Enhanced chronicle learning for process supervision
John William Vásquez, Louise Travé-Massuyès, Audine Subias, Fernando Jimenez

To cite this version:
John William Vásquez, Louise Travé-Massuyès, Audine Subias, Fernando Jimenez. Enhanced chronicle learning for process supervision. 20th IFAC WORLD CONGRESS, Jul 2017, Toulouse, France. pp.5191-5196. hal-01847010

HAL Id: hal-01847010
https://hal.laas.fr/hal-01847010
Submitted on 23 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Enhanced chronicle learning for process supervision

J. W. Vásquez*,** L. Travé-Massuyès* A. Subias* F. Jimenez**

* LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France.
  e-mail: (jwvasque@laas.fr, louise@laas.fr, subias@laas.fr)
** GIAP, Engineering department, Andes University
  Bogotá, Colombia. fjimenez@uniandes.edu.co

Abstract: Process alarm management can be approached as a pattern recognition problem in which temporal patterns are used to characterize different typical situations, particularly at startup and shutdown stages. This paper focuses on learning the temporal patterns, in the form of chronicles, by extending the previously proposed Heuristic Chronicle Discovery Algorithm Modified HCDAM. The proposed extension incorporates knowledge, in particular in the form of so called temporal runs, to focus the learning process and produce less conservative chronicles. The resulting Chronicle Based Alarm Management (CBAM) approach is hence based on a diagnosis process which permits situation recognition and provides the operators with relevant information about the failures inducing alarms flows in the startup and shutdown stages. The event sequences that represent a process situation are generated by simulation and including temporal runs, the chronicles are extracted using the extended version of HCDAM. Finally, the conclusion and future work are presented.

Keywords: Alarm management, Diagnosis, Chronicles, Pattern recognition

1. INTRODUCTION

In industrial environments, it is common for the plant operators to perform their duties by silencing process alarms. This is so because these alarms are too numerous and become noise rather than helping to detect and diagnose abnormal situations. Currently, it is very easy to set alarms on modern electronic control systems, and operators are inundated with “alarms” that do not help them to perform their tasks but, on the contrary, hinder them (Stanley and Vaidhyanathan (1998)). However, the plant alarms should be administered according to: a philosophy that includes the purpose of the alarm system; procedures associated with the alarm system and other plant procedures; methods for prioritization; alarm classes; roles and operator responsibilities with respect to alarms; principles of the alarms; documentation required for each alarm; training; rates of key system performance; change management, and preservation of history of alarms (ISA 18.2).

The nature of alarms and other procedure actions makes that many of the aspects of industrial processes can be captured by discrete events. The supervision problem thus must fit the formal recognition of behaviors in the context of Complex Event Processing (CEP). This corresponds to developing reliable tools that support the analysis of event streams to recognize activities that can generate normal or abnormal situations in complex flows. The dynamics of a process can be represented by an approach that depicts the process behavior using the events that occurs. Then, identifying the transition from normal operation to abnormal operation corresponds to detect symptoms. For do this, the systems must be monitored and the diagnostic algorithms are therefore based on observations that the system provides. It must have a sufficient set of strategically placed sensors, providing useful indicators. This information is then used to reconstruct the state of the system and this observations coupled with a diagnostic algorithm allow to introduce the concept of diagnosability. A system is diagnosable with a diagnostic algorithm if it detects with certainty the occurrence of faults in the system, and is able to discriminate them in finite time. The formal definition of diagnosability may be different if one is interested in continuous systems (Basseville (2001)) or discrete events (Pencole (2004)). Diagnosability provides an efficient criterion to assess the relevance of the observations in each situation.

In this context, chronicles (Dousson (1996),Ghallab (1996)) have been applied in many diagnosis applications but one of the biggest difficulties is to obtain automatically a base of chronicles that represents each situation. To obtain relevant chronicles from a set of event sequences representing a given situation, it is often necessary incorporate expert knowledge. This paper enhances the results of the chronicle learning algorithm proposed in (Subias et al. (2014)) by incorporating expert knowledge in the form of temporal runs, as well as additional information that allows us to limit the conservatism of chronicles.

The paper is divided into 5 sections. After the introduction, Section 2 presents the chronicles representation with its formal concepts. Section 3 presents the learning approach. An extension of the HCDAM is described with
an example. In the section 4, a case study related with an
illustrative application in the petrochemical sector is used
to apply the tree steps of the CBAM. Finally, in Section
5 the conclusion and future work are presented.

