# Chronicle Based Alarm Management

John William Vasquez Capacho

# THÈSE

**En vue de l'obtention du**

## DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

**Délivré par :**
*l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*
Cotutelle *Universidad de los Andes, Colombie*

**Présentée et soutenue le** *13/10/2017* **par :**

JOHN WILLIAM VÁSQUEZ CAPACHO

## CHRONICLE BASED ALARM MANAGEMENT

**JURY**

| | | |
|---|---|---|
| CLAUDIA ISAZA | Professor | Universidad de Antioquia |
| MARC LE GOC | Professeur d'Université | Université Aix-Marseille |
| FELIPE MUÑOZ | Professor | Universidad de los Andes |
| AUDINE SUBIAS | Maître de conférences | INSA Toulouse |
| FERNANDO JIMENEZ | Professor | Universidad de los Andes |
| LOUISE TRAVÉ-MASSUYÈS | Directeur de recherche | LAAS-CNRS |

**École doctorale et spécialité :**
*EDSYS : Automatique 4200046*
**Unité de Recherche :**
*Laboratoire d'Analyse et d'Architecture des Systèmes LAAS-CNRS*
**Directeur(s) de Thèse :**
*Audine SUBIAS*, *Fernando JIMENEZ* et *Louise TRAVÉ-MASSUYÈS*
**Rapporteurs :**
*Claudia ISAZA* et *Marc Le GOC*

Quiero dedicar esta tesis especialmente a mi amada esposa, que pacientemente tuvo el coraje de soportar mis prolongadas ausencias... siendo su amor, el motor de mi energía.

A mi madre querida, cuyas oraciones y buenos ánimos me mantuvieron siempre adelante. A mi papá por sus valiosos consejos y palabras de apoyo en todas mis metas.

# Acknowledgements

Quiero agradecerle a Dios la oportunidad, que en un universo de infinitas posibilidades, hoy pueda escribir estas palabras. Agradezco a COLCIENCIAS y COLFUTURO por financiar mis estudios. A mis asesores, los profesores Louise, Audine y Fernando, quiero agradecerles inmensamente su disposición y ayuda en el desarrollo exitoso de este trabajo de grado. Doy infinitas gracias al profesor Felipe por todo su apoyo y todas las sugerencias oportunas que me hizo durante este trabajo. A Carlos, mi amigo y compañero, agradezco sus recomendaciones y sugerencias en el desarrollo de esta tesis. Todo mi agradecimiento a los profesores Marc y Claudia por aceptar ser los revisores de este trabajo. Agradezco la amistad a mi parcero Gustavo, con quien tenemos una visión de trabajo mancomunado entre la academia y la industria. Finalmente, agradezco a todas y cada una de las presonas que de una u otra forma me ayudaron a cumplir esta meta.

# Abstract

Ce travail de thèse a été réalisé dans le cadre d'une thèse en co-tutelle entre l'INSA, Toulouse, et l'Université des Andes, Colombie, avec un financement de Colciencias. Ce travail est motivé par la nécessité pour l'industrie de détecter des situations anormales pendant les phases de démarrage et d'arrêt des installations. La sécurité des installations industrielles implique une gestion intégrée de tous les facteurs et événements pouvant causer des accidents. La gestion des alarmes peut être formulée comme un problème de reconnaissance de motifs événementiels dans lequel des modèles temporels sont utilisés pour caractériser différentes situations typiques, en particulier pendant les phases de démarrage et d'arrêt. Dans cette thèse une nouvelle approche de gestion des alarmes basée sur un processus de diagnostic est proposée. En supposant que les alarmes et les actions du mode opératoire standard sont des événements discrets, l'étape de diagnostic repose sur la reconnaissance de situation pour fournir aux opérateurs des informations pertinentes sur les défaillances induisant le flux d'alarmes.

La reconnaissance de situation est basée sur des chroniques qui caractérisent les situations d'interdit et qui sont apprises de manière automatique. Les chroniques sont apprises à partir de séquences d'événements représentatives obtenues par simulation et constituant l'entrée d'une version étendue de l'Algorithme de Découverte de Chroniques Heuristique Modifié ($HCDAM$). HCDAM a été étendu dans cette thèse pour prendre en compte des connaissances expertes sous la forme de restrictions temporelles spécifiques. Un modèles hybride causal du procédé est utilisé pour vérifier les séquences d'entrée et pour expliquer et donner du sens aux chroniques apprises.

La méthodologie de gestion des alarmes basée sur des chroniques CBAM (comme *Chronicle Based Alarm Management* ) proposée dans cette thèse fusionne différentes techniques pour tenir compte de l'aspect hybride et des procédures opérationnelles standard des processus concernés. Comparée aux autres approches de gestion d'alarmes, cette approche se caractérise par l'utilisation de l'information sur les actions procédurales liées au comportement des variables continues dans un processus formel de diagnostic. Des informations spécifiques sont obtenues à chaque étape de la méthodologie CBAM qui se résume en trois étapes :

1. Étape 1 : Identification du type d'événement
   à partir des procédures d'exploitation standard et de l'évolution des variables continues, cette étape détermine l'ensemble des types d'événements pendant les phases de démarrage et d'arrêt.

2. Étape 2 : Génération de séquence d'événements
   à partir de l'expertise et d'une procédure d'abstraction événementielle, cette étape détermine la date d'apparition de chaque type d'événement pour la construction des séquences d'événements représentatives. Une séquence d'événements représentatifs est l'ensemble des types d'événements avec leurs dates d'occurrence qui peuvent être associées à un scénario spécifique du processus. Cette étape se conclut avec la vérification des séquences d'événements représentatives à l'aide du modèle causal hybride.

3. Étape 3 : Construction de la base de chroniques
   à partir des séquences d'événements représentatives et des restrictions temporelles dans chaque scénario, cette étape détermine la base de chroniques à l'aide de l'algorithme $HCDAM$.

La methode proposée pour la gestion des alarmes est illustrée par deux cas d'etude representatifs du domaine pétrochimique.

# Abstract

This thesis work was carried out in the framework of a co-tutelle between INSA, Toulouse, and the University of the Andes, Colombia, with financial support of Colciencias. This work is motivated by the need of the industry to detect abnormal situations in the plant startup and shutdown stages. Industrial plants involve integrated management of all the events that may cause accidents and translate into alarms. Process alarm management can be formulated as an event-based pattern recognition problem in which temporal patterns are used to characterize different typical situations, particularly at startup and shutdown stages. In this thesis, a new approach for alarm management based on a diagnosis process is proposed. Considering 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 about the failures inducing the alarm flow. The situation recognition is based on chronicles that characterize the situations of interest and are learned automatically. The chronicles are learned from representative event sequences obtained by simulation and given as input to an extended version of the Heuristic Chronicle Discovery Algorithm Modified ($HCDAM$). HCDAM has been extended in this thesis to account for expert knowledge in the form of specific temporal restrictions. A hybrid causal model of the process is used to verify the input event sequences and to explain and provide semantics to the learned chronicles.

The Chronicle Based Alarm Management (CBAM) methodology proposed in this thesis involves different techniques to take the hybrid aspect and the standard operational procedures of the concerned processes into account. Compared to other approaches of alarm management, this approach uses information about the procedural actions related to the continuous variables behavior in a formal diagnosis process. Specific information is obtained in each step of the CBAM methodology, and it is summarized in three steps:

1. Step 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. Step 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. This step concludes verifying the representative event sequences using the hybrid causal graph.

3. Step 3: Chronicle database construction
   From the representative event sequences and temporal restrictions of each scenario, this step determines the chronicle database using the extended $HCDAM$ algorithm.

The proposed framework for alarm management is illustrated with two case studies representative of the petrochemical field.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

## 1.1 Global overview and motivation

The increasing automation of industrial production processes has resulted also in an increase of the complexity of the control systems. Such systems are based on digital technologies that required increase their monitoring capacity in terms of the number of variables that can be treated with its processing speed and communication capacity [10], [11]. This complexity makes extremely difficult the diagnosis of failures that may occur. Currently, on highly automated systems fault diagnosis performed automatically with automatic reconfiguration on embedded control system is an usual requirement [55], [70],[78]. The ultimate goal is to optimize the availability, reliability and safety of production processes [69].

The operation of many industrial processes, especially in the petrochemical sector, involves inherent risks due to the presence of dangerous materials like gases and chemicals; which in some conditions can cause emergencies. In these types of industrial processes, safety is supplied by layers of protection, which begin with a safe design (*Process design level*) and an effective process control (*Process Control level)*, followed by the manual (*Operator interventions level*) and automatic (*Safety Instrumented System level*) prevention layers, and concluded with layers to mitigate the consequences of a critical event (*Active protection level*, *Passive protection level*, *Plant emergency response level* and *Community emergency response level*) as shown Figure. 1.1. The petrochemical industrie's losses have been estimated at 20 billion dollars in the U.S. alone each year, and the AEM (Abnormal Events Management) has been classified as a critical problem [95], [122], [111]. An integrated management of the critical factors in the process, ensures an optimum reliability level in the industrial plants [50]. Factors such as the control of the process variables, procedures and steps followed in transitional stages try to keep the plants within the operating established "limits" [46]. While, on starting or shutdown procedures, the quantity of signals increases, the plant safety needs to involve an integrated management of those factors analyzing the causes of the accidents.

In other words, these factors must be managed together, and not separately, because if any of them is left outside, unattended or decreased, the security would be threatened [1]. The critical factors of the process work that must be managed together are:

- Facilities safely,
- Control of process variables,
- Safe behaviors,
- Valid procedures.

This raises the need not only of a diagnosis system that helps to maintain safe the process increasing the availability of the installation, but also of new alarm management methodologies [108]. Industrial plant safety involves an integrated management of all the factors that may cause accidents. Hence alarm management is one aspect of great interest in the safety planning for different plants.

In process state transitions such as startup and shutdown stages, the alarm flood increases and it generates critical conditions in which the operator does not respond efficiently; moreover, it is commonly reported that 70% of plant incidents occur at startup or shutdown stages [6]. Due to this alarm flood, dynamic alarm management is required. Currently, many fault detection and diagnosis techniques for multimode processes have been proposed; however, these techniques cannot indicate fundamental faults in the basic alarm system [125]. On the other hand, the technical report ”Advance Alarm System Requirements” EPRI (The Electric Power Research Institute) suggests both cause - consequence and event-based processing. Today, it is very easy to set alarms on modern electronic control systems, and operators are inundated with "alarms" that actually hinder the performance of their tasks [93]. In industrial environments, it is common for plant operators to perform their duties silencing process alarms. This situation arises because these alarms become noise rather than an indicator of abnormal situations. Nonetheless, the plant alarms should be administered according to: 1) a philosophy that includes the purpose of the alarm system; 2) procedures associated with the alarm system and other plant procedures; 3) methods for prioritization; 4) alarm classes; 5) roles and operator responsibilities with respect to alarms; 6) principles

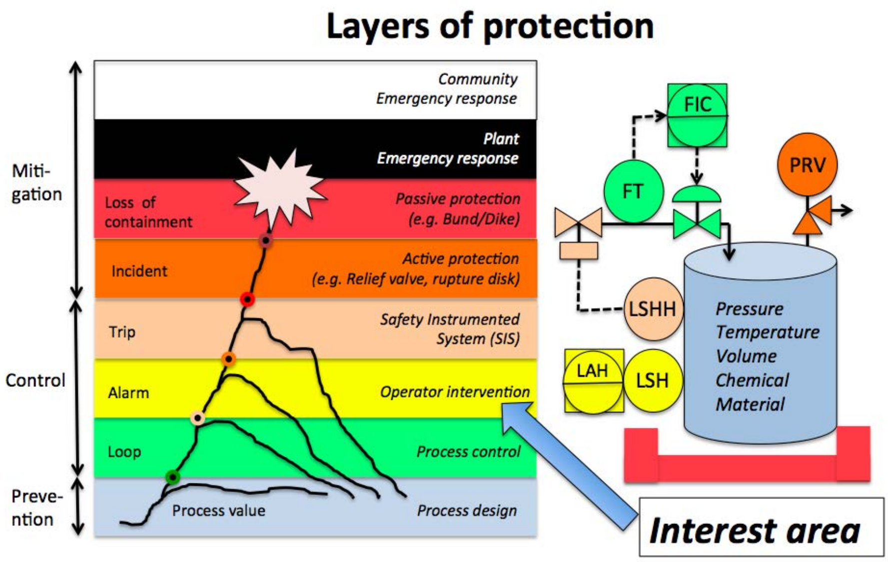


Figure 1.1: Safety layers of protection

of the alarms; 7) documentation required for each alarm; 8) training; 9) rates of key system performance; 10) change management, and 11) preservation of history of alarms (ISA 18.2).

In this thesis we propose to address the problem of alarm management by developing reliable tools that support the analysis of event streams to recognize activities that can generate normal or abnormal situations in complex flows. The challenge is then to fit the formal recognition of behaviors in the context of Complex Event Processing. The dynamics of a process can be represented by an approach that depicts the process behavior using the events that occur during the process evolutions. In this context, the *chronicle approach* [31] has been applied in many applications of situation recognition and often with a diagnosis objective. Chronicles are temporal pattern supported by a set of observable events and a set of temporal constraints between pairs of events. One of the main difficulties of situation recognition based on chronicles is to obtain automatically a base of chronicles that represents each situation of interest. Our proposal is then to use a chronicle recognition approach to analyse the behavior of the process and to use learning techniques for the chronicles design. Diagnosis by situation recognition (*chronicle based diagnosis*) in startup and shutdown stages of chemical /petrochemical processes as a support to human operators is the principal goal of this thesis and resumed in Fig. 1.2.

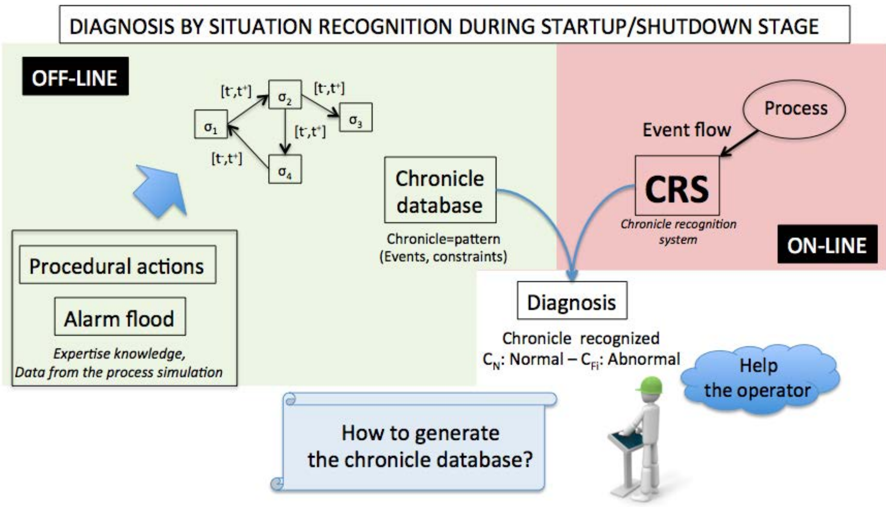


Figure 1.2: Motivation: Diagnosis by situation recognition

A formal methodology called **Chronicle Based Alarm Management** (CBAM) to generate offline the chronicle database using the alarm flood, procedural actions and expert knowledge is proposed. As the efficiency of alarm management approaches depends on the operator expertise and process knowledge, our final objective is to develop a diagnosis approach as a decision tool for operators. For this, we propose to enhance the chronicle learning stage by incorporating expert knowledge. In this thesis the chronicle learning algorithm proposed in [96] has been extended to incorporate expert knowledge in the form of temporal restrictions, as well as additional information that allows us to limit the conservatism of chronicles.

The global approach $CBAM$ provides a dynamic alarm management system for the transition stages of chemical processes. To consider both continuous and discrete features of chemical plants, CBAM is fed by the hybrid system framework. The Chronicle Based Alarm Management approach is then at the boarder between the alarm management area, the fault diagnosis research field and hybrid models field (see Fig. 1.3). In this approach the simultaneous occurrence of events is not considered.

## 1.2 Alarm management

Alarm management is an important aspect in the safety of the industrial processes. In years past (60's, 70's) the integration of a new alarm on the systems had a high cost and required a careful study and analysis before deploying. Each alarm had to be wired

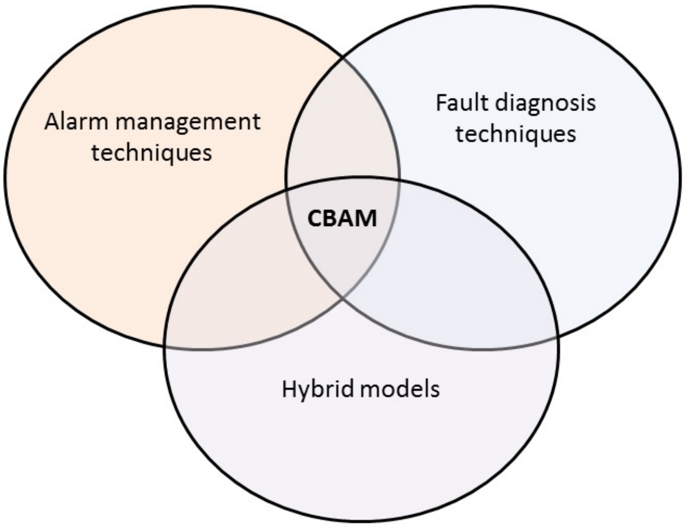


Figure 1.3: Chronicle Based Alarm Management CBAM

given the limited space on the panels of the control room. Today, advances in hardware and software have made possible the implementation of alarms at a minimum cost, without limits of space and with less review. Therefore, in many cases unnecessary alarms arise. Due this, an important advance has been the appearance of alarm systems, in which alarms are installed and configured considering the amount of existing signals (analog and discrete) and the rate of alarms that an operator can respond efficiently. Alarm systems can induce many alarms that cannot be evaluated by the operator which is a serious threat to the safety of the process. Therefore, now, the question is: *Which alarms can be ignored without compromising the integrity of the process?* This at the extreme can lead to sub-alarm systems, which is as bad as having a system over-alarmed [76]. Alarm management systems must deal with two main difficulties:

- A very high rate of alarms,
- A lack of criteria for assigning the priority of an alarm.

Alarm rate indicates the load that produces the alarm system to the operator. If the operator is supposed to respond to all alarms, the system must not produce more alarms that the operator can respond effectively. The most important factors that affect the rate of alarms are:

- The number of alarms settled,

- The deadband analog alarms (pressure, temperature, flow, level, etc.),
- The analog alarm limits,
- Alarms packages equipment (compressors, furnaces, etc.).

The alarm priority determines the order in which the operator must respond to the alarm, i.e. it determines the relative importance of alarms. Frequently it can be found that all alarms have the same priority, or there are a large percentage for one priority and a few for other priorities. It is important that alarms are prioritized correctly because in a scenario in which the operator receives a sequence of alarms in a short period, the priority is the only factor that the operator owns to determine to which alarm he has to respond in priority [95].

Alarm management is a process by which the alarms are designed, monitored and managed to ensure more reliable and secure operations. The first mistake is to assume that the alarm management has to do with reducing alarms. The aim of an alarm management system is to improve the quality process acting on the rate of alarms during normal operation, on the rate of alarms during abnormal situations, on the priority of alarms and on problems related to maintenance and Operation / Control. The motivation of alarm management is based on improving the work environment of the operator (ergonomics) preventing overload of the same, to avoid unexpected stops, make operation safer thereby achieving improved plant reliability.

On the other hand, many failings by operators have been recorded as incidents that have been the major contributing cause of major accidents. For controlling and mitigate these events, it is necessary to provide clear, concise and accurate operating procedures. Operating procedures must declare the instructions for the correct operation of the process plant regarding aspects such as the *Control of Substances Hazardous to Health* (COSHH), manual handling, Personal Protective Equipment (PPE) regulations, quality, the *Hazard and Operability study* (Hazop), and the *Safety Health and Environment* (SHE) requirements. A *Standard Operating Procedure* (SOP), is a set of instructions step-by-step structured to help the operators carry out routine operations, and each company or organization defines theirs SOP as they believe that is more convenient. The principal objective of the SOPs is to achieve efficiency, quality output and uniformity of performance, reducing delays and failures. Therefore, the standard operating procedures should depict a definition of the best practice that can do to at any momment.

Summarizing, the fundamental purpose of an *alarm* is 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".* This means that an alarm is more than a message or an event; an alarm indicates a condition requiring the operator's attention towards plant conditions requiring timely assessment or action. In addition, alarm management corresponds to determining, documenting, designing, operating, monitoring, and maintaining alarm systems and it has recently focused the attention of many researchers in themes such as:

- Alarm history visualization and analysis,
- Process data-based alarm system analysis and rationalization,
- Plant connectivity and process variable causality analysis (causal methods).

The proposal in this thesis seeks to exploit the causal relationships between process variables and procedure actions issued of Standard Operating Procedures. Additionally our objective is to take the temporal dimension of the alarm management into account. More precisely we want to exploit the temporal information of the alarm sequences (i.e the time between alarms occurrence). In this context the choice we have made of a chronicle based approach for situation recognition is well suited as time is an intrinsic feature of chronicles.

## 1.3 Fault diagnosis techniques

The knowledge that we can acquire about the behavior of a physical system is based primarily on the acquisition and valuation of two types of information: *Quantitative*, which is acquired through various measuring instruments variables that characterize the system operation. *Qualitative*, which is acquired by humans through the sensory organs and processed by the brain, usually provided in the form of linguistic information [91],[111]. In fault detection process we cannot neglect any kind of information because both are essential for the generation of fault indicators. We present next a summary of fault detection techniques based on two types of approach: **Data - driven techniques** and **Model - based techniques**.

### 1.3.1 Data - driven techniques

In diagnosis theory, there exist promising methods of fault diagnosis in technical systems described by linear and nonlinear models; methods noted as "model-free" or "data-driven" methods [8], [29]. This type of techniques consider continuous measurements, and a set of measurements of the process using sensors is represented as a pattern [92],[124]. A list of some data-driven techniques is presented below.

(a) Quantitative techniques

i. Statistical: These methods use analysis as the Principal Component Analysis (PCA) which is a statistical procedure that convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (or sometimes principal modes of variation) [75],[115].

ii. Neuronal networks (Multivariable models built from a set of input/output data), Neural networks are a computational model based on a large set of simple neuronal units (artificial neurons), approximately similar to the behavior observed in axons of neurons in biological brains [112].

(b) Qualitative techniques

i. Expert systems (Rule-based feature extraction): These systems represent the knowledge of the experts through determining rules and patterns which can be implemented with Neuronal networks -Pattern classification approaches [84], Fuzzy logic [46],[26], Genetic algorithm [103] etc.

ii. Qualitative trend analysis (QTA) (Abstraction of trend information): These type of techniques are an useful tool for process data analysis, process monitoring, fault diagnosis, and data mining. Techniques which can analyze the data applying triangulation [20], finite difference method [54], syntactic pattern recognition approach [83], Gaussian filter [123], etc.

### 1.3.2 Model - based techniques

Different approaches for fault detection use mathematical and graph models. The mission corresponds to the detection of faults in the process, faults in the actuators and sensors by using the dependencies between different measurable signals. These dependencies are expressed by mathematical process models. The basic structure of a model-based fault detection is based on measured input signals and output signals. The detection methods generate residuals, parameter estimates or state estimates, which are called features. By comparison with the normal features, changes of features are detected, leading to analytical symptoms. A list of the most popular model-based techniques is presented below.

(a) Quantitative techniques

i. Residual generation methods: These methods include the generation of residual signals analyzing measurements of input/output elements. These methods use analytical redundancy [4],[21],[45], residuals and parameterization of residual generators [44], Fault detection filter (Observer) [18], etc.

ii. Another method based on quantitative techniques is the Kalman filter [119] which uses data from the process predicting the next values in the variables to detect failures.

(b) Qualitative techniques

i. Causal models: These models represent a description of the process. In a graphical form, the following techniques can indicate the relationships between components, variables and procedural actions: Diagraph (Graph with directed arcs between the nodes) [67], Bond graphs [87], SDG - Signed direct graph [121], ESDG - Extended SDG [63]. Other techniques can involve information from a risk analysis, time and the date of occurrences of the events: PCEG - Possible cause and effect graph models [117],[118], HDG - HAZOP-digraph models [107], Causal Graphs [17], Chronicles [31],[96],[110], Fault trees [102].

ii. Qualitative physics: These techniques can predict and explain the behavior of mechanisms in qualitative terms. For example, we can obtain the qualitative behavior from the ordinary differential equations (ODEs) [86] [57] or simulate a process in a qualitative form (Qualitative simulation -QSIM) [57].

iii. Abstraction hierarchies: The complex processes can be divided hierarchically for its analysis. This division may be structural [82], from multilevel flow Models (MFM) [62], from functional areas [40], etc.

In this thesis as said previously, we consider a chronicle based diagnosis approach. This diagnosis approach is a data driven approach as the chronicles are designed through data (in our case event sequences) by a learning technique. It can also be considered as a qualitative model based approach as chronicles can be viewed as observable abstractions of the process behaviors.

## 1.4 Hybrid models and causal graph

The **Chronicle Based Alarm Management** (CBAM) methodology proposed in this thesis uses a formal framework based on hybrid systems to formally model the chemical/petrochemical processes that have both continuous dynamics and discrete dynamics. In addition, to capture the causal relationship between the continuous variables the hybrid modeling of the process integrates causal graphs. This hybrid framework will be described in details in the chapter 3.

## 1.5 Objectives of the thesis

The general objective of the thesis is to design and to develop a new methodology for alarm management in startup and shutdown stages.

More precisely we aim to:

- design an alarm management method based on a diagnosis process during startup and shutdown stages
- structure the diagnosis based on situation recognition
- include normal and abnormal situations captured by chronicles
- generate the chronicle database automatically using a learning approach
- apply the methodology in the petrochemical sector

## 1.6 Contributions

The principal contribution of the thesis is a new proposal for alarm management based on a diagnosis process. An analysis of alarm management in startups and shutdowns for oil refining processes was presented in an international conference [108], and the Chronicle Based Alarm Management methodology has been exposed in international workshops and conferences [109],[110]. This methodology includes the generation of event sequences used for learning chronicles in a formal framework. Additionally, to limit the conservatism and restrictiveness in the chronicles the expertise knowledge in form of temporal restrictions has been included extending the Heuristic Chronicle Discovery Algorithm Modified ($HCDAM$) (see Fig. 1.4). These results gave rise to the following publications:

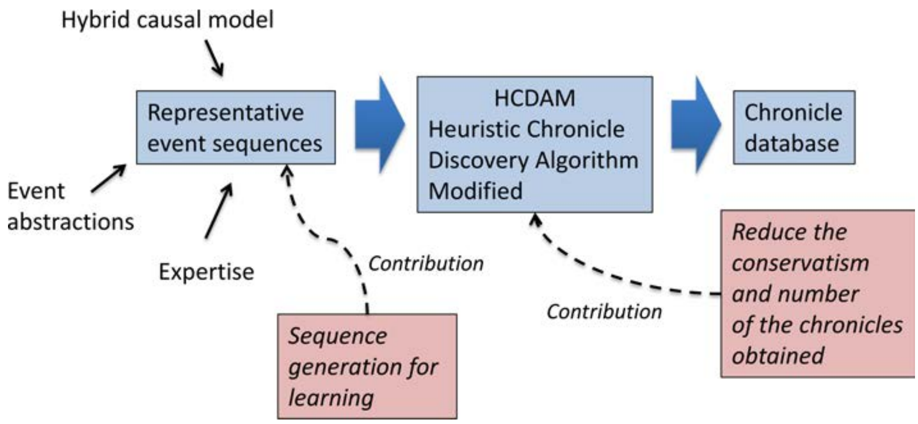


Figure 1.4: Chronicle learning proposal - Contributions

- Vásquez, J., Subias, A., Travé-Massuyès, L.,and Jimenez F. (2017). Alarm management via pattern recognition. Engineering Applications of Artificial Intelligence, Volume 65, October 2017, Pages 506-516.
- Vásquez, J., Travé-Massuyès, L., Subias, A., Jimenez, F. (2017). Enhanced chronicle learning for process supervision. IFAC 2017 World Congress, Toulouse France, pp. 5191 to 5196.
- Vásquez, J., Travé-Massuyès, L., Subias, A., Jimenez, F., and Agudelo, C. (2016). Alarm management based on diagnosis. 4th IFAC International Conference on Intelligent Control and Automation Sciences (ICONS 2016), Reims, France.
- Vasquez, J., Travé-Massuyès, L., Subias, A., Jimenez, F., and Agudelo, C. (2015). Chronicle based alarm management in startup and shutdown stages. International Workshop on Principles of Diagnosis (DX-2015), Paris, France.

## 1.7 Thesis structure

The thesis manuscript is organized in eight chapters with the following contents:

**Chapter 1**

It is dedicated to the presentation of the motivations and objectives of the thesis. Aspects such as safety layers of protection and the proposal of diagnosis by situation recognition are presented in this chapter. Moreover, it presents a global overview positioning our work at the border between alarm management, hybrid modeling and diagnosis techniques.

**Chapter 2**

In this chapter an analysis of fault diagnosis in industrial processes is presented. The concept of SUPER ALARM and the new layer of protection on process safety are explained. In addition, the following aspects are described: Reliability & Risk management, hazard analysis and finally control & safety systems (SIS).

**Chapter 3**

This chapter considers hybrid models and causal graphs which are used in the proposed methodology as tool that help human operators understand the behavior of the physical components of the process and to verify the representative event sequences that are generated in the CBAM methodology. This chapter gives the principles of causal modeling and the generation of causal graphs including an example with an illustrative application applying the five steps for the generation of a causal graph.

**Chapter 4**

In chapter 4 theoretical aspects on chronicles are given. The chronicle and temporal restriction definitions are provided. The concepts of chronicle recognition and diagnosis based on chronicles are introduced. This chapter concludes with a small example that shows how a chronicle represents a temporal pattern of an industrial process.

**Chapter 5**

This chapter exposes the chronicle learning process and the Heuristic Chronicle Discovery Algorithm Modified $HCDAM$ [96]. Interactive chronicle discovery is presented with related work, in particular, Dousson & Vudoung [34] and Cram [25] algorithms. The organization and the three phases of $HCDAM$ are presented. The chapter concludes with an example in which the algorithm $HCDAM$ is applied to an industrial process. The same example uses also the extended version of $HCDAM$.

**Chapter 6**

The new methodology "Chronicle Based Alarm Management" (CBAM) is presented in this chapter. The Hybrid Causal Model and the Qualitative abstraction of continuous behavior are described. Additionally the three steps of CBAM are described: "event type identification", "event sequences generation" and "construction of the chronicle data base".

**Chapter 7**

The Cartagena Refinery is described and two case studies related to the petrochemical sector are considered in this chapter: a Hydrostatic Tank Gauging System and a Vacuum Oven system. Each step of the Chronicle Based Alarm Management methodology is illustrated on these real cases of application.

**Chapter 8**

The contributions of this thesis are presented and future work directions are pointed out in relation with the thesis.

# Chapter 2

# Diagnosis in industrial processes

## 2.1 Introduction

Diagnosis in industrial processes corresponds to the procedures, activities, and tools that help operators to recognize the real plant situation, especially at transitional stages in which increases the risk of accidents. In Figure 2.1 is presented the process safety relationships. The layers of protection (Loop, Alarm, and Trip) involve the components of supervision scheme in which the first level includes the instrumentation and actuators of the system, also the Safety Instrumented System (SIS). The next level contains the acquisition and control equipment followed by the supervision stage, in which the tools of diagnosis are implemented. To determine the events and signals of a procedure it is necessary to analyze and consider the initial conditions of the process and to identify possible failure modes. Hence, a complex system requires a division into subsystems for a reliable analysis. The goal of the technology used maintains the process variables on their limits of operation.

In terms of process safety, the principal characteristics of a good protective barrier are specificity, independence, reliability, and audit. Specificity: Barrier capable of detecting and preventing or mitigating consequences of a potentially dangerous specific event (e.g. explosion). Independence: A barrier is independent of all other layers which are associated with the potentially dangerous event, when there is no potential for common cause failures and the protection layer is independent of the initiating event. Reliability: The protection provided by the barrier reduces the risk identified by a specific and known quantity determined by its probability of failure. Audit: A barrier must be designed to allow inspections and periodic and regular testing of the protection function.

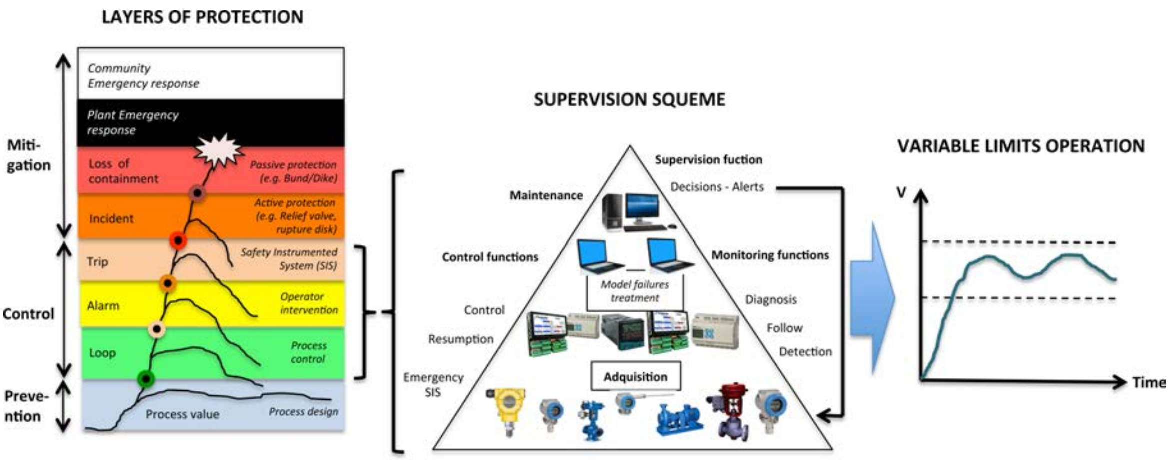


Figure 2.1: Process safety relationships

This thesis proposes an extension of protection barrier "Alarm", see Figure 2.2. The extension in this layer comes from a diagnosis process and the concept of SUPER ALARM corresponds to a new alert to the operators resulted from a diagnosis procedure representing a "Superior" alarm.

Consequently, in automatic control systems, the supervision functions serve to indicate undesirable or not permitted processes states and take appropriate actions that maintain performance and avoid damage or harm states. From supervision we can discriminate the following functions:

- **Monitoring**: The measurable variables are checked respect their tolerances and alarms are generated to alert the operators.

- **Supervision**: Supervision with fault diagnosis this action is developed from the analysis of the measurable variables detecting the symptoms of a possible failure [2], [60].

- **Automatic protection**: Actions for counteract the possible damages. A system is said to be diagnosable if whatever the behavior of the system, we will be able to determine without ambiguity a unique diagnosis.

The diagnosability of a system is generally computed from its model [5], and in industrial applications using model-based diagnosis, such a model is generally present and does not need to be built from scratch. The fault diagnosis in general consists in the following three important aspects: Fault detection: it consists in discovering the existence of faults in the most useful units in the process, Fault isolation: it is

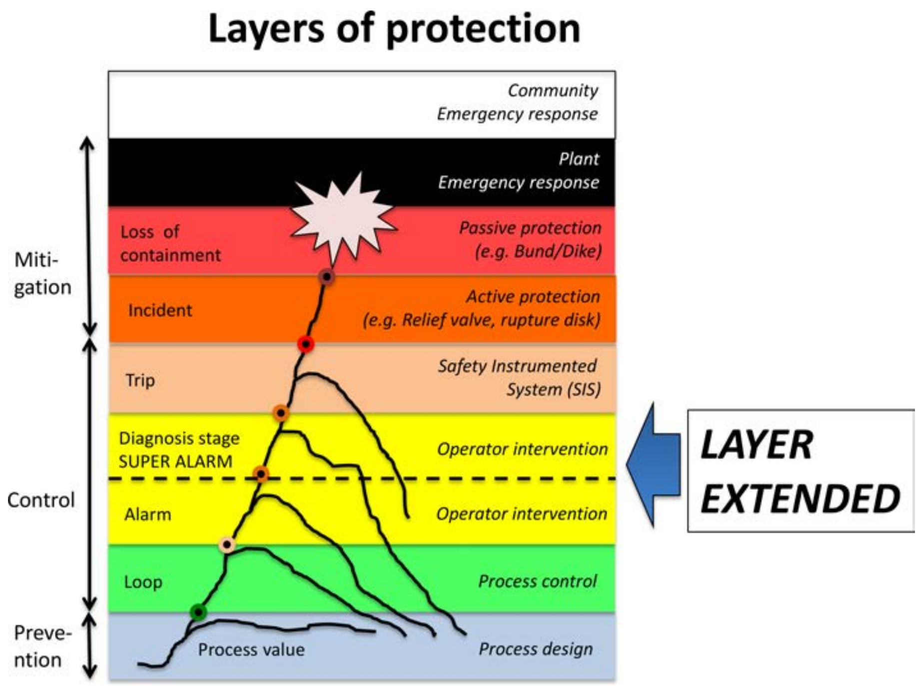


Figure 2.2: SUPER ALARM layer of protection

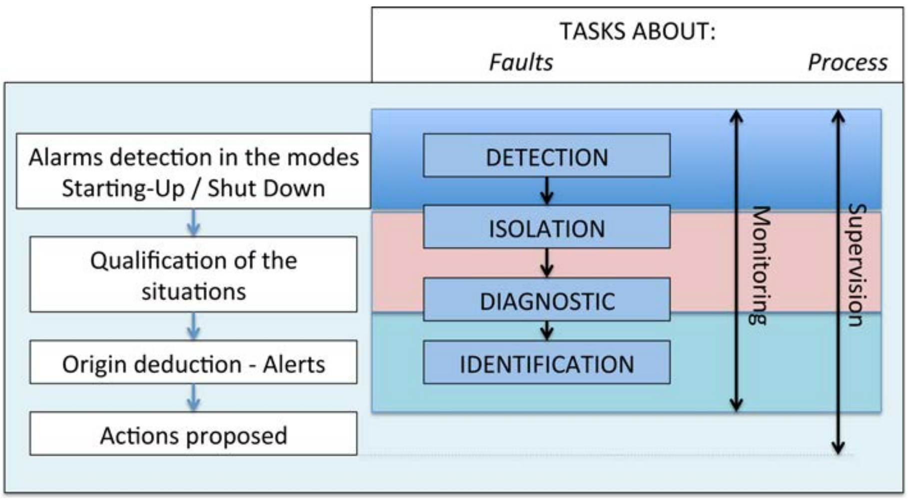


Figure 2.3: Reduction of alerts to the operators

referred to localize (classified) the different faults, and Fault analysis or identification: it consists in determining the type, degree and origin of the fault [28].

Safety requirements and the increasing efficiency in monitoring, control, and management of complex systems motivates great interest and efforts devoted to the development of fault detection and isolation techniques. Many popular approaches are available for identifying faults. Among them, methods based on signals are widely used and try to extract useful information from the analysis of specific signals through a comprehensive and rigorous analysis of the main statistical methods used to detect changes [64]. The model-based methods, like parity or space-based approaches observers [77], used a mathematical model of the plant to explore the implicit analytical redundancy relations model to monitor inconsistencies between the model and data measured. However, these methods suggest a big demanding of computational load. Other popular methods as those based on fault trees [113] or causal graphs and propagation [120] based on a qualitative model of the plant. Other approaches have been developed by expert systems based on artificial intelligence techniques [89]. On the other hand, hierarchical clustering methods were used to carry out pattern matching correlation [19] in which some frequent patterns of multiple alarm correlation may be discovered to have the ability to reflect the sequence of normal operation. Any change in the pattern may indicate abnormal alterations sensor degradation or malfunction. Meanwhile, the Professor Ali Zolghadri expresses that currently there is a valley of the death between the diagnosis theory and the industrial process applications, the work [126] presented several examples of model and signal based fault detection in aircraft Electrical Flight Control System (EFCS).

Expert systems have also been widely used in industrial monitoring systems for the diagnosis of impairment, in particular, by IFP (Institut Francaise du Petrole) in ALEXIP software [15] and by France Telecom for monitoring TRANSPAC network [104]. These systems set alarm linked directly to a typical situation which must be identified in the form of rules. Once of these rules are acquired in the knowledge base, the task is to analyze the flow of alarms and perform validation rules. In these fields, there are recent works of the Colombian Petroleum Institute (intelligent alarm management) [46] and the French Institute of Petroleum presents a diagnostic module developed and tested offline in a pilot plant [17], works which integrate techniques such as causal graphs and fuzzy logic for performing the fault diagnosis. Another case [90] used the techniques of fuzzy clustering and neuronal networks in the identification and estimation of functional states for a power transmission line and in the monitoring of a boiler system. Moreover, [106] used expert knowledge-guided feature selection for data-based industrial process monitoring.

Hybrid intelligent systems are an important future direction to develop in diagnosis systems. The disadvantage of diagnosis systems based on a single method is not to be versatile enough to control systems on a large scale, so these systems need to integrate various techniques to make an efficient diagnosis. The integration of diagnosis methods combines many types of techniques of fault detection; in particular, the complementary combination of quantitative and qualitative models can greatly reduce the false alarm rate [67]. All these methods have their advantages and their specific fields of application, which can be implemented efficiently on a general approach for fault detection in compliance with the following characteristics:

- *The modularity and flexibility*: The model must describe any element of the process and environmental assessment. This model should be implemented in a library for use it in the construction of diagnosis algorithm.
- *Hierarchical design*: According to a top-down approach the model of fault detection could be created from diagnosis procedures for different extraction levels.
- *Data fusion*: The diagnosis system should be able to extract information from many different sources: local signal analysis, intelligent instrumentation, empirical knowledge, logical conditions.
- *Temporal Analysis*: The diagnosis system can provide useful information to complete the diagnosis analyzing the dynamic events detection in abnormal behavior of the system.
- *Compatibility with industry standards*: The failure mode effects (FMEA) is a method for determining the possible failure occurrences in the industry. Then, the diagnosis system for the industry must apply this method in the determination of the failures to detect.

Concluding, the goal to obtain an efficient and reliable methodology in a safe process must include the following two aspects: Reliability & risk management and Control & safety systems. Consequently, the Chronicle Based Alarm Management methodology (CBAM) can integrate these aspects as follows:

- Reliability & risk management: CBAM can use analysis such as HAZOP, fault tree among others, for the determination of the most important scenarios and events in the process.

- Control & safety systems: The algorithm and tool for evaluating the event sequences must be integrated into the control system of the process and in some cases must interact with the safety system. For example, to check that the safety system had been activated when there is a failure.

## 2.2 Reliability and risk management

Reliability may be assumed as the ability of a system or component to perform its required functions under stated conditions for a specified time; meanwhile, safety is the state of being "safe". In other words, it is the condition of being protected from harm or other non-desirable outcomes. Consequently, if a system is reliable, supposedly it is safe also. For example, a new pistol is reliable, but it is safe? so if exist a risk of hurt or damage, the use of a pistol needs strict conditions for its management. Likewise, happen in the industrial processes, if there exists the risk of that something wrong occurs, this situation or risk needs a management. It is common confuse safety with security; security is the degree of resistance to, or protection from, harm. It applies to any vulnerable and/or valuable asset, such as a person, dwelling, community, item, nation, or organization. In short, safety is the minimization of the risk of occurrence of accidents and serious incidents in the system, equipment (prevention). Security, on the other hand, is responsible for the control of incidents of infrastructure, property, and persons against acts of unlawful interference (protection).

