal framework for this analysis with the representation of the hybrid causal model and the qualitative abstraction of continuous behavior. Finally, a case study is given in section 6 where an illustrative application in the petrochemical sector is presented.

2. Alarm management review

An *alarm* aims to alert the operator of deviations in the process variables from normal operating conditions, i.e. abnormal operating situations. ISA-18.2 defines an alarm as *"An audible and/or visible means of indicating to the operator an equipment malfunction, process deviation, or abnormal condition requiring a response."* . From this definition it appears clearly that an alarm is not a simple message or event but rather a condition directing the operator's attention to plant functioning in order to generate a timely assessment or action. Because of the fundamental role of an alarm management, the attention of many researchers has recently focused in themes such as alarm history visualization and analysis, process data based alarm system analysis and plant connectivity and causality analysis that are further presented below.

2.1. Alarm historian visualization and analysis

A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system is presented in [8]; the aim of the work is to reduce the number of alerts presented to the operator. If alarms are related one to another, those alarms should be grouped and presented as one alarm problem. Graphical tools for routine assessment of industrial alarm systems are proposed by [9]; two new alarm data visualization tools for the performance evaluation of the alarm systems are presented. These tools are called the high density alarm plot and the alarm similarity color map. In [10], event correlation analysis and two-layer cause-effect model are used to reduce the number of alarms and a Bayesian method is introduced for multimode process monitoring in [11]. These approaches allow to recognize alarm chattering, to group many alarms or to estimate the alarm limits in transition stages, but the dates of the alarm occurrences and the procedure actions are not considered.

2.2. Data based analysis of alarm system

In [12] an operator model is used as a virtual subject to evaluate plant alarm systems under abnormal situations. Another proposal [13] introduced a technique for optimal design of alarm limits by analyzing the correlation between process variables and alarm variables. In 2009 a framework based on the receiver operating characteristic curve was proposed to optimally design alarm limits, filters, dead bands, and delay timers; this work was presented in [14] and a dynamic risk analysis methodology that uses alarm databases to improve process safety and product quality was presented in [15]. In [16], the Gaussian mixture model is employed to extract a series of operating modes from the historical process data. Then local statistics and its normalized contribution chart are derived for detecting abnormalities early and for isolating faulty variables. These approaches require numerous simulations and/or historical data, and are not well suited in case of new plants for which historical data is not yet available.

2.3. Plant connectivity and causality analysis

In the literature, transition monitoring of chemical processes has been reported by many researchers. In [17] a dynamic alarm management strategy is presented for chemical process transitions in which the artificial immune system-based fault diagnosis method and a Bayesian estimation based dynamic alarm management method are integrated. In another proposal [18], a fault diagnosis strategy for startup process based on standard operating procedures is presented. This approach proposes a behavior observer combined with dynamic PCA (Principal Component Analysis) to estimate process faults and operator errors at the same time. One can also mention the work related to direct causality detection via a transfer entropy approach in [19]. [20] overviews the modeling methods for capturing process topology and causality. [21] proposes fault detection during process transitions: a model-based approach in which extended Kalman filters, Kalman filters, and open-loop observers are used to estimate process states during the transition and to generate residuals. [22] presents a framework for managing transitions in chemical plants where a trend analysis-based approach for locating and characterizing the modes and transitions in historical data is proposed. Finally, in [23] a hybrid model-based framework is used for alarm anticipation where the user is preparing for the possibility of a single alarm occurrence. For transition monitoring, these types of techniques are used in industrial processes and the hybrid model based framework is a possible representation of a petrochemical system. A causal model allows to identify the root of the failures and to check the correct evolution in a transitional stage. Our proposal is closer to this third type of approach as it seeks to exploit the causal relationships between process variables and procedure actions as explained in the next sections.

3. Problem statement and method overview

As explained in the previous section, alarm floods are an important aspect of safety for industrial plants. Therefore, the operators need a tool that help them recognize the plant situation, specially in the transitional stages such as startup and shutdown. In this work, we propose to generate models that describe the process evolution on a discrete level. These models, which can be used to perform diagnosis, can take the form of temporal patterns. In this paper we have chosen to work with chronicles [24].

Chronicles represent temporal patterns of situations for specific scenarios. Designing the chronicles involves difficulties as the generation of the event sequences for learning and the use of expert knowledge. This paper proposes a comprehensive methodology that permits the event sequence generation for learning chronicles representing different situations of the plant. The Chronicle Based Alarm Management (CBAM) methodology that is proposed merges different techniques to take the hybrid aspect and the standard operational procedures of the concerned process into account. These two features stand out of the literature ([18],[23],[22],[21]). Another important aspect is the analysis of dynamic alarm management as most of the time, the alarm is assumed to be a static indicator. In our proposal an alarm is an event with an occurrence date and the expected alarm flow is formally modeled by a chronicle [25],[26]. The position of our approach with respect to other approaches stands in that we use information about the procedural actions related to the behavior of continuous variables for the situation awareness process. Specific information is obtained in each step of the CBAM methodology and it is summarized in three steps: *event type identification*, *event sequence generation* and *chronicle database construction*:

1. *Event type identification*: From the standard operating procedures and from the evolution of the continuous variables, this step determines the set of event types in startup and shutdown stages.
2. *Event sequence generation*: From the expertise and an event abstraction

procedure this step determines the date of occurrence of each event type for constructing the representative event sequences. A representative event sequence is the set of event types with their dates of occurrence that can be associated to a specific scenario of the process.

3. *Chronicle database construction*: From the representative event sequences in each scenario, this step determines the chronicle database using the *Heuristic Chronicle Discovery Algorithm Modified* ($HCDAM$).

In a general way, chronicle learning requires a lot of representative event sequences of each scenario. In our case no historical information related to startup or shutdown stages is available, as these types of scenarios do not occur frequently. Therefore, it is by simulation using a fault injection framework that the representative event sequences of each scenario are obtained. The different steps of this methodology, detailed further in the article, base their formalization on the works [7],[27],[28].

4. Background on chronicles

Chronicles correspond to temporal patterns. A chronicle is associated to each situation to recognize, normal or abnormal. During the operation of the system, several sensors are used to retrieve information about the system's status over time. This record is then decomposed into a series of discrete events to generate an event sequence. Then, a chronicle recognition algorithm fed by this sequence looks for the chronicles that are recognized. The situation in which the system is then deduced accordingly. Let us consider time as a linearly ordered discrete set of instants. The occurrence of different events in time represents the system dynamics and a model can be determined to diagnose correct evolution. E is defined as the set of *event types* and an *event* is defined as a pair (e_i, t_i), where $e_i \in E$ is an event type and t_i is a variable of integer type called the event date. Without loss of generality, we assume that two events cannot occur at the same instant, i.e. simultaneously. In the following, we may refer to an *event type* as an *event* for short. A *temporal sequence* on E is denoted as an ordered set of events S = $\langle (e_i, t_i)_j \rangle$ with $j \in\ N_l$ where l is the size of the temporal sequence S and N_l is a finite set of linearly ordered instants of cardinal l. $l{=}\mid S \mid$ is the size of the temporal sequence, i.e. the number of event type occurrences in S.

Definition 1 (Chronicle). *A chronicle is defined as a triplet $C = (\mathcal{E}, \mathcal{T}, \mathcal{G})$ such that:*

- *$\mathcal{E} \subseteq E$, where $\mathcal{E}$ is called the typology of the chronicle,*
- *$\mathcal{T}$ is the set of temporal constraints of the chronicle,*
- *$\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is a directed graph where:*
    - *$\mathcal{V}$ is a set of indexed event types, i.e. a finite indexed family defined by $\nu : K \to \mathcal{E}$, where $K \subset \mathbb{N}$,*
    - *$\mathcal{A}$ is a set of edges between indexed event types; there is an edge $(e_{i_{k_1}}, e_{j_{k_2}}) \in \mathcal{A}$ if and only if there is a time constraint between $e_{i_{k_1}}$ and $e_{j_{k_2}}$.*

Given a set of event types E, the space of possible chronicles can be structured by a *generality relation.*

Definition 2 (Generality relation among chronicles). *A chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T}, \mathcal{G})$ is* more general than *a chronicle $\mathcal{C}' = (\mathcal{E}', \mathcal{T}', \mathcal{G}')$, denoted $\mathcal{C} \sqsubseteq \mathcal{C}'$, if $\mathcal{E} \subseteq \mathcal{E}'$ or $\forall \tau_{ij} \in \mathcal{T}, \tau_{ij} \supseteq \tau'_{ij}$. Equivalently, $\mathcal{C}'$ is said stricter than $\mathcal{C}$.*

If the event e_1 occurs t time units after e_2, then it exists a directed link $\mathcal{A}$ from e_1 to e_2 associated with a time constraint. Considering the two events (e_i, t_i) and (e_j, t_j), we define the time interval as the pair $\tau_{ij} = [t^-, t^+] \in \mathcal{T}$, where $t^-, t^+ \in \mathbb{Z}$ correspond to the lower and upper bounds on the temporal distance between the two event dates t_i and t_j. For instance, the constraint $e_i[-3, 1]e_j$ allows e_i to preceed e_j by 1 time unit while it also allows e_i to follow e_j up to 3 time units.