2. CHRONICLE REPRESENTATION
A chronicle is a set of events linked by temporal constraints
and the occurrence of which is subject to a certain context.
A chronicle is expressed as a constraint graph in which
events are represented by nodes and time constraints are
the labels of the arcs that connect the nodes. Time is
represented as a linearly ordered discrete set of instants,
whose granularity is fine enough compared to the dynamics
of the environment and the means of observation. The
"Chronicle Recognition System" named CRS was intro-
duced in Dousson (1996) and extended in Dousson (2002).
A system of chronicles is composed of three parts:
(1) A set of predicates
(2) A set of temporal constraints concerning these pred-
icates
(3) A set of actions to apply when chronicle is recognized
(optional).

The latter part is used to react following the recognition
of a chronicle; for example, the reaction may be a main-
tenance operation. This part does not concern the subject
of training and therefore is not addressed in the following.
To recognize chronicles, CRS must detect all subsets of
occurrences of events in the input stream. These subsets
correspond to all the sets of patterns of the chronicle taking
into account the various constraints that compose it. To
achieve this result, CRS creates partial instances. A partial
instance corresponds to a partial subset of the chronicle
event pattern. It is only when this subset is complete that
the chronicle is recognized.

2.1 Concepts and formalization

Before defining a chronicle, it is good to remember that it
is based on the verification of time constraints. The check
is directly inspired by a Simple Temporal Problem (STP)
introduced by Dechter et al. (1991). The occurrence of
different events in time represents the system’s dynamics
in a given situation. 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 event types. Note $\sigma$
an event type where $\sigma \in E$. We can thus define the concept of
event.

Definition 1: An event $e$ is defined as a pair $e = (\sigma, t)$,
where $\sigma \in E$ is an event type and $t$ is a variable of integer
type called the event date.

Several events can have the same event type, but do not
necessarily have the same date, for instance $e_1 = (a, 3)$
and $e_2 = (a, 6)$ are two events carrying the same type of
event $a$.

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

Definition 2: A sequence is defined as an ordered set of
events denoted $S = \langle e_i \rangle$ where $i \in N$, $i = 1, \ldots, l$ and
$t_i < (t_{i+1})$, $i = 1, \ldots, l - 1$ where $l$ is the length of the time
sequence $S$.

An example of sequence representing an activity stream
may be given by a sequence $S_1 = \langle e_1, e_2, e_3, e_4, e_5, e_6 \rangle =
\{(a, 2), (b, 4), (c, 5), (a, 8), (b, 9), (a, 10)\}$ with $l_1 = 6$.

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

Definition 3: A chronicle is defined as a triplet $C =
(\xi, T, G)$ such that:

• $\xi \subseteq E$. Where $\xi$ is called the typology of the chronicle.
• $T$ is the set of temporal constraints of the chronicle.
• $G = (V, A)$ is a directed graph where:
  • $V$ represent the event types of $\xi$
  • The arcs $A$ represent the different time con-
    straints between event types. The edge $i \rightarrow j$
is associated to the constraint $\tau_{ij} \in T$

Considering the two events $(e_1, t_1)$ and $(e_2, t_2)$, we define
the time interval as the pair $\tau_{ij} = [t^-, t^+], \tau_{ij} \in T$
corresponding to the lower $t^-$ and upper $t^+$ bounds on
the temporal distance between the two event dates $t_1$ and
$t_2$. If the event $e_2$ occurs after $e_1$, then it exists a directed
link $A$ from $e_1$ to $e_2$ associated with a time constraint.

A chronicle $C = (\xi, T, G)$ is recognized in a given sequence
$S$ when $\xi \subseteq \xi' \subseteq (\xi'$ set of event types associated to $S$ ) and
all temporal constraints $T$ are satisfied. $G_{inst} = (\xi, T_c)$
where $T_c$ is a valuation of $T$, is an instance of $C$.

3. LEARNING CHRONICLES
The chronicle exploration process corresponds in discover-
ing all the chronicles, whose instances occur in a given
sequence. In many cases, the same situation does not
implies perfectly identical sequences. This is the reason
of that $HCDAM$ uses several sequences as input.