The relationship between safety and reliability had been enhanced since the Industrial Revolution. The use of new sources of power, using water or steam, nuclear plants, and petrochemical industry not only gave great potential for the rapid development of manufacturing technology, likewise provided a terrible potential for death and injury when processes went wrong. Due to the demand for new machinery and the use of chemical elements such as oil and gas, the number of fatal accidents has increased. Although design new machines make possible the growing scientific knowledge, designers still lean strongly on past experience [24]. In addition, based on this experience, some risks can be accepted.

Risk management is a technique widely applied in organizations to increase its safety and reliability minimizing losses. This technique involves the identification, evaluation, and control of risks; moreover, risk evaluation includes the measurement and assessment of the risk. The risk is the potential of gaining or losing something important and valuable. Things such as physical health, social status, emotional well-being or financial wealth. In an ideal world, a good evaluation and assessment of the risks permits an effective error reduction. In practice, decisions on the acceptability of risk depend upon many factors; these include social, economic, political and legislative concerns. Referencing strategies, risk control strategies may be classified into four main areas:

1. Risk avoidance: Risk avoidance involves a conscious decision on the part of the organization to avoid a particular risk by discontinuing the operation that is producing the risk.

2. Risk retention: Risk retention may occur with or without knowledge:

    - With knowledge: It is a deliberate decision made to retain the risk, maybe by self-financing.
    - Without knowledge: Occurs when risks have not been identified

3. Risk transfer: Risk transfer is the conscious transfer of risk to another organization, usually via insurance.

4. Risk reduction: Risk reduction is the management of systems to reduce risks.

The engineers and technologists who design complex and high-risk systems, are those who develop the management procedures and, above all, those who manage and control the human factors. Concluding, a complete and effective diagnostic tool can help to reduce the risk of accident occurrences.

In Fig. 2.4 is represented a framework of the risk assessment process, which contains three levels of activities: Risk analysis, Risk assessment, and Risk management. Risk analysis is a technical process that initiates defining the system continuing with the hazard identification, frequency analysis, consequence modeling, and concluding with the risk calculation. The risk assessment level presents the actions of risk acceptability, risk reduction decisions, and cost-benefit judgments. Risk management consists in the action to monitor, test, and control risk levels, and it is part of the safety management plan of the organization. Quantitative Risk Assessment (QRA) is the most sophisticated technology to calculate the risks of incidents, estimate the uncertainties of the calculated risk levels, and provide metrics for cost-effective risk minimization. Moreover, to quantifying the effects of data uncertainty, QRA uses models to estimate conditional probabilities of failure for components or layers of protection that are not mutually independent. For risk assessment related to reactive chemicals, statistical data from incidents are often insufficient and are related to specific circumstances. Consequently, it is important to develop an effective implementation of QRA methods, such as statistical inference, requiring of significant cost, time,

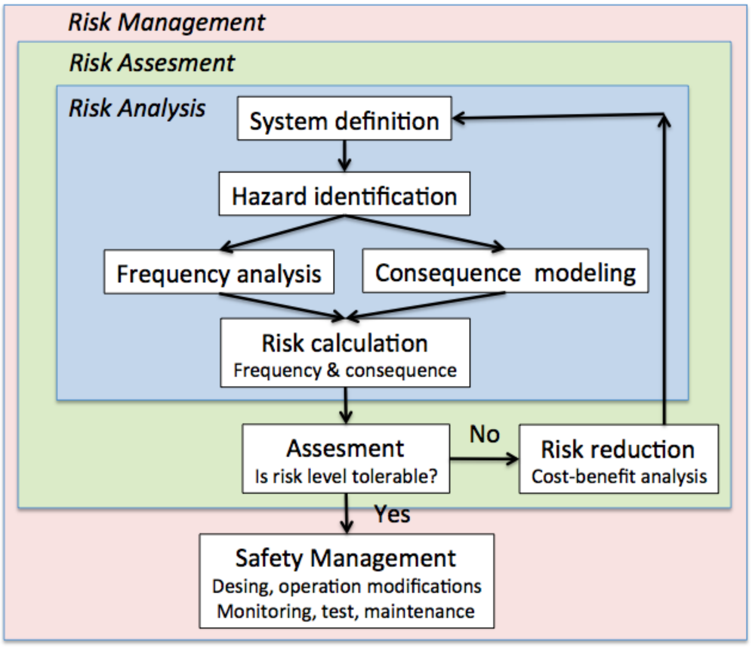


Figure 2.4: Flow diagram for a risk assessment process.

and experience. Hence, less costly qualitative and semi-quantitative risk assessment techniques can be used effectively to identify where a more quantitative analysis of the most critical components of a chemical system may be needed [97],[116].

### 2.2.1 Intrinsic safety

There are two types of processes, the processes intrinsically safe, and those for which the safety has to be adapted. An intrinsically safe process is one in which safe actions are involved in the nature of the process; a process which causes no danger, or negligible danger, under all foreseeable circumstances. The term inherently safe is often preferred to intrinsically safe, to avoid confusion with the use of this term "intrinsically safe" or "explosion proof" as is applied to electrical equipment. “Explosion proof“ is the classification for a sensor/transmitter and means that the housing has been engineered and constructed to contain a flash or explosion. These housings are usually made of cast aluminum or stainless steel and are of sufficient mass and strength to safely contain an explosion when flammable gases or vapors penetrate the housing and the internal electronics or wiring cause an ignition. This design must prevent any surface temperatures that could exceed the ignition temperature of the gases or vapors covered by its Group rating. If the sensing element is a high-temperature device (e.g.

Catalytic bead or "pellistor"), it may be protected by a flame arrestor to prevent the propagation of high-temperature gases to the ambient atmosphere. On the other hand, an "intrinsically safe" classification and design mean that an electronic circuit and its wiring will not cause any sparking or arcing and cannot store sufficient energy to ignite a flammable gas or vapor, and cannot produce a surface temperature high enough to cause ignition. If an element is not explosion proof, nor does it need to be. For permanent installations, such an installation may include "intrinsically safe barriers" that are located outside the hazardous location and limit the amount of energy available to the device located in the hazardous area.

Plainly, the designer should always select a process that is inherently safe whenever it is practical, and economic, to do so. Nonetheless, most chemical manufacturing processes are a greater or lesser extent, inherently unsafe, and many times dangerous situations can develop if the process conditions deviate from the design values. The safe operation of such processes depends on the design and provision of engineered safety devices, and on good operating practices, to prevent a dangerous situation developing and to minimize the consequences of any incident that arises from the failure of these safeguards. The term "engineered safety" covers the provision in the design of control systems, alarms, trips, pressure-relief devices, automatic shut-down systems, duplication of key equipment services; and fire-fighting equipment, sprinkler systems and blast walls, to contain any fire or explosion. Consequently, to have a lot of alarms configured in the process not ensures to increase the reliability, especially in the startup and shutdown stages. In this moment is when a diagnosis methodology or a fault diagnostic approach can increase the reliability of the processes. But now, how guarantee that the diagnostic methodology works efficiently? For example, an operator can respond correctly when less than 10 alarms occur in 10 minutes. But, a new supervisory tool can reduce the number of alarms to maximum 2 in 10 minutes; increasing the reliability of the alarm system.

Note that there is an issue if the supervision tool can not determine if the alarm $a$ has occurred two times and the alarm $b$ has occurred one time or on the contrary, the alarm $b$ occurred two times and the alarm $a$ occurred one time, and also neither detect if both occurred at the same time. It is a problem that needs to be solved to guarantee the reliability of the alarm system. The premise in a safe process is: if something can fail, it will fail!

### 2.2.2 Hazard analysis

A hazard analysis is a systematic method for identifying, evaluating and controlling the hazards of a system. It is part of risk management, which consists of five phases:

1. Definition
2. Hazard identification
3. Hazard and risk assessment
4. Proposed hazard resolution
5. Follow-up to proposed activities

The analysis of hazards corresponds to the first two phases of this process; however, from the information obtained in these phases, we obtain the necessary information to carry out its evaluation and its proposal of resolution to an acceptable level. A brief description of each of the steps is given below. The first phase begins with the definition of analysis criteria, knowledge about the functional and physical characteristics of the evaluated system, the teams involved, and knowledge of the process. It should be noted that many engineers fail in this first phase since they assume that they fully know the operation of the process and do not take the necessary time for its definition. This phase should include not only the definition of how the system works, but also include aspects of operation and external variables that affect it as in the case of other processes, plant personnel, and technologies that make the system operational. Once the system is defined, an identification of hazards and root causes must be made. We must go step by step postulating possible sources of danger for the system evaluated under normal (and abnormal) operating conditions. It should be noted that this phase must be performed in all stages of the life cycle, as there may be very particular hazards in any of its stages. The third step is to assess the hazards identified and their effects. The majority of methodologies in this phase, implement a severity classification to evaluate the possible consequences of each the dangers identified in order to carry out a process of comparison and prioritization. However, for this phase, it is not enough to consider the scenario with the worst consequences, but one must understand the probability or the possibility that the dangerous event actually occurs. Hazard analysis methodologies employ qualitative probability classification systems, which together with the severity of the accident can be assumed as a risk. From this classification, it is determined which risks are acceptable or not, for the purpose of implementing or proposing corrective measures, elimination or control of hazards. To carry out the fourth phase there are several approaches; for example, one of the most used is the model proposed by NASA (National Aeronautics and Space Administration, 1993) from a hierarchical reduction of danger. This methodology is applicable to all types of industry and part of an out-of-danger design. For instance, three elements (flammable liquids/gases) must be present for a fire to occur: a fuel, an oxidizer (e.g. oxygen) and a source of ignition. An out-of-danger design would be to eliminate possible sources of ignition close to the fuel and oxidant mixture, thus avoiding the danger of a fire. When is not possibly develop a design out of danger either because it is very expensive or because it is intrinsic to the process, you should consider the use of safety equipment. An example is a control valve of an open-failure cooling service in an equipment with a highly exothermic reaction. This valve prevents a dangerous increase in temperature that could cause even an explosion of the equipment. Another clear example is the relief valves in a pressurized vessel, which are used in order to avoid a dangerous increase in pressure. If a design can not be developed out of danger or sufficiently controlled, the next step is to warn operators of imminent danger. Examples of these elements of caution and warning are the smoke detectors. Upon reaching a certain level an alarm is activated and the area must be evacuated immediately. Finally, phase five corresponds to the follow-up to the proposed activities. It is important to monitor the effectiveness of hazard controls and review those that are unexpected or new. A periodic review must be performed whether hazards remain adequately identified and whether control mechanisms continue to function. This phase is fundamental in processes that undergo modifications, expansions, or a reconfiguration of their operating conditions [3]. Hazard assessment techniques can be classified as scenario-based or non-scenario-based. Among the first are procedures such as Hazop, What if?, FMA, fault tree analysis, event tree analysis and cause-consequence analysis (or bowtie). For the second, there are procedures such as Preliminary Hazard Analysis (PHA), safety review, relative ranking, and checklists. Some of these analyses are briefly explained below.

Hazop provides a systematic way of identifying hazards using a number of guide works as an aid. Just how the words are interpreted depends on the circumstances, but for example 'None of' could lead to a consideration of the possibility of no liquid flow in one case or no electrical current or no pressure in others. Other guide words and some applications are presented in Table 2.1.

In each case, the guide word is used to concentrate attention on to one particular fault (no liquid flow, for example). Possible causes of lack of flow are then examined and the effects of it are enumerated. The complete set of guide words is applied in this way

| Guide words | Description |
|---|---|
| More of | Liquid flow too high, temperature, pressure or electrical current too high. |
| Less of | Liquid flow too low, temperature, pressure or electrical current too low. |
| Part of | Chemical component missing, composition wrong. |
| More than | Impurities present, extra phase present (gas in liquid, for example). |

Table 2.1: Hazop table

to each component or process in turn. The design can then be modified to avoid the associated hazards and consequent operational problems. This technique is commonly used in the chemical industry and is particularly effective if applied by a mixed team providing expertise in design, instrumentation, commissioning and operation. The zonal analysis is a method used to examine possible cascade and common mode failures in aircraft. In this case, the aircraft is sub-divided into zones and for each zone, all actuators and other items of equipment within the zone are itemized. The mutual interactions within the zone are then examined as are the interactions with similar devices outside the zone. The interactions may be anything from electrical interference to leakage of hydraulic fluids or water, or undesirable mechanical interactions. Both normal and fault conditions are considered. As in the case of Hazop, the zonal analysis provides a systematic framework for the investigation of a particular type of failure.

The checklist is another tool which is used to check compliance with standard industrial procedures. Its main objective is to identify simple hazards and to ensure compliance with regulations / operational standards. Their evaluation is of qualitative type from the implementation of a checklist, which can be applied in all stages of production. DOW / MOND indexes consist of the identification and classification of risks through the use of performance indexes and the state of the internal processes of the plant or physical characteristic. These indexes are relative and are assigned subjectively in the form of bonus or penalty. For the first case, it includes the characteristics that allow mitigating the occurrence of an accident, whereas the penalties correspond with situations that can lead to the occurrence of an accident. The results obtained are of semi-quantitative type since they allow a classification and realize a subjective assignment of values for its calculation [88].

In Preliminary Hazard Analysis (PHA), the main objective is to identify hazards in the initial stages of industrial plant design. This analysis is directed to the management of hazardous substances associated with the raw material, finished product, and intermediate products. As a result, a list of possible hazards is obtained with their respective recommendations to prevent/mitigate them. It is a qualitative analysis that strengthens the design of a process plant to make an inherently secure system [88].

Analysis What if? is an analysis that focuses on the identification of unwanted consequences caused by a possible initiating event; analysis performed by a group of experts. It is a non-formal method but has shown to be useful in the definition of potential scenarios, identifying sequences of events leading to the occurrence of the fault. It can be used to examine possible deviations from the design, construction, operation, or modifications made to the plant. The result is of qualitative type, corresponds with a list of possible scenarios and the strategy to reduce its possible consequences [88]. Similarly, the "What if?"/Checklist is a technique whose purpose is to identify hazards by considering general types of incidents that may occur in a process or activity, qualitatively assessing the effects of the same, and determining when safeguards against of this potential hazard seem appropriate [16].

Fault Mode Analysis (FMA) is a methodology aimed at industrial equipment. This analysis consists in the evaluation of possible fault mechanisms of each equipment, defining possible fault scenarios and their respective consequences. Its result is qualitative in that it classifies each of the situations obtained according to their consequences [88].

Fault tree analysis (FTA) is a deductive technique that focuses on a particular incident or major cause of failure, and provides a method for determining the causes of such an event. The purpose of the FTA is to identify the combination of operational type failures, design material or process disturbances that could result in the incident. The strength of this analysis is to identify qualitatively the combinations of basic faults that could lead to the incident. What serves for the hazard analyst to take preventive measures in basic causes to reduce the probability of occurrence of the event ( an event in a safe process is referred to an accident).

Event Tree Analysis (ETA) is an event tree that shows graphically the possible outcomes following the success or failure of a protection system, given the occurrence of a specific initial cause. It is used to study the possible events that can happen in case there is a lost event. After these sequences of events are identified, the specific combinations of faults that lead to the incident are obtained [16].

Cause-consequence analysis / Bowel analysis: A cause-and-effect analysis is a mixture of fault and event tree, where its purpose is to identify the root causes and consequences of potential incidents. A particular case of this type of analysis is the one of Bowel, which correlates the existing security barriers, and evaluates their suitability. Subsequently, additional protection and recommendations are determined if necessary.

Causal events are presented to the left of the diagram and consequences to the right. An attribute of the Boundary method is that it is a visual form that clearly represents a risk.

The selection of the technique to be implemented for the evaluation of hazards follows a process that takes into account the type of information available, the response time of the analysis, the different stages of the life of the process facilities, among others. To carry out the above, there are methodologies that suggest the most appropriate technique to implement as reported by CCPS (Center for Chemical Process Safety) [16].

## 2.3 Control and safety systems

Process control systems had been developed to monitor data and control the variables and equipment on the industrial plant. Slight installations may use electric, hydraulic or pneumatic control systems; however, larger plants with up to 50,000 signals to and from the process require a dedicated distributed control system. The purpose of this system is to read values from a large number of sensors, run programs to monitor the process and control valves, motors, switches etc. to maintain under control the process. Values, alarms, reports and other information are also presented to the operator and command inputs accepted. Nowadays, a modern Process control system basically includes the following components [27]:

- Field instrumentation: sensors and switches that sense process conditions such as temperature, pressure or flow. These are connected over single and multiple pair electrical cables (hardwired) or communication bus systems called Fieldbus, Modbus, Profibus.

- Control devices, such as actuators for valves, electrical switchgear and drives or indicators can be also hardwired or connected to an industrial net of communications.

- Controllers that execute the control algorithms so that the desired actions can be taken. The controllers will also generate events and alarms based on changes of state and alarm conditions and prepare data for operators and information systems.

- Servers that perform the data processing are required for data presentation, historical archiving, alarm processing and engineering changes.

- Clients such as operator stations and engineering stations are provided for human interfaces. Which means customers can communicate to a human being.
- The communication can be exhibited in many different configurations, often including connections to remote facilities, remote operations support and similar.

The principal activity of the control system is to ensure a safe production, maintaining the components and element working efficiently within design constraints and alarm limits in the different variables. The control system is commonly determined in programs as a mix of logic and control elements such as AND, OR, NOT, PID, FUZZY.

From a Central Control Room (CCR), the system is operated with a combination of graphical process displays, alarm lists, reports and historical data curves. In this platform, new models of fault diagnosis can be implemented, the problem is to validate and to confirm the reliability of these theoretical models of diagnosis. Otherwise, with modern systems, the information in the desk screens is available to remote locations such as an onshore corporate operations support center. Field devices in most process areas must be protected to prevent them becoming ignition sources for potential hydrocarbon leaks. These equipment are explosive hazard classified e.g. as safe by pressurization (Ex.p), safe by explosive proof encapsulation (Ex.d) or intrinsically safe (Ex.i). All areas are mapped into explosive hazard zones from Zone 0 (inside vessels and pipes), Zone 1 (risk of hydrocarbons), Zone 2 (low risk of hydrocarbons) and Safe Area. Beyond the basic functionality, the control system can be used for more advanced control and optimization functions. Some examples of applications in a central control room are presented below [27]:

- Well control may include automatic startup and shutdown of a well and/or a set of wells. Applications can include optimization and stabilization of artificial lift such as pump off control and gas lift optimization.
- Flow assurance ensures the flow from wells, in pipelines and risers is stable and maximized under varying pressure, flow and temperatures. Unstable flow can result in slug formation, hydrates etc.
- Optimization of various processes to increase capacity or reduce energy costs.
- Pipeline management modeling, leak detection, and pig tracking.
- Support for remote operations, in which facility data is available to specialists located at a central support center.

- Support for remote operations where the entire facility is unmanned or without local operators full or part time, and is operated from a remote location.

Based on a study of the process in a cause and effect chart the Emergency shutdown actions are defined. The Hazop study may establish potential failures and how they should be handled. Consequently, we could execute possible shutdown actions when there are possible emergency scenarios. For instance, at an oil and gas facility, the primary response is to isolate and depressurize. The typical action would be to close the inlet and outlet sectioning valves and open the blowdown valve. This will isolate the malfunctioning unit and reduce pressure by flaring of the gas.

These actions are handled by the Emergency Shutdown system (ESD) and Process Shut Down system (PSD). System requirements are set by official laws and regulations and industry standards such as IEC 61508/61511 which set certification requirements for process safety systems and set criteria for the safety integrity level (SIL) of each loop. Events are classified on a scale, e.g. 1 to 5, followed by an Abandon Platform (APS) level. On this scale, APS as the highest level means a complete shutdown and evacuation of the facility. The next levels (ESD1, ESD2) define emergency complete shutdown. The lower levels (i.e. PSD 3, PSD 4, and PSD 5), represent single equipment or process section shutdowns. A split between APS/ESD and PSD is done in large installations because most signals are PSD and could be handled with less strict requirements. The main requirements concern availability and diagnosis both on the system itself and connected equipment. The prime requirement is an on-demand failure, or the system's ability to react with minimum probability to an undesirable event within a certain time. The second criterion is not to cause actions due to a false alarm or malfunction. Smaller ESD systems, e.g. on wellhead platforms, can be hydraulic or hardware (non-programmable) [27].

### 2.3.1 Safety instrumented systems SIS

A Safety Instrumented System (SIS) is a new term used in the standards that also has been known by the majority as Emergency stop system (ESD), System of safety stop, the system of interlocks, emergency firing system, security systems, etc. It could also be defined as the ultimate preventive security layer if the control system and operator performance are insufficient. In this case, it must exist a system that automatically takes the appropriate actions (partial or total stops of equipment and plants) in order to avoid the risk. These safety instrumented systems are normally separate and independent from control systems, including logic, sensors, and valves on the field. Unlike control systems, which are active and dynamic, SIS are basically passive and "sleepy", so they usually require a high degree of safety and fault diagnosis, as well as to prevent inadvertent changes and manipulations and good maintenance [37]. Therefore, to involve fault diagnosis methodologies is one important aspect of process safety that needs to be developed continuously.

When an accident occurs, it is usually due to a number of causes or their combinations that produce a dangerous event. In the industry are implemented the Emergency Stop Systems (ESD) for the protection of humans, the environment, and equipment. Therefore, is not a new concept, the novelty is the way to treat it. In other words, emergency shutdown systems will have a life cycle, which we will call Security Life Cycle. This cycle will begin in its definition phase and will end in the dismantling.

The variety of names assigned to Emergency Stop Systems seems unlimited: Interlocking System (IS), Instrumented Security System (SIS), Emergency Stop System (ESD), etc. Within the Process Industry, the debate continues on the meaning of each one of them. Even in the ISA SP84 Committee, there were ongoing discussions of the terminology, definition, and meaning of each of these terms. Nonetheless, the confusion in the industry goes beyond its own meaning, it affects the own design, installation, commissioning, maintenance, modifications, etc. There are many examples and questions that do not are easy to answer or the answer is not the same, depending on the standard or the person who gives it. Some typical doubts are set out as an example:

- Selection of the technology to use
  - What technology should be used: relays, solid state, a microprocessor (PLC)?
  - Does that selection depend on the application?
  - Relays are still used in small applications but would you design a system 500 relay inputs/outputs?
  - Is it economical to design a system with 20 inputs/outputs with redundant PLCs?

    Some prefer not to use software-based systems in security applications. Is it a good recommendation?
- Selection of redundancy
  - How redundant should an instrumented security system be designed?

  - Does it depend on the technology or the level of risk?
  - If most relay-based systems are simple, why are programmable triple redundancy systems so popular today?

- Field elements

  - Should the initiating elements be transmitter type or switch type?
  - If we use transmitters. Which type, analog or digital?
  - Redundancy or not in the field elements?
  - Can the same field elements be used for interlocks and for control?
  - Which is the best frequency of proof of these elements?

Industrial accidents rarely occur because of a single cause. Typically, they are a consequence of a combination of unusual events that are thought to be independent and should not happen at the same time. Taking; for example, the worst chemical accident so far in Bhopal (India) in a pesticide plant. Some 3,000 people died immediately and at least 12,500 died in the weeks afterward from inhaling gas and drinking contaminated water. Since then an estimated 25,000 people have been killed in the aftermath and some 150,000 are affected in some way. It happened as follows: The material that escaped in this plant was methyl isethionate (MIC). This leak (in the order of 40 tons) occurred in a storage tank containing more than the company's safety procedures (one of the causes). The operating procedure indicated the use of a refrigeration system to maintain the temperature in the product of said tank at 5°C with an alarm when the temperature rises from 11°C. The cooling system was switched off, and the MIC had been stored at a temperature close to 20°C and the alarm had been reset to 20°C. One worker was commissioned to flush pipes with water and the filters that were clogged. The water passed to the storage tank of the MIC through the leak of a valve producing a violent reaction with greater gas production. The tank pressure and temperature gauges that indicated the situation abnormal were not taken into account when thinking that they were imprecise. The flushing gas separator/washer that could have neutralized the leak was out of service because the MIC production was suspended and thought that it was not therefore necessary. Also, the own torch that could have burned part of these gases was out of service for maintenance. Finally, there were a series of events and errors in the emergency plans that completed the fatal scenario of that accident. As explained above, it is clear that accidents are the combination of rare events that are usually assumed to be independent and difficult to match over time. One of the methods of protecting against them is to implement multiple independent layers of protection that make it more difficult for such events to lead to dangerous conditions. It is therefore fundamental that from the beginning of a project and in its stage of operation and maintenance are available such layers of protection perfectly structured, subject to procedures and maintained with a very simple idea: "Do not put all the eggs in the same basket"

As it can be seen, the Safety Instrumented Systems constitute the last layer of preventive security and there lies its great importance and necessity within the industrial safety of the process industries. Nonetheless, it is important to clarify the difference between what is required by law and what is a good design and work practice in specifications, standards, and norms. Moreover, to claim that what may be mandatory in one country (example: US) may not be to others or vice versa. Regarding Safety Instrumented Systems (SIS), there is no directive to enforce compliance. But there are European standards, such as EN-746-2, which obliges a certain SIL (Safety Instrumented Layer) in some security ties, establishing also the test interval and the architecture to be implemented. There are standards and norms whose fulfillment is considered advisable and with a future vision should be put into practice in projects and modifications since, as in other fields, the directive will finally appear, obliging its fulfillment. Before that a SIS will be activated, a fault diagnosis technique needs to be implemented to help the operators of taking early decisions that carrying the process to a safe state. Concluding, diagnosis in industrial processes involves many aspects as was presented. The importance of the reliability, Hazard analysis and control & safety systems in a new methodology were indicated.

## 2.4 Conclusion

In this chapter, we presented an analysis of the diagnosis in industrial processes and the extension of the protection layer "Alarm" (SUPER ALARM) was explained. It was also explained how the expert and hybrid intelligent systems are an important direction to develop future research in the field of supervision of systems and safe processes. The implementation of a methodology in the safe process requires involving aspects such as: reliability and risk management including the intrinsic safety and the identification of the hazards. In addition, the control and safety systems with the description of the Safety Instrumented Systems SIS was considered.

# Chapter 3

# Hybrid models and Causal graphs

## 3.1 Introduction

The methodology proposed in this thesis uses hybrid models and causal graphs as tools that help human operators to understand the behavior of the physical components of the process and to verify the representative event sequences that are generated in step 2 of the CBAM methodology. These models can also be used to explain and provide semantics to the chronicles that will be learned from the event sequences.

Complex systems involve continuous and discrete variables that need be gathered in a hybrid model. A hybrid model represents a combination of continuous evolution with discrete dynamics whose changes are triggered by process supervisory actions or internal changes. A hybrid system develops its behavior in a continuous manner, but the continuous dynamics change when the system mode changes.

On the other hand, causal models are appropriate to explicit the influences among variables. This is why, in the alarm management methodology proposed in this thesis, a Hybrid Causal Model as proposed in [79] is used to represent and to identify the variables and the relationships between the modes of operation, procedural actions and the alarms of the system. In addition to explanatory purposes already mentioned, this model can also be used as a fault injection tool that can be used to generate event sequences for the scenarios to be learned.

### 3.1.1 Hybrid Causal Model

This section presents the representation that supports the understanding of the analyzed processes in the Chronicle Based Alarm Management (CBAM). It is based on a hybrid causal model and a qualitative abstraction process of the continuous behavior.

As it was explained before, industrial processes can be represented by a hybrid model which expresses in a formal framework the continuous and discrete dynamics involved in the system. The role of the Hybrid Causal Model is to represent and to identify the variables and the relationships between the modes of operation, procedural actions and the alarms of the system. The hybrid system is represented by an extended transition system, whose discrete states represent the different modes of operation for which the continuous dynamics are characterized by a qualitative domain. Formally, a hybrid causal system is defined like in [79] as a tuple:

$$\Gamma = (\vartheta, D, Tr, E, CSD, Init, COMP, DMC) \tag{3.1}$$

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[1] noted $\sigma$, where:
  - $\Sigma$ is the set of events associated to the procedure actions in a startup or shutdown stages.
  - $\Sigma^c$ is the set of events associated to the behavior of the continuous process variables.

  **Note**: The set of observable (unobservable) events is denoted by $\Sigma_o(\Sigma_{uo})$
- $Tr : Q \times \Sigma \rightarrow Q$ is the transition function. The transition from mode $q_i$ to mode $q_j$ with associated event $\sigma$ is noted $(q_i, \sigma, q_j)$.

[1] To be rigorous, we should refer to "event types", however this term is not used in the hybrid systems formalisms because event dates are not necesary.

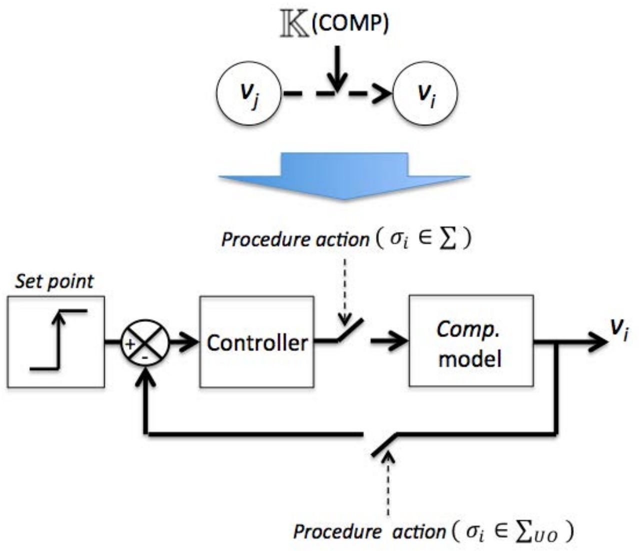


Figure 3.1: Dynamic Continuous Model $DMC$

- $CSD \supseteq \bigcup_i CSD_i$ 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 Fig. 3.1. 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.

## 3.2 Causal graphs

In social sciences, visual representations of causal models had been used many times and in the 60s the first contributions in the form of path diagrams appeared for linear structural equation models [9] [35]. Graphical causal models can be thought as directed acyclic graphs (DAGs). Various closely related (but not identical) bridges between DAGs and causation exist [36] [85]. A directed acyclic graph (DAG) is a finite directed

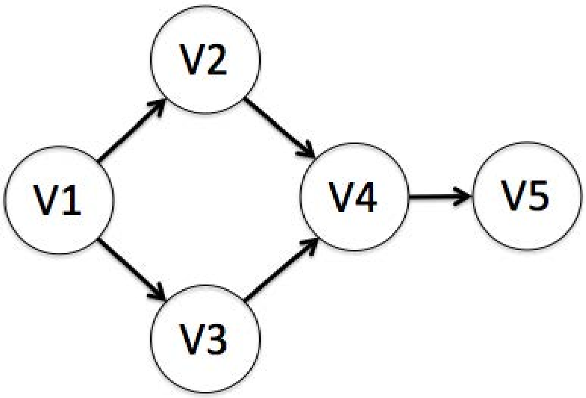


Figure 3.2: DAG example

graph with no cycles. A DAG has vertices (nodes) and edges (arcs), and each edge is directed from one vertex to another, such that there is no way to start at any vertex $V$ and follow a consistently-directed sequence of edges that eventually loops back to $V$ again. Therefore, a DAG is a directed graph that has a topological ordering. It has a sequence of the vertices such that every edge is directed from earlier to later in the sequence. Fig. 3.2 presents an example of a DAG in which the causal influence between the variables $V1$: Seasons, $V2$: Rain, $V3$: Sprinkler, $V4$: Wet pavement, and $V5$: Slippery is showed. A causal model can capture the influences between the variables of a process and supports qualitative and quantitative knowledge that may be interpreted by a diagnosis module. Particularly, each influence is marked in terms of physical component(s) of the process, which establishes a link between the knowledge of the process behavior and the process hardware [99].

### 3.2.1 Principles of causal modeling

The basic structure of a causal model is a directed graph, named the causal graph. The causal graph is composed of a set of nodes $V$ and a set of arcs $I$. Nodes depict variables and arcs represent influences among the variables. In [73], the graphs are a mighty mathematical tool and have been used to depict physical system properties. State-space representations of linear structured systems can be easily transformed into a graph. Now, [30] expresses in graph theoretic terms the classical system properties useful for control such as controllability, finite and infinite zero structure. In these control approaches, the state-space representation of the system is given, and the graph is generated easily: nodes correspond to state variables and edges are associated to the non zero parameters in the state and input matrices. The proposals [56] and [66] used qualitative digraphs for fault detection in which the arcs contain knowledge about the signs of the influences. Influences from which propagation of faults is deduced by higher or lower values of the variables. These approaches can be extended to batch processes and in this situation, the model is dynamic and quantitative. Constructing a causal graph of a specific industrial process requires many types of knowledge sources. One is the empirical knowledge of operators and experts of the process behavior, which can be difficult to extract [51],[61]. Although some authors qualify this type of knowledge as subjective and sketchy, in other works this knowledge is important to the construction of tools for diagnosis [74], [101]. Another type of process knowledge is related to the description of the process by a set of differential-algebraic equations that define its behavior. Using these equations, the causal graph can be generated automatically [100], [98], but obtaining this kind of knowledge involves all the difficulties of physical modeling and needs further processing to generate the causal graph.

### 3.2.2 Generation of Causal Graphs

Several techniques can be adopted for the generation of causal graphs. For example, we can mention the causal ordering framework of [53], the graph theoretic framework of [81] or the framework for multiple mode systems in [100]. This last approach has a great advantage of operating from the equations structure, hence only requiring a structural relation model (SRM) as initial knowledge. In the causal graph, a set of influences from variables $v_i, ..., v_n$ to the variable $\rho$ means that a relationship $r(v_i, ..., v_n, \rho)$ exists between these variables and that this relationship is expressed in such a way that $\rho$ is computed from $v_i, ..., v_n$ values [17]. In a causal model, it is possible to find qualitative and quantitative information. Moreover, there are approaches in which each influence is labeled with the physical components that underlie the relationship. The label is called the "influence/relation support" and provides the "causal model structure" [22]. The three following properties are commonly accepted to characterize causality:

- Necessity (effects have unique causes)
- Locality (the effect is structurally close from the cause)
- Temporality (the cause precedes the effect)

Causality appears naturally in differential or difference equations in canonical form [98], *i.e.*:

$$\frac{dx_{n+1}}{dt} = f(x_1, ....x_n) \quad or \quad x_{t+1}^{n+1} = g(x_t^1, ...x_t^n) \tag{3.2}$$

The left-side variable is causally dependent on the right side variables. This choice is not arbitrary; it is due to physical considerations. The same reasoning can be made for relations containing delays. If a variable $A$ influences a variable $B$ with a delay $d$, then $A$ is causally dependent on $B$. Let's consider a set of equations $E_c$, and a set of variables $Var$. A exogenous variable is to a system $\Gamma$ if it cannot be described with the help of the other variables of $\Gamma$. A variable is denominated endogenous (set $V_{en}$) if its behavior is described within the system model. On the other hand, a variable is denominated exogenous (set $V_{ex}$) if its behavior do not depend of the system model. The following equations constitute the Structural Relation Model (SRM).

$$E_c = \{e_1 : e_1(V_1, V_2, V_3), e_2 : e_2(V_4, V_5, V_1), e_3 : e_3(V_1, V_2)\} \tag{3.3}$$

$$Var = \{V_1, V_2, V_3, V_4, V_5\} \tag{3.4}$$

$$V_{en} = \{V_1, V_2, V_3, V_4\} \tag{3.5}$$

$$V_{ex} = \{V_5\} \tag{3.6}$$

Five steps are necessary to produce the Causal Model Structure CMS from the previously obtained structural relation model (SRM) [17]. The first step consists in generating a preliminary bipartite graph. A bipartite graph is an undirected graph in which nodes can be divided into two sets such that no edge connects nodes within the same set. In this example, the two sets are the set of equations $E_c$ and the set of variables $Var$. The bipartite graph $G = (Var \cup E_c, A)$ is hence defined, in which a non-directed edge $A(V_i, e_j)$ between $V_i$ and $e_j$ exists if, and only if, the variable $V_i$ is involved in equation $e_j : V_i \in Var(e_j)$ as shown in Fig. 3.3.

The objective is to determine for each equation $e_j$ which variable is causally dependent on the other variables involved in $e_j$. This means that for instance an equation such that $e_j(V_1, V_2, V_3)$ is rearranged as :

$$V_2 = g(V_1, V_3) \tag{3.7}$$

In this case, the variables on the right side $V_1$ and $V_3$ are the direct causes of the variable on the left side $V_2$, which can also be interpreted as: the values of $V_2$ that

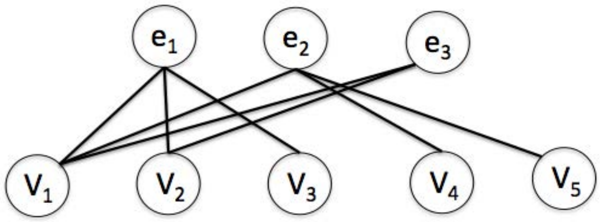


Figure 3.3: Bipartite graph

can be computed from the values of $V_1$ and $V_3$. A system of $n$ algebraic equations is selfcontained if any suitable subset of $k(k \leq n)$ involves at least $k$ variables. This notion can be compared to the definition of a just determined system that was introduced in [14]. Therefore, causal ordering requires first of all to specify the exogenous variables of the SRM. Additionally causal ordering also requires the SRM to be non degenerated, *i.e.* $n_{Ec} = n_{Var}$ (equal number of variables and equations) and selfcontained. This restriction expresses the number of endogenous variables that can be computed with the model, which means that some variables need to be considered as exogenous even if they are not so in reality. These variables are referred to as pseudo-exogenous variables in the following. They constitute the set $V_{pseudo.exo}$. This is an important point for a practical application. More than one causal graph can be built for the same SRM depending on the choice of the pseudo-exogenous variables. If the system is not self contained, the model has to be modified. The proposal of [105] gives a structural method to get a workable model from a set of Differential Algebraic Equation (DAE).

For each exogenous or pseudo-exogenous variable in $V_{ex}$ and $V_{pseudo.exo}$, $E_c$ must be increased with a so-called exogenous equation which affects a constant value to the variable, meaning that this variable is controlled by the system's environment. In the example, $(n_{Var} = 5) \neq (n_{Ec} = 3)$. For real applications, practical considerations guide the choice of pseudo-exogenous variables. In the example, $E_c$ is increased by 2 exogenous equations relative to variables $V_1$ and $V_5$ to obtain a just-determined bipartite graph $G_j$, see Fig. 3.4.

Causal ordering is the result of determining a perfect matching in $G_j$. Therefore, the perfect matching in a bipartite graph is a set of edges such that each edge is connected to only one node of each set of the bipartite graph and each node is connected to only one edge. In the just determined bipartite graph (Fig. 3.4), some edges obviously belong to the perfect matching. For instance when an equation involves only one variable (this is the case for instance of the pseudo-exogenous equations) and when a variable is involved in only one equation (case of variables $V_4$ and $V_5$) (Fig. 3.5). This

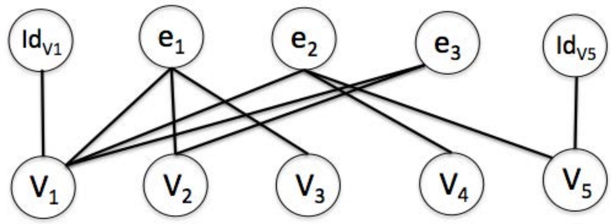


Figure 3.4: Just-determined bipartite graph

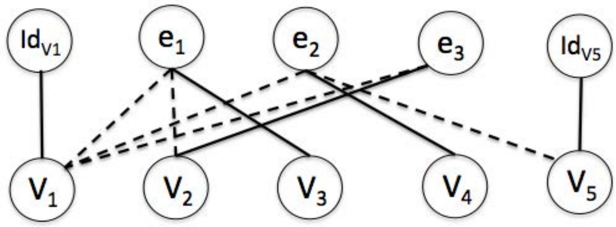


Figure 3.5: Edges belonging to the perfect matching in solid line

is also the case of dynamic relations, since their causal interpretation is predefined, as mentioned above. If the equation set $E_c$ does not contain any algebraic loop, then the perfect matching is unique. On the contrary, several perfect matching exist, which will result in the different causal interpretations around the loops. In the previous example, considering $V_5$ as exogenous variable, thus $e_2$ matches $V_4$ or $V_1$. In this case, two solutions are available. If $e_2$ is matched to $V_4$ then $Id_{V1}$ is matched to $V_1$. But, if $e_2$ is matched to $V_1$, then $Id_{V1}$ is matched to $V_4$ then no perfect matching can be found. In Fig. 3.6 is presented an example of perfect matching. We can mention that in the 50s, Ford and Fulkerson proposed an algorithm that can be used to determine the perfect matching [43].

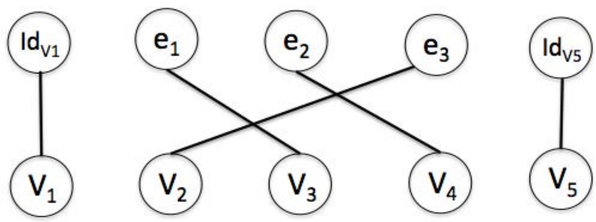


Figure 3.6: Perfect-matching

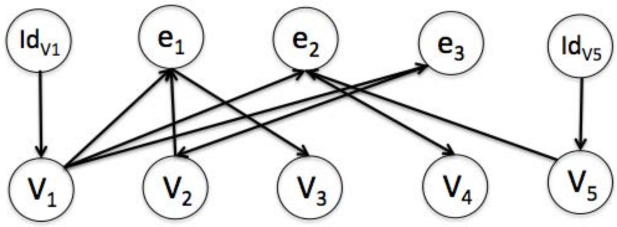


Figure 3.7: Directed graph $G'$

A directed graph $G'$ is derived from the perfect matching in $G$. The edges belonging to the perfect matching are directed from $E_c$ to $Var$. The other edges are directed from $Var$ to $E_c$ (Fig. 3.7). The causal graph $G_{ca} = (Var, I)$ is derived from the directed graph $G'$ by aggregating the matched nodes.

## 3.3 Example

The Cartagena Refinery (Colombia) is composed of several units and processes. In this example, we analyze the water injection process. This process is a HTG (Hydrostatic Tank Gauging) system composed by: one tank ($TK$), two valves ($V1$ and $V2$), one pump ($Pu$), a level sensor ($LT$), a pressure sensor ($PT$) and a flow sensor ($FT$) as shown in Fig. 3.8.