Definition 3 (Chronicle instance). *A chronicle $C = (\mathcal{E}, \mathcal{T}, \mathcal{G})$ is recognized in a temporal sequence S of event types $\mathcal{E}'$, such that $\mathcal{E} \subseteq \mathcal{E}'$ when all the*

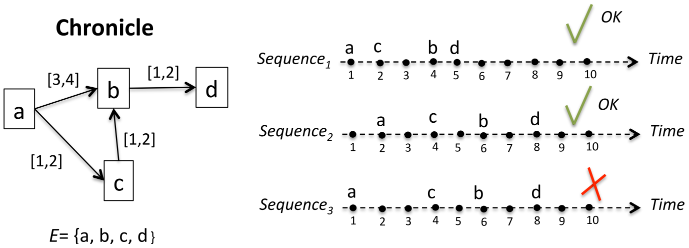


Figure 2: Example of chronicle instances (marked green)

temporal constraints $\mathcal{T}$ are satisfied. Then $C_{inst} = (\mathcal{E}, \mathcal{T}_v)$ is an instance of C, where $\mathcal{T}_v$ is a valuation of $\mathcal{T}$.

Figure 2 illustrates the above definition: the chronicle on the left is recognized in the first and second sequences. Nevertheless, it is not recognized in the third sequence because the only set of constraints relating a, b, c, and d in this sequence is $\mathcal{T}_v = \{a[5,5]b, a[3,3]c, c[2,2]b, b[2,2]d\}$ and $\mathcal{T}_v$ is not a valuation of $\mathcal{T} = \{a[3,4]b, a[1,2]c, c[1,2]b, b[1,2]d\}$.

4.1. Chronicle discovery

One of the major problems associated with chronicle-based diagnosis is to obtain chronicles characterizing the situations. Chronicle discovery is the problem of exhibiting the strictest chronicles present in a trace. One wants to obtain the strictest chronicles, which are therefore the most likely to correctly characterize the situation (and therefore the traces) that we want to detect. In practice, these are often built ”by hand” by experts. How to acquire and update automatically chronicles is an issue. Model based chronicle generation approaches have been developed in the last decades. For instance, in [28] the runs of the monitored system are described in the temporal tiles formalism. The authors propose an algorithm inspired of Petri net unfolding to build all the temporal runs of the system. Then, the projection of these runs on the observable part allows to define the chronicles. Other approaches have been investigated from learning theory for unearthing patterns from input data. One can consider for instance learning techniques based on Inductive Logic Programming (ILP) ([29],[4]), case-based chronicle learning of ([30],[31]) that is a characteristic supervised method by reinforcement learning but also ([32],[33],[34]) that adapt a clustering method to learn chronicles in an unsupervised way by projecting chronicle instances into a normative space. Finally, chronicles are also acquired from approaches that analyze logs and extract the significant patterns by temporal data mining techniques ([35]). The objective of temporal data mining techniques is to discover all patterns of interest in the input data, by means of an unsupervised approach.

There are several ways to define the relevance of a pattern. Among these methods, the frequency criterium is widely used ([36],[37],[38]). In [6],[39], the chronicle learning problem is motivated by discovering the most frequent alarm patterns in telecommunication alarm logs and their correlations. The tool, called FACE (Frequency Analyzer for Chronicle Extraction), extracts the frequent patterns by carrying out a frequency-based analysis on sublogs, defined on time windows of fixed duration.

The learning algorithm integrated in this article is also based on a frequency criterium ([7],[26]) and can be related to [36] and [37]. The proposal in [36] makes it possible to discover, given a trace S and a threshold frequency f_t, chronicles of frequency $f \geq f_t$ in S. We do not detail this algorithm in depth, but the principle is the following. For each pair of event types e_i,$e_j \in E$, they associate a temporal constraint (that one can deduce from S, several heuristics are possible). They then construct for each of these pairs the associated chronicle. Those whose frequency is $\geq f_t$ are kept and completed by adding an event type, as well as the associated constraints. One repeats on the latter. Little by little, more and more constrained chronicles are being built up until $f < f_t$, in which case the branch is stopped. The problem with this algorithm is that it does not allow the complete discovery of chronicles. It can find traces from the base of Dousson & Vuduong, but it does not generate all the frequent chronicles.

Cram's algorithm solves Dousson & Vuduong's non-completeness problem by adding more possibilities for temporal constraints attached to pairs of event types [37]. The idea is also to build the chronicles little by little, but this time from a base of constraint graphs. For each pair of E (pair of event types present in the trace S used for learning) a constraint graph is constructed, that is to say a set of intervals ordered by the relation $\subseteq$. The objective is then to build chronicles by adding event types as in the Dousson algorithm or by further constraining one of the constraints guided by the constraint graph.

4.2. Learning chronicles with HCDAM

In many cases the same situation does not imply temporal sequences perfectly identical. $HCDAM$ has been proposed in [7] to learn the chronicles whose instances occur in all event sequences representing the same situation. Its principles are briefly reminded in this section and the reader is referred to [7] for a detailed presentation.

Given a set of sequences $\mathbb{S}$ and a minimum frequency threshold, it finds all minimal frequent chronicles presented in all temporal sequences. The chronicle learning algorithm has the following three phases:

1. Filtering operation
2. Building a constraint database from the temporal sequences
3. Generating a set of candidate chronicles

We briefly describe the steps that make up HCDAM using 3 sequences to illustrate each of them. The sequences are described below:

$$S_1 = \langle (b,1),(a,3),(b,4),(b,5) \rangle \tag{1}$$

$$S_2 = \langle (a,1),(b,2),(b,3),(a,6) \rangle \tag{2}$$

$$S_3 = \langle (a,1),(b,2),(b,4),(c,5),(b,7) \rangle \tag{3}$$

4.2.1. Phase 1

The $filtering$ *operation* is a preliminary process on sequences and it can be summarized as two possible actions:

- Filtering the event types that are not present in all sequences $\mathbb{S}$. If $\exists S_k \in \mathbb{S}$ such as $\exists e_i \notin S_k$, then e_i is removed of all other sequences.
- Filtering on a given set of event types $\{e_{i_1}, e_{i_2}, ..e_{i_r}\} \subset E$ if we are interested only in those event types during processing.

After, filtering the sequences, the set of occurrences $O = \{O_{ij}^k\}$ that contains all the instances of a pair of event types (e_i,e_j) in S_k is determined. Back to the example, the set of occurrences for the pair (a,b) in the sequences S_1, S_2 and S_3 are:

$$O_{ab}^1 = \{\langle(a,3),(b,1)\rangle, \langle(a,3),(b,4)\rangle, \langle(a,3),(b,5)\rangle\} \tag{4}$$

$$O_{ab}^2 = \{\langle(a,1),(b,2)\rangle, \langle(a,1),(b,3)\rangle, \langle(a,6),(b,2)\rangle, \langle(a,6),(b,3)\rangle\} \tag{5}$$

$$O_{ab}^3 = \{\langle(a,1),(b,2)\rangle, \langle(a,1),(b,4)\rangle, \langle(a,1),(b,7)\rangle\} \tag{6}$$

In addition, the set of durations $D_u = \{D_{ij}^k\}$ is computed. D_u contains the time intervals between the occurrence dates for each pair of event types. This interval is calculated as follows:

$$D_{ij}^k = \{d_{ij}^k = (t_j - t_i) \mid < (e_i, t_i), (e_j, t_j) > \in O_{ij}^k\} \tag{7}$$

The sets of durations for the pair (a,b) in the sequences S_1, S_2 and S_3 are: $D_{ab}^1 = \{-2, 1, 2\}$, $D_{ab}^2 = \{1, 2, -4, -3\}$ and $D_{ab}^3 = \{1, 3, 6\}$.

The frequency f_{ij}^k of each pair (e_i, e_j) in the sequence S_k corresponds to the maximal number of occurrences of the pair in the sequence S_k. The maximal frequency f_{max} of each pair (e_i, e_j) is the maximal number of occurrences of the pair considering all the sequences $S_k \in S$. The example of frequency f_{ij}^k and f_{max} for the pair (a,b) in the sequences S_1, S_2 and S_3 is: f_{ab}^1=3, f_{ab}^2=4, f_{ab}^3=3 and f_{max}=3.