3.1 Learning chronicles with $HCDAM$
The algorithm $HCDAM$ learns the chronicles, whose
instances occur in all event sequences representing the
same situation (Subias et al. (2014)). Given a set of
sequences $S$ and a minimum frequency threshold, it finds
all minimal frequent chronicles present in all sequences.
The chronicle learning algorithm has the following three
phases:

(1) Filtering
(2) Building a constraint database from the input se-
quences.
(3) Generating a set of candidate chronicles.

3.1.1. Phase 1: The filtering operation is a preliminary
process on sequences and it can be summarized as follows:

• Filter the event types that are not present in all input
  sequences $S$; if $\exists S_k \in S$ such as $\exists \sigma_i \notin S_k$, then remove
  $\sigma_i$ from all the other sequences in $S$.
• Filter on a given set of event types $\Psi = \{\sigma_1, \sigma_2, \ldots, \sigma_l\}$
  if we are only interested in those event types for the
  learning process.
3.1.2. Phase 2: Phase two builds a constraint data base \( \mathbb{D} \) by considering each pair of event types and its temporal constraints in a constraint graph with tree structure.

In this constraint tree, time constraints are nodes and arcs represent the relationship is parent of defined as:

**Definition 5:** A node \( e_i[t^-, t^+]e_j \) corresponding to the time constraint between \( e_i \) and \( e_j \) 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 \([t^-, t^+] \subset [t^-, t^+] \subset [t^-, t^+] \).

3.1.3. 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 (Cram et al. (2012)).

- The set of candidates initiates with the set of tree roots
- Use the operator "add \( \varepsilon \)". This operator, checks at the constraint graphs in order to find the restrictions of \( \varepsilon \) with all elements of \( E \)
- Determine the minimal number of occurrences of the candidate in \( S \)

The details of the algorithm HDCAM can be found in (Subias et al. (2014)).

3.2 Improving restrictiveness/conservatism

3.2.1. Integration of process knowledge Expert knowledge can be represented by temporal runs that express a known time constraint between two event type dates. These temporal runs are gathered in an expert data base \( D_e \). To integrate this knowledge, Phase 2 of HDCAM is modified. One first checks the existence of a temporal run \( TR_{ij} \) for each pair of event types \((e_i, e_j)\). Temporal run, which replaces the tree root of this pair of event types. The effect of the integration of temporal runs corresponds to focus the learning process and produce less conservative chronicles; it means that the number of chronicles that represents a specific scenario could be reduced using the expertise knowledge.

**Definition 6:** A temporal run is defined as a partial order of time points with time restrictions between each pair. For the pair of event types \((e_i, e_j)\), a temporal run is defined as \( TR_{ij} = e_i[t^-, t^+]e_j \).

*Example of the "charge oven" activity: the results of learning obtained without the inclusion of temporal runs and with the inclusion of temporal runs are given in a simple example of activity consisting in a charging an oven as represented in Figure 1. The event types are given by \( E = \{a, b, c, d\} \), where \( a \) is the event of detecting product \( a \) entering the oven, \( c \) is the event corresponding to putting the heaters on, and \( d \) is the event of setting the heater to high temperature. The event sequences that express normal startup of this process are:

\[
S_1 = \{(a, 2), (b, 4), (a, 5), (c, 7), (d, 11)\}
\]

\[
S_2 = \{(a, 2), (b, 3), (a, 4), (c, 7), (d, 10)\}
\]

\[
S_3 = \{(a, 2), (b, 3), (a, 5), (c, 8), (d, 11)\}
\]

The temporal runs that indicate the expertise knowledge are \( TR_{ab} = a[-2, 2]b \) and \( TR_{cd} = c[2, 6]d \). Using the

3.2.2. Integration of event information Another type of expert knowledge that is often available is the occurrence frequency \( f(e) \) of a single event \( e \). This information is not taken into account in HDCAM. Nevertheless it can be very useful to reduce the number of learned chronicles.

With the event \( \Phi \) we can obtain the occurrence frequency of each event type that occurs in \( S_k \). This occurrence frequency permits improved the restrictiveness in the system of chronicles; for example, in a scenario represented by 7 chronicles of frequency 1 and 3 chronicles of frequency 2, the use of the event \( \Phi \) enables to analyze the system with only one unique chronicle, not 10.

**Definition 7:** We define the event \( \Phi \) as the initial event in all the event sequences \( S \) such that the occurrence frequency \( f(e) \) for each event type \( e \) in the sequence \( S_k \) is determined from this event \( \Phi \).