### 3.3.1 Identification of causal relationships

The level in the tank ($e_1$:$L$) is related to the weight of the liquid inside ($m$), his density ($\rho$) and the tank area ($A$). The density ($e_2$:$\rho$) is the relationship of the pressures ($P_{med}$,$P_{inf}$) in separated points ($h$). Based on the global material balance, we define that the input flow is equal to the output flow ($e_3$). Then, the variation of the weight ($dm(t)/dt$) in the tank ($e_4$) is proportional to the difference of the inflow ($q_i$) an outflow ($q_o(TK)$) in there. The differential pressure in the pump and in $V2$ ($\Delta P_{Pu}$, $\Delta P_{V2}$) are specified in $e_5$ and $e_6$. In $e_7$, the outlet pressure in the pump ($P_o$) is related with the outlet flow tank ($q_o(TK)$), the revolutions per minute in the pump ($RPM_{Pu}$), his capacity ($C$) and the radio ($r$) of the outlet pipe. The outflow ($e_8$: $qo(V2)$) and inflow ($e_{11}$: $q_i$) control are related with the percentage aperture of the valves $V1$ ($e_9$: $LV1$) and $V2$ ($e_{10}$: $LV2$) and differential pressures ($\Delta P_{V1}$,$\Delta P_{V2}$). The equations that

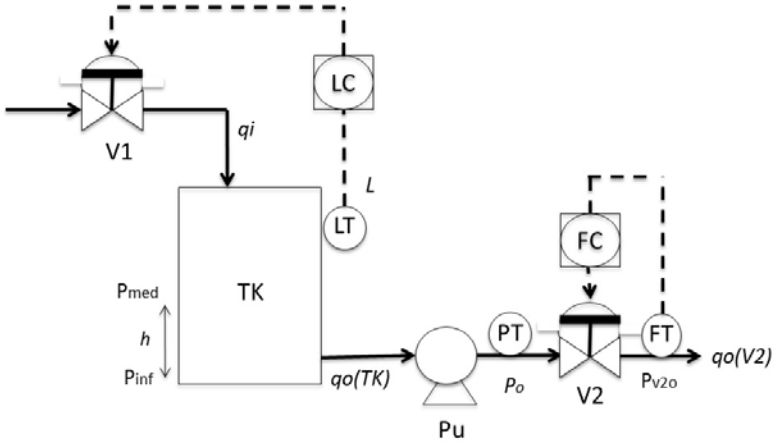


Figure 3.8: Process diagram

describe the behavior of the system are described below.

$$e_1 : L = m/(\rho * A) \tag{3.8}$$

$$e_2 : \rho = (P_{inf} - P_{med})/h \tag{3.9}$$

$$e_3 : q_o(TK) = q_o(V2) \tag{3.10}$$

$$e_4 : dm(t)/dt = q_i - q_o(TK) \tag{3.11}$$

$$e_5 : \Delta P_{Pu} = P_{inf} - P_o \tag{3.12}$$

$$e_6 : \Delta P_{V2} = P_o - P_{V2_o} \tag{3.13}$$

$$e_7 : P_o = \frac{qo(TK)}{RPMPu * C(in^3/rev)/2r(in)} * K \tag{3.14}$$

$$e_8 : q_o(V2) = f_1(LV2) * \sqrt{\Delta P_{V2}} \tag{3.15}$$

$$e_9 : LV1 = f_2(SP_L - L) \tag{3.16}$$

$$e_{10} : LV2 = f_3(SP_{q_o} - q_o(V2)) \tag{3.17}$$

$$e_{11} : RPM_{Pu} = f_4(SP_{P_o} - P_o) \tag{3.18}$$

$$e_{12} : q_i = f_5 * LV1 * \sqrt{\Delta P_{V1}} \tag{3.19}$$

$f_1$ : non linear function influence to $q_o(V2)$

$f_2$ : transfer function control level

$f_3$ : transfer function control flow $q_o(V2)$

$f_4$ : non linear function influence of $P_o$

$f_5$ : non linear function influence of $q_i$

$K$ : Factor (constant) units conversion.

The process conditions are:

- Area tank: 4 $m^2$
- $h$ tank: 2 $m$
- Volume: 8 $m^3$
- Capacity max: 7000 $kg$
- $q_i$: 1.66 $l/s \pm 10\%$; 1,41 $kg/s$
- $q_o$: 1.66 $l/s$; 1,41 $kg/s$
- Level: 1 $m$ (50%)
- $P_o$: 20 $psi$

The generation of the causal graph is arranged into five steps as recommended by [17].

### 3.3.2 Step 1

In this example, let's consider a set of equations $E_c$, and a set of variables $Var$. The following equations constitute the Structural Relation Model (SRM) of the HTG system.

$$E_c = \begin{array}{l} \{e_1 : e_1(L, m, \rho, A), \\ e_2 : e_2(\rho, P_{inf}, P_{med}, h), \\ e_3 : e_3(q_o(TK), q_o(V2)), \\ e_4 : e_4(m, q_i, q_o(TK)), \\ e_5 : e_5(\Delta P_{Pu}, P_{inf}, P_o), \\ e_6 : e_6(\Delta P_{V2}, P_o, P_{V2_o}), \\ e_7 : e_7(q_o(TK), RPM_{Pu}, C), \\ e_8 : e_8(q_o(V2), LV2, \Delta P_{V2}), \\ e_9 : e_9(LV1, SP_L, L), \\ e_{10} : e_{10}(LV2, SP_{q_o}, q_o(V2)), \\ e_{11} : e_{11}(RPM_{Pu}, SP_{P_o}, P_o), \\ e_{12} : e_{12}(q_i, LV1, \Delta P_{V1})\} \end{array} \tag{3.20}$$

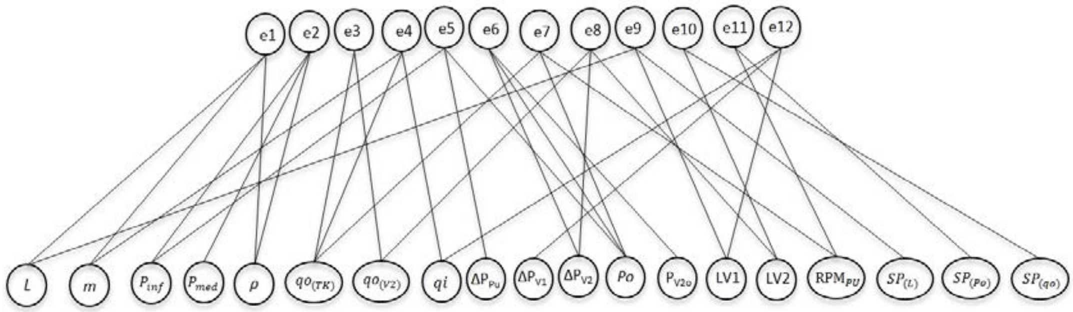


Figure 3.9: Bipartite graph of the HTG system

$$Var = \begin{array}{l}\{L, m, P_{inf}, P_{med}, \rho, q_o(TK), q_o(V2), q_i, \Delta P_{Pu}, \Delta P_{V1}, \Delta P_{V2}, P_o, P_{V2_o}, LV_1, LV_2, \\ SP_{P_o}, SP_L, SP_{q_o}, RPM_{Pu}\}\end{array} \tag{3.21}$$

$$V_{en} = \begin{array}{l}\{L, m, P_{inf}, P_{med}, \rho, q_o(TK), q_o(V2), q_i, \Delta P_{Pu}, \Delta P_{V1}, \Delta P_{V2}, P_o, P_{V2_o}, LV_1, \\ LV_2, RPM_{Pu}\}\end{array} \tag{3.22}$$

$$V_{ex} = \{SP_{P_o}, SP_L, SP_{q_o}\} \tag{3.23}$$

The bipartite graph $G$=$(Var \cup E_c, A)$ is such that a non directed edge $A(V_i, e_j)$ between $V_i$ and $e_j$ exists if, and only if, the variable is involved in equation $e_j$ : $V_i \in Var(e_j)$. The objective is to determine for each equation $e_j$ which variable is causally dependent on the other variables involved in $e_j$. For example the equation $e_1(L, m, P_{med})$ it can be rearranged as $L = f(m, P_{med})$. The variables on the right hand side are the direct causes of the variable on the left hand side, see Fig. 3.9.

### 3.3.3 Step 2

Causal ordering requires that the number of equations $n_{Ec}$ involves at least the same number of variables $n_{Var}$ . As the number of variables is less than equations we need to associate an exogenous or influence equations (Id) for each exogenous variable. In practice some variables can be considered as exogenous variables even though they are not so in reality. They are referred as pseudo exogenous. The choice can be based

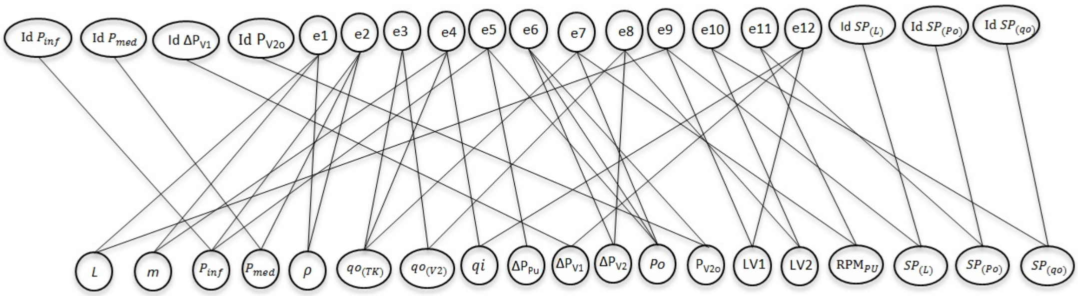


Figure 3.10: Just-determined bipartite graph of the HTG system

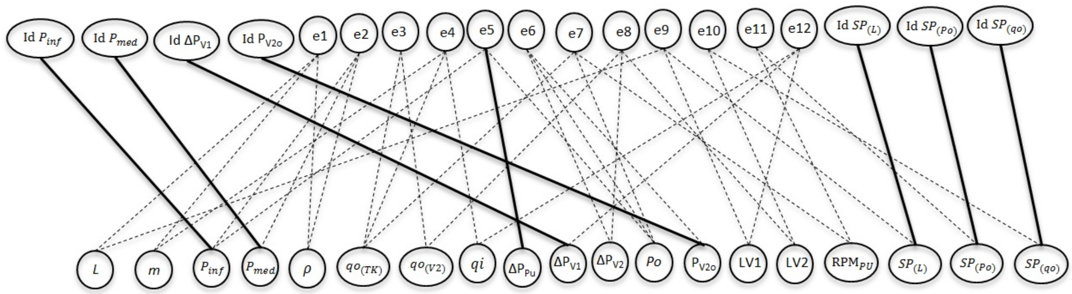


Figure 3.11: Edges belonging to the perfect matching of the HTG system

on different time scale dynamics. In our example, the variables $P_{inf}$,$P_{med}$, $\Delta P_{V1}$ and $P_{V2o}$ are considered pseudo exogenous and an influence equation is associated for each exogenous variable $SP_{P_o}$,$SP_L$,$SP_{q_o}$. This is illustrated in Fig. 3.10.

### 3.3.4 Step 3

The perfect matching in a bipartite graph indicates the unique relationship between a unique node with an unique edge. In this step we need to identify which equations involve just one variable and when a variable is involved in only one equation (See Fig. 3.11).

### 3.3.5 Step 4

We need to identify the perfect matching. For example the equation $e_1(L,m,\rho)$ can be rearranged as $e_1 : L = f(m,\rho)$. Then there exists a match between $e_1$ and $L$ (see Fig. 3.12).

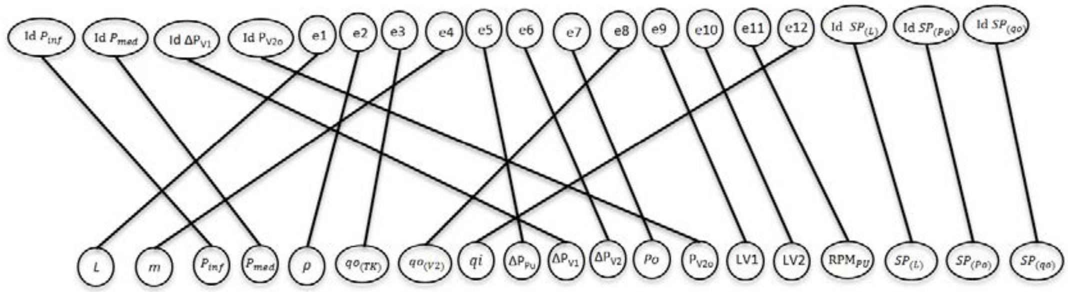


Figure 3.12: Perfect matching of the HTG system

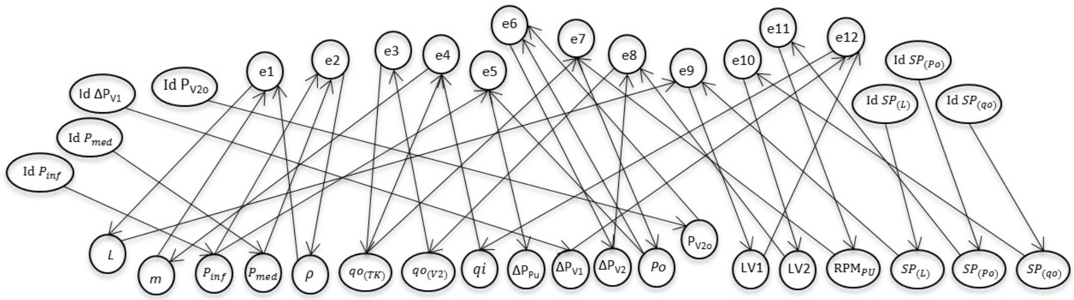


Figure 3.13: Directed graph of the HTG system

### 3.3.6 Step 5

The directed graph is derived from the perfect links by orienting the links depending on whether they have been chosen in the perfect matching or not (see Fig. 3.13).

#### 3.4.6.1 Generating the causal graph

The directed graph allows one to construct the causal graph. The nodes of the equations are eliminated and the nodes of the variables are reorganized as shown in Fig. 3.14.

#### 3.4.6.2 Suppression of unmeasured variables

To quantify each influence of the causal graph is often impossible. In such cases, the only solution is to resort to identification methods to determine the differential or difference relationship, which is only possible if data are available for the variables. But the causal model structure contains known variables (measured variables, controller set-points, etc.) as well as unknown variables. This is why a reduction operation may be used and the procedure consists in eliminating unknown variables, keeping the

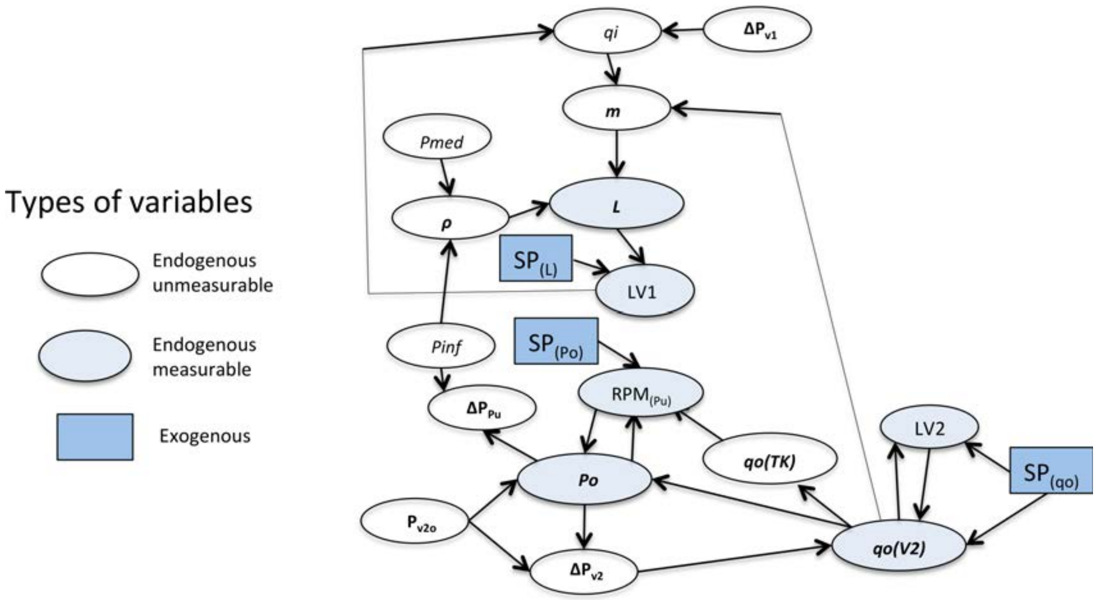


Figure 3.14: Causal graph of the HTG system

influence of physical components. Therefore, it provides the reduced causal model [80] and this procedure is similar to variable elimination theory [94] (see Fig. 3.15).

## 3.4 Conclusion

In this chapter, hybrid models and causal graphs are presented as tools to help the operators understand the behavior of physical processes and to verify the representative event sequences that are generated in the step 2 of the CBAM. These can also be used to verify the learned chronicles in the step 3 of CBAM. This chapter also explains how complex systems can be represented by hybrid causal models and describes a method that generates the causal graph associated to a set of algebro-differential equations. This chapter concludes with an example of generation of causal graph for a Hydrostatic Tank Gauging system.

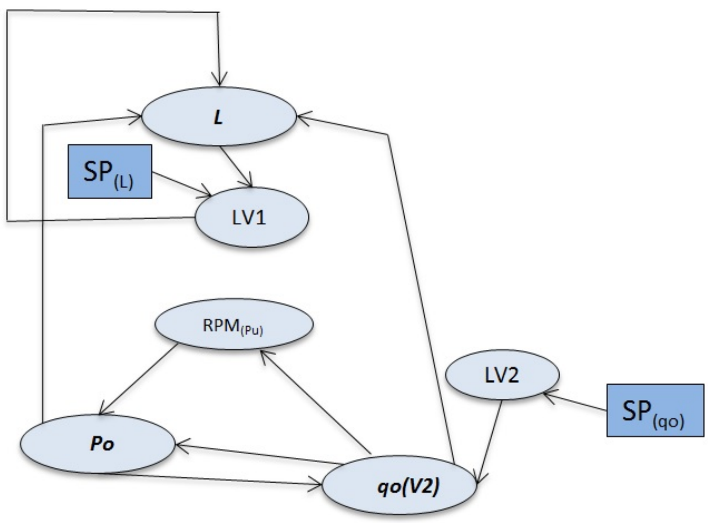


Figure 3.15: Reduced directed graph of the HTG system

# Chapter 4

# Chronicles

## 4.1 Introduction

In many fields, including aerospace, medical, financial or nuclear, very large amounts of data are produced by systems that can be real or simulated. For handling these huge masses of data, analysis support tools are necessary. Moreover, given the increasing task automation, the systems in question are increasingly critical and increasingly complex. These systems put to interact persons with machines and environments increasing risk of hazards. By analyzing the events that occurred on the process, it is possible to detect and to recognize dangerous situations. In this context the problem is to fit the formal recognition of behaviors in the context of Complex Event Processing (CEP). This corresponds to develop reliable tools that support the analysis of event streams to recognize activities associated to normal or abnormal behaviors of the process. For this, the dynamic of a process can be represented by a model that depicts the abstracted process behavior in terms of events occurring during the process evolutions. Chronicles [31] are an efficient tool for such a modeling. A chronicle is a formal model that includes the events of the systems and the time restrictions between the event occurrence dates. The design of the chronicles can be extremely complicated. There exists several approaches notably those based on learning approaches as explained in Chapter 5. The chronicle approach has been widely applied in diagnosis applications; We can cite for instance applications such as diagnosis of network telecommunication [23], cardiac arrhythmia detection [13] and intrusion detection systems [72]. 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 [52].

One of the main difficulties of chronicle design is to guarantee robustness to variations. The inclusion of expertise knowledge in the chronicles is a way to increase robustness as it permits to include a wide range of situations into the model. In this thesis the expert knowledge is expressed as temporal restrictions, which represent at a discrete level the behavior of a specific situation expressed as time points with time constraints.

In the remainder, the basic principles related to chronicles are exposed.

## 4.2 Chronicle and chronicle recognition

The concept of chronicle was developed by Christophe Dousson in his thesis [31],[32],[34],[33]. A chronicle is a set of events linked by relationships or temporal constraints and the occurrence of which will be subject to a certain context. Chronicles can also be expressed as constraint graphs where events are represented by nodes, and the time constraints are the labels of arcs. For the time, C. Dousson considers a discrete totally ordered set $\mathcal{T}$, whose granularity is fine enough compared to the observed dynamics of the environment and to the precision allowed by the means of observation.

Figure 4.1 gives a chronicle example: an event $a$ followed by an event $b$ between 1 and 5 time units, and by an event $c$ between 2 and 6 time units.

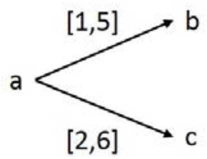


Figure 4.1: A chronicle example

A chronicle description is made through a specific description language based on predicates. A chronicle model is then composed of three parts [34]:

1. A set of predicates
2. A set of temporal constraints concerning these predicates
3. An optional set of actions to launch when the chronicle is recognized e.g a maintenance operation.

Predicates are used to define the events that are necessary for the chronicle recognition and those which are forbidden. To be recognized, a chronicle must meet all of the predicates it contains.

1. The predicate *event*: this predicate corresponds to a change of the value of an attribute or an instance of a certain message.

   Syntax: *event* (A:($a_1$, $a_2$), $t$), *event*($M$, $t$)

   The first predicate event corresponds to a change of the value of attribute A from $a_1$ to $a_2$ at time $t$. The second is an instance of the message $M$ at time $t$.

2. The predicate *noEvent*: this predicate reflects the forbidden event. It can be used both for attributes and messages.

   Syntax: *noEvent* (A:($a_1$, $a_2$), ($t_1$, $t_2$)), *noEvent* ($M$, ($t_1$, $t_2$))

   The first predicate prohibits the change of the attribute ($A$) value from $a_1$ to $a_2$ in the time interval [$t_1$, $t_2$]. The second prohibits the occurrence of a message $M$ in the time interval [$t_1$, $t_2$].

3. The predicate $OCCURS$: this predicate is a counting predicate.

   Syntax: $OCCURS$ (($n$, $m$) $a$, ($t_1$, $t_2$))

   It represents $n$ to $m$ occurrences of an event $a$, between $t_1$ and $t_2$.

### 4.2.1 Chronicle recognition

After her thesis, C. Dousson developed a chronicle recognition system called Chronicle Recognition System (CRS) which was introduced in [32]. Then later, out comes an extension in [34].

The events which are observed by the chronicle recognition system are assumed instantaneous and a chronicle is a description of a time relation between these events as explained previously. Given an event input stream handled on-the-fly, the recognition algorithm identifies all the observed event sets (called chronicle instances) that match the chronicle event patterns in respect with the time constraints. For this, CRS creates what is called *the partial instances.* A partial instance corresponds to a partial subset of the chronicle event patterns. It is only when this subset is complete that the chronicle is recognized. If a time constraint of a partial instance is violated then the instance is discarded. Figure 4.2 illustrates this notion of partial instances for the chronicle of Figure 4.1 and an input event sequence: a(at t=2),a(at t=4),b(at t=7), c(at t=9),c(at t=10) and a(at t=12). The chronicle recognition system manages the set of partial instances of chronicle as a set of time windows (one for each forthcoming event) that is gradually constrained by each new matched event: this system is predictive in the sense that it predicts forthcoming events that are relevant to instances of chronicles

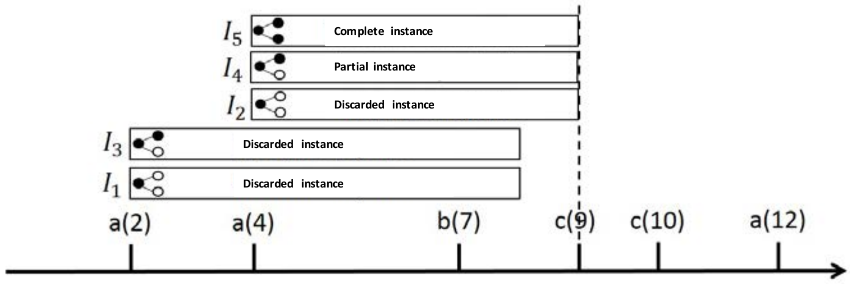


Figure 4.2: Partial and complete instances of a chronicle

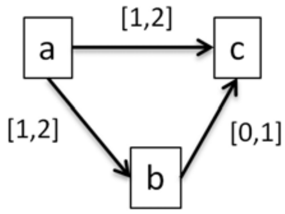


Figure 4.3: Chronicle example

currently under development; it focuses on them and it maintains their time window. The figure 4.3 presents another chronicle example which contains three event types (a,b and c) linked by three time constraints. Let us suppose that the event flow that feeds CRS is constituted by an event $a$ at t=2, a second event $a$ at t=4 an event $b$ at t=3 and an event $c$ at t=4. In this input flow, the chronicle is matched one time: the chronicle instance is: $\{a(t=2), b(t=3), c(t=4)\}$. Figure 4.4 shows the management of the time windows associated to the different events. In grey the possible dates for the remaining expected events once the occurence of an event at a specific date noted by a black box.

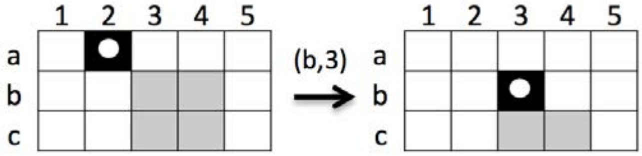


Figure 4.4: Partial instance evolution. The time windows of a partial instance {(a, 2)} (left hand) and the effect of the time constraint propagation due to the integration of (b, 3) (right hand).

When one use chronicle for diagnosis purposes and develop a **chronicle based diagnosis approach** there are two main ways to consider the problem. Chronicles can model the normal behavior of the system to diagnose. The diagnosis problem is then tackled as a consistency problem between the observations and the model of the system. In this case, the chronicle recognition allows to detect any discrepancy between the normal behavior of the system and the real behavior given through the observations (that are supposed safe). Another possibility is to consider chronicles of faulty behaviors. The efficiency of such an approach relies on the direct link between the symptom of a fault and the fault itself. Nevertheless, diagnosis based on chronicle recognition differs from classical abductive diagnosis systems as time aspects are dominant. We will see in the chapter 6 that in the **Chronicle Based Alarm Management** (CBAM) methodology proposed in this thesis, the objective is to design both chronicles describing normal behaviors and chronicles associated to faulty behaviors of the process. In this way we aim to detect any faults *anticipated or not*.

## 4.3 Chronicles: a formal framework

In this section we present the formal framework we consider to deal with chronicles.

### 4.3.1 Event, event type and sequences

The concept of **event type** expresses a change in the value of a given domain feature or set of features. $E$ expresses the set of all types of event. Let us consider time as a linearly ordered discrete set of instants.

**Definition 1:** 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. Several events can have the same type of event, but do not necessarily have the same date, for instance $(a, 3)$ and $(a, 6)$ are two events carrying the same type of event $a$.

Without loss of generality, in this thesis is assumed that two events cannot occur at the same instant, i.e. simultaneously. In the following, it may refer to an event type as an event for short.

A flow of activity generated by a system is represented by a temporal sequence. A temporal sequence (or sequence for short) consists of several events in an orderly manner, which leads us to the following definition:

**Definition 2:** A **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 $l = |S|$ is the size of the temporal sequence, i.e. the number of events in $S$, and $N_l = \{1, ...l\} \subset \mathbb{N}^*$.

An example of sequence representing an activity may be given by $S_1 = \langle (a,2)_1, (b,4)_2, (c,5)_3, (a,8)_4, (b,9)_5, (a,10)_6 \rangle$ with $l = 6$.

### 4.3.2 Chronicles and temporal restrictions

Finally, a chronicle is a set of event types associated with time variables and a set of temporal constraints between these variables.

**Definition 3:** A chronicle is defined as a triplet $\mathcal{C} = (\xi, \mathcal{T}, \mathcal{G})$ [96] such that:

- $\xi \subseteq E$. Where $\xi$ 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 $v : K \to E$, where $K \subset \mathbb{N}$,
  - $\mathcal{A}$ is a set of edges between indexed event types; there is an edge $(e_{i_\alpha}, e_{j_\beta}) \in \mathcal{A}$ where $\alpha$ and $\beta$ are integers such that $\alpha \in [1, v^{-1}(e_i)]$ and $\beta \in [1, v^{-1}(e_j)]$, if and only if there is a time constraint between $e_{i_\alpha}$ and $e_{j_\beta}$.

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

**Definition 4 (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 5 (Chronicle instance):** A chronicle $C = (\xi, \mathcal{T}, \mathcal{G})$ is recognized in a temporal sequence $S$ involving event types $\xi'$, such that $\xi \subseteq \xi'$ when all temporal

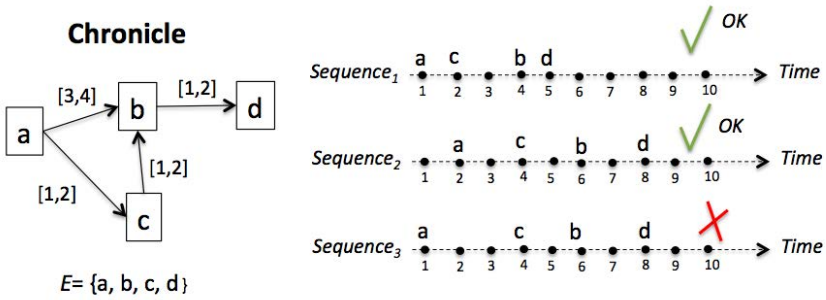


Figure 4.5: Example of chronicle instances

constraints $\mathcal{T}$ are satisfied. Then $C_{inst} = (\xi, \mathcal{T}_v)$ where $\mathcal{T}_v$ is a valuation of $\mathcal{T}$. If the sequence $S$ has finished, and at least, one event that occurs violates some temporal constraint, this chronicle is not recognized.

Fig. 4.5 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
$\mathcal{T} = \{a[3,4]b, a[1,2]c, c[1,2]b, b[1,2]d\}$.

**Definition 6 (Frequency of a chronicle):** The frequency of a chronicle $C$ in a temporal sequence $S$, noted $f(C|S)$, is the number of instances of $C$ in $S$.

For example, let us consider a chronicle $C$ defined by the event types $a$ and $b$ with a time restriction of [-10,10] between them. If we said that the frequency in this chronicle is 2, then in the temporal sequences that lead to its recognition, the event type $a$ can occur one time and the event type $b$ must occur two times. Or on the contrary, the event type $a$ occurs two times and the event type $b$ occurs only one time. In both cases, the frequency of occurrence for the pair of event types is the same $f_{ab} = 2$. In another scenario for the same example, if the frequency of this pair is six ($f_{ab} = 6$), the quantity of repetitions of the events can vary in many forms.

We will see in chapter 5 how these notions of chronicle frequency and event type frequency can be considered to guide the chronicle discovery (i.e learning) and hence to reduce the set of recognized chronicles in an input event sequence.

As explained in chapter 1, our objective is to capture the expertise of the operator when he knows something about the behavior of the process and to integrate this knowledge in the diagnosis step. For this purpose, we allow the user to specify *temporal restrictions* for event type pairs. This knowledge will then be integrated into the chronicles during the learning stage of the chronicles (see chapter 5). A *temporal restriction* expresses a known time constraint between two event type dates.

**Definition 7 (Temporal restriction):** A temporal restriction for a pair of event types $(e_i, e_j)$ is a given temporal constraint between their event dates $TR_{ij} = e_i[t^-, t^+]e_j$.

## 4.4 Example

In the following example is presented a chronicle that describes the possible behavior of an oven system shown on Figure 4.6. The oven is charged with two products ($a$ and $b$) before that the heaters are turned on initializing the startup procedure.

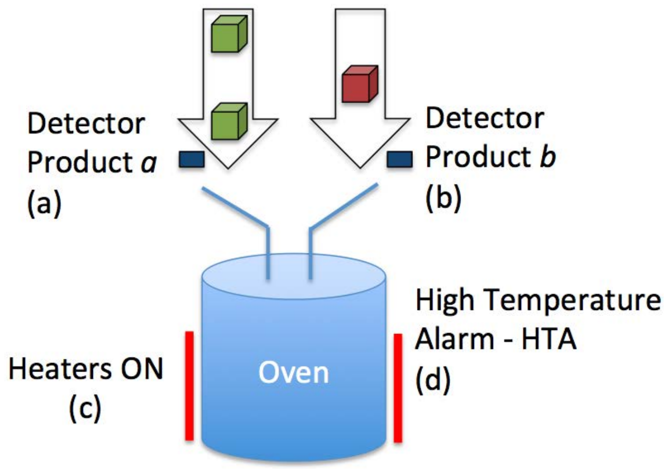


Figure 4.6: Oven charge system

To describe the oven behavior we consider four event types:

- The event type $a$ indicates that the product "a" has passed.
- The event type $b$ indicates that the product "b" has passed.
- The event type $c$ expresses that the heaters of the oven are "ON".
- The event type $d$ expresses that the temperature into the oven has arrived to its high limit.

The chronicle $C$ represents the temporal pattern of a normal startup for this process (see Fig. 4.7). This chronicle has the following event types $\xi = \{a, b, c, d\}$ and the following time constraints $\mathcal{T} = \{\tau_{ab} : [-2,-1], \tau_{ac} : [5,6], \tau_{ad} : [8,9], \tau_{bc} : [3,5], \tau_{bd} : [7,8], \tau_{dc} : [2,6]\}$. For instance, the event type $a$ only can occurs between 1 and 2 time units after that the event type $b$ has occurred. This chronicle is assumed with a frequency of 1. The directed graph ($\mathcal{G}$) is given Figure 4.7. This process has also a temporal restriction that represents the expertise of the human operator: $TR_{ab} = a[-2,2]b$.

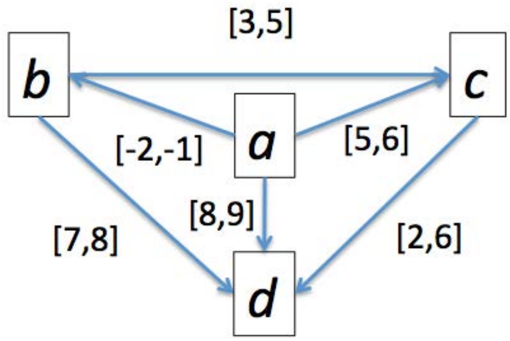


Figure 4.7: Directed graph $\mathcal{G}$ of the chronicle $C$

## 4.5 Conclusion

This chapter gives a brief overview of the chronicle framework that supports our proposal. Definitions of *event type*, *event*, *event sequences*, *chronicle* and *temporal restrictions* were described. A background on chronicle recognition was presented to better understand the basic priniciple of a chronicle based diagnosis approach. Finally an illustrative example of a chronicle modeling the normal startup of an oven charge system was given.

# Chapter 5

# Chronicle learning

## 5.1 Introduction

In numerous areas, finding temporal patterns hidden in a sequence of events has important implications for the analysis of processes. [25] presented an approach for the discovery of chronicles hidden in a sequence of events that represents a specific scenario. Chronicles are a special type of temporal pattern, in which the temporal order of events are quantified with numerical bounds. Our idea is to associate one chronicle to each situation to be recognized (normal or faulty) and obtain a well-designed chronicle data base after analysis of the different chronicles [7]. During the operation of the system, several sensors are used to retrieve information about the system‘s status over time. This record is transformed into a sequence of discrete events. We then apply a chronicle recognition algorithm to this trace and look for the chronicles of the data base that are recognized. We deduce in which situation the system is accordingly.

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 2008, [59] proposed a global model of a set of alarm sequences that are generated by a knowledge based system monitoring a dynamic process. A work aiming at showing a method to discover signatures (or models of chronicles) from a discrete event sequence (alarms) generated by a monitoring cognitive agent (MCA) [58] was also proposed in 2004. In another work [47], 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) ([68],[12]), case-based chronicle learning ([42],[41]) that is a characteristic supervised method by reinforcement learning but also ([48],[114][33],[49]) 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 ([71]).

Among these methods, the frequency criteria is widely used ([34],[25],[65]). In [39],[38], 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 thesis is also based on a frequency criteria ([96],[110]) and can be related to [34] and [25]. The proposal in [34] makes it possible to discover, given a trace $S$ and a threshold frequency $f_t$, chronicles of frequency $f \geq f_t$ in $S$. The algorithm proposed by [34] limits the search and it does not generate all the frequent chronicles. The algorithm of [25] solves the non-completeness problem by adding more possibilities for temporal constraints attached to pairs of event types. The idea is also to build the chronicles little by little 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 algorithm of [34] or by further constraining one of the constraints guided by the constraint graph.

The algorithm proposed by [25], named $HCDA$, is interesting because it provides the complete set of chronicles but it is designed to accept only a unique event sequence representing a specific situation as input. The chronicle exploration process discovers all the chronicles whose instances occur in the input temporal sequence of event types . In many cases however, the same situation does not imply perfectly identical event sequences. This is why $HCDA$ was extended in [96] to account for a set of input event sequences representing possible variants for one situation. The proposed $HCDAM$ algorithm learns the chronicles, whose instances occur in *all* event sequences

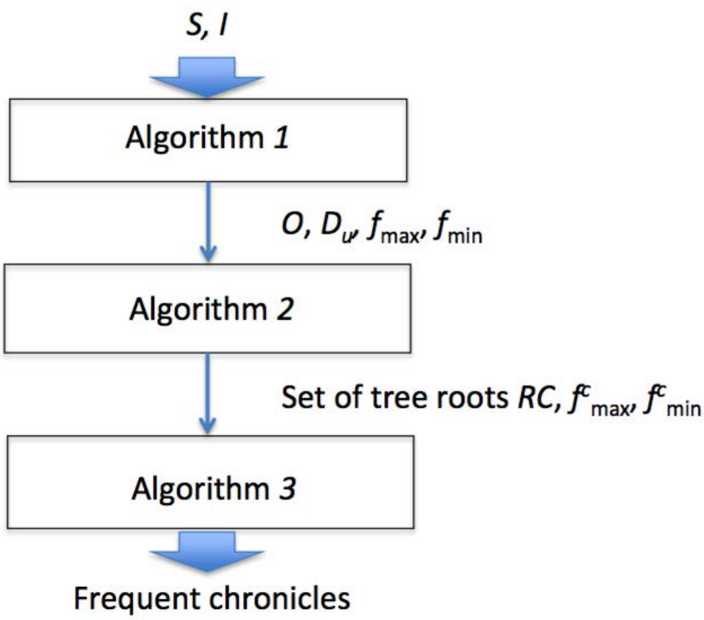


Figure 5.1: HCDAM organization

representing the same situation. In the following subsections, *HCDAM* is described with its different phases.

## 5.2 Heuristic Chronicle Discovery Algorithm Modified

The *HCDAM* algorithm [96] aims at discovering frequent chronicles common to multiple sequences representing variations of a unique situation. Given a set of sequences $\mathbb{S}$ and a minimum frequency threshold, *HCDAM* finds all minimal frequent chronicles presented in all temporal sequences. The fact that the event sequences arising from the same situation generally present variants that must be accounted for is the principal interest of this algorithm.

The principles of *HCDAM* are briefly reminded in this section and the reader is referred to [96] for a detailed presentation.

*HCDAM* is organized in three phases supported by three algorithms as presented in Fig. 5.1.

The three phases are the following:

1. Filtering the event types that are not present in all sequences of $\mathbb{S}$,

2. Building a constraint database from the temporal sequences where the temporal constraints for each pair of event types are stored in a constraint graph structure. In this graph, time constraints are nodes of an acyclic oriented graph whose arcs represent the relationship between constraints.

3. Generating a set of candidate chronicles initialized with a set of chronicles that were proved to be frequent from the constraint database to explore the chronicle space.

The following sequences are used in the following to illustrate the three phases of the $HCDAM$ algorithm described below:

$$S_1 = \langle (b,1),(a,3),(b,4),(b,5) \rangle \tag{5.1}$$

$$S_2 = \langle (a,1),(b,2),(b,3),(a,6) \rangle \tag{5.2}$$

$$S_3 = \langle (a,1),(b,2),(b,4),(c,5),(b,7) \rangle \tag{5.3}$$

### 5.2.1 Phase 1

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

- Filtering the event types that are not present in all input sequences $\mathbb{S}$: if $\exists S_k \in \mathbb{S}$ such that $\exists e_i \notin S_k$, then $e_i$ will be removed of all other sequences in $\mathbb{S}$.

- Filtering on a given set of event types $\Psi = \{e_1, e_2, ..e_r\}$ if we are interested only in those event types during processing.

After, filtering the sequences, the set of occurrences $O = \{O^k_{ij}\}$ that contain all the instances of a pair of event types $(e_i,e_j)$ 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:

$$O^1_{ab} = \{\langle (a,3),(b,1) \rangle, \langle (a,3),(b,4) \rangle, \langle (a,3),(b,5) \rangle\} \tag{5.4}$$

$$O^2_{ab} = \{\langle (a,1),(b,2) \rangle, \langle (a,1),(b,3) \rangle, \langle (a,6),(b,2) \rangle, \langle (a,6),(b,3) \rangle\} \tag{5.5}$$

$$O^3_{ab} = \{\langle (a,1),(b,2) \rangle, \langle (a,1),(b,4) \rangle, \langle (a,1),(b,7) \rangle\} \tag{5.6}$$

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

$$D^k_{ij} = \{d^k_{ij} = (t_j - t_i) \mid < (e_i, t_i), (e_j, t_j) > \in O^k_{ij}\} \tag{5.7}$$

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

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

**Algorithm 1** determines the sets of event type occurrences $O$, durations $D_u$ and the maximal frequency $f_{max}$ from the event sequences.

### 5.2.2 Phase 2

In a second phase, $HCDAM$ builds the so-called $constraint\ data-base$ $\mathbb{D}$ 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 $T^\alpha_{ij}$ for each pair of event types $(e_i, e_j)$ with $i, j = 1... \mid E \mid, i \leq j$ and $\alpha = 1, ...n_{ij}$. In the trees, time constraints are nodes and arcs represent the relationship $is\ parent\ of$ defined as below:

**Definition 10** ($is\ parent\ of$ **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 $T^\alpha_{ij}$ 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 $\Upsilon = \{e_i, e_j\}$ 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 [96] for the details of the method used in $HCDAM$ 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, the pair $(a, b)$ and the pair $(b, b)$ each give rise to one tree. These are represented in Fig. 5.2, with the mention of the frequency associated to the constraints of each level of the trees. The tree for

**Algorithm 1:** Sets definition

**Input**: $l$, $\mathbb{S}$
**Output**: $O$, $D_u$, $f_{max}$
**for** $k$=1 *to* $n$; $S_k \in \mathbb{S}$ **do**
    $i \leftarrow 0$;
    $j \leftarrow 0$;
    **while** $i < l_k$ **do**
        $\epsilon \leftarrow (e)_i$;
        $\epsilon' \leftarrow (e)_j$;
        **if** $\epsilon \leq \epsilon'$ **then**
        $O^k_{\epsilon\epsilon'} \leftarrow (\epsilon, t_i)(\epsilon', t_j)$;
        $Du^k_{\epsilon\epsilon'} \leftarrow (t_i - t_j)$;
        $f^k_{\epsilon\epsilon'}$+=1;
        **else**
        $O^k_{\epsilon'\epsilon} \leftarrow (\epsilon', t_i)(\epsilon, t_j)$;
        $Du^k_{\epsilon'\epsilon} \leftarrow (t_i - t_j)$;
        $f^k_{\epsilon'\epsilon}$+=1;
        **end**
        $j$+=1;
        **if** $j > l_k$ **then**
        $i$+=1;
        $j \leftarrow i+1$;
        **end**
    $f_{max} = min(f^k_{\epsilon\epsilon'})$; $k$=1 to n
$O = \{O_{ij}\}$; $D_u = \{Du_{ij}\}$; $f_{max} = \{f_{(\sigma_i\sigma_j)}\}$
**return**: $O$, $D_u$, $f_{max}$

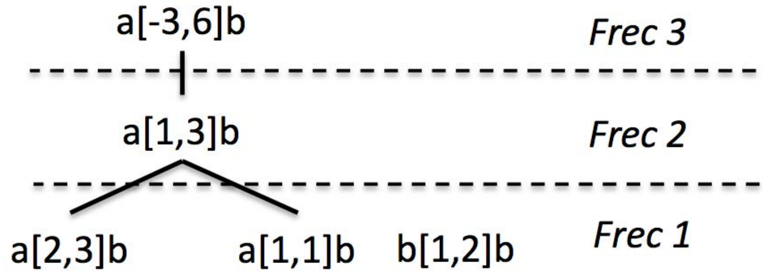


Figure 5.2: Constraint trees for the pairs $(a,b)$ and $(b,b)$

$(a,b)$ 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.