4.2.2. Phase 2

In a second phase, HDCAM builds the so-called *constraint database* $\mathbb{D}$ that stores every temporal constraint $\tau_{ij} = e_i[t^-, t^+]e_j$ that is frequent in all the sequences of $\mathbb{S}$. $\mathbb{D}$ is organized as a set of trees T_{ij}^{α} for each pair of event types (e_i, e_j) with $i, j = 1, \ldots, \mid E \mid, i \leq j$ and $\alpha = 1, \ldots, n_{ij}$.

In the trees, time constraints are nodes and arcs represent the relationship $is\ parent\ of$ defined as below:

Definition 4 (*is parent of* **relation).** *The node $e_i[t^-, t^+]e_j$ is parent of $e_i[t'^-, t'^+]e_j$ if and only if $[t'^-, t'^+] \subset [t^-, t^+]$ and there does not exist $e_i[t''^-, t''^+]e_j$ such that this $[t'^-, t'^+] \subset [t''^-, t''^+] \subset [t^-, t^+]$.*

The root of a tree T_{ij}^{α} is a temporal constraint $e_i[t^-, t^+]e_j$ such that the number of occurrences of the pair (e_i, e_j) is maximal in all sequences of $\mathbb{S}$. It represents the 2-length chronicle with topology $\mathcal{E} = \{e_i, e_j\}$ that is the most general for all temporal sequences of $\mathbb{S}$ and the child nodes are stricter 2-length chronicles with the same typology.

The reader can refer to [7] for the details of the method used in HDCAM for determining the trees for each pair and in particular their roots.

Considering the case of the three sequences S_1, S_2, and S_3 of the running example given by (1), (2), and (3) respectively, the pair (a, b) and the pair (b, b) each give rise to one tree (T_{ab}^1 and T_{bb}^1 respectively). These are represented in Figure 3, with the mention of the frequency associated to the constraints of each level of the trees. The tree for (a, b) has three levels for frequency 3 to 1 from top to bottom and the tree for the pair (b, b) has only one level for frequency 1. Any of the constraints of a given level guarantees that the pair appears exactly with the frequency associated to the level in all the sequences.

4.2.3. Phase 3

The generation of a set of candidate chronicles initializes with a set of chronicles that were proved to be frequent and it uses the constraint database $\mathbb{D}$ to explore the chronicle space. This can be resumed by the following steps:

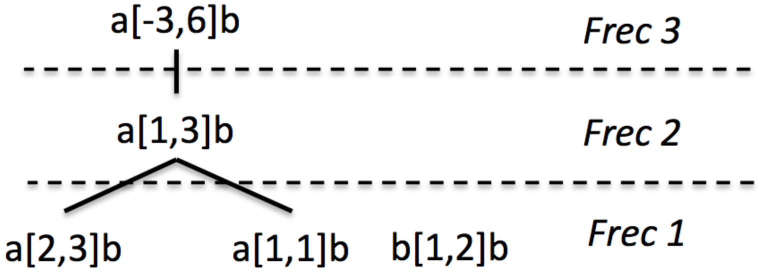


Figure 3: Constraint tree for the pairs (a,b) and (b,b)

- The set of candidates initiates with the set of tree roots
- Use the operator "*add_event*". This operator, checks at the constraint trees in order to find the restrictions of an event type ε with all elements of E.
- Count the minimal number of occurrences of the candidate in $\mathbb{S}$

Once the constraint tree is generated chronicles are extracted according to two thresholds: f_{min} (or f=1 when not defined) and f_{max}. The search starts from a constraint of maximum frequency. i.e root of the tree, which is the initial chronicle. This chronicle is then completed according to the frequency specification by the use of an operator (*add_event*) for adding the event type ε. The operator searches the constraint graph for all the constraints between ε and all the event types of the chronicle under construction in accordance with the frequency. To avoid the counting phase, the structure of the tree is changed: it no longer depends on couples of events but of the frequency of the time constraints between pairs of events.

4.3. Integration of expert knowledge in chronicle learning

The expert knowledge is important and specific information can be integrated into the algorithm $HCDAM$. Our objective is to capture the expertise of the operator when he knows something about the behavior of the process. For this purpose, we allow the user to specify *temporal restrictions* for event type pairs. This knowledge is incorporated in $HCDAM$ as additional input information to the algorithm. In [26] an alarm management strategy was proposed using the $HCDAM$ for chronicle learning. Now in this proposal, an extension of this algorithm is presented where the expertise knowledge is included. In addition, a way for reduce the quantity of possible event sequences to be recognized by a chronicle is also presented in this section as a contribution to the chronicle learning theory.

4.3.1. Integration of process knowledge

As was mentioned before, expert knowledge can be represented by *temporal restrictions* that express a known time constraint between two event type dates. These temporal restrictions are gathered in an expert database $\mathcal{D}_e$.

Definition 5 (Temporal restriction). *A temporal restriction for a pair of event types (e_i, e_j) is a given temporal constraint between their event dates $TR_{ij} = e_i[t^-, t^+]e_j$.*

To integrate this knowledge, Phase 2 of $HCDAM$ is modified. One first checks the existence of a temporal restriction TR_{ij} for each pair of event types (e_i, e_j). The temporal restriction then replaces the tree root for this pair of event types.

The effect of the integration of temporal restrictions is to focus the learning process and produce less chronicles; it means that the number of chronicles that are learned for a specific situation is reduced using the expertise knowledge.

Example of the "charge oven" activity: the results of the learning process without the inclusion of temporal restrictions and with the inclusion of temporal restrictions are given in a simple example of activity consisting of charging an oven as represented in Figure. 4. The event types are given by $E = \{a, b, c, d\}$, where a (resp. b) is the detection of the product a (resp. b) entering the oven, c is the event corresponding to putting the heaters on, and d is the event of

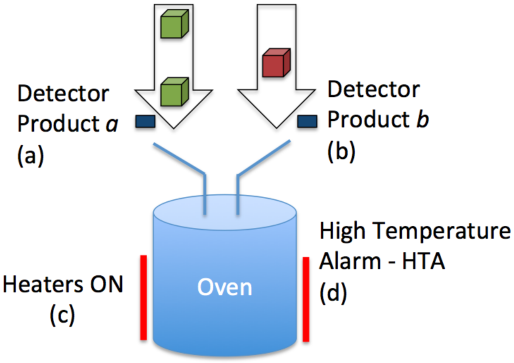


Figure 4: Example of the "charge oven" activity

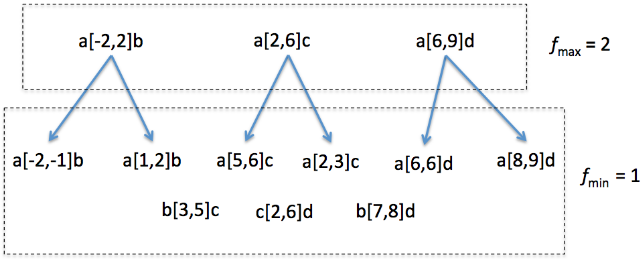


Figure 5: Tree roots without temporal restrictions

setting the heaters to high temperature. Three event sequences that express normal startup of this process are:

$S_1 = \langle (a, 2), (b, 4), (a, 5), (c, 7), (d, 11) \rangle$

$S_2 = \langle (a, 2), (b, 3), (a, 4), (c, 7), (d, 10) \rangle$

$S_3 = \langle (a, 2), (b, 3), (a, 5), (c, 8), (d, 11) \rangle$

The temporal restrictions that indicate the expertise knowledge are $TR_{ab} = a[-2, 2]b$ and $TR_{cd} = c[2, 6]d$.

The results obtained by HDCAM without the use of temporal restrictions provide 8 chronicles for a frequency 1. The tree roots are given in Fig. 5. The 8 chronicles C_1 to C_8 are given in Fig. 6 and 7.