The **Algorithm 2** determines the tree roots $\mathbf{RC}_{ij}$, the minimal supports $\underline{\mathbf{I}}_{ij}$ (minimal temporal interval for the pair $(e_i,e_j)$ to appear with a given frequency in all sequences) and the maximal supports $\bar{\mathbf{I}}_{ij}$ (maximal temporal interval for the pair $(e_i,e_j)$ to appear with a given frequency in all sequences) from $f_{max}$, $O$, $Du$ and $\mathbb{S}$.

---

**Algorithm 2:** Tree roots

---

**Input**: $f_{max}$,$O$,$D_u$, $\mathbb{S}$
**Output**: Tree roots $RC$, minimal supports $\underline{\mathbf{I}}_{ij}$, maximal supports $\bar{\mathbf{I}}_{ij}$
**for** *all* $O_{ij} \in O^k_{ij}$, *k=1 to n* **do**
  $\underline{\mathbf{I}}^k_{ij}$:$\{\underline{I}^k_{ij} = [\underline{t}^-,\underline{t}^+] \mid f^k_{ij} = f_{max}$ and $\forall[t^-,t^+] \subseteq [\underline{t}^-,\underline{t}^+] f^k_{ij} < f_{max}\}$;
  $\bar{\mathbf{I}}^k_{ij}$:$\{\bar{I}^k_{ij} = [\bar{t}^-,\bar{t}^+] \mid f^k_{ij} = f_{max}$ and $\forall[t^-,t^+] \supseteq [\bar{t}^-,\bar{t}^+] f^k_{ij} > f_{max}\}$;
  **for** *k=1 to n* **do**
    $\underline{\mathbf{I}}^{comb}_{ij}$:$\{\underline{I}^{comb}_{ij} = [\underline{I}^1,...,\underline{I}^n] \mid \underline{I}^k_{ij} \in \underline{\mathbf{I}}^{comb}_{ij}\}$;
    $\bar{\mathbf{I}}^{comb}_{ij}$:$\{\bar{I}^{comb}_{ij} = [\bar{I}^1,...,\bar{I}^n] \mid \bar{I}^k_{ij} \in \bar{\mathbf{I}}^{comb}_{ij}\}$;
    **for** $\alpha$*=1 to card*$(\underline{\mathbf{I}}^{comb}_{ij})$ **do**
      $\mathbf{RC}_{ij}$:$\{r^\alpha_{ij} = \bigcup_k \underline{I}^k_{ij}, \underline{I}^k_{ij} \in \underline{\mathbf{I}}^k_{ij}\}$;
    **for** $\beta$*=1 to card*$(\bar{\mathbf{I}}^{comb}_{ij})$ **do**
      $\mathbf{MCI}_{ij}$:$\{MCI^\beta_{ij} = \bigcap_k \bar{I}^k_{ij}, \bar{I}^k_{ij} \in \bar{\mathbf{I}}^k_{ij}\}$;
    **if** $r^\alpha_{ij} \subseteq MCI^\beta_{ij}$ **then**
      $RC_{ij}$ is valid with $f$=$f_{max(ij)}$;
**return**: $RC$

---

### 5.2.3 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 to explore the chronicle space.

- The set of candidates initiates with the set of tree roots
- The operator "*add* $\varepsilon$" is used. This operator, checks at the constraint graphs in order to find the constraints of the event type $\varepsilon$ with all elements of $E$.
- The minimal number of occurrences of the candidate in $\mathbb{S}$ is counted.

Now that the constraint trees are generated, the next step is to extract the chronicles. The chronicles are extracted according to two thresholds: $f_{min}$ (or $f$=1 when not defined) and $f_{max}$. The search starts from a pair 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 for adding the event type $\varepsilon$. The operator searches the constraint graph for all the constraints between $\varepsilon$ and all the event types of the chronicles under construction in accordance with the frequency. To avoid the counting phase, the structure of the tree can be changed to no longer depend on couples of events but on the frequency of the time constraints between pairs of events.

**Algorithm 3** determines the frequent chronicles from $\mathbb{S}$, $f_{min}$,$f_{max}$. Therefore, this algorithm allows us to build chronicles complying with the thresholds $f_{min}$ and $f_{max}$.

Line 3 initializes the chronicle list.

Line 6 to line 10 of the algorithm tries in the space of couples of event types the constraints respecting the frequency $f$.

Line 10 allows to build every possible chronicle from the lists of constraints.

On Line 13 to line 16, the operator verifies if the other pairs have constraints with the event types.

Line 17 allows to insert the event types to find the chronicles.

### 5.2.4 Example

The following example illustrates the chronicle learning algorithm $HCDAM$ with the oven charge process system of Fig. 5.3. For this system, the different event types are the following:

**Algorithm 3:** Chronicle frequents

**Input**: $\mathbb{S}$,$f_{min}$,$f_{max}$
**Output**: Frequents chronicles
**for** $f \in [f_{min}, f_{max}]$ **do**
  $ChronicleList \leftarrow \emptyset$;
  $ListOfListOfConstraints \leftarrow \emptyset$;
  **for** $(\epsilon, \epsilon') \in E \times E$ **do**
    $TemporalList \leftarrow GetConstraintsFromStruct(\mathbb{S}, f, \epsilon, \epsilon')$;
    $ListOfListOfConstraints$.append(
    $TemporalList$);
  $ListOfListOfConstraints \leftarrow$
  CartesianProduct(
  $ListOfListOfConstraints$)
  **for** $ListOfConstraints \in ListOfListOfConstraints$ **do**
    $ChronilceList$.append(
    BuiltChronicleList(
    $ChronicleList$))
  Frequents.put($f$,$ChronicleList$)
**return**: $Frequents$

- The event type $a$ indicates that the product "a" has passed.
- The event type $b$ indicates that the product "b" has passed.
- The event type $c$ expresses that the heaters of the oven are ON.
- The event type $d$ expresses that the temperature in the oven has reached its high limit.

Therefore, this system has the following event types $E = \{a, b, c, d\}$
For a normal startup stage, this process has the following three event sequences:

$$S_1 = \langle (a,2), (b,4), (a,5), (c,7), (d,11) \rangle \tag{5.8}$$

$$S_2 = \langle (a,2), (b,3), (a,4), (c,7), (d,10) \rangle \tag{5.9}$$

$$S_3 = \langle ((a,2), (b,3), (a,5), (c,8), (d,11) \rangle \tag{5.10}$$

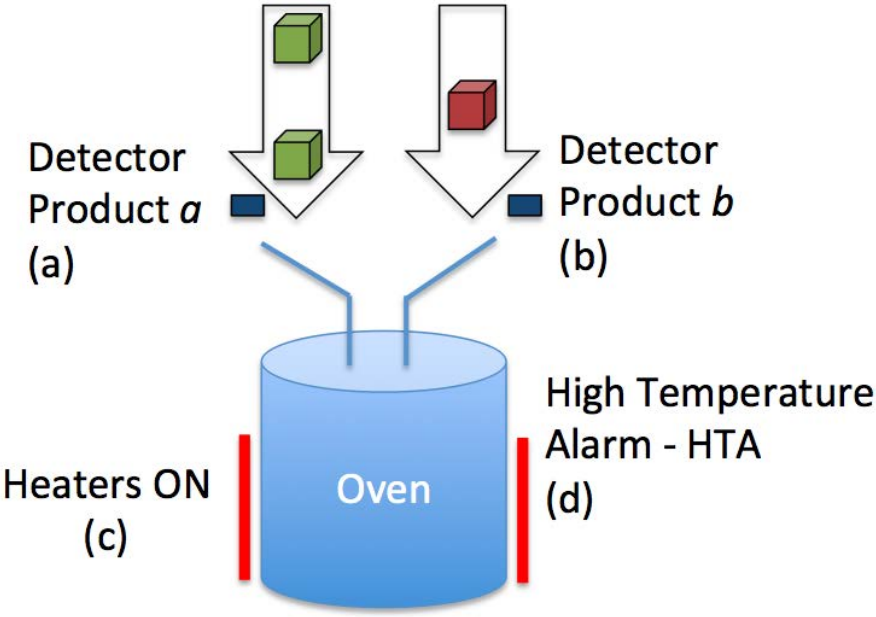


Figure 5.3: Oven charge system

The inputs to the HCDAM algorithm as well the results retruned by the three algorithms Algorithm 1, Algorithm 2, and Algorithm 3 are given below.

**Input** :
$\mathbb{S} = \{S_1, S_2, S_3\}$
$I = \{I_1, I_2, I_3\}$
$The \quad sequences \quad of \quad input \quad are:$
$S_1 = \langle (a,2);(b,4);(a,5);(c,7);(d,11) \rangle; l_1 = 5$
$S_2 = \langle (a,2);(b,3);(a,4);(c,7);(d,10) \rangle; l_2 = 5$
$S_3 = \langle (a,2);(b,3);(a,5);(c,8);(d,11) \rangle; l_3 = 5$

**Return Algorithm 1** :

$f_{max(ab)} = 2$
$f_{max(ac)} = 2$
$f_{max(ad)} = 2$
$f_{max(aa)} = 1$
$f_{max(cd)} = 1$
$f_{max(cb)} = 1$
$f_{max(bd)} = 1$
$O^1_{ab} = \{\langle(a,2)(b,4)\rangle; \langle(a,5)(b,4)\rangle\}$
$O^2_{ab} = \{\langle(a,2)(b,3)\rangle; \langle(a,4)(b,3)\rangle\}$
$O^3_{ab} = \{\langle(a,2)(b,3)\rangle; \langle(a,5)(b,3)\rangle\}$
$O^1_{aa} = \{(a,2)(a,5)\}$
$O^2_{aa} = \{(a,2)(a,4)\}$
$O^3_{aa} = \{(a,2)(a,5)\}$
$Du^1_{ab} = \{2,-1\}$
$Du^2_{ab} = \{1,-1\}$
$Du^3_{ab} = \{1,-2\}$
$Du^1_{aa} = \{3\}$
$Du^2_{aa} = \{2\}$
$Du^3_{aa} = \{3\}$
$O^1_{ac} = \{\langle(a,2)(c,7)\rangle; \langle(a,5)(c,7)\rangle\}$
$O^2_{ac} = \{\langle(a,2)(c,8)\rangle; \langle(a,4)(c,8)\rangle\}$
$O^3_{ac} = \{\langle(a,2)(c,7)\rangle; \langle(a,5)(c,7)\rangle\}$
$O^1_{ad} = \{\langle(a,2)(d,11)\rangle; \langle(a,5)(d,11)\rangle\}$
$O^2_{ad} = \{\langle(a,2)(d,10)\rangle; \langle(a,4)(d,10)\rangle\}$
$O^3_{ad} = \{\langle(a,2)(d,11)\rangle; \langle(a,5)(d,11)\rangle\}$
$Du^1_{ac} = \{5,2\}$
$Du^2_{ac} = \{6,4\}$
$Du^3_{ac} = \{5,2\}$
$Du^1_{ad} = \{9,6\}$
$Du^2_{ad} = \{8,6\}$
$Du^3_{ad} = \{9,6\}$
$O^1_{cd} = \{(c,7)(d,11)\}$
$O^2_{cd} = \{(c,8)(d,10)\}$
$O^3_{cd} = \{(c,7)(d,11)\}$
$O^1_{cb} = \{(c,7)(b,4)\}$
$O^2_{cb} = \{(c,8)(b,5)\}$
$O^3_{cb} = \{[(c,7)(b,4)\}$
$Du^1_{cd} = \{-4\}$
$Du^2_{cd} = \{-2\}$
$Du^3_{cd} = \{-4\}$
$Du^1_{cb} = \{-3\}$
$Du^2_{cb} = \{-3\}$
$Du^3_{cb} = \{-3\}$

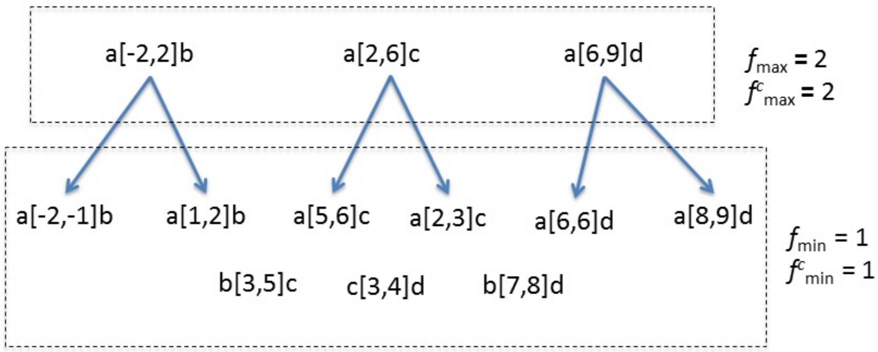


Figure 5.4: Tree roots

Table 5.1 presents the frequent chronicles obtained from the three input event sequences.

| Frec=1 | Frec=2 |
|---|---|
| a[6 ,6]d | a[-2 ,2]b |
| a[-2,-1]b | a[2,6]c |
| a[8,9]d | a[6,9]d |
| a[1,2]b | — |
| a[5,6]c | — |
| a[2,3]c | — |
| b[3,5]c | — |
| b[7,8]d | — |
| c[3,4]d | — |

Table 5.1: Frequent chronicles

The tree roots provided by Algorithm 2 are given in Fig. 5.4. Fig. 5.5 and Fig. 5.6 represent the frequent chronicles with frequency 1. Fig. 5.7 represents the frequent chronicle with frequency 2.

## 5.3 Extending HCDAM

### 5.3.1 Integration of expert knowledge in chronicle learning

Expert knowledge is important and represents specific information which can be integrated into the algorithm *HCDAM*. Our objective is to capture the expertise of the operator when he knows some time restriction about the behavior of the process.

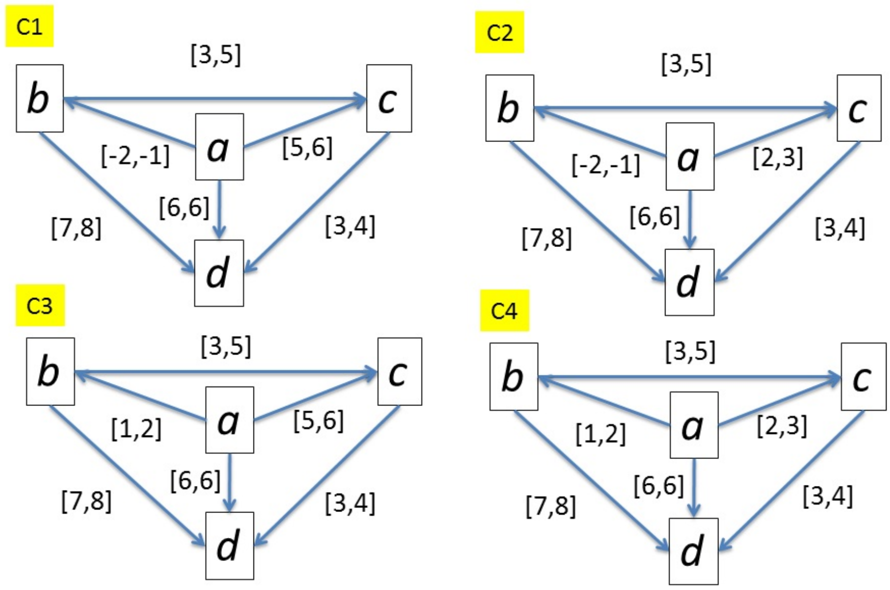


Figure 5.5: Chronicles C1 to C4 with frequency 1

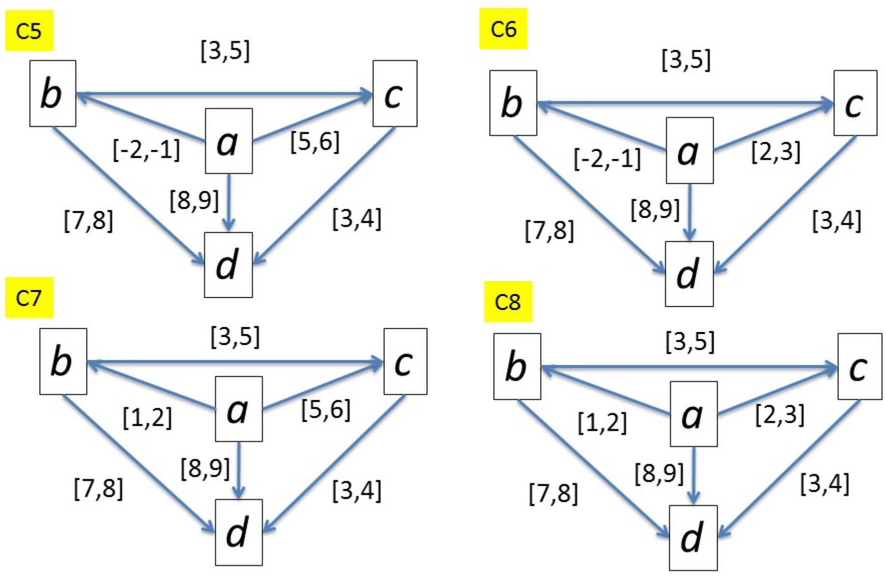


Figure 5.6: Chronicles C5 to C8 with frequency 1

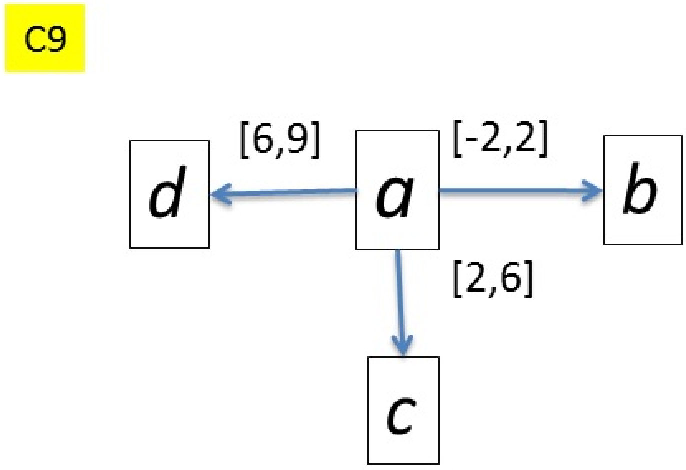


Figure 5.7: Chronicle C9 with frequency 2

For this purpose, we allow the user to specify these temporal restrictions for event type pairs. This knowledge is incorporated in $HCDAM$ as additional input information to the algorithm. In the following, an extension of $HCDAM$ 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 proposed as a contribution to the chronicle learning theory.

#### 5.4.1.1 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 data base $D_e$.

Let us remind that 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 in $D_e$ the existence of a temporal restriction $TR_{ij}$ for each pair of event types $(e_i, e_j)$. If found, the temporal restriction then replaces the tree root for this pair of event types. The integration of these temporal restrictions aims at focusing 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.

**Algorithm 4**, that represents **Algorithm 2** extended with expert knowledge determines the tree roots $\mathbf{RC}_{ij}$, the minimal supports $\underline{\mathbf{I}}_{ij}$ and the maximal supports

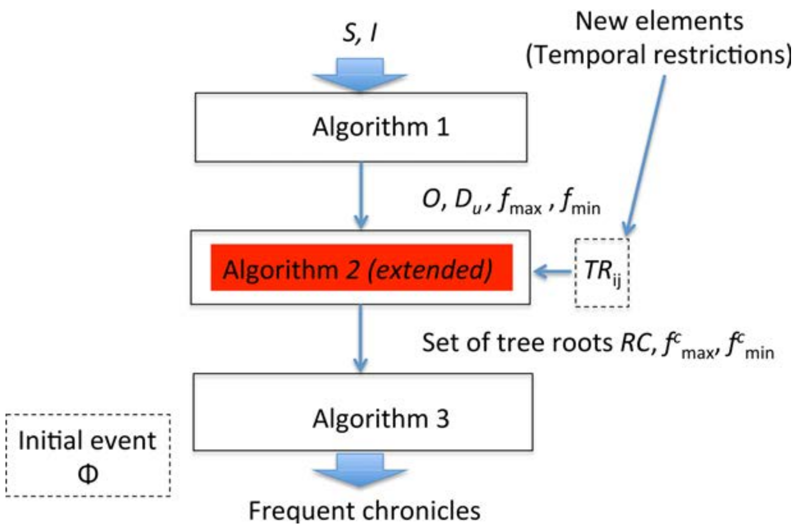


Figure 5.8: *HCDAM* extended

$\bar{\mathbf{I}}_{ij}$ from the $f_{max}$, $\mathbb{S}$ and the temporal restrictions $\mathbf{TR}_{ij}$ that represent the expertise knowledge. Line 10 of this algorithm expresses the action of checking the existence of a temporal restriction $TR_{ij}$ for a pair of event types $(e_i, e_j)$.

The *HCDAM* algorithm extended is organized in three algorithms as represented in Fig. 5.8. The extension of *HCDAM* is included in **Algorithm 2** where new elements (Temporal restrictions) are involved.

#### 5.4.1.2 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 *HCDAM*. Nevertheless it can be very useful to reduce the number of learned chronicles.

**Definition 8 (Initial event):** We define the event type $\Phi$ as the 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 $\Phi$ as the frequency of the pair $(\Phi, e_i)$.

The virtual initial event $\Phi$ allows us, without modifying the *HCDAM* algorithm, to identify the frequency of each event type whereas the original *HCDAM* only identifies the frequency of event type pairs. Fig. 5.9 illustrates the role of the event type $\Phi$.

### 5.3.2 Example

For the example of the oven charge system, **Algorithm 1** remains the same as in the non extended version of *HCDAM* (cf.section 5.3.4).

**Algorithm 4:** Tree roots

**Input**: $l$, $f_{max}$, $\mathbb{S}$, Temporal restrictions $\mathbf{TR}_{ij}$
**Output**: Tree roots $\mathbf{RC}_{ij}$, minimal supports $\underline{\mathbf{I}}_{ij}$, maximal supports $\bar{\mathbf{I}}_{ij}$
**for** $S_k \in \mathbb{S}$ **do**
    $i \leftarrow 0$;
    $j \leftarrow 0$;
    **while** $i < l_k$ **do**
        $e_i \leftarrow (Event.type)_i$;
        $e_j \leftarrow (Event.type)_j$;
        $j{+}{=}1$
        **if** *for the pair* $(e_i, e_j)$ $\nexists TR_{ij}$ **then**
            $\underline{\mathbf{I}}^k_{ij}$:$\{\underline{I}^k_{ij} = [\underline{t}^-, \underline{t}^+] \mid f^k_{ij} = f_{max}$ and $\forall [t^-, t^+] \subseteq [\underline{t}^-, \underline{t}^+] f^k_{ij} < f_{max}\}$;
            $\bar{\mathbf{I}}^k_{ij}$:$\{\bar{I}^k_{ij} = [\bar{t}^-, \bar{t}^+] \mid f^k_{ij} = f_{max}$ and $\forall [t^-, t^+] \supseteq [\bar{t}^-, \bar{t}^+] f^k_{ij} > f_{max}\}$;
            $\underline{\mathbf{I}}^{comb}_{ij}$:$\{\underline{I}^{comb}_{ij} = [\underline{I}^1, ..., \underline{I}^n] \mid \underline{I}^k_{ij} \in \underline{\mathbf{I}}^{comb}_{ij}\}$;
            $\bar{\mathbf{I}}^{comb}_{ij}$:$\{\bar{I}^{comb}_{ij} = [\bar{I}^1, ..., \bar{I}^n] \mid \bar{I}^k_{ij} \in \bar{\mathbf{I}}^{comb}_{ij}\}$;
            **for** $\alpha$*=1 to card*$(\underline{\mathbf{I}}^{comb}_{ij})$ **do**
                $\mathbf{RC}_{ij}$:$\{r^\alpha_{ij} = \bigcup_k \underline{I}^k_{ij}, \underline{I}^k_{ij} \in \underline{\mathbf{I}}^k_{ij}\}$;
            **for** $\beta$*=1 to card*$(\bar{\mathbf{I}}^{comb}_{ij})$ **do**
                $\mathbf{MCI}_{ij}$:$\{MCI^\beta_{ij} = \bigcap_k \bar{I}^k_{ij}, \bar{I}^k_{ij} \in \bar{\mathbf{I}}^k_{ij}\}$;
            **if** $r^\alpha_{ij} \subseteq MCI^\beta_{ij}$ **then**
                $RC_{ij}$ is valid with $f{=}f_{max(ij)}$;
            **else**
            $RC_{ij} = TR_{ij}$ with $f{=}f_{max(ij)}$;
        **if** $j > l$ **then**
            $i{+}{=}1$;
            $j \leftarrow i{+}1$;

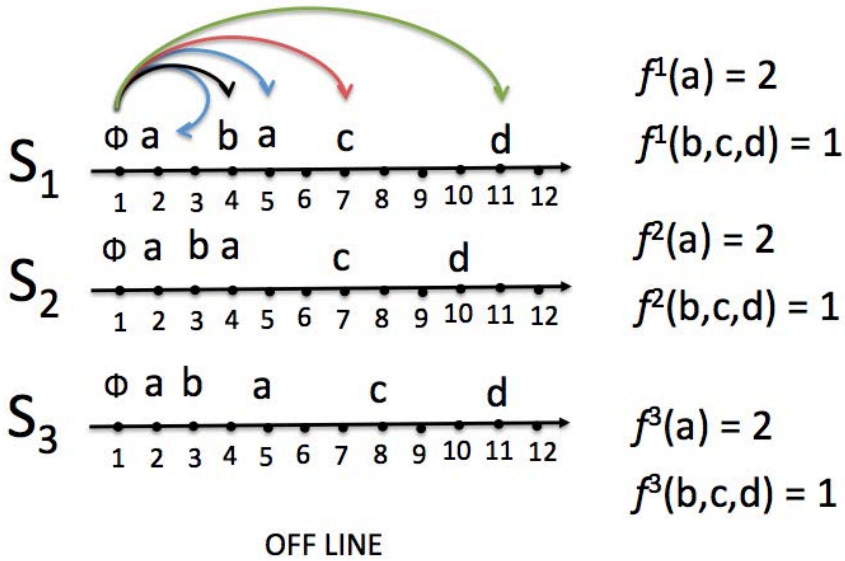


Figure 5.9: Event Φ

Nevertheless **Algorithm 4** is now run instead of **Algorithm 2**.

Table 5.2 presents the results of frequent chronicles obtained from the event sequences and the temporal restrictions.

| Frec=1 | Frec=2 |
|---|---|
| a[6 ,6]d | a[-2 ,2]b |
| c[2,6]d | a[2,6]c |
| a[8,9]d | a[6,9]d |
| a[5,6]c | — |
| a[2,3]c | — |
| b[3,5]c | — |
| b[7,8]d | — |

Table 5.2: Frequent chronicles, Algorithm 4

The tree roots are given in Fig. 5.10 and Fig. 5.11 represents the frequent chronicles with frequency 1. Let us notice that with the integration of the temporal restrictions coming from the expert, only four chronicles with frequency 1 are obtained instead of the 8 previously obtained. Fig. 5.12 represents the frequent chronicle with frequency 2.

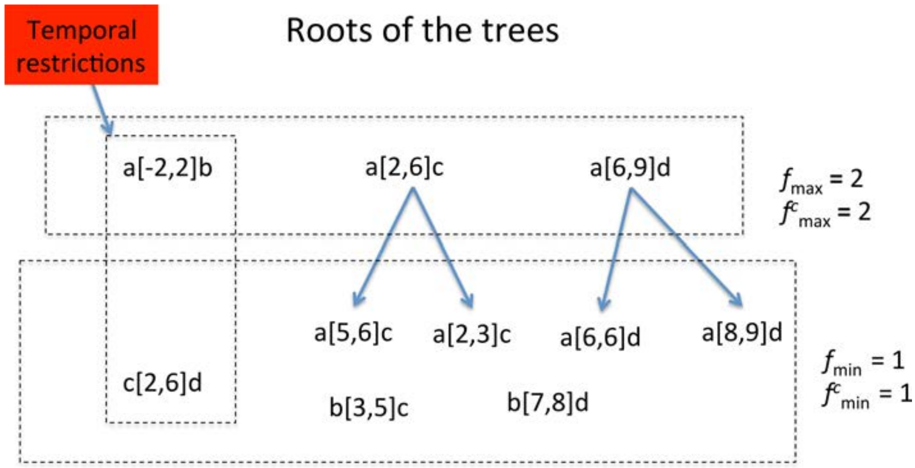


Figure 5.10: Tree roots of the oven charge system using the extended *HCDAM*

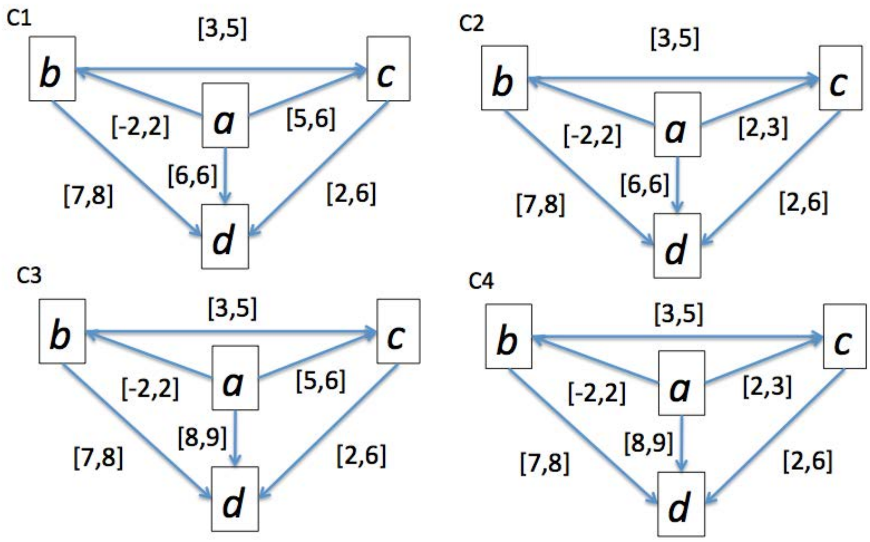


Figure 5.11: Chronicles C1 to C4 with frequency 1 using the extended *HCDAM*

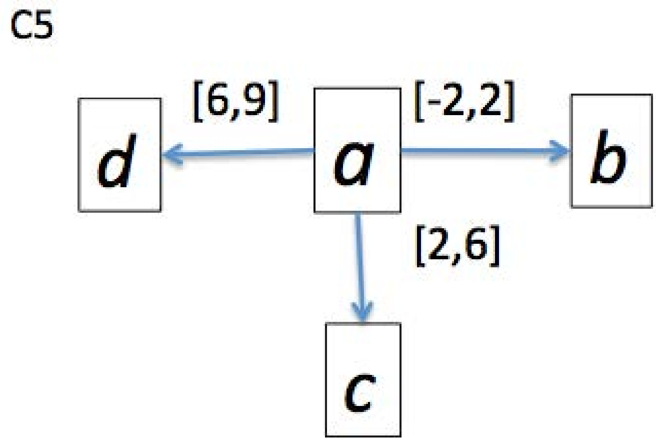


Figure 5.12: Chronicle C5 with frequency 2 using the extended $HCDAM$

Let us now introduce event information by adding the virtual initial event type Φ. Rerunning the chronicle learning algorithm now leads to a unique chronicle, as shown in Fig. 5.13, therefore reducing the number of chronicles by 90%.

## 5.4 Conclusion

This chapter first presented a state of the art of the chronicle learning theory and motivated the choice of the $HCDAM$ (Heuristic Chronicle Discovery Algorithm Modified) algorithm from [96]. This algorithm was presented, exhibiting three algorithmic phases that were illustrated with an oven charge process. The second part of the chapter proposed two improvements of $HCDAM$ related to the integration of expert knowledge. The first improvement relies on temporal restrictions that may be known by the expert and can therefore avoid to build the corresponding tree root. The second improvement proposes a way to control the frequency of every single event by the introduction of a virtual initial event. Interestingly, this does not require any change in the HCDAM algorithm.

It is important to notice that the impact of the two improvements is quite significant on the number of learned chronicles that is drastically reduced as well as on the conservatism of the learned chronicles. They hence improve the algorithm complexity, they ease the interpretation of the results and make the use of the learned chronicles simpler in practice.

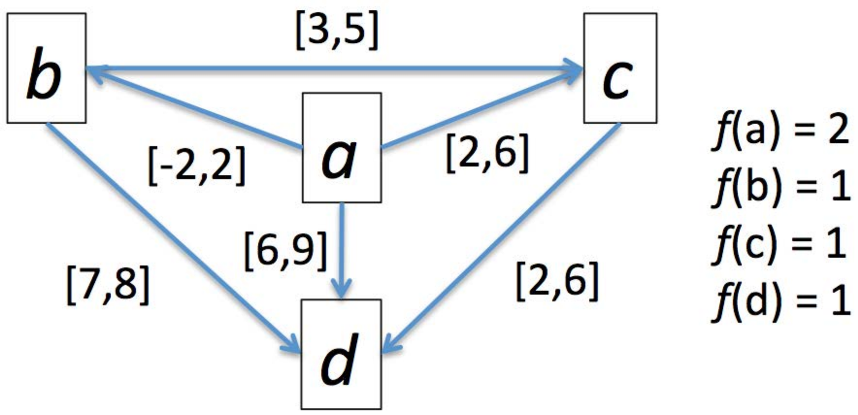


Figure 5.13: Unique chronicle of the oven charge system

# Chapter 6

# A chronicle based approach for Alarm Management

## 6.1 Overview of the Chronicle Based Alarm Management

As explained in the previous sections, 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 thesis we propose to use *chronicles* to represent the plant situations under interest and to integrate a diagnosis step based on chronicle recognition in the global schema of alarm management (see Chapter 1). The chronicle design is a tricky step and we have proposed in Chapter 5 to use a learning technique and more precisely the $HCDAM$ algorithm to generate the different chronicles that capture the process evolutions. This approach of alarm management constitutes what we have called the *Chronicle Based Alarm Management* (CBAM). The principle of $CBAM$ is to consider several process situations (normal or abnormal) during startup and shutdown stages and to model each of these situations through a learned chronicle. For this, given a situation one want to model, the $HCDAM$ algorithm is fed by a set of event sequences issued from the process and associated to the situation. The global objective of $CBAM$ is to generate a chronicle database on which a diagnosis process based on chronicle recognition is then performed. In this approach the simultaneous occurrence of events is not considered.

The Chronicle Based Alarm Management (CBAM) relies then on three main steps resumed as below:

1. STEP 1: Event type identification: The aim is to determine the event types that define the chronicles. For this step, information from the standard operating procedures and from the evolution of the continuous variables are exploited.

2. STEP 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 used by the learning algorithm $HCDAM$. 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. The representative event sequences are then verified using the hybrid modeling of the system and the hybrid causal graphs.

3. STEP 3: Chronicle database construction: For each scenario, the representative event sequences and temporal restrictions given by expert are considered to learn chronicles using the extended algorithm HCDAM. The set of chronicles learned for each scenario and each process element constitutes the chronicle database.

The methodology proposed in this thesis merges different techniques to take the hybrid features of the system into account. Notably, the information about the procedural actions and the continuous variables behavior are considered to extract the representative event sequences. Another important aspect of this work is the dynamic alarm management. Indeed, most of the time the alarm is assumed to be a static indicator (see chapter 2). In this proposal an alarm is an event with an occurrence date and the alarm flow is formally modeled by a chronicle [109],[110].

The different steps of $CBAM$ are described more deeply in the next sections.

### 6.1.1 Step 1: Event type identification

There are several types of events that occur during the transitional stages in an industrial process. The goal of this step is to identify the most important event types that represent all the highly significant evolutions of the process. The event type identification is a crucial step to have a good characterization of the situations that will be then captured by chronicles.

The set of event types $E$ considered in the chronicles is defined by $E = \Sigma \cup \Sigma^c$ where:

- $\Sigma$ is the set of event types associated to the procedure actions in a startup or shutdown stages. Procedure actions such as *Open valve*, *close valve*, *turn on the pump* and *turn off the pump* are some actions that can be taken as event types.

- $\Sigma^c$ is the set of event types associated to the behavior of the continuous process variables. These event types related to the behavior of the continuous variables are the signals obtained when the variable has passed its limits (alarms) of high or low. They are defined through a **qualitative abstraction** of the continuous behavior of the system.

### Qualitative abstraction of continuous behavior

As presented in Chapter 3 the process is modeled by an hybrid causal model. 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 $f_{qual} : D_o(v_i) \rightarrow D_o(V_i)$ that maps the continuous values of variable $v_i$ to ranges defined by limit values (High $H_i$ and Low $L_i$), alarm values determined by experts in the alarm system.

$$f(v_i)_{qual} = \begin{cases} V_i^H & if \quad v_i \geq H_i \\ V_i^M & if \quad L_i < v_i < H_i \\ V_i^L & if \quad v_i \leq L_i \end{cases} \tag{6.1}$$

The behavior of these qualitative variables is represented by the automaton $G_{V_i} = (V_Q, \Sigma^c, \gamma)$ in Fig. 6.1 where $V_Q$ is the set of the possible qualitative states ($V_i^L : Low$, $V_i^M : Medium$, $V_i^H : High$) of the continuous variable $v_i$, $\Sigma^c$ 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 $f_{V_Q \rightarrow \sigma}$

$$\begin{aligned} & f_{V_Q \rightarrow \sigma} : V_Q \times \gamma(V_Q, \Sigma^c) \rightarrow \Sigma^c \\ & \forall V_i \in V_Q, (V_i^n, V_i^m) \rightarrow \begin{cases} L(v_i) & if \quad V_i^L \rightarrow V_i^M \\ l(v_i) & if \quad V_i^M \rightarrow V_i^L \\ H(v_i) & if \quad V_i^M \rightarrow V_i^H \\ h(v_i) & if \quad V_i^H \rightarrow V_i^M \end{cases} \\ & V_i^n, V_i^m \in \{V_i^L, V_i^M, V_i^H\} \end{aligned} \tag{6.2}$$

$$\Sigma^c = \bigcup_{v_i \in \vartheta} \{L(v_i), l(v_i), H(v_i), h(v_i)\} \tag{6.3}$$

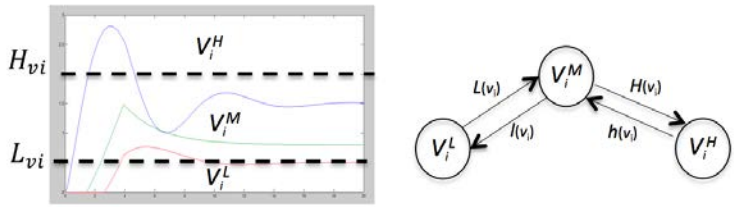


Figure 6.1: Behavior of the qualitative variables

Let us come back to the oven charge system example (see Fig. 6.2). The system is composed of two components. One passive component: the Oven (**OV**), and one active component: the heater (**R**). The continuous variables are the intensity or current ($Int$) that passes for the heater and the temperature inside the oven ($T$). In this example, the unique variable from which we obtain event types is the temperature $T$. The event types related with the procedural actions are $\Sigma = \{a, b, c, o\}$. The procedural actions to charge the oven are represented when the operators introduce the products "a" and "b" into the oven. The other procedural actions are related to the activation of the heater.

- The event type $a$ indicates that the product "a" has passed
- The event type $b$ indicates that the product "b" has passed.
- The event type $c$ expresses that the heaters of the oven be ON.
- The event type $o$ expresses that the heaters of the oven be OFF.

$\Sigma^c = \{d\}$ is related to the behavior of the continuous variable of temperature.

- The event type $d$ expresses that the temperature into the oven has arrived to its high limit.This limit is specified by the expertise and represents a high limit of temperature into the oven around 100ºC.

### 6.1.2 Step 2: Event sequence generation

The event sequences are obtained by simulation based on the knowledge of the procedural actions. Although the event types related with the procedural actions are the first events that occur, sometimes another event types can take place before.

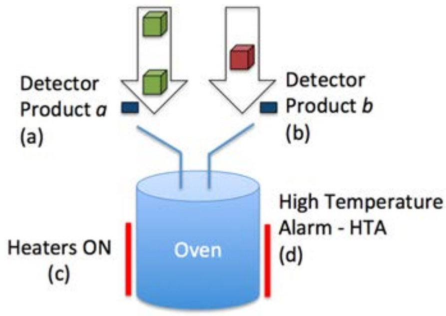


Figure 6.2: Oven charge system

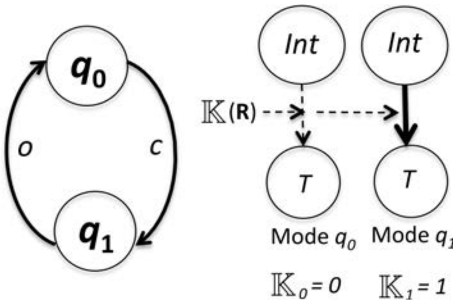


Figure 6.3: Startup stage of the oven charge system: underlying DES and Causal System Description

Let us consider again the charge oven system. This system has two modes of operation: $q_0$ when the heater of the oven is OFF and $q_1$ when the heater is ON. Figure . 6.3) gives the underlying DES associated to the charge oven system (on the left) and the causal graphs of each mode of operation according to the configuration of the heater active component R (on the right). Before the heater in the oven is turned on, three pieces must be charged. First, one type of product $a$, followed by $b$ and finish with another $a$. This order is defined by expertise. And the evolution of the temperature is determined by simulation of its transfer function.

In Fig. 6.4 are expressed the three representative event sequences for a normal startup of the charge oven system. The event type evolution is represented on a time line for each event sequence; notice that the three initial event type occurrences correspond to the pass of the products $a$ and $b$. After that, when the event type $c$ occurs, the system passes to the mode $q_1$ finishing with the occurrence of $d$. Furthermore, the variable of temperature depends on the heater activation, so the occurrence of $d$ depends of that $c$ had occurred before.

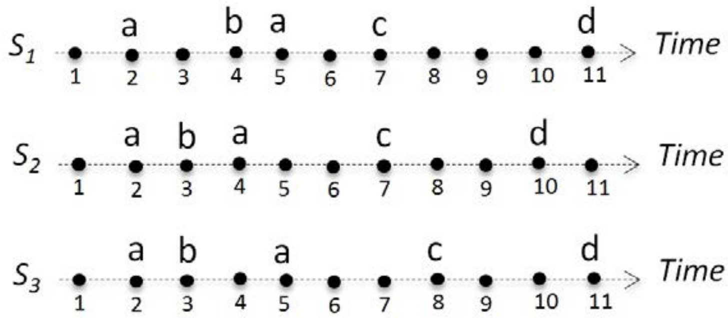


Figure 6.4: Representative event sequences of the oven charge system

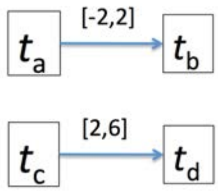


Figure 6.5: Temporal restrictions of the charge oven system

Once generated by simulation, the representative event sequences are verified using the hybrid causal model to check the relevance of the performed simulations. For example, the sequence $S_1$ initiates with the event types $a$, $b$ and $a$ according to the standard procedure. After that, the event type $c$ occurs and the system passes from the operation mode $q_0$ to the mode of operation $q_1$. When the system is in the mode $q_1$, the causal relationship between $Int$ (current) and $T$ (temperature) is activated (see Fig. 6.3). An event type $d$ is then expected as the mode $q_1$ is activated. When the event type $d$ occurs, this event sequence is verified. For the other sequences, the same procedure is applied.

The expertise knowledge plays an important role in the determination of these representative event sequences. This knowledge also can be exposed as the identification of the most dangerous situations in the process; for example, using *HAZOP*, *fault tree* or *bow tie analysis* which were explained in the Section 2. For the charge oven example the expertise knowledge is represented by temporal restrictions, which are shown in Fig. 6.5

### 6.1.3 Step 3: Chronicle database construction

From the representative event sequences and the partial temporal restrictions in each scenario, the chronicle database is constructed using the extended algorithm $HCDAM$.

#### 6.2.3.1 Construction of the 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 $Ar_m$, $m = \{1,2,...n\}$ has $K \in N$ operational modes (e.g. startup, shutdown) noted $O_i$, $i = \{1,2,...K\}$. 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 $\Delta = f_0 \cup \Delta f$, where $f_0$ is the faultless behavior. 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. The set of chronicles $\{C_{ij}^m\}$ for each area $Ar_m$ is presented in the matrix below, where the rows represent the operating modes (i.e. $O_1 : Startup$, $O_2 : Shutdown$, etc) and the columns the normal and abnormal situations. For a given area $m$, a learned chronicle $C_{ij}^m$ is associated to each couple $(O_i, l_j)$ where $l_j \in \Delta$: When $l_j$=$f_0$, 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 $f_i$.

$$C_{Ar_m} = \begin{array}{c c} & \begin{array}{cccc} f_0 & f_1 & f_2 & \dots f_r \end{array} \\ \begin{array}{c} O_1 \\ O_2 \\ \dots \\ O_k \end{array} & \begin{bmatrix} C_{10}^m & C_{11}^m & C_{12}^m & \dots C_{1r}^m \\ C_{20}^m & C_{21}^m & C_{22}^m & \dots C_{2r}^m \\ \dots & \dots & \dots & \dots \\ C_{k0}^m & C_{k1}^m & C_{k2}^m & \dots C_{kr}^m \end{bmatrix} \end{array} \tag{6.4}$$

This chronicle database, is to be submitted to a chronicle recognition system that identifies 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.

## 6.2 Conclusion

In this chapter our proposal of a Chronicle Based Alarm Management ($CBAM$) was presented. This methodology aims to integrate in the alarm management process a diagnosis process based on chronicle recognition. For this $CBAM$ relies on three main steps allowing at the end the construction of a chronicle database. The first step is the event type identification, the second is the event sequences generation and the last step is the chronicle database construction using the Heuristic Chronicle Discovery Algorithm Modified. $CBAM$ is based on an hybrid modeling of the system to integrate information issued from procedural actions and information issued from the continuous variable behaviors, and to capture at a discrete level the process evolutions in terms of chronicles.

# Chapter 7

# Case Studies

## 7.1 Introduction

The Cartagena Refinery is located in the Industrial Zone of Mamonal, one of the most important in Colombia and Latin America. This is a core of companies mostly chemical, petrochemical and services, which were installed in the area after the refinery, visualizing its enormous potential. The industrial area of Mamonal, located half an hour from Cartagena, today has more than 60 plants, several of which belong to Companies listed among the 100 largest in the country: GRC, Exxon Mobil, Texaco, Colombian Petrochemical and Propilco. Currently, the refinery is composed of six units or principal areas, which are described below:

- **Storage of Raw Materials and Products**: This unit is responsible for receiving crude oil through the Coveñas - Cartagena Pipeline, supplying the charge to the Combined Distillation Unit, and handling the product flows from the different units. The purpose of the unit is to store them within specifications and to ensure the necessary inventory to the normal fuel supply in the northern part of the country.

- **Unit of Crude**: The first step in petroleum refinement is the separation of the oil into several fractions or "cuts" using the atmospheric distillation towers, the vacuum tower, and the vacuum oven. The fractions or cuts obtained during this process are obtained thanks to the different ranges of boiling. The ranges can be classified based on a decrease in volatility of gases: light distillates, medium distillates, liquid gases, and waste.

  The Crude Unit has a design capacity of 78 kB/d (Kilo barrels per day) of crude oil through a combined distillation process. In the first stage (the atmospheric distillation) the crude is subjected to be heated in furnaces. Subsequently, it is divided in the hot tower, where the atmospheric gas oil and ACPM are obtained. The gases from above pass to the atmospheric tower to continue the distillation and obtain kerosene, turbocharged fuel, gasoline, and gas. The bottoms of the hot tower are called reduced crude.

The second stage of the process is vacuum distillation. At this stage, the reduced oil passes through some kilns where they are heated and then fractioned in the vacuum tower. After that, this is combined with the oil from the atmospheric hot tower on the vacuum oven. In this way, the light and heavy gas oils are recovered. The product of funds or heavy residue is sent as a load to the Visor Rearing Unit for further use. The gas oils produced with crude serve as raw material in the Catalytic Cracking Unit.

- **Viscous-reducer Unit**: The unit has the capacity to process 25 KB/d of empty bottoms from the crude plant. The viscous reduction is the process by which a heavy charge composed of heavy chain heavy hydrocarbons is partially decomposed into other hydrocarbon chains of lower and higher molecular weight (condensation) relative to the original charge. This is achieved by a thermal cracking reaction with secondary condensation reactions. This process receives its energy from the viscous breaking furnace. Its result is the conversion levels that determine the passage of the charge to fractions of naphtha, kerosene, and gas oil (distillates).

- **Cracking unit**: Cracking is a refining process whereby several gas oils are cracked into simpler hydro carbon compounds through the use of extreme heat, pressure, and exposure to catalytic chemicals. Essentially, this process changes the long chains of the hydrocarbon molecules (less value) into smaller chains which increase in value to produce high octane types of gasoline, light fuel oils, and olefins rich gases.

- **Industrial Services Unit**: This unit produces the services that the refinery requires for process units such as water, steam, electricity, air, and fuel gas. In the first four, the district is self-sufficient and in the latter, it is complemented by the purchase of natural gas. This area includes the following systems: Water (HTG Hydrostatic Tank Gauging System), Steam, Electricity, Air, and Fuel Gas. Respect to the HTG System, raw water with chemicals is treated to produce water suitable for different uses, such as cooling, steam generation, and human consumption.

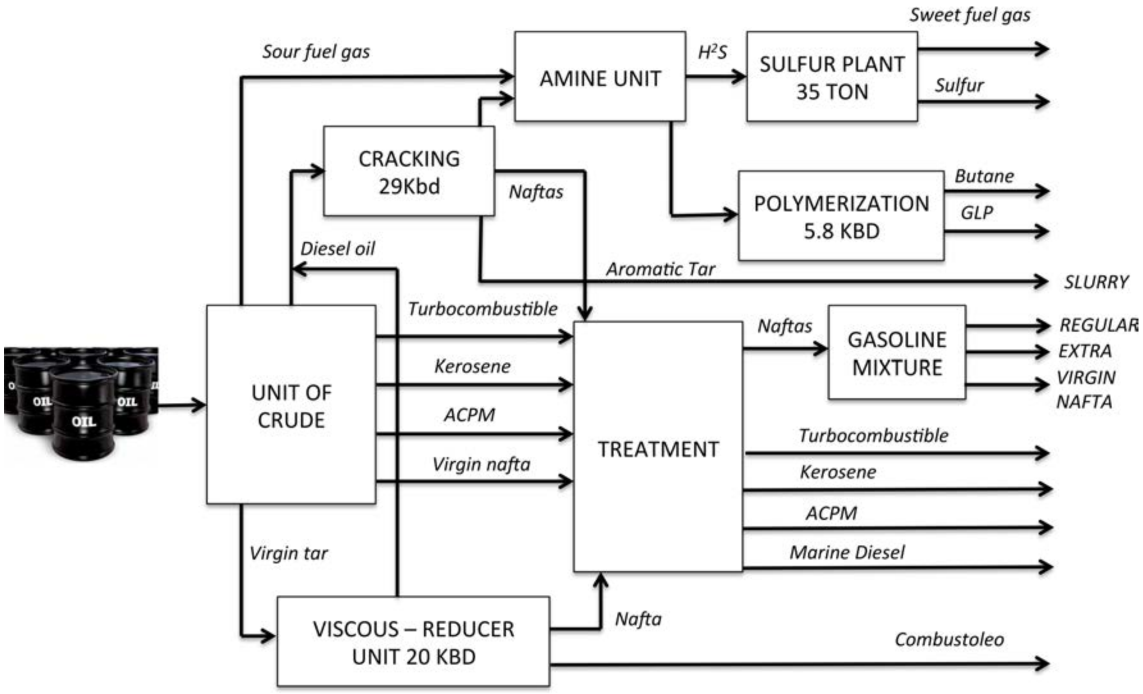


Figure 7.1: General Diagram of the Cartagena Refinery

- **Product and Wastewater Treatment Unit**: The Product and Wastewater Treatment Unit is necessary to remove contaminants from the products and the water used in the processes. Therefore, this area includes Product Treatment and Sewage Treatment. The General Diagram of the Cartagena Refinery is presented in Fig. 7.1.

The project: *Modernization of the Refinería Cartagena* consists of the expansion of the refinery to increase its capacity to 80 KB/d. The modernization will include 14 new units, including crude distillation, hydrostatic tank gauging System, Vacuum oven system, vacuum distillation, coking, hydrocracking, saturated gas, water treatment, power generation and late cooking, all with the aim of improving the value of the product and the system of production. In addition, residual fuel not is produced. Reficar (Refinery of Cartagena S.A) is in charge of the initiative. The company is formed by Ecopetrol and Andean Chemical, a subsidiary of Ecopetrol.

According to the Ecopetrol experts, ones of the most critical process areas are the Vacuum Oven and the HTG system because when the cooling system fails, the Vacuum Oven suffers critical damages. For this reason, we develop the methodology CBAM in these two cases of study, the Hydrostatic Tank Gauging system and the Vacuum Oven.

The standard operating procedure of the refinery is very constrained and specifies the standard procedural actions the operators must execute during the start-up 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 the 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 to 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 6) leading to the construction of a chronicle base. Next sections, detail each of these steps in each case study. Including also a validation of the chronicle for abnormal startup in each case study.

## 7.2 Hydrostatic Tank Gauging System

The Cartagena Refinery in Colombia has been recently enriched with news units and elements, units such as the system of Hydrostatic Tank Gauging (HTG), the atmospheric hot tower, the vacuum tower and the vacuum oven between other elements. Our proposal aims to help the operator to recognize dangerous conditions during the start-up stage of the refinery with modified equipment. The first unit that we analyze the startup and shutdown stages is the unit of water injection, see Fig. 7.2. We can see that the measured continuous variables are the level of the tank $L$, the pressure $Po$ in the pump and the outlet flow $Qo(V2)$ in the valve **V2**. For the startup stage in this process, the initial conditions are that the tank (**TK**) is empty, the valves **V1** and **V2** are closed and the pump **Pu** is off. In this situation, the alarms for low levels in all the continuous variables ($L$, $Po$ and $Qo(V2)$) are active. For the shutdown stage in this process, the initial conditions could be different each one of the others, depending on the situation in that the system is. For example, one condition is that the outlet pressure ($Po$) has passed its high limit activating the alarm $PAH$ (Pressure Alarm High), but the outlet flow ($Qo(V2)$) does not increase over its low limit after that a specific quantity of time units has passed.

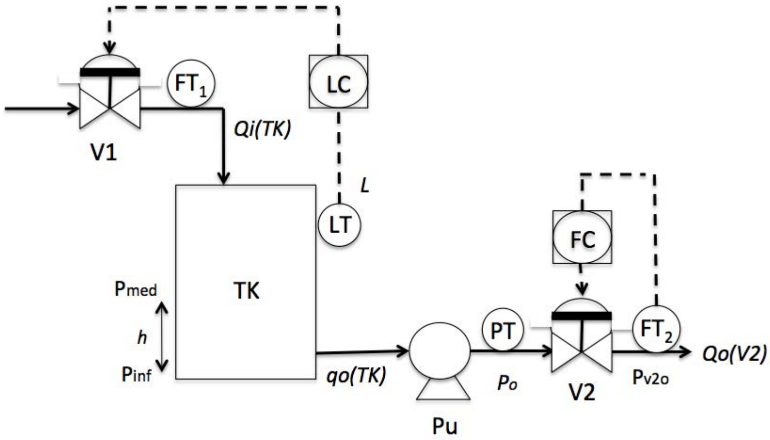


Figure 7.2: Hydrostatic Tank Gauging

### 7.2.1 Hybrid features of the HTG system

This process is a HTG (Hydrostatic Tank Gauging) system composed by the following components:

- Passive component: one tank (**TK**),
- Active components: two normally closed valves (**V1** and **V2**), one pump (**Pu**),
- Sensors: level sensor (**LT**), pressure sensor (**PT**), inflow sensor (**FT1**) and outflow sensor (**FT2**).

Since there are three active components, the HTG system obviously involves hybrid behavior. Modeling the behavior of this hybrid system involves a set of continuous variables and of a set of discrete variables. The continuous variables are level $L$, pressure $Po$, and outflow $Qo(V2)$. On the other hand, the discrete variables are:

- the states of the transition system representing the system operating modes. The HTG has thus $2^3 = 8$ configurations and operating modes denoted $q_0$ to $q_7$ due to the two valves (**V1** and **V2**) each with two possible modes (opened and closed); and the pump (**Pu**) with two possible modes (ON and OFF).
- $V_Q$ the set of qualitative variables values obtained from the behavior of continuous variables as explained Section 6. In this case study, continuous variable domain partitioning has been chosen according to expert knowledge and to limit values specified in standard operating procedures. $V_Q = \{L^L, L^M, L^H\} \cup \{Po^L, Po^M, Po^H\} \cup \{Qo(V2)^L, Qo(V2)^M, Qo(V2)^H\}$

- 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 operation mode. The configuration is defined by the state (opened or closed) of the two valves and the state of the pump. For a normal startup the HTG evolves through the modes $q_0, q_1, q_4, q_5$ and $q_7$. In the mode $q_0$ the two valves are closed and the pump is OFF, then $\mathbb{K}_0 = 0$. When the valve **V1** is opened, the system passes to the mode $q_1$ and $\mathbb{K}_1 = 1$. The system can evolve to $q_4$ if the valve **V1** is opened, then $\mathbb{K}_4 = 4$, or it can evolve to $q_5$ if the pump **Pu** is turned ON, then $\mathbb{K}_5 = 5$. Finally, for $q_7$ both valves are opened and the pump turned ON, then $\mathbb{K}_7 = 7$.

### 7.2.2 Event type identification

In the system HTG of the case of study, the set of event types $\Sigma$ that represent the procedure actions is:

$$\Sigma = \{V1, V2, PuO, v1, v2, PuF, M2A\} \tag{7.1}$$

where $V1$ (resp. $V2$) is for the action that switches the valve **V1** (resp. **V2**) from closed to opened. $v1$ ($v2$) for the action that switches the valve **V1** (resp. **V2**) from opened to closed and $PuO$ (resp. $PuF$) for the action that turns on (resp. off) the pump. The event $M2A$ corresponds to the transition from *manual* to *automatic* operation, closing the control loops. In the reminder we assume that this event is the only unobservable event of the system i.e. $M2A \in \Sigma_{uo}$.

The underlying DES (Discrete Event System) of the HTG system represents the sequence of observable procedure actions for a start-up stage (indicated by the red or green arrows on Fig. 7.3) corresponding to the evolution of the operation modes (i.e $q_0, q_1, q_4, q_5$ and $q_7$). To each operation mode $q_i$ is associated a causal system description to identify the influences between the variables $L$,$Po$ and $Qo(V2)$ see Fig. 7.3. These influences allow determining the event types $\Sigma^c$ occurrence.

$$\Sigma^c = \begin{matrix} \{L_{(L)}, l_{(L)}, H_{(L)}, h_{(L)}, L_{(Po)}, l_{(Po)}, H_{(Po)}, h_{(Po)}, \\ L_{(Qo(V2))}, l_{(Qo(V2))}, H_{(Qo(V2))}, h_{(Qo(V2))}\} \end{matrix} \tag{7.2}$$

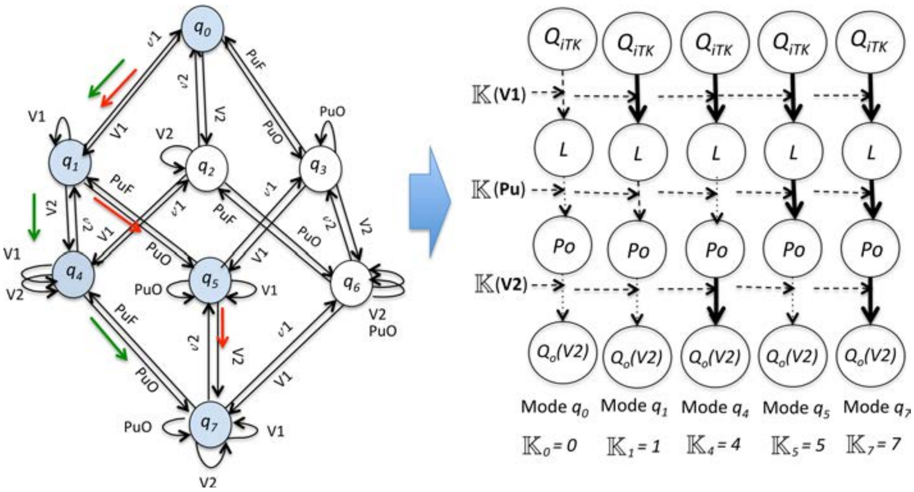


Figure 7.3: Start-up stage of the HTG System: underlying DES and Causal System Description

### 7.2.3 Event sequence generation

From simulations, the behavior of the variables is obtained and the learning event sequences are generated according to the evolution of the system in each scenario. In this manuscript are analyzed three scenarios: Normal startup, Abnormal start-up, and Normal shutdown.

#### 7.2.3.1 Scenario 1, Normal startup

According to the standard procedural actions, the first event type that must occur is $V1$ (*Open* **V1**). After this event type occurrence, the system is in the mode of operation $q_1$ where the variable $L$ increases and the event type $L_{(L)}$ must occur after that the valve **V1** is opened, indicating that the level of the liquid into the tank **TK** has passed the limit of low level. After $L_{(L)}$, the liquid into the tank must arrive to the high limit of the level and the event type $H_{(L)}$ must occur. At this time point, the ordered sequence of event types that has occurred is $V1, L_{(L)}, H_{(L)}$. The high limit of the level into the tank is the condition for continuing the procedure actions *Open* **V2** and *turn on* **Pu** ($V2$ and $PuO$). If the operator opens the valve **V2** first, the system passes to the mode of operation $q_4$, but if the pump **Pu** is turned on first, then the system passes to $q_5$. The duration between the occurrences of event types $V2$ and $PuO$ must be of 1 time unit, leaving the system in the mode of operation $q_4$ or $q_5$. At this time point, the ordered sequence of event types that has occurred must be $V1, L_{(L)}, H_{(L)}, PuO, V2$ or $V1, L_{(L)}, H_{(L)}, V2, PuO$. In the *scenario*$1_a$:

$(V1, L_{(L)}, H_{(L)}, PuO, V2)$, the outlet pressure $(Po)$ of the pump **Pu** increases first of that the outlet flow $(Qo(V2))$. Then, after of $V2$, the pressure $Po$ has passed its limit of low pressure and the event type $L_{(Po)}$ must occur. Passing the high limit of pressure $(H_{(Po)})$ occurs after of $L_{(Po)}$. In the *scenario*$1_b$: $(V1, L_{(L)}, H_{(L)}, V2, PuO)$, the event type $L_{(Po)}$ occurs after of $PuO$. Now, after of $L_{(Po)}$, $L_{(Qo(V2))}$ must occur. After of this the event type $H_{(Po)}$ must occurs. At this time point, the ordered sequence of event types that has occurred must be $V1, L_{(L)}, H_{(L)}, PuO, V2, L_{(Po)}, H_{(Po)}, L_{(Qo(V2))}$ or $V1, L_{(L)}, H_{(L)}, V2, PuO, L_{(Po)}, L_{(Qo(V2))}, H_{(Po)}$. In this situation, the unobservable event type $M2A$ occurs and the control loops are closed, carrying the system to a steady state. We assume that the control loops are closed whereas $L_{(Qo(V2))}$ occurs in the *scenario*$1_a$ or $H_{(Po)}$ in the *scenario*$1_b$. Then, the event type $h_{(Po)}$ indicates that outlet pressure decreases after that the control loops are closed. In the same way, the level of liquid in the tank **TK** decreases from the high limit of level $h_{(L)}$ after that $h_{(Po)}$ occurs. When this event type $h_{(L)}$ occurs, we assume that the startup stage finished correctly and the ordered sequences of event types must be $V1, L_{(L)}, H_{(L)}, PuO, V2, L_{(Po)}, H_{(Po)}, L_{(Qo(V2))}, h_{(Po)}, h_{(L)}$ or $V1, L_{(L)}, H_{(L)}, V2, PuO, L_{(Po)}, L_{(Qo(V2))}, H_{(Po)}, h_{(Po)}, h_{(L)}$. For this scenario, we chose the representative event sequences ($S_1$, $S_2$ and $S_3$) that represent the extreme behaviors with all the possible sequence order of event types.

$S_1 = \langle (V1,1), (L_{(L)},20), (H_{(L)},48), (PuO,50), (V2,51), (L_{(Po)},58), (H_{(Po)},71), (L_{(Qo(V2))},80), (h_{(Po)},106), (h_{(L},180) \rangle$

$S_2 = \langle (V1,1), (L_{(L)},25), (H_{(L)},55), (V2,56), (PuO,57), L_{(Po)},69), (L_{(Qo(V2))},83), (H_{(Po)},91), (h_{(Po)},115), (h_{(L},188) \rangle$

$S_3 = \langle (V1,1), (L_{(L)},31), (H_{(L)},60), (PuO,61), (V2,62), (L_{(Po)},71), (H_{(Po)},85), (L_{(Qo(V2))},91), (h_{(Po)},112), (h_{(L},182) \rangle$

The simulation of a *normal startup* is presented in Fig. 7.4 where we can see the evolution of the variables $L$, $Po$ and $Qo(V2)$. This simulation represents only one possible situation in this scenario related with the sequence $S_1$. The values of the variables on the graph are specified as follows:

- For the variable of level $(L)$ the value of 0 corresponds to 0 meters, each increase of 0.5 (vertical axis) corresponds to 0.5 meters.

- For the variable of pressure $(Po)$ the value of 0 corresponds to 0 PSI, each increase of 0.5 (vertical axis) corresponds to 10 PSI.

- For the variable of outlet flow $(Qo(V2))$ the division of 0 corresponds to 0 lts/s (Liters per second), each increase of 0.5 (vertical axis) corresponds to 1 lts/s.

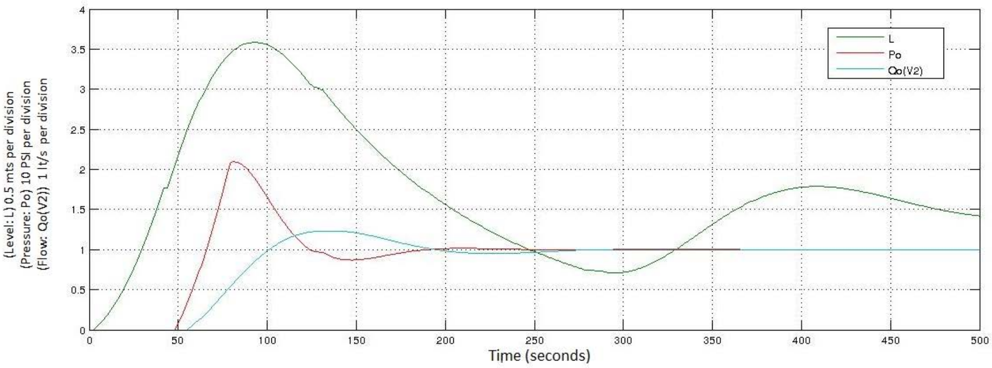


Figure 7.4: Simulation of a normal startup in the HTG system

- The time (horizontal axis) in the graph is expressed in seconds.

At this point, the representative event sequences must be verified using the hybrid causal model. For example, the sequence $S_1$ initiates with the event type $V1$. Then, the system passes to the operation mode $q_1$ and the relationship between $Q_{iTK}$ and $L$ is activated, see Fig. 7.3. We wait for the occurrence of two event types, first of the event type $L_{(L)}$ and second $H_{(L)}$. After that, the standard procedure actions $PuO$ and $V2$ must occur passing the system from the mode of operation $q_1$, in this case first to the mode of operation $q_5$ and after to $q_7$. When the system is in the mode $q_5$, the relationship of the continuous variables $L$ and $Po$ is activated and we wait for the occurrence of the two event types: first, $L_{(Po)}$ and second $H_{(Po)}$ in this order. And, when the system is in the mode $q_7$ the relationship of the continuous variables $Po$ and $Q_{(o(V2))}$ is activated. After that, the event type $L_{(Qo(V2))}$ and the no observable event $M2A$ is activated and the control loops are closed. Concluding this event sequence with the event types $h_{(Po)}$ and $h_{(L)}$ in this order. For the other sequences, the same procedure is applied.

#### 7.2.3.2 Scenario 2, Abnormal start-up

This abnormal situation is related to a failure in the valve **V2**. In this scenario the sequences of event types are similar that the event sequences of a normal startup, until that is detected that the outlet flow in the system does not increase. When the level of liquid in the tank **TK** arrived to its high limit, the ordered sequence of event types that has occurred must be $V1, L_{(L)}, H_{(L)}, PuO, V2$ or $V1, L_{(L)}, H_{(L)}, V2, PuO$. In $scenario2_a : (V1, L_{(L)}, H_{(L)}, PuO, V2)$ the event type $LP$ occurs after $V2$. In $scenario2_b : (V1, L_{(L)}, H_{(L)}, V2, PuO)$ the event type $L_{(Po)}$ occurs after $PuO$. The

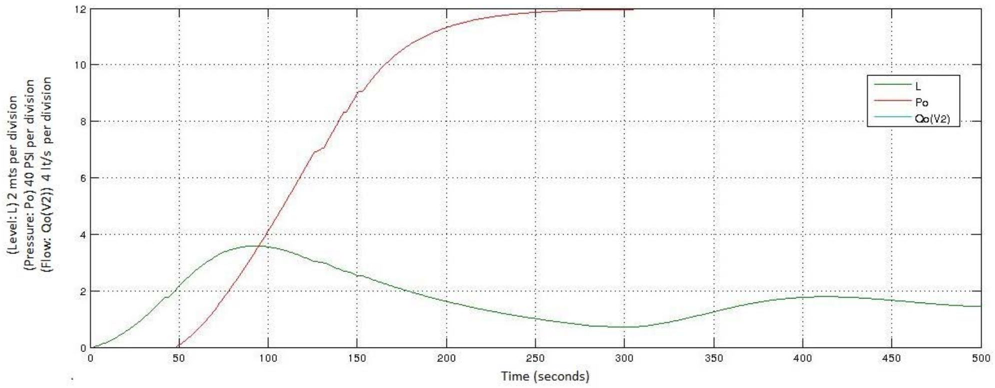


Figure 7.5: Simulation of a startup with a failure in **V2** in the HTG system

event type $H_{(Po)}$ occurs after $L_{(Po)}$. So the ordered sequences of event types must be: $V1, L_{(L)}, H_{(L)}, PuO, V2, L_{(Po)}, H_{(Po)}$ or $V1, L_{(L)}, H_{(L)}, V2, PuO, L_{(Po)}, H_{(Po)}$. For this scenario, we chose the representative event sequences ($S_4$, $S_5$ and $S_6$) that show the extreme behaviors with all the possible sequence order of event types.

$S_4 = \langle (V1,1),(L_{(L)},21),(H_{(L)},48),(PuO,50),(V2,51),(L_{(Po)},60),(H_{(Po)},75)\rangle$

$S_5 = \langle (V1,1),(L_{(L)},25),(H_{(L)},55)(V2,56),(PuO,57),(L_{(Po)},63),(H_{(Po)},78)\rangle$

$S_6 = \langle (V1,1),(L_{(L)},28),(H_{(L)},60),(PuO,61),(V2,62),(L_{(Po)},71),(H_{(Po)},85)\rangle$

The simulation of this *abnormal startup* is presented in Fig. 7.5 where we can see the evolution of the variables $L$ and $Po$. The variable $Q_o(V2)$ does not appear because the valve **V2** had failed. The values of the variables on the graph are specified as follows:

- For the variable of the level ($L$) the value of 0 corresponds to 0 meters, each increase of 2 (vertical axis) corresponds to 2 meters.
- For the variable of pressure ($Po$) the value of 0 corresponds to 0 PSI, each increase of 2 (vertical axis) corresponds to 40 PSI.
- For the variable of outlet flow ($Qo(V2)$) the division of 0 corresponds to 0 lts/s (Liters per second), each increase of 2 (vertical axis) corresponds to 4 lts/s.
- The time (horizontal axis) in the graph is expressed in seconds.

This simulation represents only one possible situation in this scenario related with the pattern sequence $S_4$.

At this point, the representative event sequences must be verified using the hybrid causal model. For example, similar that in a normal startup, the sequence $S_4$ initiates with the event type $V1$. After that, the system passes to the operation mode $q_1$ and the relationship between $Q_{iTK}$ and $L$ is activated, see Fig. 7.3. Therefore, the occurrence of two event types happen, first of the event type $L_{(L)}$ and second $H_{(L)}$. Then, the standard procedure actions $PuO$ and $V2$ must occur passing the system from the mode of operation $q_1$, in this case first to the mode of operation $q_5$ and after to $q_7$. When the system is in the mode $q_5$, the relationship of the continuous variables $L$ and $Po$ is activated and we wait for the occurrence of the two event types: first, $L_{(Po)}$ and second $H_{(Po)}$ in this order. Now, when the system is in the mode $q_7$ the relationship of the continuous variables $Po$ and $Q_{(o(V2))}$ is activated. After that, the event type $L_{(Qo(V2))}$ and the no observable event $M2A$ is activated and the control loops are closed. In this case, the event types $L_{(Qo(V2))}$, $h_{(Po)}$ and $h_{(L)}$ are not activated. This situation is assumed as a failure in the valve $V2$. For the other sequences, the same procedure is applied.

#### 7.2.3.3 Scenario 3, Normal shutdown

After of that, an abnormal start-up situation is detected, a shutdown procedure must be executed. Taking the above situation (*Scenario* 2), it is assumed that later of the abnormal startup confirmed, the standard procedure actions $v1$, $v2$, and $PuF$ must be developed. For this scenario, we chose the representative event sequences ($S_7$, $S_8$ and $S_9$) that represent the extreme behaviors with all the possible sequence order of event types.

$\mathrm{S}_7 = \langle (V1,1),(L_{(L)},20),(H_{(L)},48),(PuO,50),(V2,51),(L_{(Po)},60),(H_{(Po)},75),(PuF,77),(v1,78),(v2,79),(h_{(L)},190),(h_{(Po)},195),(l_{(Po)},240)\rangle$

$\mathrm{S}_8 = \langle (V1,1),(L_{(L)},20),(H_{(L)},48)(V2,51),(PuO,52),(L_{(Po)},63),(H_{(Po)},78),(PuF,79),(v2,81),(v1,82),((h_{(Po)},188),(h_{(L)},200)(l_{(Po)},250)\rangle$

$\mathrm{S}_9 = \langle (V1,1),(L_{(L)},28),(H_{(L)},59),(PuO,61),(V2,63),(L_{(Po)},70),(H_{(Po)},84),(PuF,85),(v1,86),(v2,87),((h_{(Po)},193),(h_{(L)},198)((l_{(Po)},231)\rangle$

The simulation of this *normal shutdown* is presented in Fig. 7.6 where we can see the evolution of the variables $L$, $Po$ and $Q_o(V2)$. The values of the variables are specified as follows:

- For the variable of the level ($L$) the value of 0 corresponds to 0 meters, each increase of 1 (vertical axis) corresponds to 1 meters.
- For the variable of pressure ($Po$) the value of 0 corresponds to 0 PSI, each increase of 1 (vertical axis) corresponds to 20 PSI.

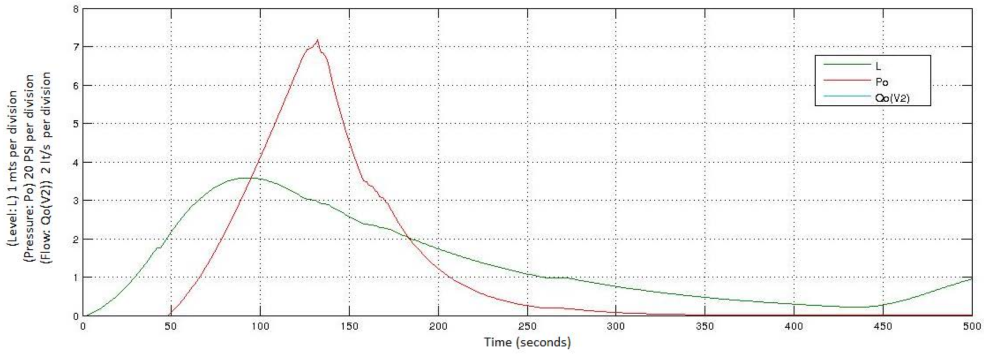


Figure 7.6: Normal shutdown in the HTG system

- For the variable of outlet flow ($Qo(V2)$) the division of 0 corresponds to 0 lts/s (Liters per second), each increase of 1 (vertical axis) corresponds to 2 lts/s.

- The time (horizontal axis) in the graph is expressed in seconds.

This simulation represents only one possible situation in this scenario related with the representative sequence $S_7$. The procedure evaluation of this event sequences is similar that the procedure developed in the other scenarios. In this scenario, the event types $v1$, $v2$ and $PuF$ are involved in the shutdown procedure.

### 7.2.4 Chronicle database construction

This chronicle database is to be submitted to a chronicle recognition system that identifies 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. In the following subsection are presented three chronicles ($C^1_{10}$, $C^1_{11}$ and $C^1_{20}$) of the set of chronicles of the HTG (Hydrostatic Tank Gauging) system i.e Area $Ar_1$ of the whole system. $C^1_{10}$ is a chronicle describing the normal start-up stage of the HTG, $C^1_{11}$ is associated with failure behavior of type $f_1$ during a startup stage and $C^1_{20}$ corresponds to a normal shutdown.

#### 7.2.4.1 Scenario 1, Normal start-up

For this scenario, we have the following temporal restrictions that represent the expert knowledge which is used in the extended version of the $HCDAM$.

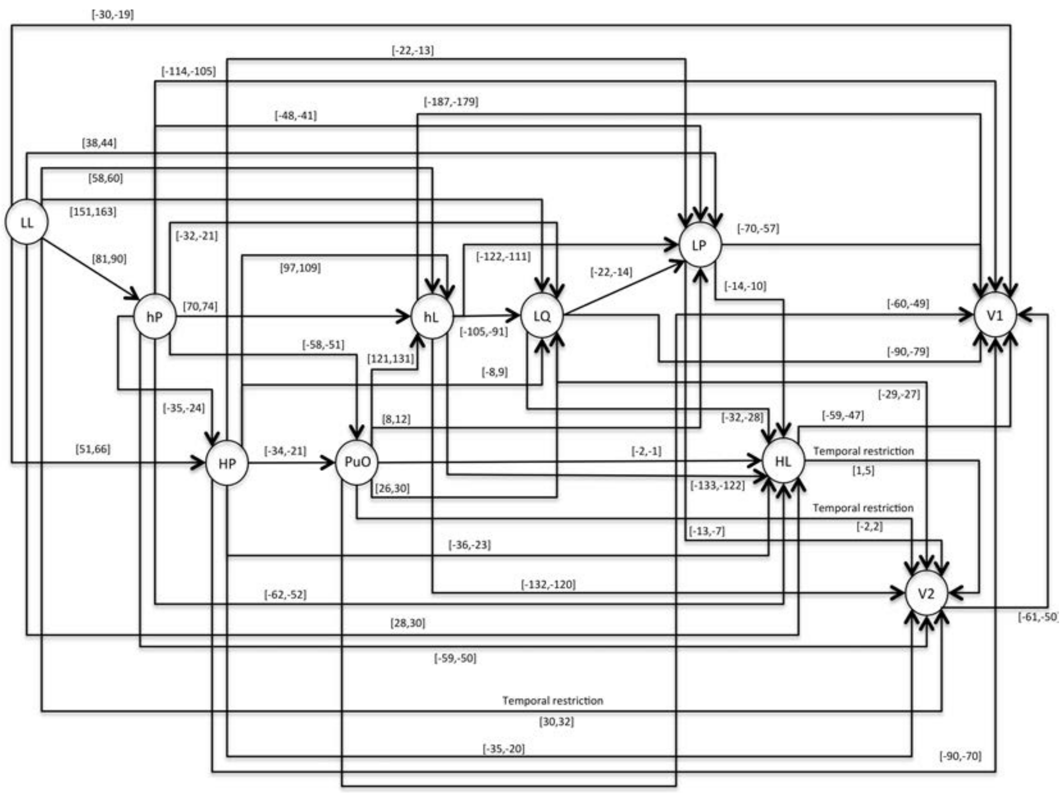


Figure 7.7: Directed graph ($\mathcal{G}$) of the chronicle $C_{01}^{1}$

$TR_{LL,V2}$=$LL[30,32]V2$, this temporal restriction expresses that the valve **V2** is opened between 30 and 32-time units after the low limit level ($LL$) into the tank happens.

$TR_{PuO,V2}$=$PuO[-2,2]V2$, this temporal restriction indicates that the valve **V2**($V2$) can be opened in between 2 time units before the pump **Pu** ($PuO$) is turned on or after $PuO$ occurs.

$TR_{HL,V2}$=$HL[1,5]V2$, this temporal restriction expresses that the valve **V2** is opened between 1 and 5-time units after the high limit level ($HL$) into the tank happens.

The directed graph of chronicle $C_{10}^{1}$ that resulted using the algorithm $HCDAM$ is presented in Fig. 7.7. The learning event sequences used are $S_1$, $S_2$ and $S_3$ generated in subsection 7.2.3.

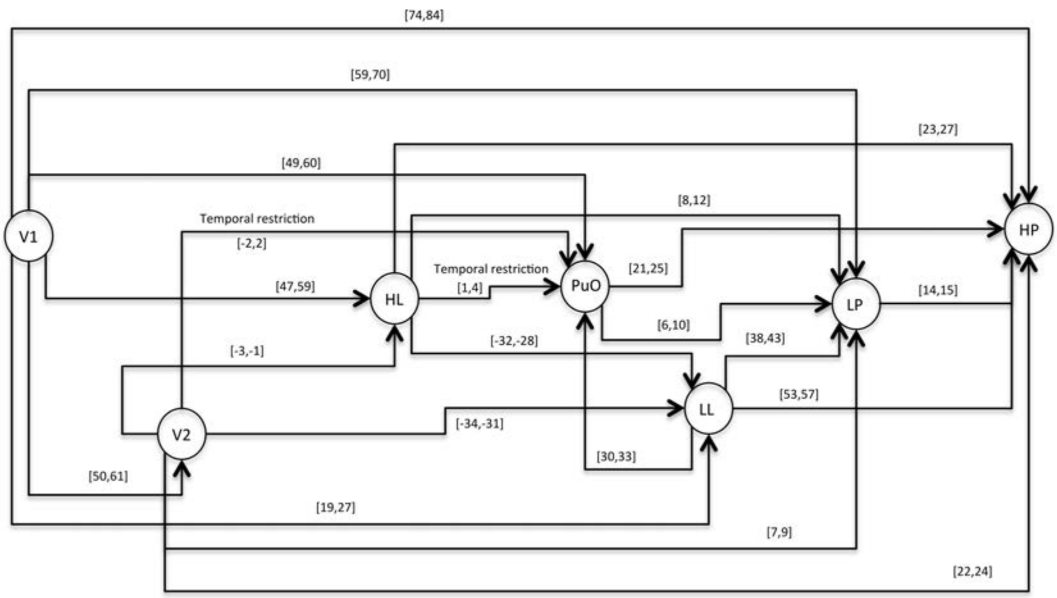


Figure 7.8: Directed graph ($\mathcal{G}$) of the chronicle $C_{11}^{1}$

### 7.2.4.2 Scenario 2, Abnormal start-up

For this scenario we have the following temporal restrictions that represent the expert knowledge which is used in the extended version of the $HCDAM$.

$TR_{PuO,V2}$=$PuO[-2,2]V2$, this temporal restriction indicates that the valve **V2**($V2$) can be opened in between 2 time units before the pump **Pu** ($PuO$) is turned ON, or after $PuO$ occurs.

$TR_{HL,PuO}$=$HL[1,4]PuO$, this temporal restriction expresses that the pump **Pu** ($PuO$) is turned ON between 1 and 4-time units after the high limit level ($HL$) into the tank happens.

The directed graph of the chronicle $C_{11}^{1}$ that resulted using the algorithm $HCDAM$ is presented in Fig. 7.8. The learning event sequences used are $S_4$, $S_5$ and $S_6$ generated in subsection 7.2.3.

### 7.2.4.3 Scenario 3, Normal shutdown

For this scenario we have the following temporal restrictions that represent the expert knowledge which is used in the extended version of the $HCDAM$.

$TR_{PuO,V2}$=$PuO[-3,4]V2$, this temporal restriction indicates that the valve **V2**($V2$) can be opened from 3 time units before that the pump **Pu** ($PuO$) is turned ON to 4 time units after $PuO$ occurs.

$TR_{HP,PuF}$=$HP[2,6]PuF$, this temporal restriction expresses that the pump **Pu** ($PuF$) is turned OFF between 2 and 6-time units after the high limit ($HP$) of the pressure $Po$ happens. The directed graph of chronicle $C_{20}^{1}$ that resulted using the algorithm $HCDAM$ is presented in Fig. 7.9. The learning event sequences used are $S_7$, $S_8$ and $S_9$ generated in subsection 7.2.3.

### 7.2.5 Validation

This section presents the evaluation of the chronicle $C_{11}^{1}$ that represents the temporal pattern for an abnormal startup in the HTG system. The sequence of evaluation is presented below:

$\mathrm{S}_{\mathrm{eval}} = \langle (V1,1),(L_{(L)},26),(H_{(L)},58),(PuO,60),(V2,62),(L_{(Po)},70),(H_{(Po)},85)\rangle$.

Fig. 7.10 to Fig. 7.16 present the recognition process of the chronicle, and the generation of one SUPER ALARM.

The new concept of SUPER ALARM is proposed in this thesis, a concept which corresponds to one "superior alarm" giving relevant information to the operators after a diagnosis process, increasing the reliability of this protective layer, (see Chapter 2).

## 7.3 Vacuum oven system

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, T3**, respectively, and then each coil is divided into two coils. The operator controls these flow with the valves **V1, V2**. Now, the control of temperature inside the oven initiates 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 **F1, F2, F3**.