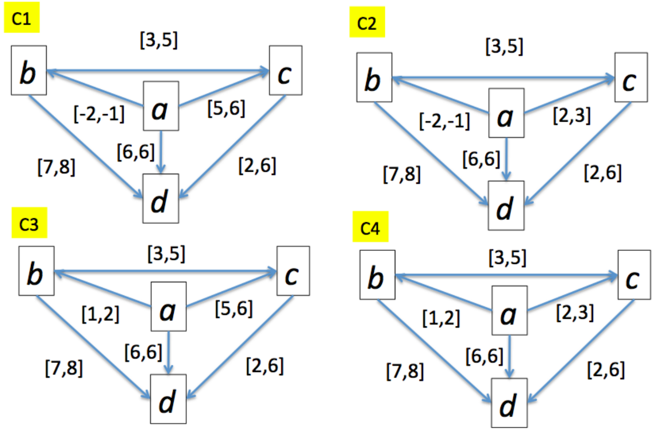


Figure 6: Chronicles (1-4) without temporal restrictions

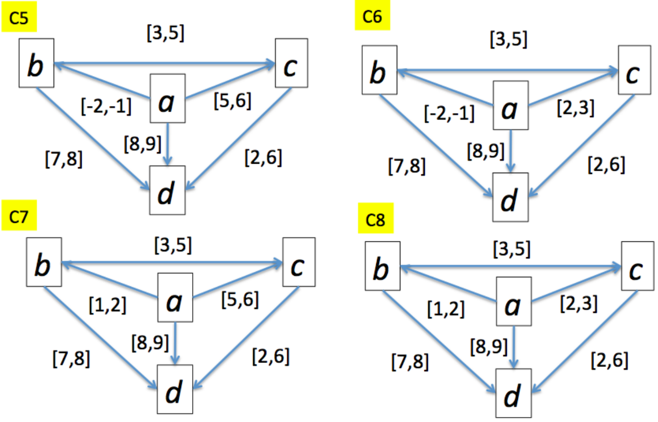


Figure 7: Chronicles (5-8) without temporal restrictions

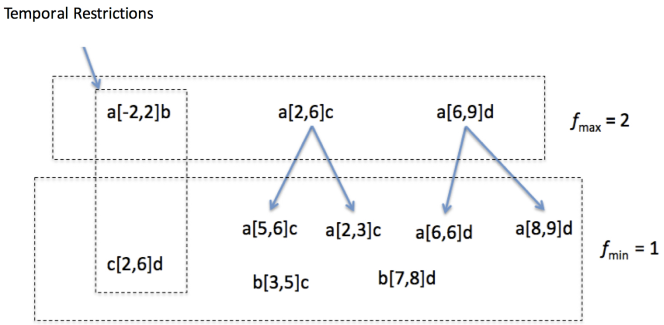


Figure 8: Tree roots using temporal restrictions

The results using temporal restrictions in $HCDAM$ provide 4 chronicles, reducing the number of chronicles by 50%. The tree roots are given in Figure 8. The 4 chronicles C_1 to C_4 are given in Figure 9.

4.3.2. Integration of event information

Another type of expert knowledge that is often available is the occurrence frequency $f_{(e_i)}$ of a single event type e_i. This information is not taken into account in $HDCAM$. Nevertheless it can be very useful to reduce the number of learned chronicles.

Definition 6 (Initial event). *We define the event Φ as a virtual initial event type in all the event sequences of $\mathbb{S}$ such that the occurrence frequency $f_{(e_i)}$ for each event type e_i in the sequence S_k is determined from Φ as the frequency of the pair (Φ, e_i).*

The virtual initial event Φ allows us, without modifying the HDCAM algorithm, to identify the frequency of each event type whereas the original HDCAM only identifies the frequency of event type pairs.

Considering the above example, we use the integration of event information and obtain a unique chronicle, reducing the number of chronicles by 90% (see Figure. 10).

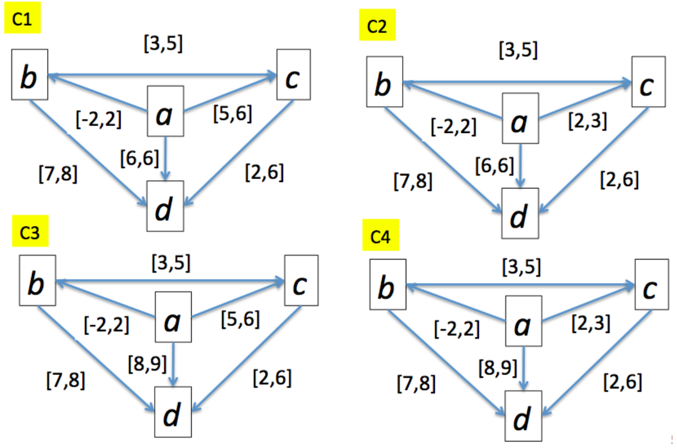


Figure 9: Chronicles using temporal restrictions

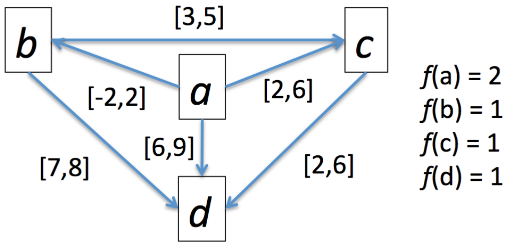


Figure 10: Unique chronicle of the Charge oven system

5. Fault injection framework

This section presents the fault injection framework of the Chronicle Based Alarm Management (CBAM) which is based on a hybrid causal model and a qualitative abstraction process of the continuous behavior. The fault injection tool is used to generate event sequences for the scenarios to be learned.

5.1. Hybrid Causal Model

The hybrid model representing the plant is based on an extended transition system, whose discrete states represent the different modes of operation for which the continuous dynamics are characterized by a causal system. Formally, a hybrid causal system is defined as a tuple:

$$\Gamma = (\vartheta, D, Tr, E, CSD, Init, COMP, DMC) \tag{8}$$

Where

- $\vartheta = \{v_i\}$ is a set of continuous *process variables* which are function of time t.
- D is a set of discrete variables $D = Q \cup \mathbb{K} \cup V_Q$.
  - Q is a set of states q_i of the transition system which represent the system operation modes.
  - The set of auxiliary discrete variables $\mathbb{K} = \{\mathbb{K}_i, i = 1, ...n_c\}$ represents the system configuration in each mode q_i, where $\mathbb{K}_i$ indicates the discrete state of the active components.
  - V_Q is a set of qualitative variables whose values are obtained from the behavior of each continuous variable v_i.
- $E = \Sigma \cup \Sigma^c$ is a finite set of events [5] noted e, where:

[5] More precisely, the elements of E are event types but this term is not used in the hybrid systems literature.

  - Σ is the set of events associated to the procedure actions in startup or shutdown stages.
  - Σ^c is the set of events associated to the behavior of the continuous process variables.

  Unobservable events form the set Σ_{uo}.

- $Tr : Q\times\ \Sigma \to\ Q$ is the transition function. The transition from mode q_i to mode q_j with associated event e is noted (q_i, e, q_j).
- $CSD \supseteq \bigcup_i CSD_i$ is the *Causal System Description* or the causal model used to represent the constraints underlying the continuous dynamics of the hybrid system.

  Every CSD_i associated to a mode q_i, is given by a graph $(G_c = \vartheta \cup \mathbb{K}, \mathbf{In})$. $\mathbf{In}$ is the set of influences where there is an edge $\mathbf{e}_d(v_i, v_j) \in \mathbf{In}$ from $v_i \in \vartheta$ to $v_j \in \vartheta$ if the variable v_i influences variable v_j. A dynamic continuous model $DMC_{In_{\mathbb{K}}}$ is associated to every influence $In_{\mathbb{K}} \in \mathbf{In}$, see Figure 11. The model of the active component corresponds to a transfer function of first order with delay.

- $Init$ is the initial condition of the hybrid system.
- $COMP$ is the set of components.

5.2. Qualitative abstraction of the continuous behavior

In each mode of operation, variables evolve according to the corresponding dynamics. This evolution is represented with qualitative values. The domain $D_o(V_i)$ of a qualitative variable $V_i \in V_Q$ is obtained through the function $f_{qual} : D_o(v_i) \to D_o(V_i)$ that maps the continuous values of variable v_i to ranges defined by limit values (High H_i and Low L_i).

$$f(v_i)_{qual} = \begin{cases} V_i^H \quad if \quad v_i \geq H_i \\ V_i^M \quad if \quad L_i < v_i < H_i \\ V_i^L \quad if \quad v_i \leq L_i \end{cases} \tag{9}$$

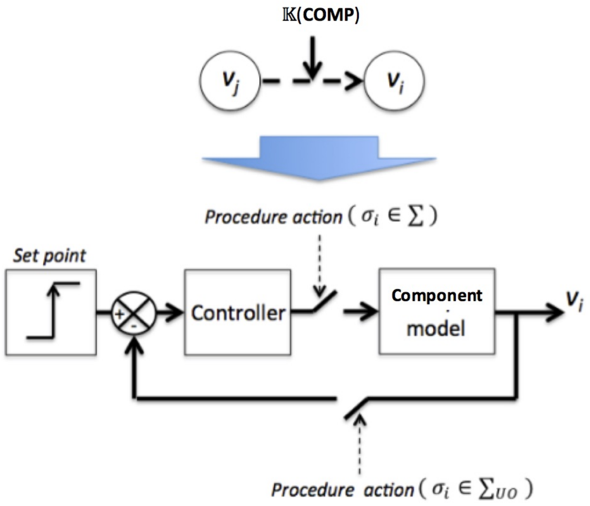

𝕂(COMP)
v_j
v_i
Procedure action ($\sigma_i \in \sum$)
Set point
+
-
Controller
Component model
v_i
Procedure action ($\sigma_i \in \sum_{UO}$)