### 7.3.1 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

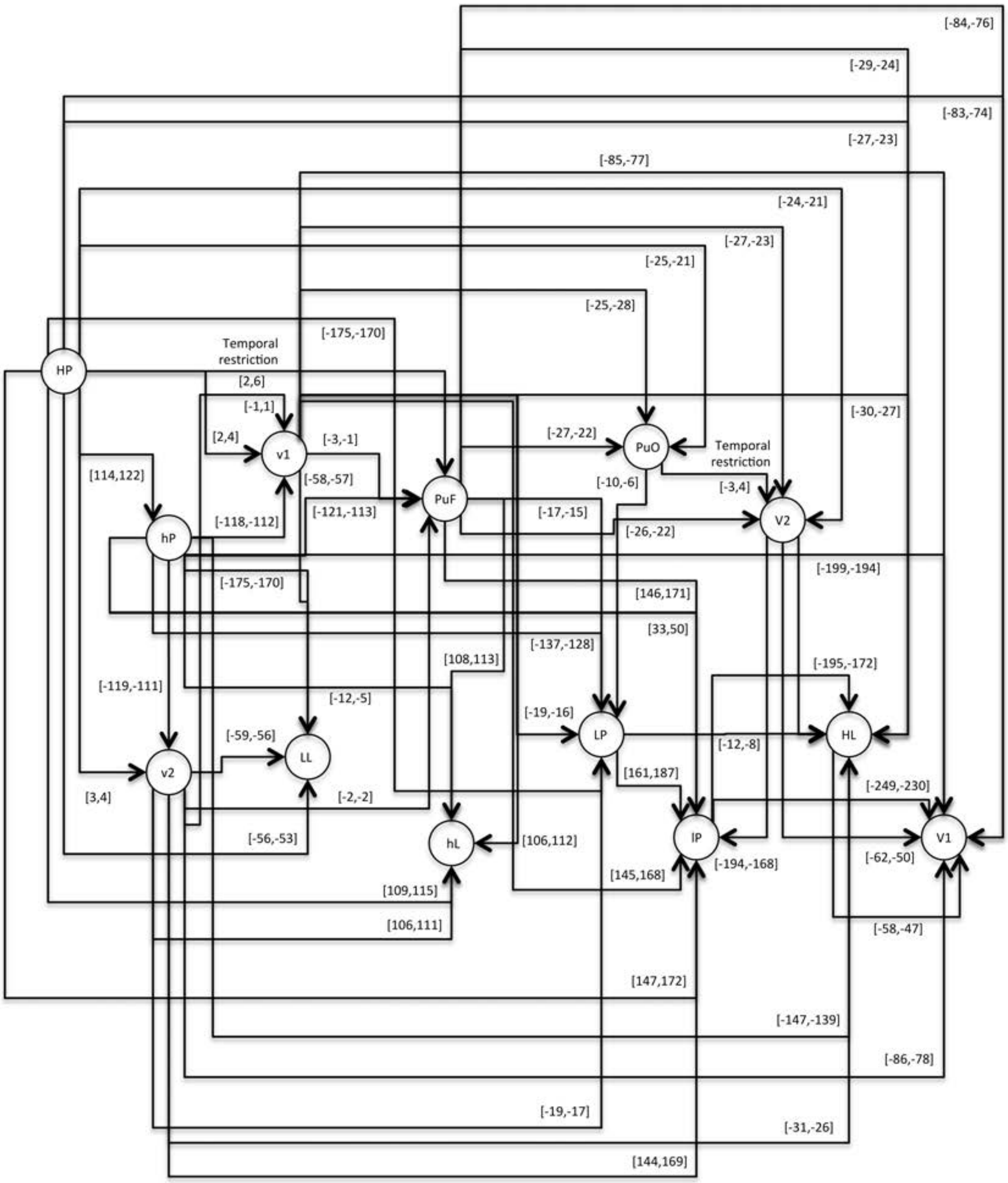


Figure 7.9: Directed graph ($\mathcal{G}$) of the chronicle $C^1_{02}$

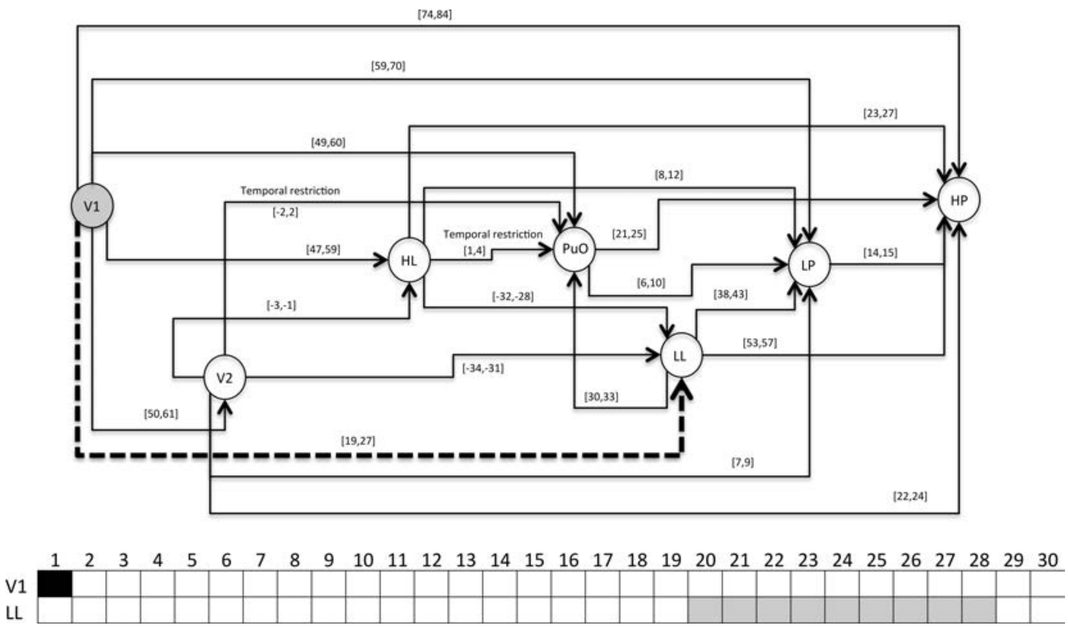


Figure 7.10: Activation of $V1$ at 1

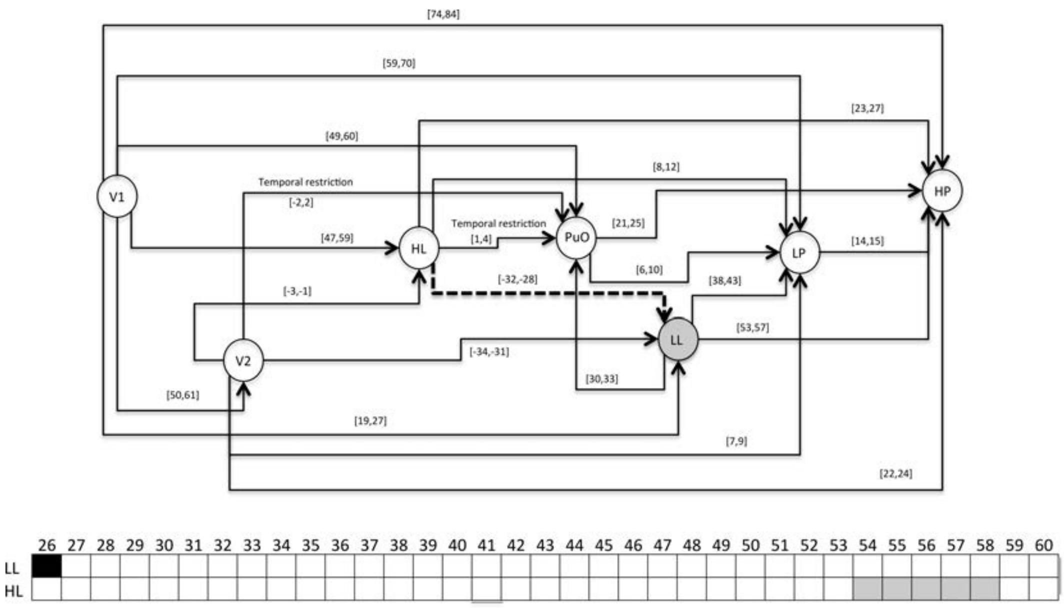


Figure 7.11: Activation of $LL$ at 26

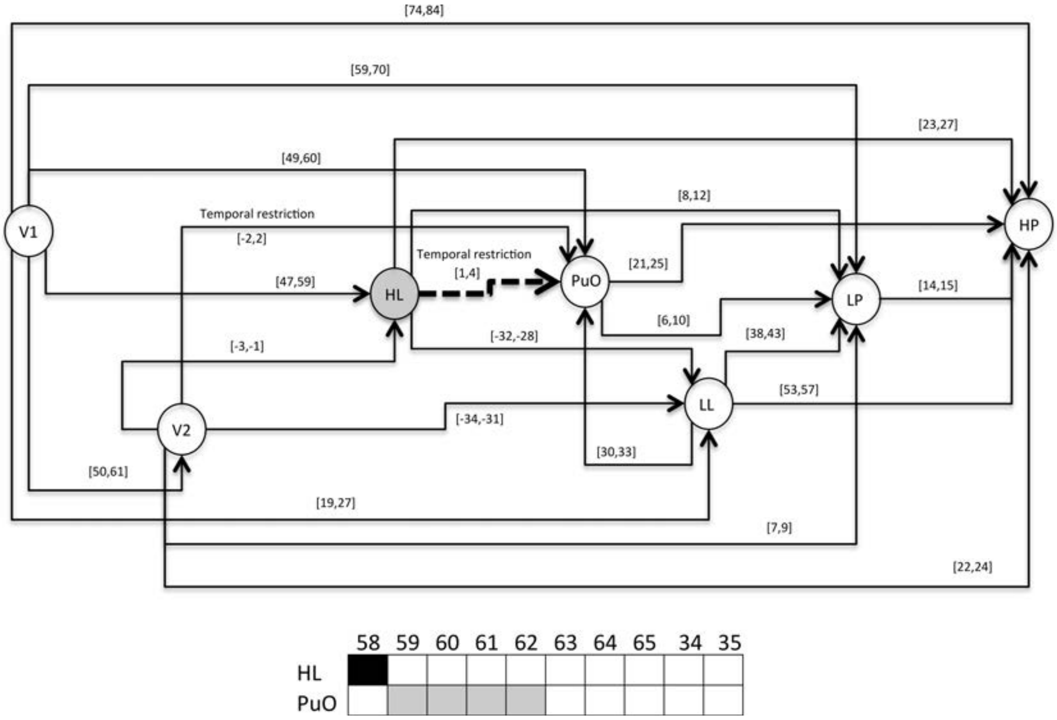


Figure 7.12: Activation of $HL$ at 58

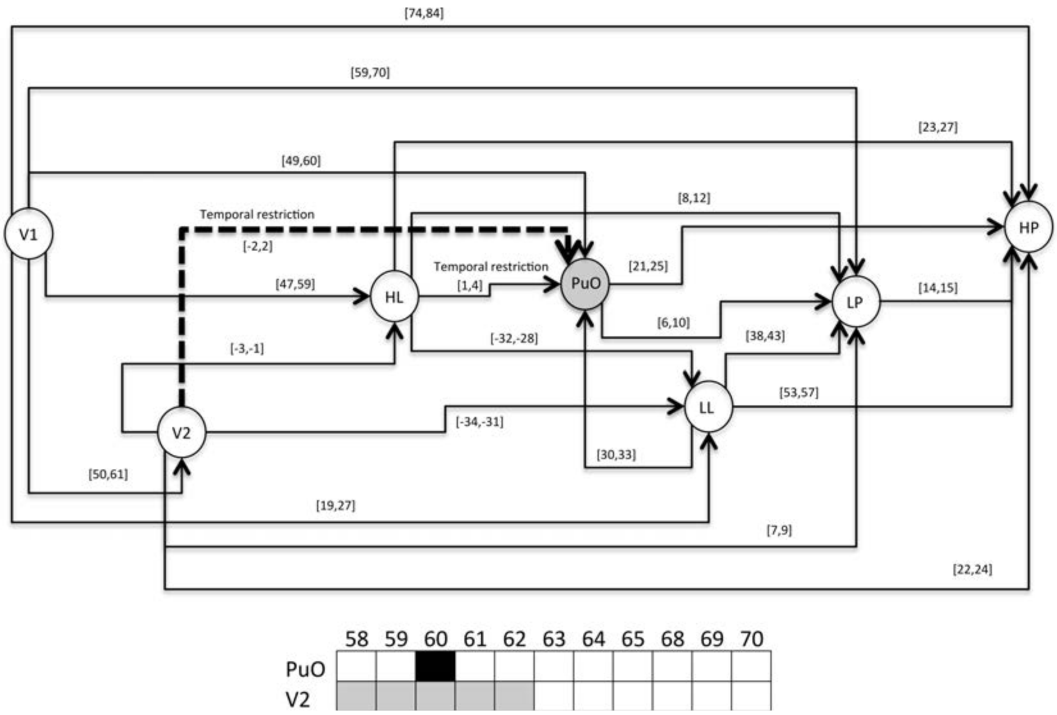


Figure 7.13: Activation of $PuO$ at 60

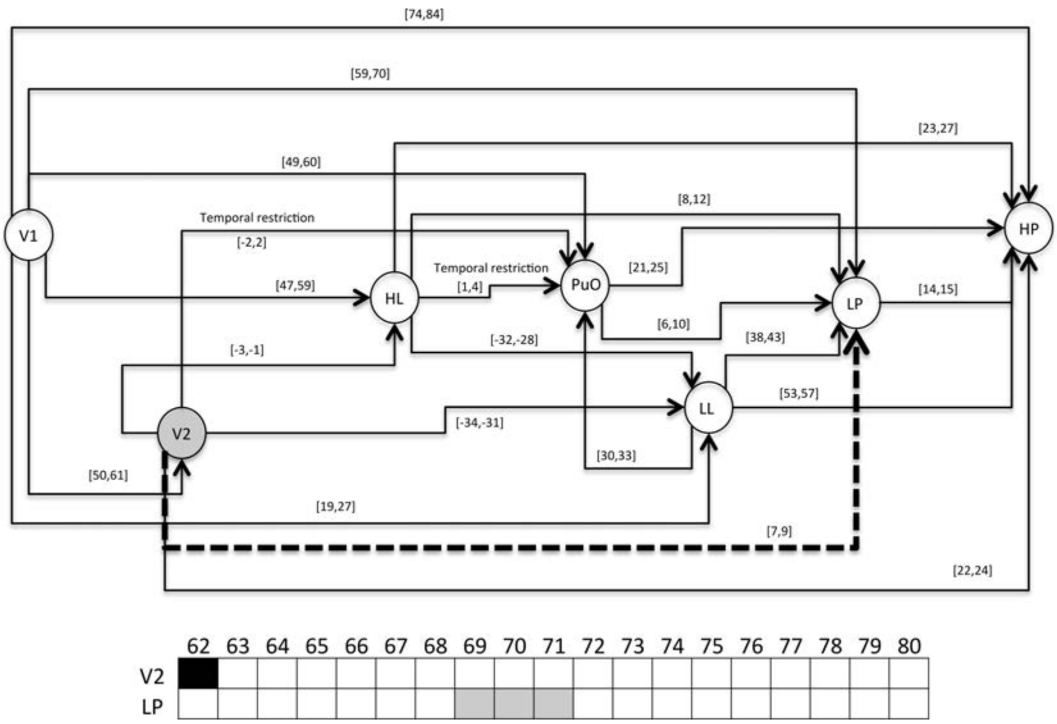


Figure 7.14: Activation of $V2$ at 62

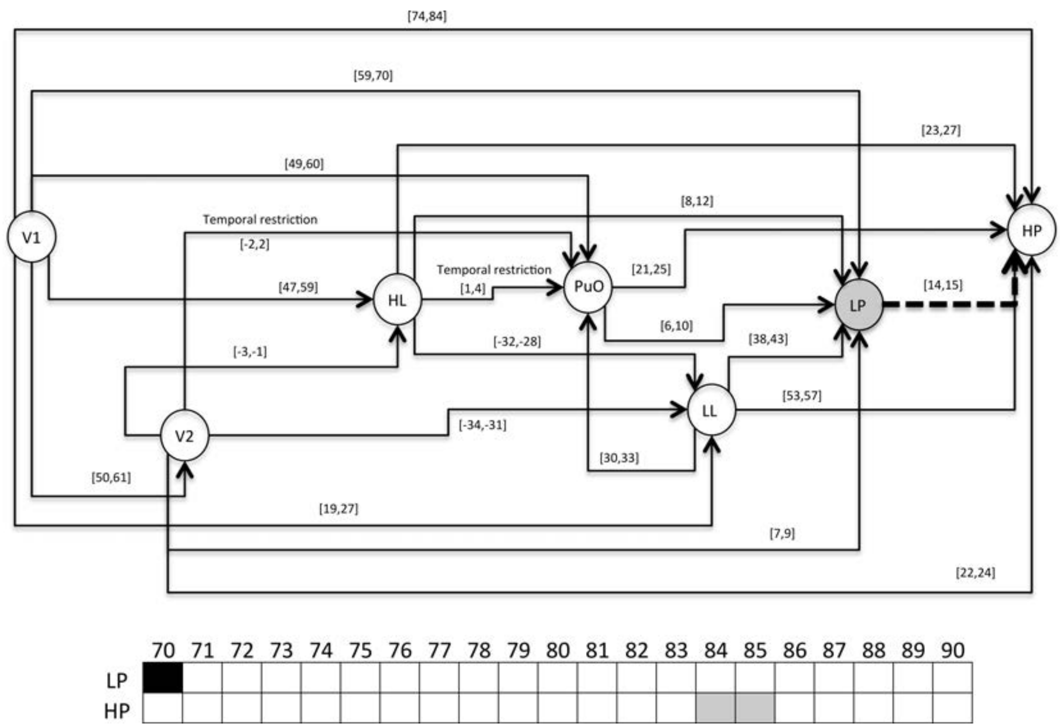


Figure 7.15: Activation of $LP$ at 70

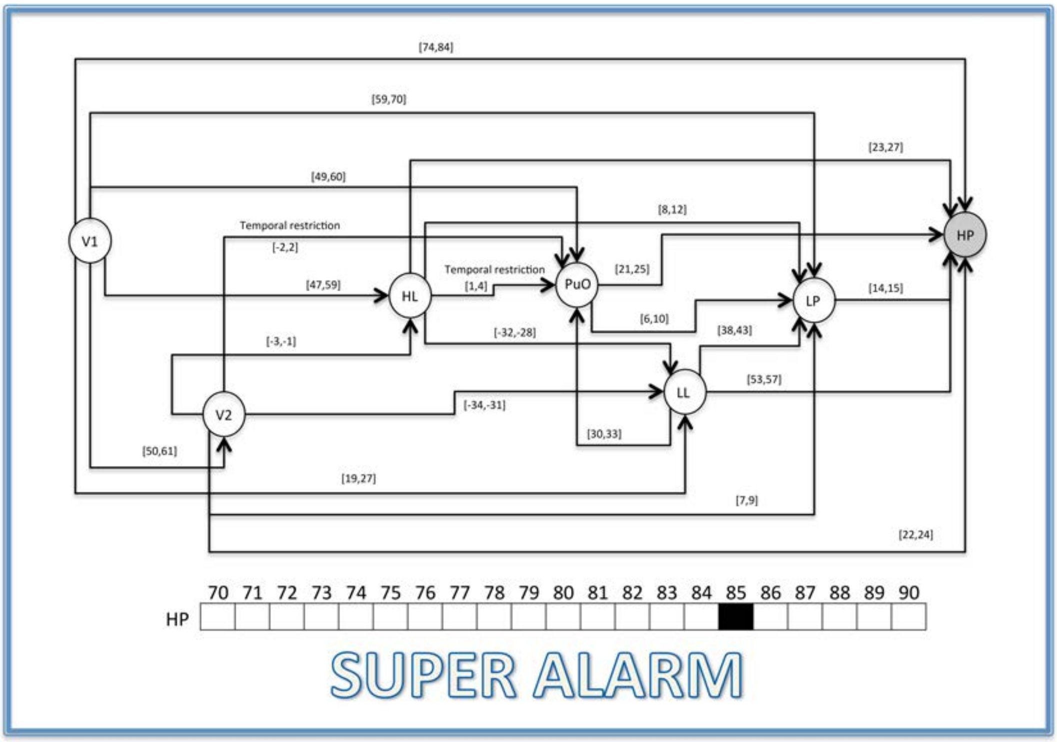


Figure 7.16: Activation of $HP$ at 85, SUPER ALARM: Recognition of the abnormal situation

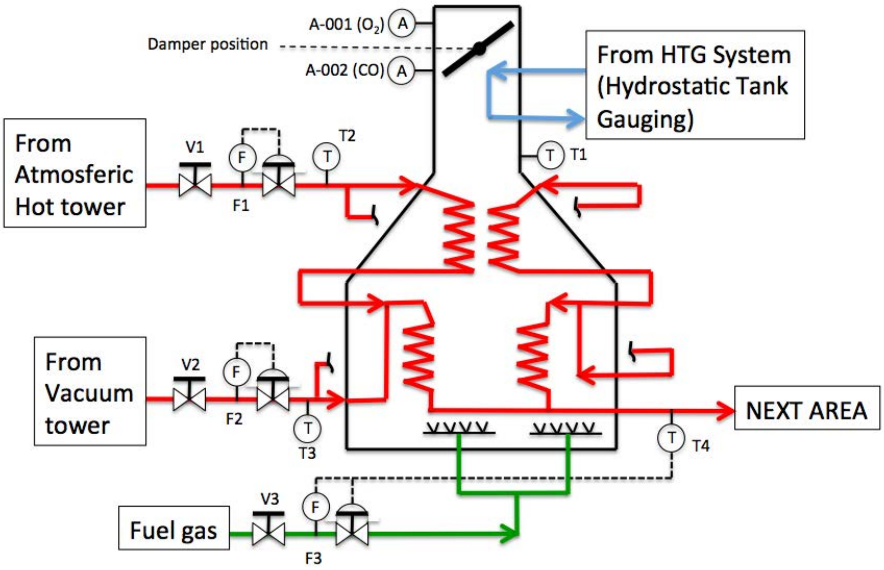


Figure 7.17: Vacuum oven

modified via an external action (e.g. the oven structure (**Ov**)) unlike active components whose states can be changed 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 sensor (**F1, F2,** and **F3**) and the temperature sensors (**T1, T2, T3** and **T4**), see Fig. 7.17. 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 of a set of discrete variables. The continuous variables are the temperature ($T1$,$T2$,$T3$ and $T4$) and the flows ($F1$,$F2$, and $F3$). On the other hand, 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 6. In this case study, continuous variable domain partitioning has been chosen according to expert knowledge and to limit values specified in standard operating procedures. $V_Q = \{\cup_{i=1}^{3}\{Fi^L, Fi^M, Fi^H\}\} \cup \{\cup_{i=1}^{4}\{Ti^L, Ti^M, Ti^H\}\}$

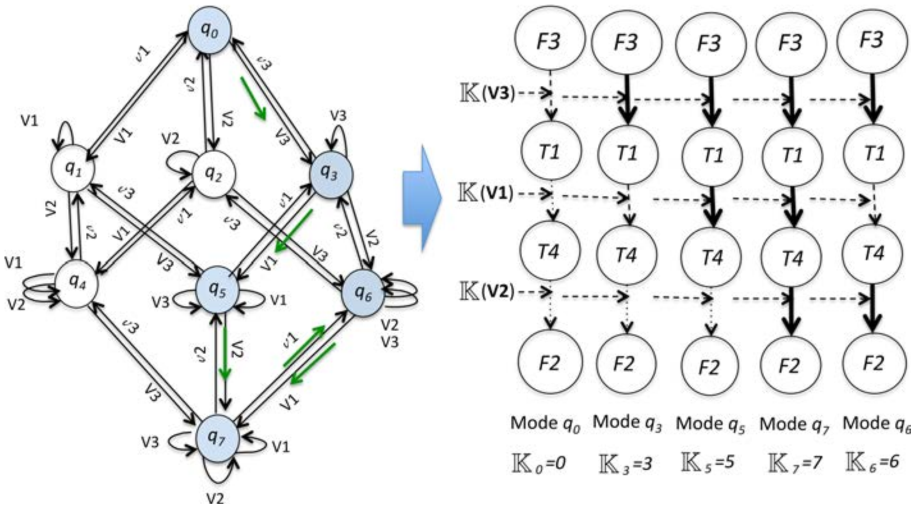


Figure 7.18: Startup stage of the vacuum oven: underlying DES and Causal System Description

- 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 operation 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 passes to the mode $q_3$ and $\mathbb{K}_3 = 3$. In $q_5$, **V3** and **V1** are opened and **V2** is closed, then $\mathbb{K}_5 = 5$. 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) (see Fig. 7.18 on the left). This model is obtained from the operating specifications described in the standard operating procedures. To each operation mode, qi 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 Fig. 7.18 on the left. Green arrows indicate the system evolutions during a start-up stage. The $CSDs$ associated to the operating modes (i.e. $q_0, q_3, q_5, q_6$ and $q_7$) involved in a start-up stage are shown Fig. 7.18 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 $\mathbb{K}$(**V3**). A bold edge indicates that the influence is active.

In the next subsections, the three steps of the methodology CBAM are detailed. Recalling, the first step is the Event type identification, the second step is the Event sequences generation and the third step corresponds to the Chronicle data base construction.

### 7.3.2 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:

$$\Sigma = \{V1, V2, V3, v1, v2, v3, M2A\} \tag{7.3}$$

where $V1$ (resp. $V2, V3$) is for the action that switches the valve **V1** (resp. **V2,V3**) from closed to opened. $v1$ (resp. $v2, v3$) for the action that switches the valve **V1** (resp. **V2,V3**) from to opened to closed. The event $M2A$ corresponds to the transition from *manual* to *automatic* operation, 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 with the behavior of the continuous variables is defined by the abstraction function (see Section 6). These influences allow determining the event types $\Sigma^c$ occurrence.

$$\Sigma^c = \begin{array}{l} \{L_{(F1)}, l_{(F1)}, H_{(F1)}, h_{(F1)}, \\ L_{(F2)}, l_{(F2)}, H_{(F2)}, h_{(F2)}, \\ L_{(F3)}, l_{(F3)}, H_{(F3)}, h_{(F3)}, \\ L_{(T1)}, l_{(T1)}, H_{(T1)}, h_{(T1)}, \\ L_{(T2)}, l_{(T2)}, H_{(T2)}, h_{(T2)}, \\ L_{(T3)}, l_{(T3)}, H_{(T3)}, h_{(T3)}, \\ L_{(T4)}, l_{(T4)}, H_{(T4)}, h_{(T4)}\} \end{array} \tag{7.4}$$

The occurrence of the event types $\Sigma^c$ depends on the influence of the continuous variables. These influences are captured in each causal system description associated with each operation mode (See Fig. 7.18 on the right).

### 7.3.3 Event sequence generation

Equal that in the other case study, from simulations the behavior of the variables is obtained and the learning event sequences are generated according to the evolution of the system in each scenario. In this manuscript are analyzed three scenarios: Normal startup, Abnormal start-up, and Normal shutdown.

#### 7.3.3.1 Scenario 1, Normal startup

For the start-up 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 its low limits (F1, F2, F3, T1, T2, T3, T4 ). According to the standard procedural actions, the first event type that must occur is $V3$ (*Open* **V3**). After of this event type occurrence, the system is in the mode of operation $q_3$ where the variable $T1$ increases and the event type $L_{(T1)}$ must occur after that the valve **V3** is opened, indicating that the internal oven temperature has passed the limit of low. After $L_{(T1)}$, the flow of the fuel gas must arrive to its low limit and the event type $L_{(F3)}$ must occur. At this time point, the ordered sequence of event types that has occurred is $V3, L_{(T1)}, L_{(F3)}$. Passing the low limit of the $F3$ is the condition for continuing the procedural action *Open* **V1** ($V1$) after $L_{(F3)}$. When the operator opens the valve **V1**, the system passes to the mode of operation $q_5$ and the internal flow in the vacuum oven begins. In this situation, the flow $F1$ and the outflow temperature $T4$ increases. Then, after of $V1$, the event type $L_{(T4)}$ must occur followed by the event type $L_{(F1)}$. The next event type that occurs is $H_{(F1)}$ indicating that the flow $F1$ has passed its high level after $L_{(F1)}$. At this time point, the ordered sequence of event types that has occurred must be $V3, L_{(T1)}, L_{(F3)}, V1, L_{(T4)}, L_{(F1)}, H_{(F1)}$. Continuing with the evolution of the process, after of $H_{(F1)}$ the following procedural action is close the valve **V1** ($v1$) and after of $v1$ the valve **V2** is opened ($V2$). Now, after of $V2$ the high limit of the temperature $T1$ must occurs and the event type $H_{(T1)}$ happens. The flow $F1$ decreases from its high limit and $h_{(F1)}$ occurs after of $H_{(T1)}$. Now, after of $h_{(F1)}$, the event type $L_{(F2)}$ occurs because the flow in the valve **V2** begins to increase. The high limit in the temperature $T4$ ($H_{(T4)}$) occurs after of $L_{(F2)}$. Carry on with the procedure, the high limit of flow in $F2$ happens and the event type $H_{(F2)}$ occurs after of $H_{(T4)}$. At this time point, the ordered sequence of event types that has occurred is $V3, L_{(T1)}, L_{(F3)}, V1, L_{(T4)}, L_{(F1)}, H_{(F1)}, v1, V2, H_{(T1)}, h_{(F1)}, L_{(F2)}, H_{(T4)}, H_{(F2)}$. In this situation, the unobservable event type $M2A$ occurs and the control loops are closed, carrying the system to a steady state. We assume that the control loops are closed immediately after that $H_{(F2)}$ occurs. Now, the system must advance to its steady state and its variables decreases. Then, after of $H_{(F2)}$ the flow $F1$ decreases and $l_{(F1)}$ occurs; after that $h_{(T1)}$ occurs, and the temperature $T4$ decreases and the event type $h_{(T4)}$ occurs after of $h_{(T1)}$. The event type $h_{(F2)}$ occurs after of $h_{(T4)}$. The final procedural action is open the valve **V1** for second time, and the event type $V1$ occurs after of $h_{(F2)}$. The last event type that occurs in a normal startup for this case of study is the second occurrence of $L_{(F1)}$, that happens after of $V1$. The final ordered sequence that must occurs in this scenario is $V3, L_{(T1)}, L_{(F3)}, V1, L_{(T4)}, L_{(F1)}, H_{(F1)}, v1$, $V2, H_{(T1)}, h_{(F1)}, L_{(F2)}, H_{(T4)}, H_{(F2)}, l_{(F1)}, h_{(T1)}, h_{(F2)}, V1, L_{(F1)}$

For this scenario, we chose the learning event sequences ($S_1$, $S_2$ and $S_3$) that represent the extreme behaviors with all the possible sequence order of event types.

$S_1$:

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

$S_3$:

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

The simulation of a $normal\ start-up$ is presented in Fig. 7.19 where we can see the evolution of the process variables. This simulation represents only one possible situation in this scenario and it is related to the representative sequence $S_1$. The values of the variables on the graph are specified as follows:

- For the variables of flow ($F1$ and $F2$) each increase in the vertical axis corresponds to 20 BPM. In this graph, the value of 25 correspond to 0 BPM.
- For the variable of flow ($F3$) each increase in the vertical axis corresponds to 2 $m^3$/min. In this graph, the value of 25 corresponds to 0 $m^3$/min (cubic meters per minute).
- For the variables of temperature ($T1$,$T2$,$T3$,$T4$) each division corresponds to 25°C.
- The time (horizontal axis) in the graph is expressed in seconds.

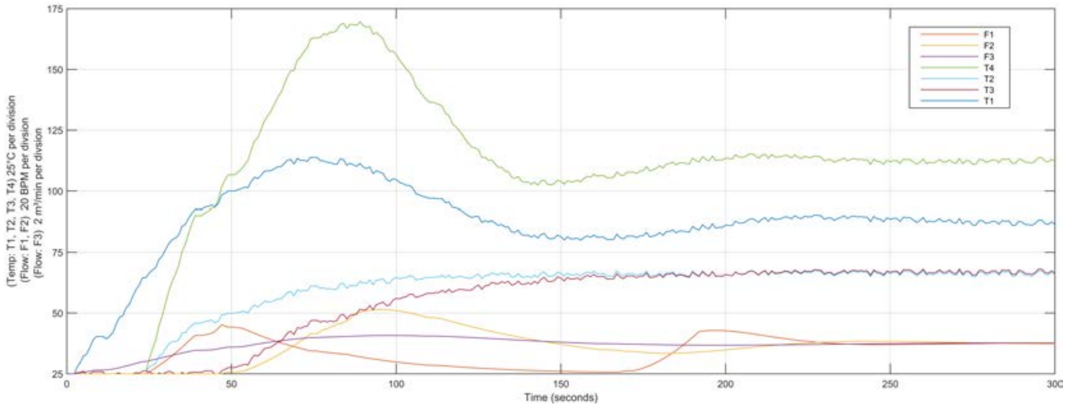


Figure 7.19: Simulation of a normal startup in the vacuum oven

At this point, the representative event sequences must be verified using the hybrid causal model. For example, the sequence $S_1$ initiates with the event type $V3$. Then, the system passes to the operation mode $q_3$ and the relationship between $F3$ and $T1$ is activated, see Fig. 7.18. We wait for the occurrence of two event types, first of the event type $L_{(T1)}$ and second $L_{(F3)}$. After that, the standard procedure actions $V1$ must occur passing the system from the mode of operation $q_3$ to the mode of operation $q_5$. When the system is in the mode $q_5$, the relationship of the continuous variables $T1$ and $T4$ is activated and we wait for the occurrence of the three event types: first $L_{(T4)}$, second $L_{(F1)}$ and third $H_{(F1)}$ in this order. Now the procedural actions $v1$ and $V2$ are activated and the system first return from $q_5$ to $q_3$ and after that, it passes to $q_6$ where is activated the relationship between $T4$ and $F2$. In this time point, the following five event types are activated: $H_{(T1)}$, $h_{(F1)}$, $L_{(F2)}$, $H_{(T4)}$, $H_{(F2)}$. Now, temperature and flow in $T1$, $T4$, $F1$ and $F2$ decrease and the event types $l_{(F1)}$, $h_{(T1)}$, $h_{(T4)}$ and $h_{(F2)}$ occur in this order. Finally, the procedural action $V1$ is executed by the second time and the event type $L_{(F1)}$ also occurs for the second time, concluding this event sequence. For the other sequences, the same procedure is applied.

#### 7.3.3.2 Scenario 2, Abnormal startup

This abnormal situation is related to a failure in the control valve of $F3$. In this scenario, the sequences of event types are similar that the event sequences of a normal startup until it is detected that the flow in $F3$ increase without control ($H_{(F3)}$) and the temperature of the oven do not decrease after that the control loops are closed. For this scenario, we chose the representative event sequences ($S_4$, $S_5$ and $S_6$) that show the extreme behaviors with all the possible sequence order of event types.

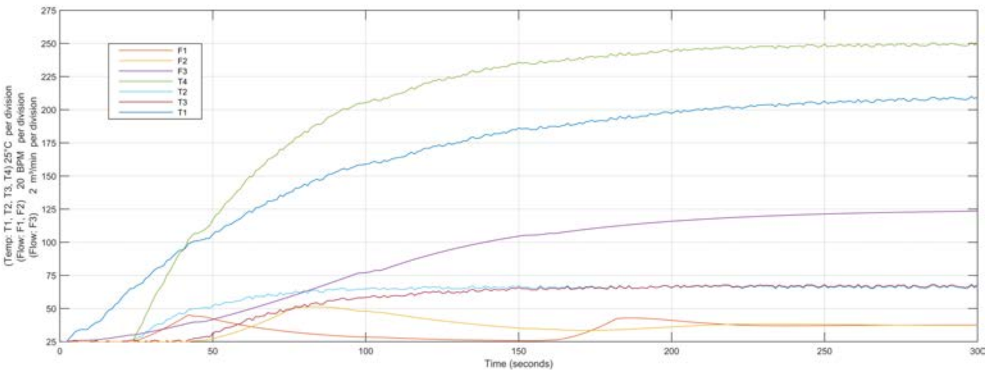


Figure 7.20: Simulation of an abnormal startup in the vacuum oven

$S_4$:
$\langle$(V3,1), $(L_{(T1)}, 3)$, $(L_{(F3)}, 5)$, $(V1, 6)$, $(L_{(T4)}, 7)$, $(L_{(F1)}, 8)$, $(H_{(F1)}, 12)$, $(v1, 13)$, $(V2, 14)$, $(H_{(T1)}, 15)$, $(h_{(F1)}, 16)$, $(L_{(F2)}, 17)$, $(H_{(T4)}, 19)$, $(H_{(F2)}, 22)$, $(l_{(F1)}, 24)$, $(H_{(F3)}, 26)$, $(h_{(F2)}, 27)$, $(V1, 42)$, $(L_{(F1)}, 45)$ $\rangle$

$S_5$:
$\langle$(V3,1), $(L_{(T1)}, 7)$, $(L_{(F3)}, 13)$, $(V1, 18)$, $(L_{(T4)}, 21)$, $(L_{(F1)}, 24)$, $(H_{(F1)}, 32)$, $(v1, 35)$, $(V2, 37)$, $(H_{(T1)}, 40)$, $(h_{(F1)}, 45)$, $(L_{(F2)}, 48)$, $(H_{(T4)}, 54)$, $(H_{(F2)}, 61)$, $(l_{(F1)}, 65)$, $(H_{(F3)}, 70)$, $(h_{(F2)}, 76)$, $(V1, 96)$, $(L_{(F1)}, 101)$ $\rangle$

$S_6$:
$\langle$(V3,2), $(L_{(T1)}, 6)$, $(L_{(F3)}, 9)$, $(V1, 12)$, $(L_{(T4)}, 14)$, $(L_{(F1)}, 16)$, $(H_{(F1)}, 22)$, $(v1, 24)$, $(V2, 25)$, $(H_{(T1)}, 27)$, $(h_{(F1)}, 30)$, $(L_{(F2)}, 32)$, $(H_{(T4)}, 36)$, $(H_{(F2)}, 41)$, $(l_{(F1)}, 43)$, $H_{(F3)}, 44)$, $(h_{(F2)}, 50)$, $(V1, 68)$, $(L_{(F1)}, 71)$ $\rangle$

The simulation of this *abnormal startup* is presented in Fig. 7.20 where we can see the evolution of the process variables. This simulation represents only one possible situation in this scenario related with the representative sequence $Sp_4$. The values of the variables on the graph are specified as follows:

- For the variables of flow ($F1$ and $F2$) each increase in the vertical axis corresponds to 20 BPM. In this graph, the value of 25 correspond to 0 BPM.

- For the variable of flow ($F3$) each increase in the vertical axis corresponds to 2 $m^3$/min. In this graph, the value of 25 corresponds to 0 $m^3$/min (cubic meters per minute).

- For the variables of temperature ($T1$,$T2$,$T3$,$T4$) each division corresponds to 25°C.

- The time (horizontal axis) in the graph is expressed in seconds.

For this abnormal start-up, the representative event sequences must be verified using the hybrid causal model. For example, similar that in the normal startup, the sequence $S_4$ initiates with the event type $V3$. Then, the system passes to the operation mode $q_3$ and the relationship between $F3$ and $T1$ is activated, see Fig. 7.18. We wait for the occurrence of two event types, first of the event type $L_{(T1)}$ and second $L_{(F3)}$. Similar to the other scenario, the standard procedure actions $V1$ must occur passing the system from the mode of operation $q_3$ to the mode of operation $q_5$. When the system is in the mode $q_5$, the relationship of the continuous variables $T1$ and $T4$ is activated and we wait for the occurrence of the three event types: first $L_{(T4)}$, second $L_{(F1)}$ and third $H_{(F1)}$ in this order. At this time point, equal that in the previous scenario, the procedural actions $v1$ and $V2$ are activated and the system first returns from $q_5$ to $q_3$ and after that, it passes to $q_6$ where is activated the relationship between $T4$ and $F2$. At this time point, the following five event types are activated: $H_{(T1)}$, $h_{(F1)}$, $L_{(F2)}$, $H_{(T4)}$, $H_{(F2)}$. In this moment, the temperature in $T4$ and $T1$ must decrease, but on the contrary the flow in the $F3$ increase. The unique three event types detected are: $l_{(F1)}$, $H_{(F3)}$, and $h_{(F2)}$ in this order. Finally, the procedural action $V1$ is executed by the second time and the event type $L_{(F1)}$ also occurs in a second time, concluding this event sequence detecting an abnormal behavior. For the other sequences, the same procedure is applied.

#### 7.3.3.3 Scenario 3, Normal shutdown

After of that, an abnormal start-up situation is detected, a shutdown procedure must be executed. Taking the above situation (*scenario* 2), it is assumed that after of the abnormal startup confirmed the standard procedure actions $v1$, $v2$, and $v3$ must be developed. For this scenario, we chose the representative event sequences ($S_7$, $S_8$ and $S_9$) that represent the extreme behaviors with all the possible sequence order of event types.

$S_7$:

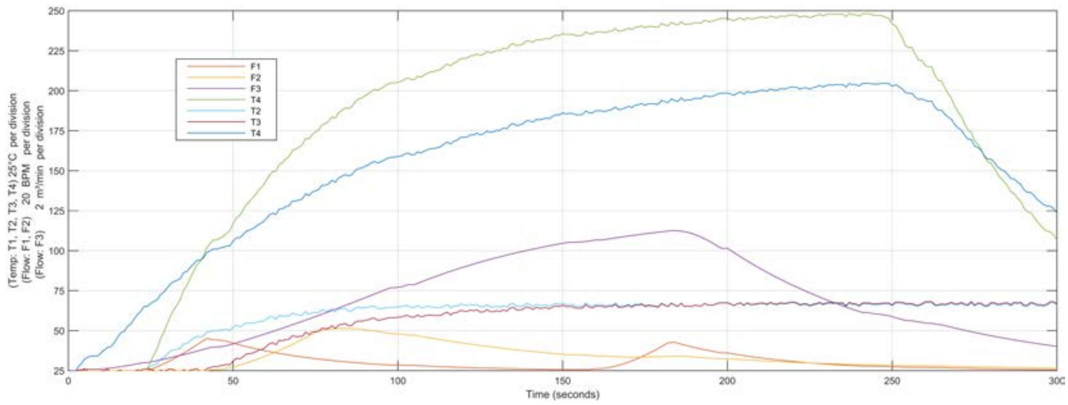


Figure 7.21: Simulation of a normal shutdown in the vacuum oven

$\langle$(V3,1), $(L_{(T1)}, 3)$, $(L_{(F3)}, 5)$, $(V1, 6)$, $(L_{(T4)}, 7)$, $(L_{(F1)}, 8)$, $(H_{(F1)}, 12)$, $(v1, 13)$, $(V2, 14)$, $(H_{(T1)}, 15)$, $(h_{(F1)}, 16)$, $(L_{(F2)}, 17)$, $(H_{(T4)}, 19)$, $(H_{(F2)}, 22)$, $(l_{(F1)}, 24)$, $(H_{(F3)}, 26)$, $(h_{(F2)}, 27)$, $(V1, 42)$, $(L_{(F1)}, 45)$, $(v3, 52)$, $(v1, 53)$, $(v2, 54)$, $(h_{(T4)}, 88)$, $(h_{(F3)}, 121)$, $(h_{(T1)}, 142)$ $\rangle$

$S_8$:

$\langle$(V3,1), $(L_{(T1)},7)$, $(L_{(F3)},13)$, $(V1,18)$, $(L_{(T4)},21)$, $(L_{(F1)},24)$, $(H_{(F1)},32)$, $(v1,35)$, $(V2,37)$, $(H_{(T1)},40)$, $(h_{(F1)},45)$, $(L_{(F2)},48)$, $(H_{(T4)},54)$, $(H_{(F2)},61)$, $(l_{(F1)},65)$, $(H_{(F3)},70)$, $(h_{(F2)},76)$, $(V1,96)$, $(L_{(F1)},101)$, $(v3,77)$, $(v2,78)$, $(v1,80)$, $(h_{(F3)},158)$, $(h_{(T1)},151)$, $(h_{(T4)},182)$ $\rangle$

$S_9$:

$\langle$(V3,2), $(L_{(T1)}, 6)$, $(L_{(F3)}, 9)$, $(V1, 12)$, $(L_{(T4)}, 14)$, $(L_{(F1)}, 16)$, $(H_{(F1)}, 22)$, $(v1, 24)$, $(V2, 25)$, $(H_{(T1)}, 27)$, $(h_{(F1)}, 30)$, $(L_{(F2)}, 32)$, $(H_{(T4)}, 36)$, $(H_{(F2)}, 41)$, $(l_{(F1)}, 43)$, $H_{(F3)}, 44)$, $(h_{(F2)}, 50)$, $(V1, 68)$, $(L_{(F1)}, 71)$, $(v3, 63)$, $(v1, 65)$, $(v2, 67)$, $(h_{(T4)}, 98)$, $(h_{(F3)}, 123)$, $(h_{(T1)}, 152)$ $\rangle$

The simulation of a *normal shutdown* is presented in Fig. 7.21 where we can see the evolution of the process variables. This simulation represents only one possible situation in this scenario related with the representative sequence $S_7$. The procedure evaluation of this event sequences is similar that the procedure developed in the other scenarios. In this scenario, the event types $v1$, $v2$ and $v3$ are involved in the shutdown procedure. The values of the variables on the graph are specified as follows:

- For the variables of flow ($F1$ and $F2$) each increase in the vertical axis corresponds to 20 BPM. In this graph, the value of 25 correspond to 0 BPM.

- For the variable of flow ($F3$) each increase in the vertical axis corresponds to 2 $m^3$/min. In this graph, the value of 25 corresponds to 0 $m^3$/min (cubic meters per minute).

- For the variables of temperature ($T1$,$T2$,$T3$,$T4$) each division corresponds to 25°C.

- The time (horizontal axis) in the graph is expressed in seconds.

### 7.3.4 Chronicle database construction

In this case study, the chronicle learning algorithm used was the $HCDAM$ extended which includes temporal runs as a new input to the algorithm. This chronicle database is to be submitted to a chronicle recognition system that identifies 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. For example, for the Refinery, we denote the system HTG (Hydrostatic Tank Gauging) as the area $Ar_1$ and the vacuum oven as the area $Ar_2$. In this subsection are presented the Chronicles $C^2_{10}$,$C^2_{11}$,$C^2_{20}$ from the set of chronicles of the vacuum oven system (Area $Ar_2$ of the whole system). $C^2_{10}$ is a chronicle that describes the normal startup stage of the Vacuum Oven, $C^2_{11}$ is associated with the failure behavior of type $f_1$ during a startup stage, $C^2_{20}$ describes a normal shutdown.

#### 7.3.4.1 Scenario 1, Normal startup:

For this scenario we have the following temporal restrictions that represent the expert knowledge which is used in the extended version of the $HCDAM$.

$TR_{V3,L_{F3}}$=$V3[6,8]L_{F3}$, this temporal restriction expresses that the lower limit of the flow in $F3$ arrives between 6 and 8-time units after that the valve **V3** is opened.

$TR_{V1,L_{F1}}$=$V1[-76,82]L_{F1}$, this temporal restriction indicates that the lower limit of the flow in $F1$ can occur 76-time units before that the valve **V1** is opened to 82-time units after that.

$TR_{L_{F2},V2}$=$L_{F2}[2,8]V2$, this temporal restriction indicates that the valve **V2** is opened between 2 and 8-time units after that the lower limit of the flow in $F2$ happen.

The directed graph of chronicle $C^2_{01}$ was obtained using the algorithm $HCDAM$ and its directed graph is presented in Fig. 7.22. The representative event sequences used were $S_1$, $S_2$ and $S_3$ which were generated in subsection *7.3.3*.

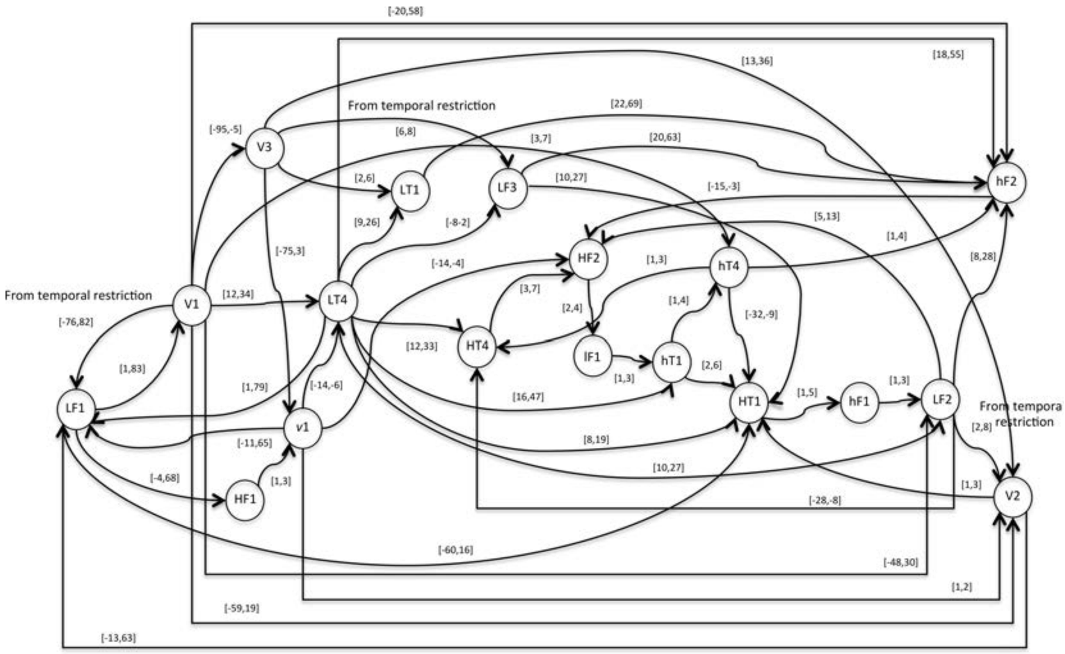


Figure 7.22: Directed graph ($\mathcal{G}$) of the chronicle $C^2_{01}$

Applying the concept of the event $\phi$, the frequency of occurrence of the event types $V1$ and $L_{(F1)}$ are $f_{(V1)}$=2, $f_{(LF1)}$=2 and for the others event types it is 1.

### 7.3.4.2 Scenario 2, Abnormal start-up:

In this case the following temporal restrictions represent the expert knowledge which is used in the extended version of the $HCDAM$.

**TR**$_{ij}$: { $TR_{(V1,LF1)}$=$V1$[-25,32]$L_{(F1)}$, $TR_{(HT1,LT1)}$=$HT1$[-40,12]$L_{(T1)}$ }

The directed graph of chronicle $C^2_{11}$ that resulted using the algorithm $HCDAM$ is presented in Fig. 7.23. The representative event sequences used are $S_4$, $S_5$ and $S_6$ generated in subsection *7.3.3*.

Applying the concept of the event $\phi$, the frequency of occurrence of the event types $V1$ and $L_{(F1)}$ are $f_{V1}$=2, $f_{LF1}$=2 and for the others event types it is 1.

### 7.3.4.3 Scenario 3, normal shutdown:

In this case the following temporal restrictions represent the expert knowledge which is used in the extended version of the $HCDAM$.

**TR**$_{ij}$: { $TR_{(V1,LF1)}$=$V1$[-20,18]$L_{(F1)}$, $TR_{(HT1,LT1)}$=$HT1$[-51,5]$L_{(T1)}$ }

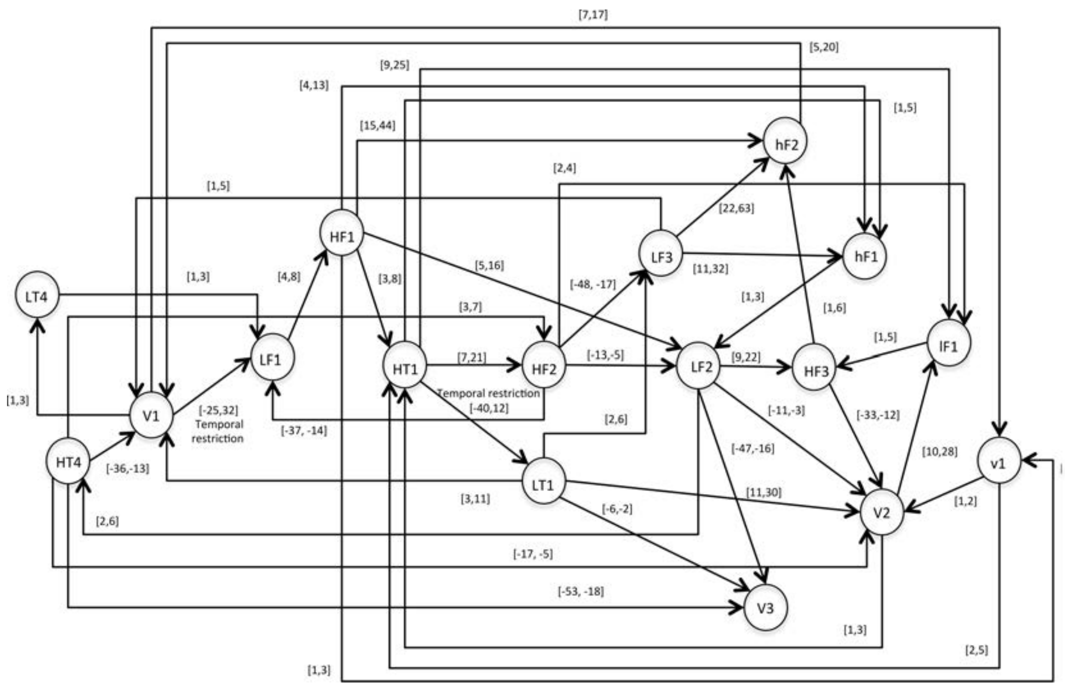


Figure 7.23: Directed graph ($\mathcal{G}$) of the chronicle $C_{11}^2$

The directed graph of chronicle $C_{02}^2$ that resulted using the algorithm $HCDAM$ is presented in Fig. 7.24. The representative event sequences used are $S_7$, $S_8$ and $S_9$ generated in subsection *7.3.3*.

Applying the concept of the event $\phi$, the frequency of occurrence of the event types $V1$ and $L_{(F1)}$ are $f_{V1}$=2, $f_{LF1}$=2 and for the others event types it is 1.

### 7.3.5 Validation

This section presents the evaluation of the chronicle $C_{11}^2$. It represents the temporal pattern for an abnormal startup in the HTG system. The sequence of evaluation is presented below:

S$_{\text{eval}}$:

$\langle$(V3, 3), $(L_{(T1)}, 9)$, $(L_{(F3)}, 15)$, $(V1, 19)$, $(L_{(T4)}, 21)$, $(L_{(F1)}, 24)$, $(H_{(F1)}, 31)$, $(v1, 33)$, $(V2, 34)$, $(H_{(T1)}, 35)$, $(h_{(F1)}, 39)$, $(L_{(F2)}, 40)$, $(H_{(T4)}, 44)$, $(H_{(F2)}, 48)$, $(l_{(F1)}, 52)$, $(H_{(F3)}, 55)$, $(h_{(F2)}, 59)$, $(V1, 78)$, $(L_{(F1)}, 83)$ $\rangle$

Fig. 7.25 to Fig. 7.43 present the recognition process of the chronicle, and the generation of one SUPER ALARM.