Figure 11: Dynamic Model of Control *DMC*

The behavior of these qualitative variables is represented by the automaton $G_{V_i} = (V_Q, \Sigma^c, \gamma)$ illustrated in Figure 12 where V_Q is the set of possible qualitative states ($V_i^L : Low$, $V_i^M : Medium$, $V_i^H : High$) of the continuous variable v_i, Σ^c is the finite set of the events associated to the transitions and $\gamma : V_Q \times \Sigma^c \rightarrow V_Q$ is the transition function. The corresponding event generator is defined by the abstraction function $f_{V_Q \rightarrow \sigma}$

$$\begin{aligned} & f_{V_Q \rightarrow \sigma} : V_Q \times \gamma(V_Q, \Sigma^c) \rightarrow \Sigma^c \\ & \forall V_i \in V_Q, (V_i^n, V_i^m) \rightarrow \begin{cases} L(v_i) \quad if \quad V_i^L \rightarrow V_i^M \\ l(v_i) \quad if \quad V_i^M \rightarrow V_i^L \\ H(v_i) \quad if \quad V_i^M \rightarrow V_i^H \\ h(v_i) \quad if \quad V_i^H \rightarrow V_i^M \end{cases} \\ & V_i^n, V_i^m \in \{V_i^L, V_i^M, V_i^H\} \end{aligned} \tag{10}$$

$$\Sigma^c = \bigcup_{v_i \in \vartheta} \{L(v_i), l(v_i), H(v_i), h(v_i)\} \tag{11}$$

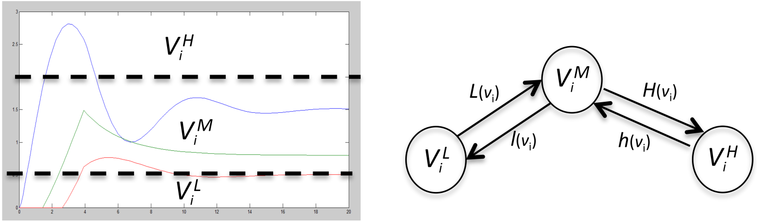


Figure 12: Behavior of the qualitative variables

5.3. Chronicle database

A complex process (Pr) is composed of $n \in N$ different units or areas $Pr = \{Ar_1, Ar_2, ...Ar_n\}$ where each area Ar_m, $m = 1, ..., n$ has $K \in N$ operational modes (e.g startup, shutdown ..) noted O_i, $i = 1, ..., K$. The process behavior in each operating mode can be either normal or faulty. We define the set of failure labels $\Delta_f = f_1, f_2, ..., f_r$ and the complete set of possible labels is $\Delta = N \cup \Delta_f$, here N means normal. To monitor the process and to recognize the different situations (normal or faulty) of the operational modes, we propose to build a chronicle base for each area (CAr_m). Then, for a given area Ar_m, a learned chronicle C_{ij}^m is associated to each couple (O_i, l_j) where $l_j \in \Delta$: When $l_j = l_0 = N$, the chronicle is a model of the normal behavior of the considered system, otherwise $(l_j = f_i)$ the chronicle is a model of the behavior of the system under the occurrence of the fault f_i.

$$CAr_m = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} N & f_1 & f_2 & \dots & f_r \end{array} \\ \begin{bmatrix} C_{10}^m & C_{11}^m & C_{12}^m & \dots C_{1r}^m \\ C_{20}^m & C_{21}^m & C_{22}^m & \dots C_{2r}^m \\ \dots & \dots & \dots & \dots \\ C_{k0}^m & C_{k1}^m & C_{k2j}^m & \dots C_{kr}^m \end{bmatrix} \end{array} \qquad (12)$$

This chronicle database is to be submitted to a chronicle recognition system. The chronicle recognition system is in charge to identify in an observable flow of events all the possible matching with the set of chronicles from which the situation (normal or faulty) can be assessed.

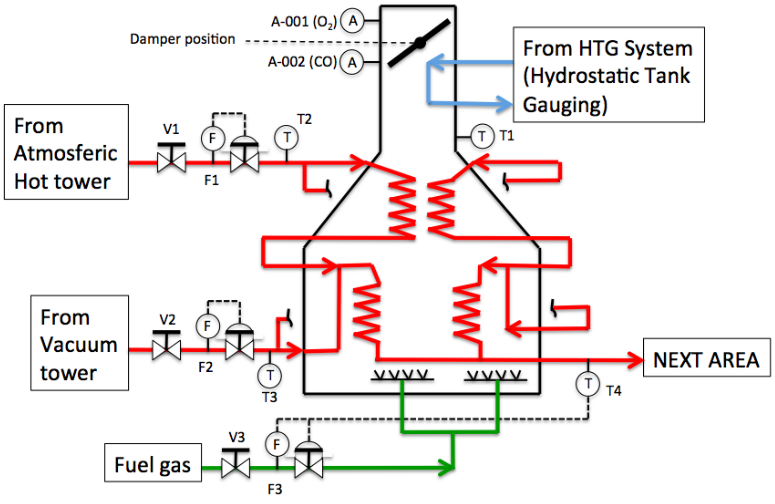


Figure 13: Vacuum oven

6. Case study - Vacuum oven

The case study is issued from the Cartagena Refinery in Colombia. This refinery has been recently enriched with news units such as an Hydrostatic Tank Gauging (HTG), an atmospheric hot tower, a vacuum tower and a vacuum oven between other elements. Our proposal aims helping the operator to recognize normal or dangerous conditions during the startup and shutdown stages of the refinery equipped with these new equipments.

Let us firstly focus on the vacuum oven unit presented Figure 13. Vacuum is a condition to protect the steel parts and heated metals from the negative influence of the air atmosphere. A vacuum oven is usually an oven in which vacuum is maintained during the process. The charge of this oven is a mixture of the reduced oil coming from the section of the hot atmospheric tower and a recycle produced in the section of the vacuum tower. This furnace has flue gas temperature indicators at the outlet of the radiation section, as well as at the outlet of the flue. The reduced oil flow through the two main coils passes through temperature sensors **T2** and **T3** and then each coil is divided into two coils. The operator controls these flows with the valves (**V1,V2**). The temperature control inside of the oven starts when the fuel gas system valve **V3** is opened. The inside temperature of the oven is monitored with **T1** and the outside temperature of the oil is monitored with **T4**. The flows in the system are monitored by three sensors (**F1,F2** and **F3**).

The standard operating procedure of the refinery is very constrained and specifies the standard procedural actions the operators must execute during the startup and shutdown stages. The correct execution of the whole operating procedure supposes that the operators execute the procedural actions planned for a normal evolution of the procedure.

Therefore, in case of an abnormal situation, the process evolution due to the procedural actions executed by operators and so the continuous variable evolutions are no more consistent with the standard operating procedure. This section shows how abnormal situations can be captured into chronicles built according the proposed Chronicle Based Alarm Management ($CBAM$) method. The so built chronicle base could be then considered by a recognition system to recognize the normal or faulty situations when they occur.

The $CBAM$ method relies on several steps (see section 3) leading to the construction of a chronicle base. Next sections, detail each of these steps.

6.1. Hybrid features of the vacuum oven

The vacuum oven process is composed of passive components, active components and sensors. Passive components are components whose operational state cannot be modified via an external action (e.g. the oven structure (Ov)) unlike active components whose states can be changed by a procedural action (e.g. the three valves (**V1, V2,** and **V3**) that can be switched from opened to closed and closed to opened). The sensors, correspond to the instrumentation that measures the continuous variables e.g. flow sensors and temperature sensors. Since there are three active components, the vacuum oven system obviously involves hybrid behavior. Modeling the behavior of this hybrid system involves a set of continuous variables and a set of discrete variables (see section 5.1). The continuous variables are the temperature (T1, T2, T3 and T4) and the flows (F1, F2, and F3) (see Figure 13).

The discrete variables are:

- the states of the transition system representing the system operating modes. The vacuum oven has thus $2^3 = 8$ configurations and operating modes denoted q_0 to q_7 due to the three valves (**V1, V2,** and **V3**) each with two possible modes (opened and closed).