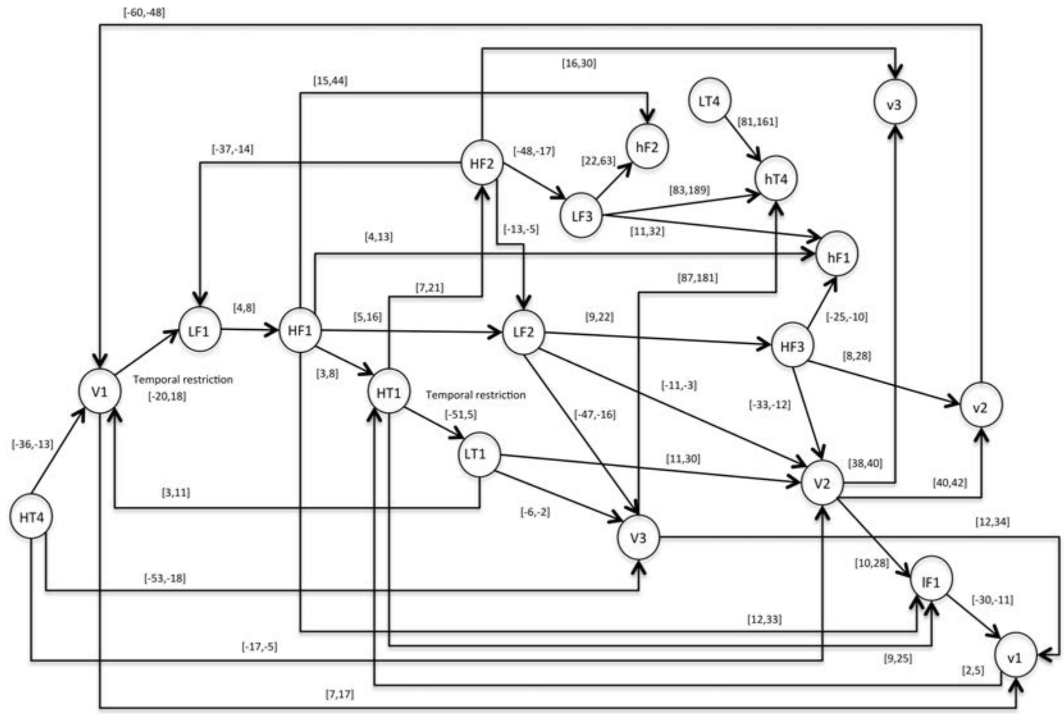


Figure 7.24: Directed graph ($\mathcal{G}$) of the chronicle $C_{02}^{2}$

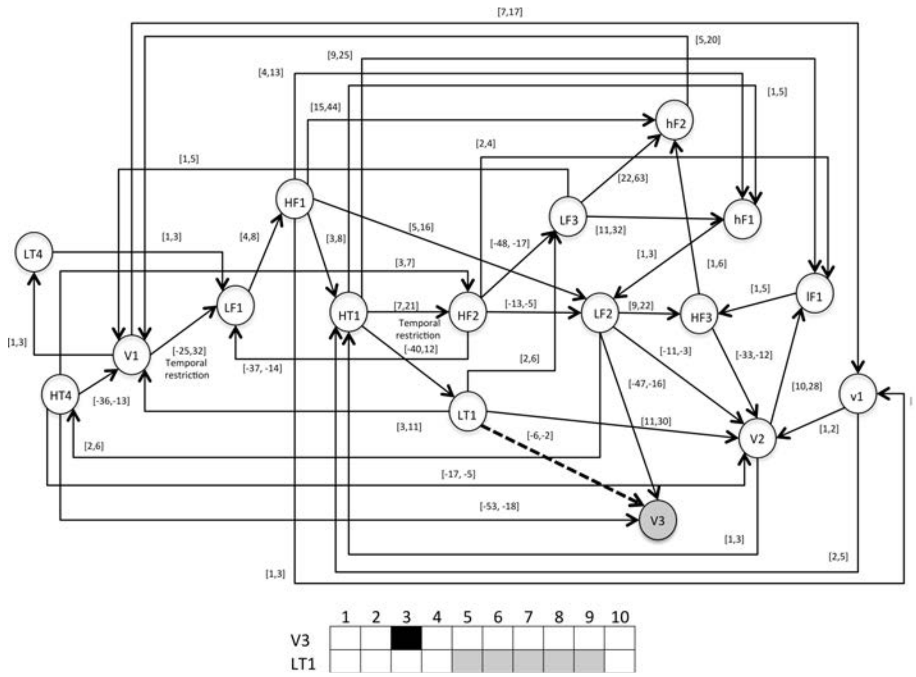


Figure 7.25: Activation of $V3$ at 3

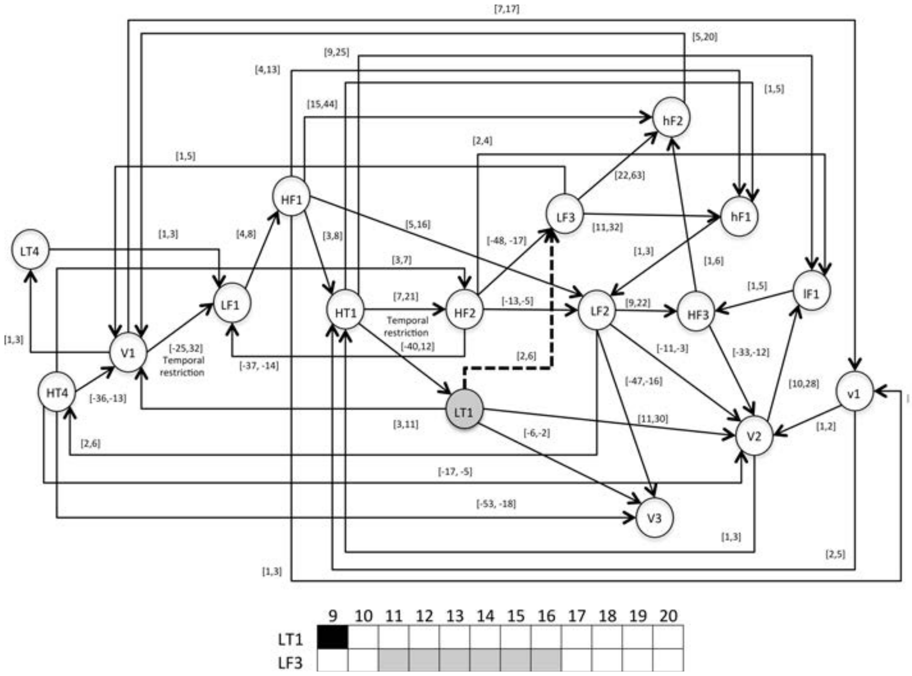


Figure 7.26: Activation of $LT1$ at 9

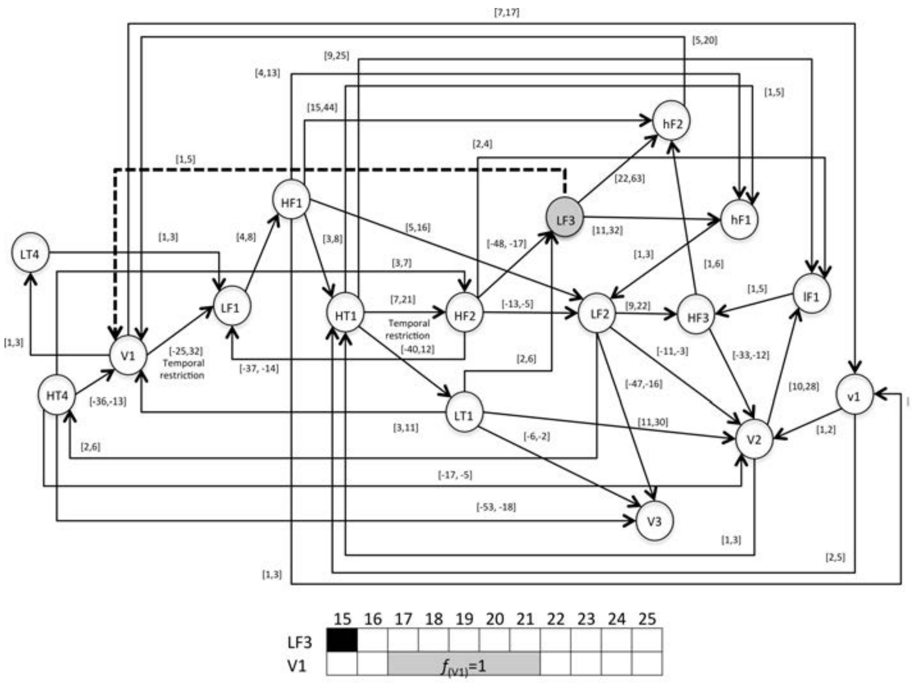


Figure 7.27: Activation of $LF3$ at 15

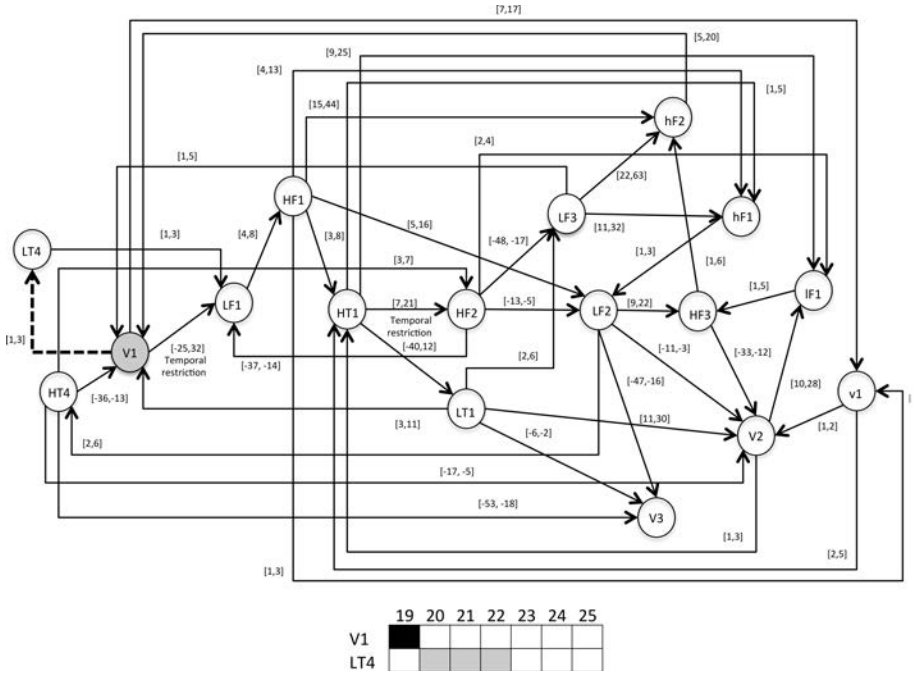


Figure 7.28: Activation for first time of $V1$ ($f_{(V1)} = 1$) at 19

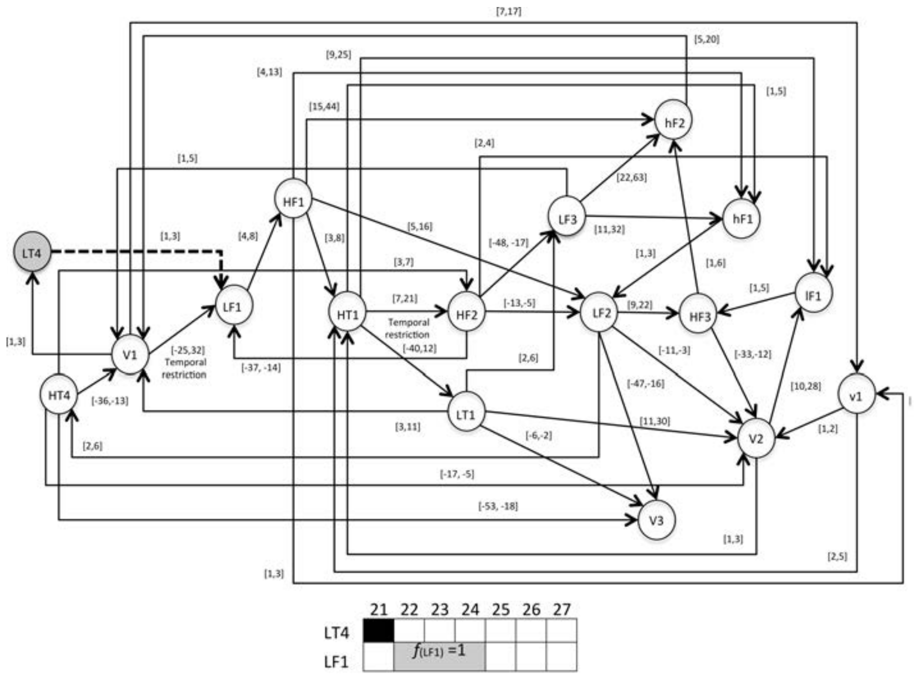


Figure 7.29: Activation of $LT4$ at 21

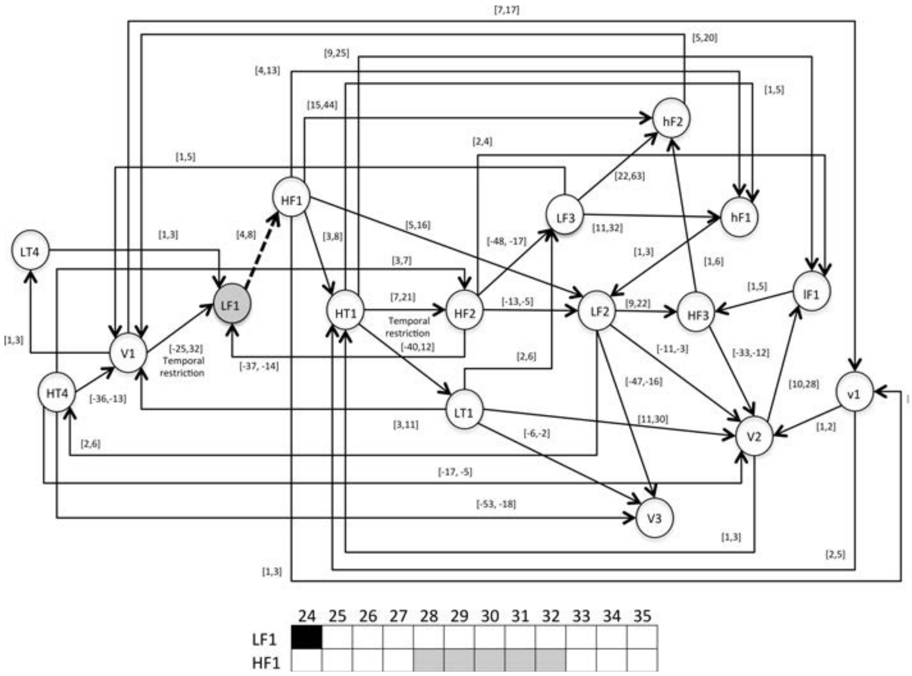


Figure 7.30: Activation for first time of $LF1$ ($f_{(LF1)} = 1$) at 24

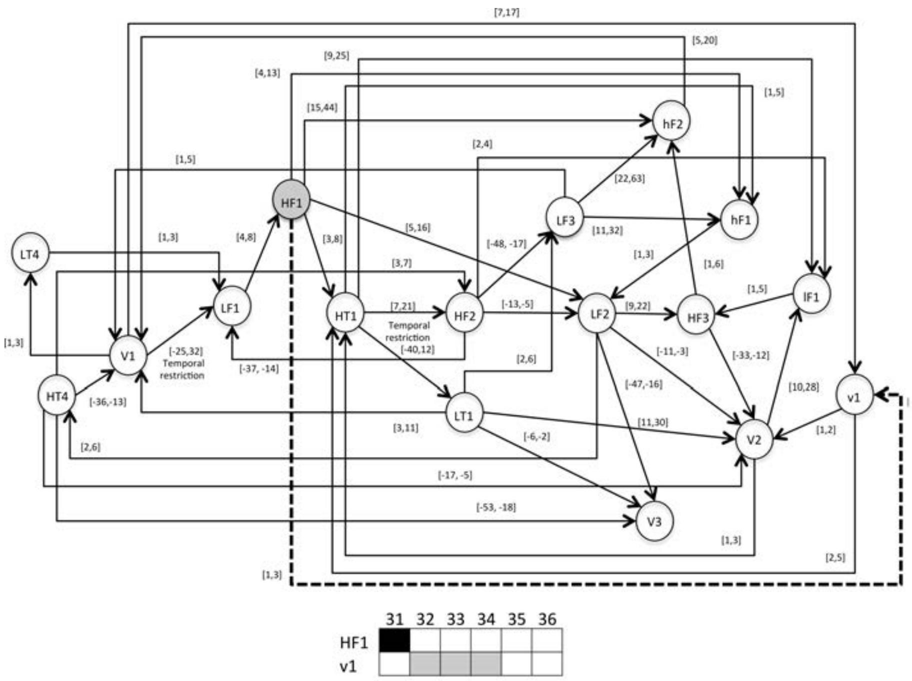


Figure 7.31: Activation of $HF1$ at 31

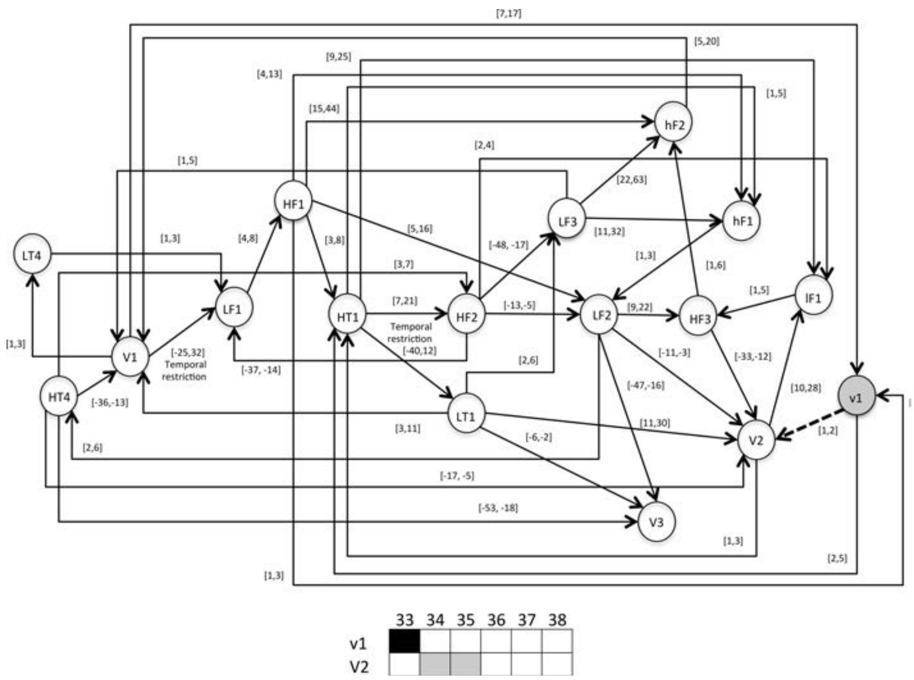


Figure 7.32: Activation of $v1$ at 33

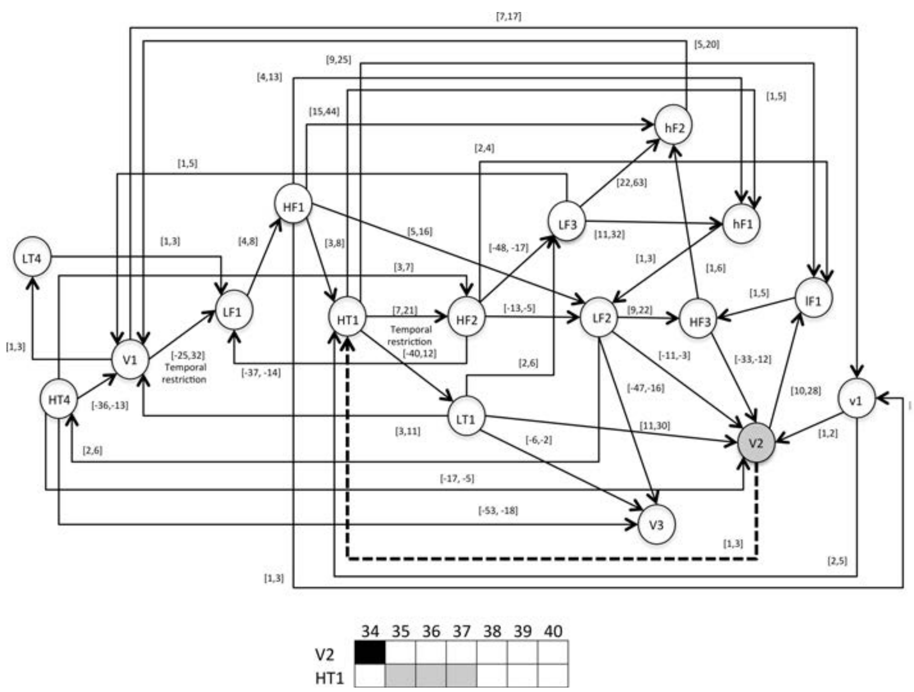


Figure 7.33: Activation of $V2$ at 34

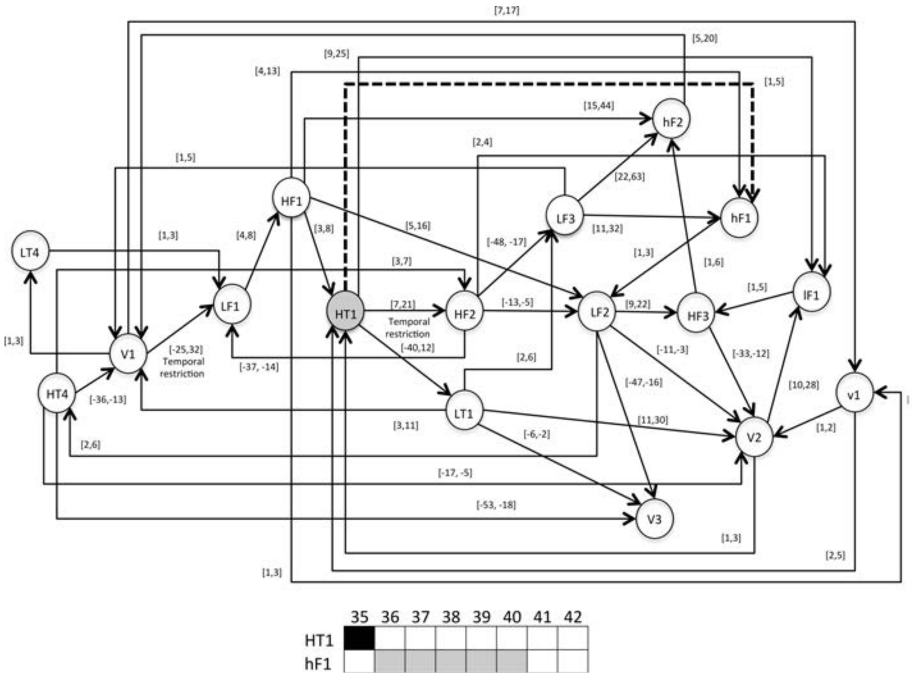


Figure 7.34: Activation of $HT1$ at 35

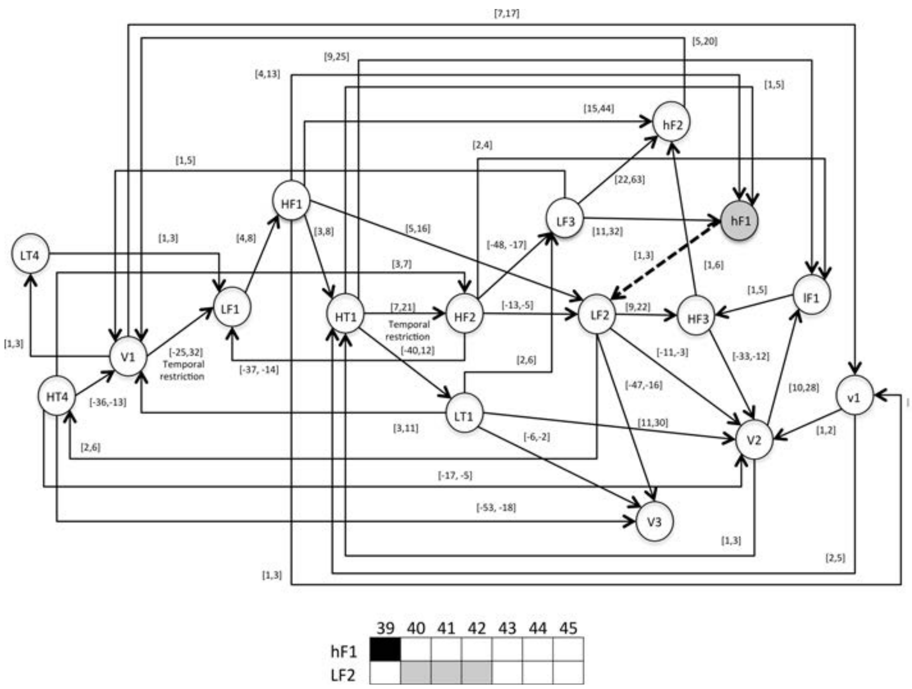


Figure 7.35: Activation of $hF1$ at 39

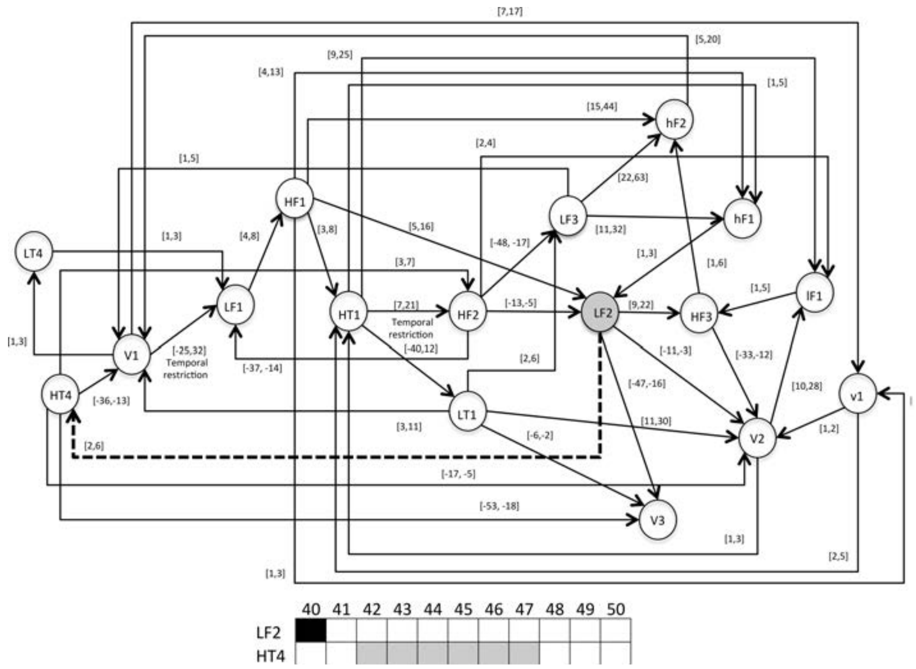


Figure 7.36: Activation of $LF2$ at 40

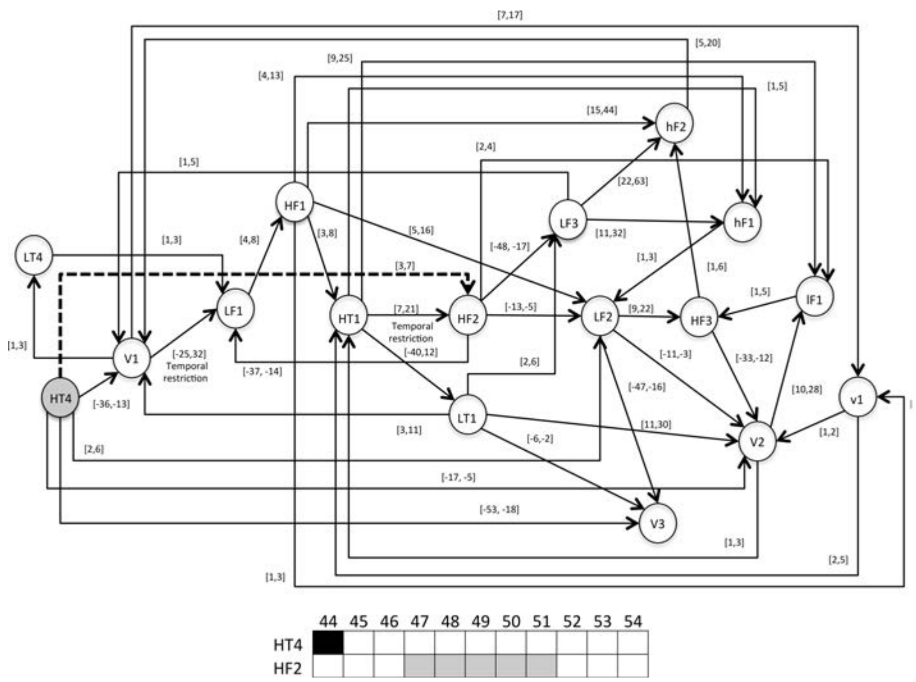


Figure 7.37: Activation of $HT4$ at 44

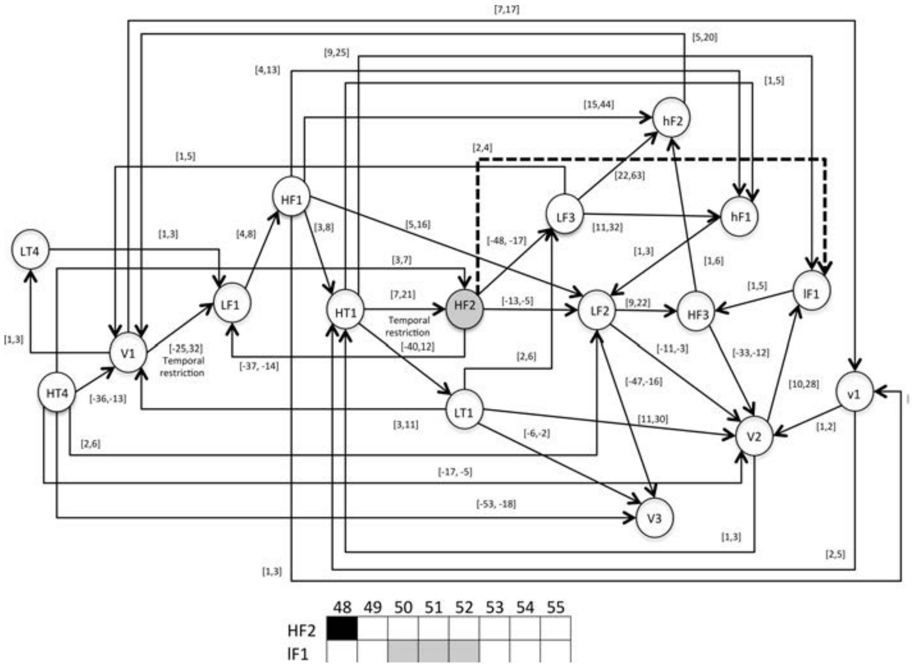


Figure 7.38: Activation of $HF2$ at 48

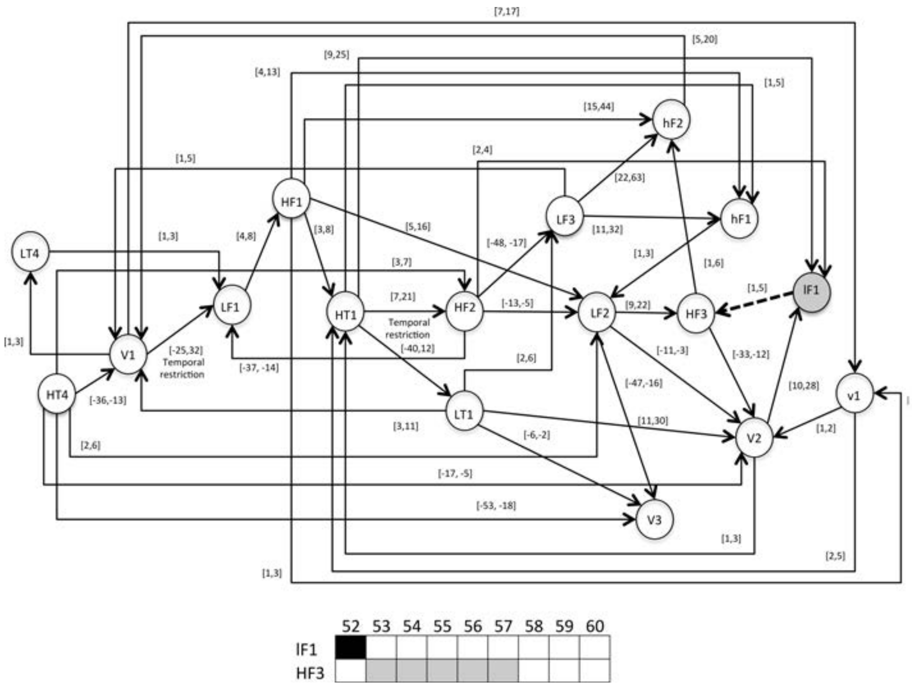


Figure 7.39: Activation of $lF1$ at 52

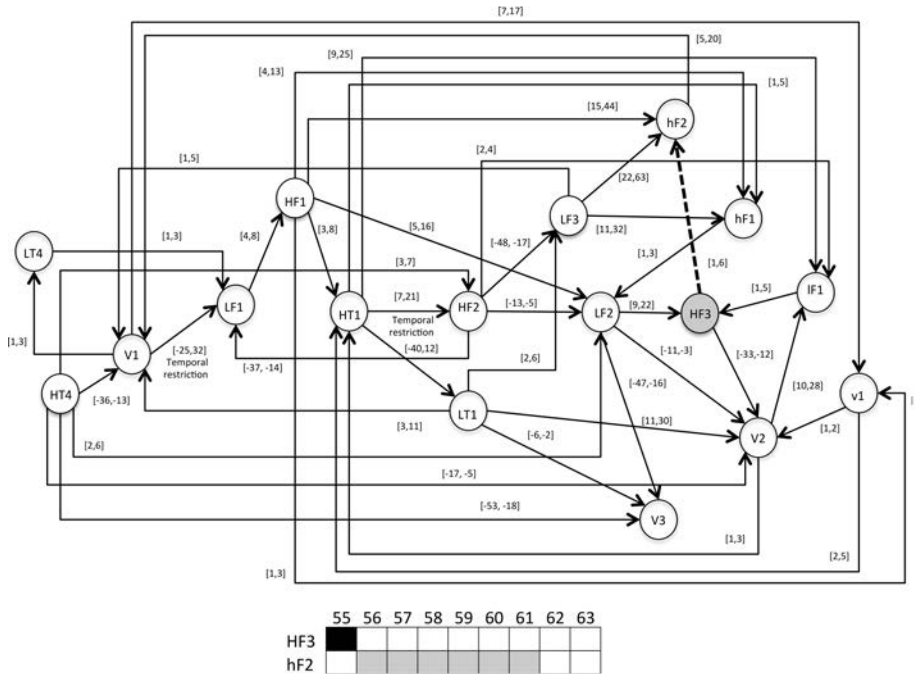


Figure 7.40: Activation of $HF3$ at 55

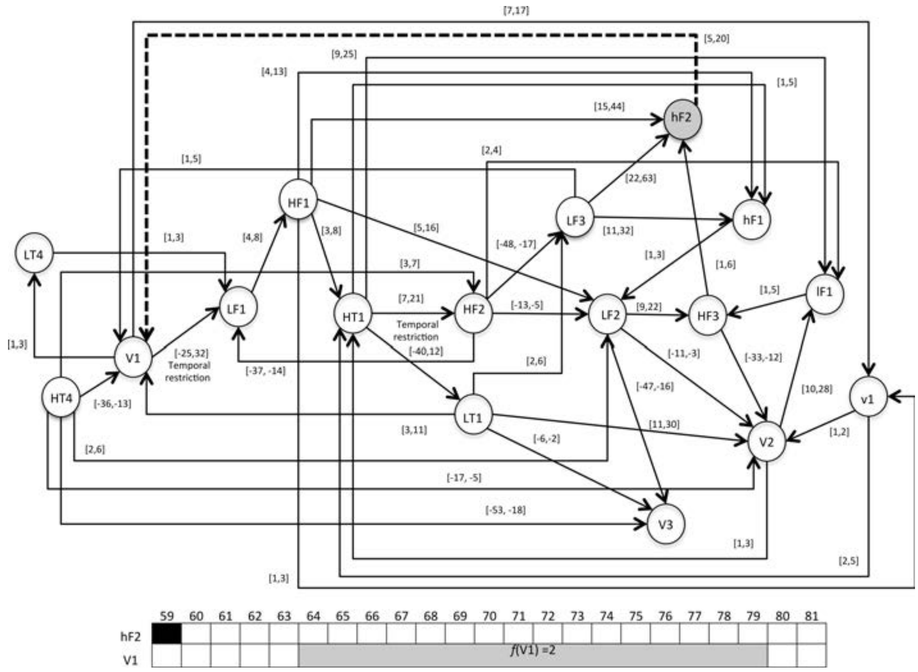


Figure 7.41: Activation of $hF2$ at 59

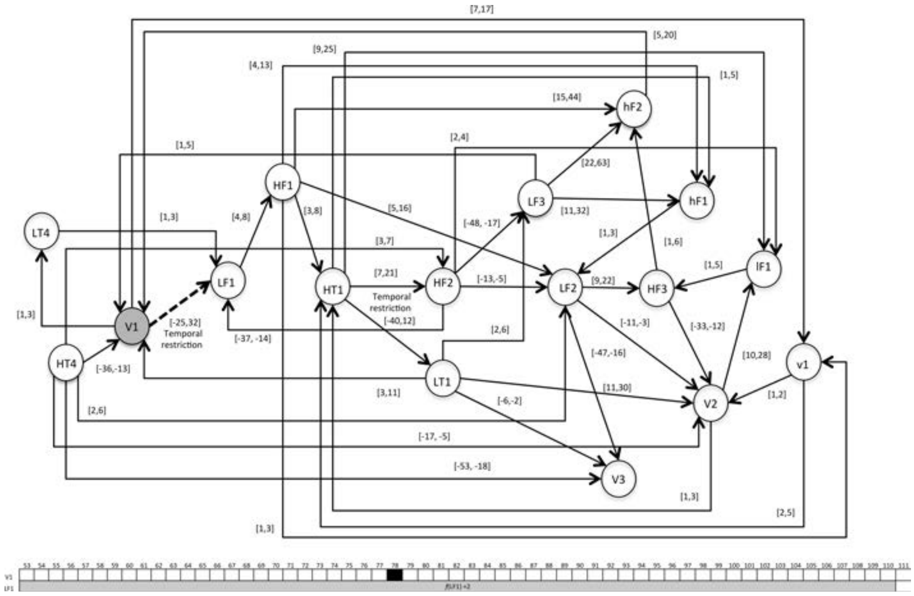

Figure 7.42: Activation for second time of $V1$ ($f_{(V1)}$=2) at 78

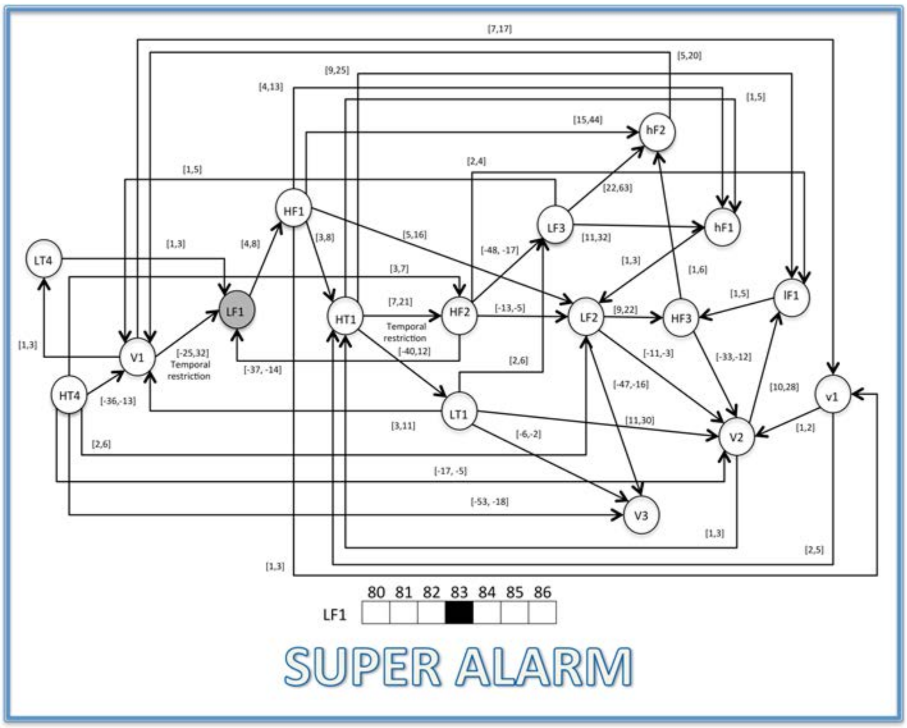


Figure 7.43: Activation for second time $LF1$ ($f_{(LF1)}$=2) at 83, SUPER ALARM: Recognition of the abnormal situation

The new concept of SUPER ALARM is proposed in this thesis, a concept which corresponds to one "superior alarm" giving relevant information to the operators after a diagnosis process, increasing the reliability of this protective layer, (see Chapter 2).

## 7.4 Discussion: How to implement CBAM

The new methodology Chronicle Based Alarm Management (CBAM) proposed in this thesis can be implemented in any industrial process, and its principal steps are enumerated below:

1. Event type identification
2. Learning event sequences identification
3. Construction of the chronicle database

For to implement this methodology, each step needs to involve all the specialties of the process such as chemistry, electrical and mechanical to ensure the maximum possible scenarios and to obtain a complete database of the normal and abnormal behavior. Before carrying out this methodology, an important action is to divide the complex process into different functional units hierarchically. As we saw in the case studies, the refinery was divided into separate units as the HTG system and vacuum oven system. In addition, each unit was conformed by active components such as valves, motors and pumps; and passive components such as tanks, oven structures, and pipes.

### 7.4.1 Event type identification

Once the complex process is divided, different tools like fault tree, the tree of events, Hazop and FMA can be used in determining the most important event types. In this methodology, an "event type" refers to a discrete event that represents a procedural action or a condition in the continuous or discrete process variables. There is a meaningful difference between an "event type" and the concept of "event" in safe processes. An "event" in safe processes corresponds to a failure or abnormal situation including an explosion, leak, burst or blow up. Therefore, an "event" can correspond to a scenario that is produced by a specific "event types" sequence. To avoid confusions with these terms, the concept of "event" in safe processes is replaced by the concept of "scenario" or "operational situation".

Although this methodology is for alarms, the alarms "fire detection" or "smoke detection" are not involved in this analysis. These types of signals are included in the fire protection systems. However, a diagnosis approach based on chronicles can be constructed to detect the correct progression of the fire protection system. Additionally, all the event types selected in this step must be observable and registered in the supervisory system HMI (Human machine interface). For example, the alarm limits for temperature, pressure or flow are observable events. Another example is a normally closed electric valve that does not indicate that it was opened, but we can read the voltage consumption there. Then, when the voltage is more than 100 Vac, it shows that the valve was opened.

Obtaining a complete list of event types allows for the construction of an efficient pattern model. Event types related to the behavior of the continuous variables are event types like low and high limit alarms. Discrete variables include a detector of elements, pressure switches and limit switches that can be involved in the temporal patterns. The events related to the standard operational actions are steps that change the mode of operation in the component. Turning on a pump, opening a valve, energizing a motor and connecting a heater, for instance, are events that represent a procedural action.

### 7.4.2 Event sequence generation

The hazard analysis can give us the scenarios or operational situations that we want to detect early, and the learning event sequences are the sequences of event types that are followed in each scenario. A hazard analysis expresses abnormal situation to avoid, yet normal situations also are required to detect. In other words, both normal and abnormal scenarios are necessary to construct a complete diagnosis model. The definition of the types and of the quantity of scenarios depends on an extensive study of the most important failures and abnormal situations to detect. The common failures and abnormal situations can be summarized in the following types:

- Failure in an actuator
  - Failure in the electrical connections and elements
  - Failure in the mechanical elements
  - Blockage
- Failure in a sensor

  - Primary element problem ( electrical connections, element position)
  - Failure in the transmitter
  - Uncalibrated sensor
- Failure in the control equipment
  - Failure in the electrical connectors (I/O)
  - Failure in the communication network
  - Non-tuned control parameters
- Failure of the safety element
- Errors in the procedural actions
  - Missing procedural action
  - Untimely procedural action
  - Wrong procedural action
- Problems in the process conditions
  - Leaks
  - Lack or excess of electric energy
  - Lack or excess of industrial air
  - Lack or excess of steam
  - Lack or excess of industrial water

The learning event sequences normally initiate with a standard operational action and the process evolves generating the event sequences. Sometimes, an event type different from a procedural action can occur first in a learning event sequence. As we saw in the example "oven charge system" (Section 6), the event types $a$ and $b$ occur before the first operational action $c$ is executed. On the other hand, in the HTG and vacuum oven systems (Section 7), the learning event sequences initiated with a standard procedural action.

The Hybrid causal model gives us a complete representation of the system, including its analog and discrete variables. In each mode of operation, the continuous and discrete variables progress and change according to the procedural actions. Depending on the causal relationships between the variables, the event type occurrence can be predicted.

For example, when an inlet valve is opened, the level of the liquid inside the tank increases and the alarm of low limit is activated. If the input flow increases, the level of the tank will increase faster than expected. Therefore, in many cases, abnormal situations can be detected by the analysis of the incidences between variables identifying which is the cause of the failure. Using the event type abstraction from the behavior of the continuous variables, the different event types with its date of occurrences are obtained. The continuous behavior of the variables can be obtained by simulation; in which each continuous variable is depicted by a transfer function of simple order with delay.

Expertise knowledge is largely used to complete the learning event sequences. In many situations, for a specific scenario, the operators know which are the most restrictive sequences of events and the most extended sequence in the time line. In the extension of the $HCDAM$, the temporal restrictions were used as the expert knowledge that represents some situations in the process. Although using temporal restrictions is not common to represent the expert knowledge in petrochemical processes, the operators normally are acquainted with the sequence order of the event types in a specific scenario.

### 7.4.3 Chronicle database construction

To construct the chronicles, the extended algorithm $HCDAM$ was used in this methodology. The procedure consists in to compile the totals of scenarios with their learning event sequences for each one, and also are required the temporal restrictions that define the expert knowledge in each scenario. Once, these sequences and temporal restrictions are defined, the algorithm generates automatically the chronicles. This algorithm works without problems for chronicles obtained from sequences with less of 15 occurrences of events. But when the sequences contain more than 15 event occurrences and there are repetitions of event types there, the algorithm works with limitations due to the complexity of the Cartesian product on the $HCDAM$.

To implement the chronicle database on the DCS system requires the conversion of the chronicles to a set of predicates. Chronicle recognition system corresponds to these sets of predicates; therefore, it is necessary a conversion of the parameters used in the predicates to a standard language like Ladder, Grafcet, Visual Basic or C++ that are commonly used in the HMI supervisory systems. However, it is necessary a license to work with a CRS - Chronicle Recognition System; moreover, this tool has never been applied in safety tools of the petrochemical sector.

Carry out a new element in safety systems requires guaranteeing its reliability, efficiency, and trust because the safety that is at stake not only is in the process but also in many people. In improving the reliability of industrial systems, the key point lies in human-machine interaction; nevertheless, it is obvious that it is much more complicated to normalize the behavior of the man than the behavior of a machine. This is a difficulty that has given rise to many lines of multidisciplinary research, especially in those industrial sectors in which the impact of possible human errors is high: nuclear plants, aviation, and the chemical industry.

## 7.5 Conclusion

This chapter presented two practical examples of the proposal "Chronicle Based Alarm management". In the first example, the Hydrostatic Tank Gauging system was described and the three steps of the methodology CBAM were applied. For the Chronicle database construction, the version extended of the $HCDAM$ was used, furthermore we described the chronicles in three scenarios. The evaluation of the chronicle $C_{11}^{1}$ tested this temporal pattern using a sequence of evaluation, which at the end generates a SUPER ALARM. In the second example, the vacuum oven system was described. This example contains expertise information that was included using the extended version of the $HCDAM$. The scenarios of normal startup, abnormal start-up, and normal shutdown were explained and temporal patterns represented as chronicles were exposed. In the process of evaluation of the chronicle $C_{11}^{2}$, we indicated how the occurrence frequency of each event type is included in the recognition of the chronicle. For instance, in this scenario, the event types $V1$ and $LF1$ occurred two times. With this methodology relevant information (SUPER ALARMS) can be given to the operators reducing a large number of alarms, increasing the reliability in the execution of the startup and shutdown procedures. This chapter ends with the discussion of how to implement the Chronicle Based Alarm Management methodology, expressing the principal requirements in each step.

# Conclusions and future work

A new methodology for alarm management of complex processes has been proposed. This methodology proposes a diagnosis process as a support to the operators during startup and shutdown stages based on situation recognition. Situations to recognize correspond to normal and/or anormal process behaviors modeled by temporal patterns called *chronicles*.

The *Chronicle Based Alarm Management* relies on an hybrid modeling of the process under study. This hybrid causal model captures the hybrid features of the process and also the causal relations between the continuous variables according to the procedural actions performed by the process operators. To design the chronicles we propose to use learning techniques. The Heuristic Chronicle Discovery Algorithm Modified $HCDAM$ learns chronicles from a set of event sequences obtained by simulation representing particular process behaviors. The event types of the representative sequences are issued from the hybrid causal modeling by an abstraction of the continuous variables evolution relatively to alarms limits.

The *Chronicle Based Alarm Management* is structured in three main steps: Event types determination, Learning event sequences generation and Chronicle database learning. The whole approach has been illustrated on two real case studies from the Cartagena Refinery: an Hydrostatic Tank Gauging system and a vacuum oven system.

Extensions of the original $HCDAM$ have been proposed. The first extension aims to integrate expert knowledge during the learning process through *temporal restrictions*. The second extension consists in considering the frequency of each event type in the representative sequences. These two extensions allow to reduce the number of learned chronicles and then the number of chronicle instances completely recognized in an input event flow, i.e the conservatism of learned chronicles.

**Future work**

The work presented in this thesis opens several promising directions for future researches.

In a first time, it would be interesting to test and validate the chronicle database by taking into account the operator feedbacks, so that to evaluate the acceptability of the proposal. This validation can be developed on the pilot plants of the ICP (Instituto Colombiano del petroleo) and on the laboratories of the Andes University.

A new model of chronicles will be proposed including the simultaneous occurrence of events, repetition of the events and to analyze the variability of the materials included into the process.

Currently the approach has been applied on two case studies that constitute two parts of the Cartagena Refinery but it has to be extended to the whole process by integrating the different areas of the system. For this and in the objective to transpose the approach to large scale systems, CBAM would benefit from a decentralized or distributed approach in which the notion of sub-chronicles would be exploited. This induces a new definition of the chronicles to introduce communication aspects through for instance the notion of shared events. This can also state a problem of communication delays in the information exchange between chronicles.

From a chronicle learning point of view several extensions could be addressed. One interesting problem is to extend the chronicle learning algorithm by integrating notably negative examples that could reduce the conservatism of the learned chronicles. Forgetting capabilities would also be an interesting feature to integrate into the learning algorithm $HCDAM$.

This is important to take the structural evolutions of the process (e.g change of components, degradation or aging of a component, ...) into account without a complete reconstruction that is to say a new learning of the chronicle database. Finally the problem of integrating into the chronicles forbidden events could be considered. Indeed, a forbidden event corresponding to the *no-event* predicate in the chronicle language description allows to design exclusive chronicles that is to say chronicles that cannot be recognized by the same input flow. This property of exclusiveness is challenging for diagnosis purposes as it can permit to conclude with certainty on a fault occurrence associated to a recognized chronicle. Another issue concerns the training event sequences. It would be interesting to consider Hazard and Operability studies or Event Tree Analysis for the determination of the scenarios and the event types identification.

Finally, another perpective is the exploitation of the chronicle recognition as a *super-alarm* (see Fig. 7.44) generator providing to the operators relevant information about the process situation. A new extension of the layer of protection corresponding to a diagnosis step based on chronicle recognition should have to be integrated in the

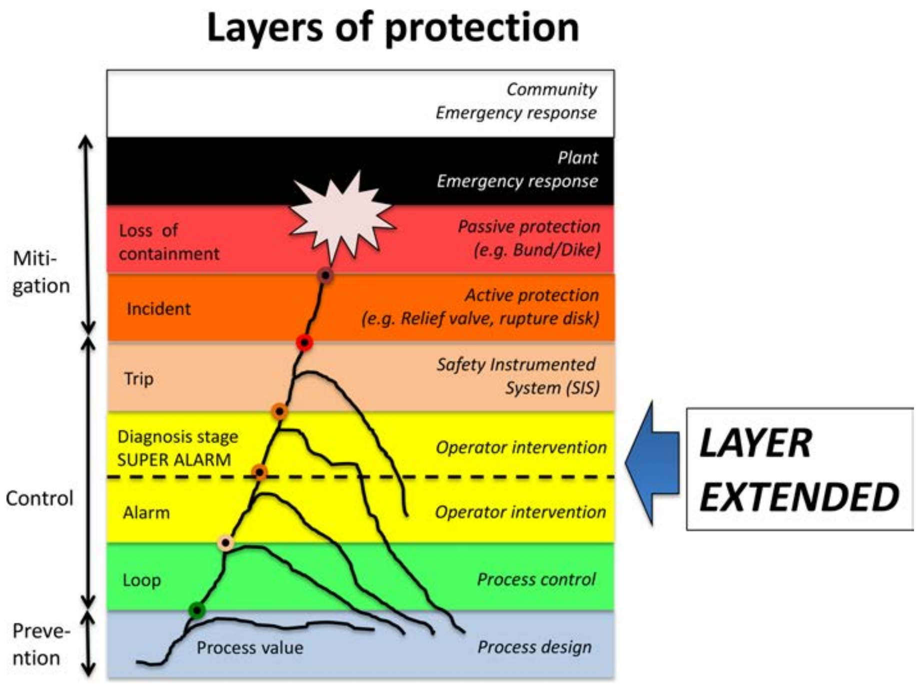


Figure 7.44: SUPER ALARM layer of protection

global safety structure increasing the reliability of the layer of protection related to the operator intervention.

# Bibliography


[1] Agudelo, C. (2015). Integración de técnicas y las secuencias de alarmas para la detección y el diagnostico de fallos. *Doctoral Thesis, Universidad Politecnica De Valencia - Spain.*

[2] Astolfi, A. and Praly, L. (2006). Global complete observability and output- to-state stability imply the existence of a globally convergent observer. *Mathematics of Control Signals and Systems, Vol:18, ISSN:0932-4194, Pages:32-65.*

[3] Bahr, N. J. (2015). System safety engineering and risk assessment. *Second ed., CRC Press, 2015.*

[4] Basseville, M. (1988). Detecting changes in signals and systems: a survey. *Automatica 24(3), 309-326.*

[5] Bayoudh, M., Travé-Massuyès, L., and Olive, X. (2006). Hybrid systems diagnosability by abstracting faulty continuous dynamics. *In Proc. of the 17th International Workshop on Principles of Diagnosis, pages 915, 2006.*

[6] Beebe, D., Ferrer, S., and Logerot, D. (2013). The connection of peak alarm rates to plant incidents and what you can do to minimize. *Process Safety Progress.*

[7] Benayadi, N., Le.Goc, M., and Bouché, P. (2006). Approche entropique pour l'analyse de modèle de chroniques. *EGC, volume RNTI-E-6 of Revue des Nouvelles Technologies de l'Information, page 511-516. Cépaduès-Éditions, (2006).*

[8] Bittencourt, A., Saarinen, K., and Sander-Tavallaey, S. (2012). A data-driven method for monitoring systems that operate repetitively - applications to wear monitoring in an industrial robot joint. *8th IFAC Symp. Safeprocess'2012 (2012) 198-203.*

[9] Blalock, H. (1964). Causal inferences in nonexperimental research. *Chapel Hill:University of North Carolina Press.*

[10] Brennan, R. (2007). Toward real-time distributed intelligent control: A survey of research themes and applications. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 37, no. 5, pp. 744–765.*

[11] Brennan, R., Fletcher, M., and Norrie, D. (2002). An agent-based approach to reconfiguration of real-time distributed control systems. *IEEE Trans. Robot. Autom., vol. 18, no. 4, pp. 444–451, Aug. 2002.*

[12] Carrault, G., Cordier, M., Quiniou, R., and Wanga, F. (2003). Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms. *Artificial Intelligence in Medicine, 28(3):231–263.*

[13] Carrault, G., Cordier, M., R. Quiniou, M. G., Bellanger, J., and Bardou., A. (1999). A model-based approach for learning to identify cardiac arrhythmias. *Artificial Intelligence in Medicine and Medical Decision Making, volume 1620 of LNAI, pages 165–174, Aalborg, Denmark. Springer Verlag.*

[14] Cassar, J. and Staroswiecki, M. (1997). A structural approach for the design of failure detection and identification systems. *Proc. of the IFAC Symposium on Control of Industrial Systems,Belfort.*

[15] Cauvin, S., Braunschweig, B., Galtier, P., and Glaize, Y. (1992). Alexip: an expert system coupled with a dynamic simulator for the supervision of the alphabutol process. *Process. Oil & Gas Science and Technology - Rev. IFP Vol. 47 (1992), No.3, pp. 375-382.*

[16] CCPS (2008). *Guidelines for Hazard Evaluation Procedures, (Center for Chemical Process Safety), New York, Wiley.* Wiley.

[17] Celse, B., Cauvin, S., Heim, B., Gentil, S., and Travé-Massuyès, L. (2005). Model based diagnostic module for a fcc pilot plant. *Oil & Gas Science and Technology – Rev. IFP, Vol. 60 (2005), No. 4, pp. 661-67.*

[18] Chen, J. and Patton, R. (1999). Robust model-based fault diagnosis for dynamic systems. *Massachusetts: Kluwer Academic Publishers.*

[19] Chen, Y. and Lee, J. (2011). Autonomous mining for alarm correlation patterns based on time-shift similarity clustering in manufacturing system. *2011 IEEE International Conference on Prognostics and Health Management.*

[20] Cheung, J. and Stephanopoulos, G. (1990). Representation of process trends part i: A formal representation framework. *Computers & Chemical Engineering,14 , 495-510.*

[21] Chow, E. and Willsky, A. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control 29 (7), 603-614.*

[22] Cordier, B., Dumas, P., Levy, F., Montmain, F., and Travé-Massuyès, L. (2000). A comparative analysis of ai and control theory approaches to model-based diagnosis. *ECAI' 00 Congress.*

[23] Cordier, M. and Dousson, C. (2000). Alarm driven monitoring based on chronicles. *4th Sumposium on Fault Detection Supervision and Safety for Technical Processes (SafeProcess), pages 286–291, Budapest, Hungary.*

[24] Cox, S. and Tait, R. (2008). *Reliability, Safety, and Risk Management.* John Wiley & Sons, Ltd.

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

[26] Dash, S., Rengaswamy, R., and Venkatasubramanian, V. (2003). Fuzzy-logic based trend classification for fault diagnosis of chemical processes. *Computers & Chemical Engineering*, 27(3):347–362.

[27] Devold, H. (2013). *Oil and Gas Production Handbook: An Introduction to Oil and Gas Production*. SRH Media, ISBN 9781105538643.

[28] Ding, S. (2008). Model-based fault diagnosis techniques design schemes, algorithms, and tools. *ISBN 978-3-540-76303-1. Springer 2008.*

[29] Ding, S., Wang, Y., Yin, S., Zhang, P., Yang, Y., and Ding, E. (2012). Date-driven design of fault-tolerant control systems. *Proc. 8th IFAC Symp. Safeprocess'2012 (2012) 1323-1328.*

[30] Dion, J., Commault, C., and der Woude, J. V. (2003). Generic properties and control of linear structured systems: a survey. *Automatica, 39, 1125-1144.*

[31] Dousson, C. (1994). Suivi d'évolution et reconnaissance de chroniques. *Thése de doctorat, Université Paul Sabatier.*

[32] Dousson, C. (1996). Alarm driven supervision for telecommunication network: On-line chronicle recognition. *In Annales Des Télécommunications, volume 51, pages 501–508.*

[33] Dousson, C., Gaborit, P., and Ghallab, M. (1993). Situation recognition: representation and algorithms. *in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93), Chambéry, France, 1993, pp. 166– 172.*

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

[35] Duncan, O. (1975). Introduction to structural equation models. *New York: Academic.*

[36] Elwert, F. (2013). Chapter 13: Graphical causal models. *Handbook of Causal Analysis for Social Research,Springer.*

[37] Fernandez, I., Camacho, A., Gasco, C., Macias, A., and M.A. Martin, G. Reyes, J. R. (2012). *Seguridad funcional en instalaciones de proceso: sistemas, instrumentados de seguridad y análisis SIL*. Ediciones Díaz de Santos, S.A.

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

[39] Fessant, F., Clérot, F., and Dousson, C. (2004). Mining of an alarm log to improve the discovery of frequent patterns. *Industrial Conference on Data Mining, 2004, pp. 144–152.*

[40] Finch, F. and Kramer, M. (1987). Narrowing diagnostic focus using functional decomposition. *American Institute of Chemical Engineers Journal 34 (1), 130/140.*

[41] Floyd, M., Bicakci, M., and Esfandiari, B. (2012). Case-based learning by observation in robotics using a dynamic case representation. *FLAIRS Conference, 2012.*

[42] Floyd, M. and Esfandiari, B. (2011). A case-based reasoning framework for developing agents using learning by observation. *ICTAI, 2011, pp. 531–538.*

[43] Ford, L. and Fulkerson, D. (1956). Maximal flow through a network. *Can. J. Math., 8, 399-404.*

[44] Frank, P. and Ding, X. (1997). Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403 – 424.

[45] Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy a survey and some new results. *Automatica, vol. 26, no. 3, pp. 459–474.*

[46] Garcia, E., Agudelo, C., and Morant, F. (2012). Secuencias de alarmas para detección y diagnóstico de fallos. *Universidad Politécnica de Valencia, 460022 –Spain. 3er Congreso internacional de ingeniería mecatrónica. UNAB 2012.*

[47] Guerraz, B. and Dousson, C. (2004). Chronicles construction starting from the fault model of the system to diagnose. *Proc. of the 15th Int. Workshop on Principles of Diagnosis (DX'04), pp 51-56.*

[48] Guyet, T. and Quiniou, R. (2008). Mining temporal patterns with quantitative intervals. *ICDM Workshops, 2008, pp. 218–227.*

[49] Guyet, T. and Quiniou, R. (2011). Extracting temporal patterns from interval-based sequences. *IJCAI, 2011, pp. 1306–1311.*

[50] Habibi, E. and Hollifiled, B. (2006). Alarm systems greatly affect offshore facilities amid high oil prices. *World Oil, septiembre de 2006, págs. 101-105.*

[51] Heim, B., Cauvin, S., and Gentil, S. (2001). A fuzzy and causal reasoning methodology coupled with an heuristic approach for fault diagnosis on a fcc pilot process. *4th Workshop on On- Line Fault Detection and Supervision in the Chemical Process Industries, Seoul.*

[52] Heintz, F. (2001). Chronicle recognition in the WITAS UAV project a preliminary report. In *Swedish AI Society Workshop (SAIS2001).*

[53] Iwasaki, S. and Simon, H. (2003). Modèles et raisonnements qualitatifs. *In: Trait IC2 Information, Commande, Communications. Hermés.*

[54] Janusz, M. and Venkatasubramanian, V. (1991). Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. *Engineering Applications of Artificial Intelligence*, 4(5):329 – 339.

[55] Khalgui, M., Mosbahi, O., Li, Z., and Hanisch, H. (2011). Reconfiguration of distributed embedded-control systems. *IEEE/ASME Trans. Mechatronics, vol. 16, no. 4, pp. 684–694.*

[56] Kramer, M. and Palowitch, J. B. (1987). A rule-based approach to fault diagnosis using the signed directed graph. *AIChE Journal, 33, 7, 1067-1078.*

[57] Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence 29 (3), pp. 289-338.*

[58] Le.Goc, M. and Bouché, P. (2004). Discovering operational signatures with time constraints from a discrete event sequence. *IEEE Explorer, Hybrid Intelligent Systems, 2004. HIS '04. Fourth International Conference.*

[59] Le.Goc, M., Bouché, P., and Coinu, J. (2008). A global model of sequences of discrete event class occurrences. *Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS'08) 12 - 16, June 2008, Barcelona, Spain.*

[60] Lew, J., Juang, J., and Keel, H. (1994). Quantification of parametric uncertainty via an interval model. *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM. 01/1994; 17(6):1212-1218.*

[61] Leyval, L., Gentil, S., and Feray-Beaumont, S. (1994). Model based causal reasoning for process supervision. *Automatica, 30, 8, 1295-1306.*

[62] Lind, M. (1991). Abstraction for modeling diagnostic strategies. *In IFAC workshop on computer software structures integrating AI/KBS systems in process control.*

[63] Liu, Y., Xie, G., Yang, Y., Chen, Z., and Chai, Q. (2014). Hierarchical method of fault diagnosis based on extended sdg. pages 3808–3812.

[64] Magni, L., Scattolini, R., and Rossi, C. (2000). A fault detection and isolation method for complex industrial systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans ( Volume: 30, Issue: 6, Nov 2000).*

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

[66] Maurya, M., Rengaswamy, R., and Venkatasubramanian, V. (2003). A systematic framework for the development and analysis of signed digraphs for chemical processes. *A. Algorithm and Analysis. Ind. Eng. Chem. Res., 42, 4789-4810.*

[67] Maurya, M., Rengaswamy, R., and Venkatasubramanian, V. (2007). A signed directed graph and qualitative trend analysis-based framework for incipient fault diagnosis. *Chemical Engineering Research and Design, Vol. 85, No.10, 1407-1422, 2007.*

[68] Mayer, E. (1994). Inductive learning of chronicles. *European Conference on Artificial Intelligence, 1998, pp. 471–472.*

[69] Mehrabi, M. G., Ulsoy, G., and Kore, Y. (2000). Reconfigurable manufacturing systems: Key to future manufacturing. *Journal of Intelligent Manufacturing, Volume 11, Issue 4, pp 403–419.*

[70] Menkhaus, G. and Andrich, B. (2005). Metric suite for directing the failure mode analysis of embedded software systems. *Inf. Syst. J. [Online]. pp. 266–273.*

[71] Mitsa, T. (2010). Temporal data mining. *CRC Press, 2010.*

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

[73] Murota, K. (2010). Matrices and matroids for system analysis. *Springer (Algorithms and Combinatorics). Series Volume 20. ISBN 978-3-642-03993-5.*

[74] Nan, C., Khan, F., and Tariq-Iqbal, M. (2008). Real-time fault diagnosis using knowledge-based expert system. *Process Safety and Environmental Protection, Volume 86, Issue 1, January 2008, Pages 55–71.*

[75] Nomikos, P. and MacGregor, J. (1994). Monitoring batch processes using multiway principal component analysis. *American Institute of Chemical Engineers Journal 40 (8), 1361/1375.*

[76] Palomeque, D. (2005). Enfoque integral y herramientas de gestión. *Petrotecnia, Vol 46, pp. 42-48.*

[77] Patton, R. J. and Chen, J. (1997). Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice, vol. 5, no. 5, pp. 671–682.*

[78] Pentti, H. and Atte, H. (2002). Failure mode and effects analysis of software-based automation systems. *VTT Ind. Systems STUKYTOTR 190 [Online]. (09), p. 37.*

[79] Pons, R., Subias, A., and Travé-Massuyès, L. (2015). Iterative hybrid causal model based diagnosis: Application to automotive embedded functions. *Engineering Applications of Artificial Intelligence*, 37:319 – 335.

[80] Porté, B. H., Boucheron, S., Sallantin, S., and Arlabosse, F. (2003). Approche ensembliste et par logique floue pour le diagnostic causal de procédés de raffinage - application à un pilote de fcc. *Ph.D Thesis, INPG Grenoble.*

[81] Porté, N., Boucheron, S., Sallantin, S., and Arlabosse, F. (1988). An algorithmic view at causal ordering. *Proc. of the 2nd International Workshop on Qualitative Physics QR'88, Paris.*

[82] Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man and Cybernetics 15 (2), 234/243.*

[83] Rengaswamy, R. and Venkatasubramanian, V. (1995). A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engineering Applications of Artificial Intelligence*, 8(1):35 – 51.

[84] Rengaswamy, R. and Venkatasubramanian, V. (2000). A fast training neural network and its updation for incipient fault detection and diagnosis. *Computers and Chemical Engineering*, 24:431 – 437.

[85] Robins, J. and Richardson, T. (2011). Alternative graphical causal models and the identification of direct effects. *Causality and psychopathology: Finding the determinants of disorders and their cures (pp. 103–158). New York: Oxford University Press.*

[86] Sacks, E. (1988). Qualitative analysis of piecewise linear approximation. *Journal of Artificial Intelligence in Engineering 3 (3), pp 151-155.*

[87] Samantaray, A., Medjaher, K., Ould-Bouamama, B., Staroswiecki, M., and Dauphin-Tanguy, G. (2006). Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice and Theory*, 14(3):237 – 262.

[88] Sánchez, M. (2010). Introducción a la confiabilidad y evaluación de riesgos, bogotá.

[89] Sarmiento, H. and Isaza, C. (2012). Identification and estimation of functional states in drinking water plant based on fuzzy clustering. *22st European Symposium on Computer Aided Process Engineering. pp 1317 to 1327.*

[90] Sarmiento, H., Isaza, C., Kempowsky-Hamon, T., and LeLann, M. (2013). Estimacion de estados funcionales en procesos complejos con base en agrupamiento difuso. *Informacion tecnologica*, 24:79 – 98.

[91] Shui, A., Chen, W., Zhang, P., Hu, S., and Huang, X. (2009). Review of fault diagnosis in control systems. *978-1-4244-2723-9/09/c 2009 IEEE.*

[92] Shumsky, A. (2007). Data driven method for fault detection and isolation in nonlinear uncertain systems. *Proc. IFAC Conf. on Control Applications in Marine Systems, 2007.*

[93] Stanley, G. and Vaidhyanathan, R. (1998). Generic fault propagation modeling approach to on-line diagnosis and event correlation. *3rd IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries. June 4-5 Solaize, France.*

[94] Staroswiecki, M. and Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebric dynamic systems. *Automatica, 3, 5, 697-699.*

[95] Stauffer, T., Sands, N., and Dunn, D. (2000). Alarm management and isa-18 a journey, not a destination. *Texas A & M Instrumentation Symposium.*

[96] Subias, A., Travé-Massuyès, L., and Le.Corronc, E. (2014). Learning chronicles signing multiple scenario instances. *In IFAC World Congress.*

[97] Tixier, J., Dusserre, G., Salvi, O., and Gaston, D. (2002). Review of 62 risk analysis methodologies of industrial plants. journal of loss prevention in the process industries 15 (2002) 291–303.

[98] Travé-Massuyès, L. and Dague, P. (2003). Modèles et raisonnements qualitatifs. *In: Trait IC2 Information, Commande, Communications. Hermés.*

[99] Travé-Massuyès, L., Escobet, L., Pons, R., and Tornil, R. (2001). The ca-en diagnosis system and its automatic modeling method. *Computacion in Sistemas Journal, 5, 2, 128-143.*

[100] Travé-Massuyès, L. and Pons, R. (1997). Causal ordering for multiple mode systems. *Proc. of the 11th Int. Workshop on "Qualitative Reasoning about Physical Systems", Cortona.*

[101] Treyer, D. and Zogg, D. (2013). Model and expert knowledge based fault diagnosis for a heat exchanger. *IEEE - Conference on Control and Fault-Tolerant Systems (SysTol) 9-11 Oct. Nice, France.*

[102] Ulerich, N. and Powers, G. (1988). Online hazard aversion and fault diagnosis in chemical processes: the digraph +fault tree. *IEEE Transactions on Reliability 37 (2), pp 171-177.*

[103] Unal, M., Onat, M., Demetgul, M., and Kucuk, H. (2014). Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network. *Measurement*, 58:187 – 196.

[104] Ungauer, C. (1993). Problematique d'utilisation de techniques de supervision a base de connaissances profondes: L'exemple de la supervision du réseau transpac. *Rapport technique, CNET, Octobre 1993.*

[105] Unger, J., Kroner, A., and Marquadt, W. (1995). Structural analysis of differential-algebraic equation systems-theory and applications. *Computers chem. Eng, 19, 8, 867-882.*

[106] Uribe, C. A. and Isaza, C. (2012). Expert knowledge-guided feature selection for data-based industrial process monitoring. *Revista Facultad de Ingenieria Universidad de Antioquia*, pages 112 – 125.

[107] Vaidhyanathan, R. and Venkatasubramanian, V. (1995). Digraph-based models for automated hazop analysis. *Reliability Engineering and Systems Safety 50 (1), 33/49.*

[108] Vásquez, J., Prada, J., Agudelo, C., and Jimenez, F. (2013). Analysis of alarm management in startups and shutdowns for oil refining processes. *IEEE Explorer. Engineering Mechatronics and Automation (CIIMA), 2013 II International Congress, Bogota.*

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

[110] Vásquez, J., Travé-Massuyès, L., Subias, A., Jimenez, F., and Agudelo, C. (2016). Alarm management based on diagnosis. *4th IFAC International Conference on Intelligent Control and Automation Sciences (ICONS 2016), Reims.*

[111] Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003). A review of process fault detection and diagnosis. part i: Quantitative model-based methods. *Computers and Chemical Engineering, 27 (2003) 293-311.*

[112] Venkatasubramanian, V., Vaidyanathan, R., and Yamamoto, Y. (1990). Process fault detection and diagnosis using neural networks part i: Steady state processes. *Computers and Chemical Engineering, 14 (2003) 699-712.*

[113] Vries, R. (1990). An automated methodology for generating a fault tree. *IEEE Transactions on Reliability.*

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

[115] Wang, Z., Zhu, C., Niu, Z., Gao, D., and Feng, X. (2014). Monitoring batch processes using multiway principal component analysis. *Knowledge-Based Systems 6583-95.*

[116] Wei, C., Rogers, W., and Mannan, M. (2008). Layer of protection analysis for reactive chemical risk assessment. journal of hazardous materials. vol 159, pg 19–24.

[117] Wilcox, N. and Himmelblau, D. (1994a). Possible cause and effect graphs (pceg) model for fault diagnosis i. methodology. *Computers and Chemical Engineering 18 (2), 103/116.*

[118] Wilcox, N. and Himmelblau, D. (1994b). Possible cause and effect graphs (pceg) model for fault diagnosis ii. applications. *Computers and Chemical Engineering 18 (2), 117/127.*

[119] Willsky, A. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica, 12, 601-611.*

[120] Yang, F. and Xiao, D. (2012). Progress in root cause and fault propagation analysis of large scale industrial processes. *Journal of Control Science and Engineering. Volume 2012 (2012), Article ID 478373, 10 pages.*

[121] Yang, F., Xiao, D., and Shah, L. (2010). Qualitative fault detection and hazard analysis based on signed directed graphs for large-scale complex systems. *Tsinghua University University of Alberta China Canada 2010, ISBN 978-953-307-037-7.*

[122] Yélamos, I. (2008). A global approach for supporting operators decision-making dealing with plant abnormal events. *Doctoral Thesis, Universidad Politecnica de Cataluña – Barcelona.*

[123] Yu, J. (2012). A particle filter driven dynamic gaussian mixture model approach for complex process monitoring and fault diagnosis. *Journal of Process Control*, 22(4):778 – 788.

[124] Zhirabok, A. and Pavlov, S. (2014). Data-driven method of fault detection in technical systems. *25th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM 2014, Procedia Engineering 100 ( 2015 ) 242 – 248.*

[125] Zhu, J., Shu, Y., Zhao, J., and Yang, F. (2013). A dynamic alarm management strategy for chemical process transitions. *Journal of Loss Prevention in the Process Industries. Volume 30, July 2014, Pages 207-218*.

[126] Zolghadri, A., Cieslak, J., Efimov, D., Henry, D., Goupil, P., Dayre, R., Gheorghe, A., and Leberre, H. (2015). Signal and model-based fault detection for aircraft systems. *IFAC PapersOnLine. 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2015. Paris, 2–4 September 2015*, 48(21):1096 – 1101.