- V_Q the set of qualitative variables values are obtained from the behavior of continuous variables as explained Section 5.2. In this case study, continuous variable domain partitioning has been chosen according to expertise knowledge and to limit values specified in standard operating procedures. $V_Q = \{\bigcup_{i=1}^{3}\{Fi^L, Fi^M, Fi^H\}\} \cup \{\bigcup_{i=1}^{4}\{Ti^L, Ti^M, Ti^H\}\}$.

- the set of auxiliary discrete variables indicating the state of active components is given by: $\mathbb{K} = \{\mathbb{K}_i, i = 0, ...7\}$ i.e the system configuration associated to an operating mode. The configuration is defined by the state (opened or closed) of the three valves. For a normal startup the vacuum oven evolves through the modes q_0,q_3,q_5,q_6 and q_7. In the mode q_0 the three valves are closed and then $\mathbb{K}_0 = 0$. When the two first valves are closed and the valve **V3** is opened, the system in the mode q_3 and $\mathbb{K}_3 = 3$. In q_5, **V3** and **V1** are opened and **V2** is closed then $\mathbb{K}_5 = 2$. For q_7 all the valves are opened and $\mathbb{K}_7 = 7$.

The discrete part of the model is given by the underlying DES (Discrete Event System). This model is obtained from the operating specifications described in the standard operating procedures. To each operation mode q_i is associated a Causal System Description (CSD_i) to identify the influences between the continuous variables $F1$, $F2$, $F3$, $T1$, $T2$, $T3$ and $T4$. For the vacuum oven, the underlying DES is shown Figure 14 on the left. Green arrows indicate the system evolutions during a startup stage. The $CSDs$ associated to the operating modes (i.e q_0, q_3, q_5, q_6 and q_7) involved in a startup stage are shown

Figure 14 on the right. In each CSD, the edges are labeled by the influences between the variables. These influences are defined by the configuration of the valves. For instance the influence between $F3$ and $T1$ depends on the configuration of the valve **V3** noted $\mathbb{K}(\mathbf{V3})$. A bold edge indicates that the influence is active.

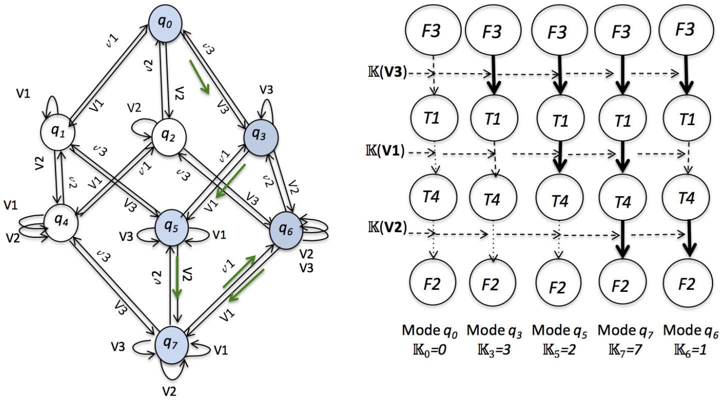
6.2. Event type identification

The set of event types E considered into the chronicles is defined by $E = \Sigma \cup \Sigma^c$ and corresponds to the set of event types of the vacuum oven hybrid system. The set of event types associated to procedural actions concern mainly the valves of the oven: $\Sigma = \{V1, V2, V3, v1, v2, v3, M2A\}$, where $V1$ (resp. $V2, V3$) denotes the switch of the valve **V1** (resp. **V2**,**V3**) from closed to opened. $v1$ (resp. $v2, v3$) the switch of the valve **V1** (resp. **V2**,**V3**) from opened to closed. The event $M2A$ corresponds to the change from *manual* to *automatic* operating, closing the control loops. In the reminder we assume that this event is the only unobservable: $M2A \in \Sigma_{uo}$ and $\Sigma_o =${V1,V2,V3,v1,v2,v3}.

The set of event types associated to the behavior of the continuous variables is defined by the abstraction function (see Section 5.2).

$$\Sigma^c = \begin{cases} L_{(F1)}, l_{(F1)}, H_{(F1)}, h_{(F1)}, \\ L_{(F2)}, l_{(F2)}, H_{(F2)}, h_{(F2)}, \\ L_{(F3)}, l_{(F3)}, H_{(F3)}, h_{(F3)}, \\ L_{(T1)}, l_{(T1)}, H_{(T1)}, h_{(T1)}, \\ L_{(T2)}, l_{(T2)}, H_{(T2)}, h_{(T2)}, \\ L_{(T3)}, l_{(T3)}, H_{(T3)}, h_{(T3)}, \\ L_{(T4)}, l_{(T4)}, H_{(T4)}, h_{(T4)} \end{cases} \tag{13}$$

The occurrence of the event types Σ^c depends on the influence between the continuous variables. These influences are captured in each causal system description associated to each operation mode (See Figure 14 on the right).



q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
V1
V2
V3
F3
T1
T4
F2
$\mathbb{K}$(V3)
$\mathbb{K}$(V1)
$\mathbb{K}$(V2)
Mode q_0 $\mathbb{K}_0$=0
Mode q_3 $\mathbb{K}_3$=3
Mode q_5 $\mathbb{K}_5$=2
Mode q_7 $\mathbb{K}_7$=7
Mode q_6 $\mathbb{K}_6$=1

Figure 14: Startup stage of the vacuum oven: underlying DES and Causal System Description

6.3. Event sequence generation

The event sequences used as input for the chronicle learning stage are generated according to the behavior of the system in a given scenario. In this case study, only a normal scenario is considered.

For the startup stage, the initial conditions are that the oven is empty and the valves **V1**, **V2** and **V3** are closed. In this situation, the values for all the continuous variables are below their low limits ($F1$, $F2$, $F3$, $T1$, $T2$, $T3$, $T4$). Then according to the standard procedure description, the scenario starts with the opening of the valve **V3** that is to say the occurrence of an event of type $V3$. After this event type occurrence, the system is in the mode of operation q_3 where only the valves **V1** and **V2** are closed. The variable $T1$ increases and an event of type L_{T1} must occur indicating that the internal oven temperature has passed the low limit. Then the flow of the fuel gas reaches its low limit and an event of type L_{F3} occurs. So, the ordered sequence of event types that has occurred is $V3, L_{T1}, L_{F3}$. Passing the low limit of $F3$ is the condition for continuing the procedure by the opening action of the valve **V1** (i.e occurrence of an event of type $V1$). When the operator opens the valve **V1**, the system evolves to the mode of operation q_5 where the internal flow in the vacuum oven starts. In this state, the flow $F1$ and the outflow temperature $T4$ increase (event of type L_{T4} followed by an event of type L_{F1}). The next event that occurs is of type H_{F1} indicating that the flow $F1$ has passed its high level. At this stage, the ordered sequence of event types is given by: $V3, L_{T1}, L_{F3}, V1, L_{T4}, L_{F1}, H_{F1}$. The next procedural action is the closing of the valve **V1** (occurrence of an event of type $v1$) followed by the opening of the valve **V2** ($V2$). Then, the high limit of the temperature $T1$ is reached and an event of type H_{T1} occurs. The flow $F1$ decreases from its high limit (event type h_{F1}). An event of type L_{F2} occurs because the flow from **V2** increases. The high limit in the temperature $T4$ induced an event of type (H_{T4}). Following up with the procedure, due to the high limit of $F2$ an event of type H_{F2} occurs. At this instant, the ordered sequence of event types is $V3_{co}, L_{T1}, L_{F3}, V1,\ L_{T4}, L_{F1}, H_{F1}, v1, V2, H_{T1}, h_{F1}, L_{F2},H_{T4}, H_{F2}$. In this situation, an unobservable event of type $M2A$ occurs and the control loops are closed, carrying the system to a steady state. Note that in this study, we assume that the control loops are closed immediately after an event of type H_{F2}. Then, $F1$, $T1$ and $T4$ decrease (events l_{F1}, h_{T1} and h_{T4}). Finally, the sequence ends by and event of type h_{F2} and by the opening of the valve **V1** so that the last event that occurs in this normal startup is of type L_{F1}. The final ordered sequence that must occur in this scenario is $V3, L_{(T1)}, L_{(F3)}, V1, L_{(T4)}, L_{(F1)}, H_{(F1)}, v1, V2, H_{(T1)}, h_{(F1)}, L_{(F2)}, H_{(T4)}, H_{(F2)}, l_{(F1)},h_{(T1)}, h_{(F2)}, V1, L_{(F1)}$.

By simulation three different event sequences (S_1, S_2 and S_3) have been obtained all of them associated with the same scenario i.e a normal startup of the vacuum oven. These sequences differ only from the event occurrence dates:

- $S_1 = \langle (V3, 1), (L_{(T1)}, 3), (L_{(F3)}, 5), (V1, 6), (L_{(T4)}, 7), (L_{(F1)}, 8), (H_{(F1)}, 12),$ $(v1, 13), (V2, 14), (H_{(T1)}, 15), (h_{(F1)}, 16), (L_{(F2)}, 17), (H_{(T4)}, 19), (H_{(F2)}, 22),$ $(l_{(F1)}, 24), (h_{(T1)}, 25), (h_{(T4)}, 26), (h_{(F2)}, 27), (V1, 42), (L_{(F1)}, 45)\rangle$

- $S_2 = \langle (V3, 1), (L_{(T1)}, 7)(L_{(F3)}, 13), (V1, 18), (L_{(T4)}, 21), (L_{(F1)}, 24), (H_{(F1)}, 32),$ $(v1, 35), (V2, 37), (H_{(T1)}, 40), (h_{(F1)}, 45), (L_{(F2)}, 48), (H_{(T4)}, 54), (H_{(F2)}, 61),$ $(l_{(F1)}, 65), (h_{(T1)}, 68), (h_{(T4)}, 72), (h_{(F2)}, 76)$ $(V1, 96), (L_{(F1)}, 101)\rangle$

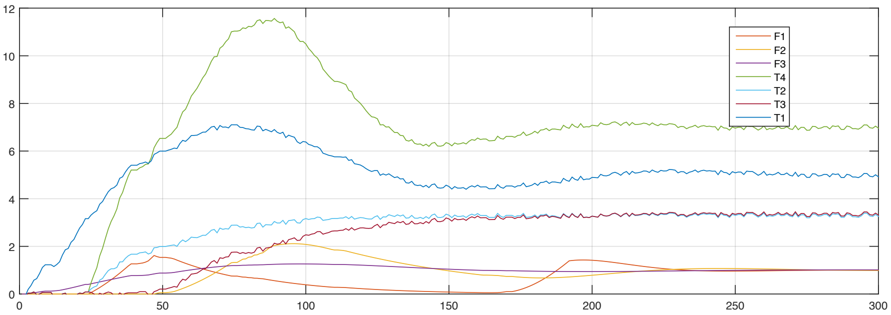


Figure 15: Simulation of a normal startup

- $S_3 = \langle (V3,2), (L_{(T1)},6), (L_{(F3)},9), (V1,12), (L_{(T4)},14), (L_{(F1)},16)$ $(H_{(F1)},22), (v1,24), (V2,25), (H_{(T1)},27), (h_{(F1)},30), (L_{(F2)},32)$ $(H_{(T4)},36), (H_{(F2)},41), (l_{(F1)},43), (h_{(T1)},45), (h_{(T4)},48), (h_{(F2)},50)$ $(V1,68), (L_{(F1)},71)\rangle$

Figure 15 shows one simulation of the scenario leading to the generation of the sequence S_1. The values of the flow variables are normalized between 0 and 1 and for temperature, the value of 0 corresponds to the ambient temperature. The time unit is in second.

6.4. Construction of the chronicle database

The vacuum oven is associated to the area Ar_2. The learning algorithm learns one unique chronicle C^2_{10} for a normal startup as shown in Figure 16.

During the learning stage expert knowledge as been integrated through three temporal restrictions: $TR_{(V3,LF3)}$=$V3$[6,8]$L_{(F3)}$, $TR_{(V1,LF1)}$=$V1$[-76,82]$L_{(F1)}$, $TR_{(LF2,V2))}$=$L_{(F2)}$[2,8]$V2$. This information is reported on the corresponding edges of the chronicle graph on Figure 16. While the temporal restrictions coming from expertise do not save much effort for building the constraint data base, it is important to notice that they achieve to cut down the number of chronicles to one.

The three input sequences S_1,S_2 and S_3 used for the learning stage constitute three possible instances of the chronicle C^2_{10}. However, the chronicle has an interesting generalization power and it defines partial orders on the occurrence of event types, which results in a graph.

7. Conclusion and future work

This paper addresses the problem of alarm management based on chronicles recognition. The process situations are modeled by chronicles. Chronicles are obtained via a chronicle learning algorithm working on multiple input sequences and integrating expert knowledge if available. The method designed for hybrid systems relies on a theoretical framework in which hybrid features are captured. The event types of the chronicles are defined from an abstraction of the continuous behavior.

The paper provides experimental results from a petrochemical process real case study, a vacuum oven. This kind of processes is well-suited to chronicle based supervision. Indeed, the temporal sequences corresponding to start up and shut down are of reasonable size in terms of number of events. This is important because the HDCAM algorithm is exponentially dependent on the number of events in the input sequences. Like for the algorithms of [37] or [36], this is the price to pay for a complete chronicle discovery algorithm. HDCAM returns discovered chronicles in highest frequency order, so the user can stop the discovery at anytime. The user can also specify a particular frequency range. This paper also provides some ideas to mitigate this issue by integrating expert knowledge (Section 4.3).

One interesting problem that we are planning to address is to extend the chronicle learning algorithm by integrating notably negative examples and forgetting capabilities. Another research perspective is to consider Hazard and Operability studies or event tree analysis for the scenarios determination and the event types identification. Finally, another issue is the exploitation of the chronicles recognition as a *super-alarm* generator providing to the operators relevant information about the process situation, increasing the reliability of this layer of protection.

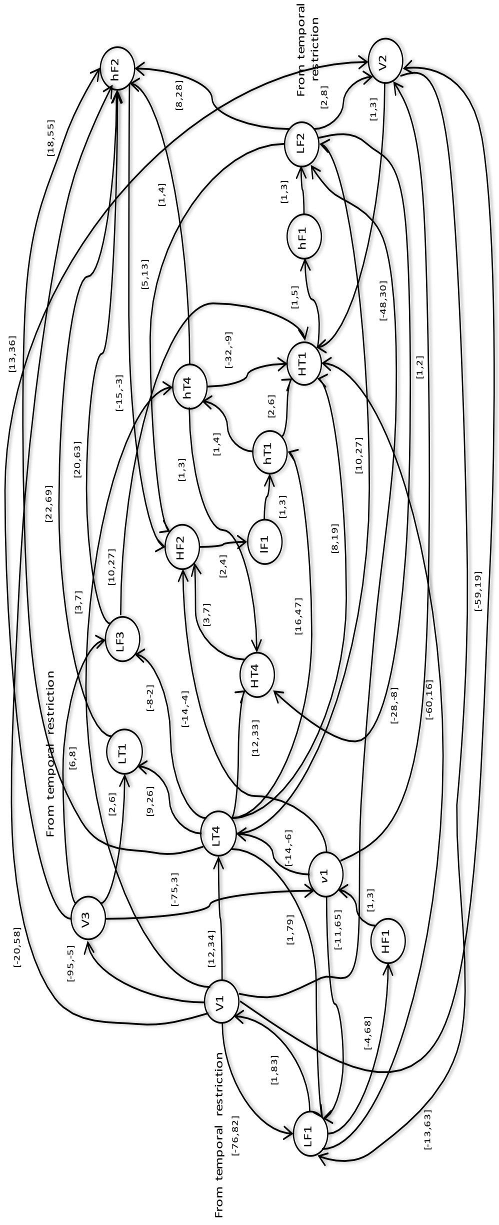


Figure 16: Vacuum oven: chronicle of the normal startup, C^2_{10}

Acknowledgments

The ECOPETROL - ICP engineers Carlos Agudelo, Jorge Prada, Francisco Cala and Gladys Valderrama help us to develop and validate the simulations.

References

[1] T. Stauffer, N. Sands, D. Dunn, Alarm management and isa-18 a journey, not a destination, in: Texas A & M Instrumentation Symposium, 2000.

[2] D. Beebe, S. Ferrer, D. Logerot, The connection of peak alarm rates to plant incidents and what you can do to minimize, Process Safety Progress 32 (1) (2013) 72–77. `doi:10.1002/prs.11539`.
URL `http://dx.doi.org/10.1002/prs.11539`

[3] M.-O. Cordier, C. Dousson, Alarm driven monitoring based on chronicles, in: 4th Sumposium on Fault Detection Supervision and Safety for Technical Processes (SafeProcess), Budapest, Hungary, 2000, pp. 286–291.

[4] G. Carrault, M. Cordier, R. Quiniou, M. Garreau, J. Bellanger, A. Bardou, A model-based approach for learning to identify cardiac arrhythmias, Lecture Note on Artificial Intelligence 1620 (1999) 165–174.

[5] B. Morin, H. Debar, Correlation on intrusion: an application of chronicles, in: 6th International Conference on recent Advances in Intrusion Detection RAID, Pittsburgh, USA, 2003.

[6] F. Fessant, F. Clérot, C. Dousson, Mining of an alarm log to improve the discovery of frequent patterns, in: Industrial Conference on Data Mining, 2004, pp. 144–152.

[7] A. Subias, L. Travé-Massuyès, E. LeCorronc, Learning chronicles signing multiple scenario instances, IFAC World Congress, Le Cap, South Africa, 26-29 August.

[8] M. Schleburg, L. Christiansen, N. F. Thornhill, A. Fay, A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system, Journal of Process Control 23 (6) (2013) 839 – 851. `doi:http://dx.doi.org/10.1016/j.jprocont.2013.03.010`. URL `http://www.sciencedirect.com/science/article/pii/S0959152413000541`

[9] S. R. Kondaveeti, I. Izadi, S. L. Shah, T. Black, T. Chen, Graphical tools for routine assessment of industrial alarm systems, Computers and Chemical Engineering 46 (2012) 39 – 47. `doi:http://dx.doi.org/10.1016/j.compchemeng.2012.06.042`. URL `http://www.sciencedirect.com/science/article/pii/S0098135412002268`

[10] F. Higuchi, I. Yamamoto, T. Takai, M. Noda, H. Nishitani, Use of event correlation analysis to reduce number of alarms, in: C. A. O. d. N. Rita Maria de Brito Alves, E. C. Biscaia (Eds.), 10th International Symposium on Process Systems Engineering: Part A, Vol. 27 of Computer Aided Chemical Engineering, Elsevier, 2009, pp. 1521 – 1526. `doi:http://dx.doi.org/10.1016/S1570-7946(09)70644-3`. URL `http://www.sciencedirect.com/science/article/pii/S1570794609706443`

[11] Z. Ge, Z. Song, Multimode process monitoring based on bayesian method, Journal of Chemometrics 23 (12) (2009) 636–650. `doi:10.1002/cem.1262`. URL `http://dx.doi.org/10.1002/cem.1262`

[12] X. Liu, M. Noda, H. Nishitani, Evaluation of plant alarm systems by behavior simulation using a virtual subject, Computers and Chemical Engineering 34 (3) (2010) 374 – 386. `doi:http://dx.doi.org/10.1016/j.compchemeng.2009.11.017`. URL `http://www.sciencedirect.com/science/article/pii/S0098135409003020`

[13] F. Yang, S. Shah, D. Xiao, T. Chen, Improved correlation analysis and visualization of industrial alarm data, {ISA} Transactions 51 (4) (2012) 499 – 506. doi:http://dx.doi.org/10.1016/j.isatra.2012.03.005.
URL http://www.sciencedirect.com/science/article/pii/S0019057812000341

[14] I. Izadi, S. L. Shah, D. S. Shook, S. R. Kondaveeti, T. Chen, A framework for optimal design of alarm systems, IFAC Proceedings Volumes 42 (8) (2009) 651 – 656. doi:http://dx.doi.org/10.3182/20090630-4-ES-2003.00108.
URL http://www.sciencedirect.com/science/article/pii/S1474667016358517

[15] A. Pariyani, W. D. Seider, U. G. Oktem, M. Soroush, Dynamic risk analysis using alarm databases to improve process safety and product quality: Part iibayesian analysis, AIChE Journal 58 (3) (2012) 826–841. doi:10.1002/aic.12642.
URL http://dx.doi.org/10.1002/aic.12642

[16] J. Liu, D.-S. Chen, Nonstationary fault detection and diagnosis for multimode processes, AIChE Journal 56 (1) (2010) 207–219. doi:10.1002/aic.11999.
URL http://dx.doi.org/10.1002/aic.11999

[17] J. Zhu, Y. Shu, J. Zhao, F. Yang, A dynamic alarm management strategy for chemical process transitions, Journal of Loss Prevention in the Process Industries 30 (2014) 207 – 218. doi:http://dx.doi.org/10.1016/j.jlp.2013.07.008.
URL http://www.sciencedirect.com/science/article/pii/S0950423013001393

[18] Z. Jing, L. Boang, Y. Hao, Fault diagnosis strategy for startup process based on standard operating procedures, in: 2013 25th Chinese Control

and Decision Conference (CCDC), 2013, pp. 4221–4226. doi:10.1109/CCDC.2013.6561692.

[19] P. Duan, F. Yang, T. Chen, S. L. Shah, Direct causality detection via the transfer entropy approach, IEEE Transactions on Control Systems Technology 21 (6) (2013) 2052–2066. doi:10.1109/TCST.2012.2233476.

[20] F. Yang, D. Xiao, Progress in root cause and fault propagation analysis of large-scale industrial processes, J. Control Sci. Eng. 2012 (2012) 8:8–8:8.
URL http://dl.acm.org/citation.cfm?id=2215493.2336517

[21] A. Bhagwat, R. Srinivasan, P. Krishnaswamy, Fault detection during process transitions: a model-based approach, Chemical Engineering Science 58 (2) (2003) 309 – 325. doi:http://dx.doi.org/10.1016/S0009-2509(02)00520-1.
URL http://www.sciencedirect.com/science/article/pii/S0009250902005201

[22] R. Srinivasan, P. Viswanathan, H. Vedam, A. Nochur, A framework for managing transitions in chemical plants, Computers & Chemical Engineering 29 (2) (2005) 305 – 322. doi:http://dx.doi.org/10.1016/j.compchemeng.2004.09.024.
URL http://www.sciencedirect.com/science/article/pii/S0098135404002868

[23] S. Xu, A. Adhitya, R. Srinivasan, Hybrid model-based framework for alarm anticipation, Industrial & Engineering Chemistry Research 53 (13) (2014) 5182–5193. arXiv:http://dx.doi.org/10.1021/ie4014953, doi:10.1021/ie4014953.
URL http://dx.doi.org/10.1021/ie4014953

[24] C. Dousson, P. Gaborit, M. Ghallab, Situation recognition: representation and algorithms, in: IJCAI: International Joint Conference on Artificial Intelligence, Chambéry, France, 1993, pp. 166–172.

[25] J. Vasquez, L. Travé-Massuyès, A. Subias, F. Jimenez, C. Agudelo, Chronicle based alarm management in startup and shutdown stages, International Workshop on Principles of Diagnosis (DX-2015), Paris.

[26] J. Vsquez, L. Trav-Massuys, A. Subias, F. Jimenez, C. Agudelo, Alarm management based on diagnosis, IFAC-PapersOnLine 49 (5) (2016) 126 – 131, 4th {IFAC} Conference on Intelligent Control and Automation SciencesICONS 2016Reims, France, 1-3 June 2016.

[27] R. Pons, A. Subias, L. Travé-Massuyès, Iterative hybrid causal model based diagnosis: Application to automotive embedded functions, Engineering Applications of Artificial Intelligence, Volume 37, January 2015, Pages 319 - 335.

[28] B. Guerraz, C. Dousson, Chronicles construction starting from the fault model of the system to diagnose, in: International Workshop on Principles of Diagnosis (DX04), Carcassonne, France, 2004, pp. 51–56.

[29] E. Mayer, Inductive learning of chronicles, in: European Conference on Artificial Intelligence, 1998, pp. 471–472.

[30] M. W. Floyd, B. Esfandiari, A case-based reasoning framework for developing agents using learning by observation, in: ICTAI, 2011, pp. 531–538.

[31] M. Floyd, M.Bicakci, B. Esfandiari, Case-based learning by observation in robotics using a dynamic case representation, in: FLAIRS Conference, 2012.

[32] T. Guyet, R. Quiniou, Mining temporal patterns with quantitative intervals, in: ICDM Workshops, 2008, pp. 218–227.

[33] W. Wang, T. Guyet, R. Quiniou, M.-O. Cordier, F. Masseglia, Online and adaptive anomaly detection: detecting intrusions in unlabelled audit data streams, in: EGC, 2009, pp. 457–458.

[34] T. Guyet, R. Quiniou, Extracting temporal patterns from interval-based sequences, in: IJCAI, 2011, pp. 1306–1311.

[35] T. Mitsa, Temporal data mining, CRC Press, 2010.

[36] C. Dousson, T. V. Duong, Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems, in: IJCAI 99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, 1999, pp. 620–626.

[37] D. Cram, B. Mathern, A. Mille, A complete chronicle discovery approach: application to activity analysis, Expert Systems 29 (4) (2012) 321–346.

[38] H. Mannila, H. Toivonen, A. I. Verkamo, Discovery of frequent episodes in event sequences, Data Mining and Knowledge Discovery 1 (1997) 259–289.

[39] F. Fessant, F. Clérot, An efficient som-based pre-processing to improve the discovery of frequent patterns in alarm logs, in: DMIN, 2006, pp. 276–282.