Assuming the above example, we use the integration of event information for structure an unique chronicle reducing the number of chronicles in 90% see Figure 2. In this chronicle, the concept of frequency is assumed as the occurrence frequency for each event type in a sequence, not as the occurrence frequency of a pair of events.

4. CASE STUDY - VACUUM OVEN

This section presents a case study from the Cartagena Refinery in Colombia. Our proposal aims at helping the operator to recognize specific operation (i.e. normal and/or faulty operation) during the startup and shutdown stages.
of the vacuum oven unit described Figure 3. This section
includes the learning of the chronic base that will be
considered by a recognition system to recognize these
normal or faulty situations when they occur, and in the
next subsections the CBAM methodology is applied with
its three steps: 1. Event type identification, 2. Learning
event sequences generation and 3. Construction of the
chronicle database. Vasquez et al. (2015), Vasquez et al.
(2016).

4.1 Event type identification

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

- $\Sigma$ is the set of procedural actions performed during
standard operating procedures
- $\Sigma_e$ is the set of event types associated to the behavior
of the continuous variables.

Procedural actions concern mainly the valves of the oven
and $\Sigma = \{V1, V2, V3, v1, v2, v3, M2A\}$, where V1 (resp.
V2, V3) denotes the switch of the valve V1 (resp. V2, V3)
from closed to opened. v1 (resp v2, v3) the switch of the
valve V1 (resp V2, V3) from opened to closed. The event
M2A corresponds to the change from manual to automatic
operating, closing the control loops. In the reminder
we assume that this event is the only unobservable
event of the system i.e. $\Sigma_{uo} = \{M2A\}$.

Continuous variables are associated to the different flows
($F1, F2, F3$) and different temperatures ($T1, T2, T3$
and $T4$) (see Figure 3). For each evolution of a continuous
variable (obtained by simulation) upper and lower bounds
are defined from experience feedback. Then the continuous
values of the variable are mapped to ranges defined by
these limits. We propose to define three ranges: High when
the value of the continuous variable is higher than the
upper bound, Medium when the value is between the
upper and lower bounds, and Low when the value is under
the lower limit bound. Form this qualitative abstraction of
the continuous variable evolution we consider that an event is
generated each time a transition between qualitative domains
(i.e. ranges) occurs. So, for a continuous variable noted $i$, four event types can be considered: $L_{vi}$ (resp. $l_{vi}$)
for a transition from the range L to the range M (resp. M
to L). $H_{vi}$ (resp $h_{vi}$) for a transition from the range M to
the range H (resp. H to M) (Vasquez et al. (2016)). For
the vacuum oven the set of event types associated to the
behavior of the continuous variables is then defined by:

$$\Sigma_e = \{L_{F1}, l_{F1}, H_{F1}, h_{F1}, L_{F2}, l_{F2}, H_{F2}, h_{F2}, L_{F3},
H_{F3}, h_{F3}, L_{T1}, H_{T1}, L_{T2}, H_{T2}, L_{T3}, H_{T3}, L_{T4}, H_{T4},
h_{T4}\}$$

(1)

4.2 Learning event sequences generation

The learning event sequences are generated according to
the behavior of the system in a given scenario (scenario
of normal operating, faulty scenario with a fault on valve,
etc...). In this section we consider a scenario of normal behavior during the start up of the oven. By simulation we have obtained three different event sequences ($S1$, $S2$ and $S3$) all of them associated with the same scenario. Figure 4 shows one simulation of the scenario leading to the generation of the sequence $S1$ given by:

$S1 = ((V3,1), (LT1,3), (LF3,5), (V1,6), (LT4,7),
(LF1,8), (HT1,12), (V1,13), (V2,14), (HT1,15),
(h_{F1},16), (LF2,17), (HT4,19), (HF2,22), (l_{F1},24),
(h_{T1},25), (HT4,26), (HF2,27), (V1,42), (LF1,45))$

The two others sequences $S2$ and $S3$ are identical to $S1$ in
term of sequence of event types but differ to $S1$ in term of
event occurrence dates.

$S2 = ((V3,1), (LT1,7), (LF3,13), (V1,18), (LT4,21),
(LF1,24), (HT1,32), (V1,35), (V2,37), (HT1,40),
(h_{F1},45), (LF2,48), (HT4,54), (HF2,61), (l_{F1},65),
(h_{T1},68), (HT4,72), (HF2,76), (V1,96), (LF1,101))$

$S3 = ((V3,2), (LT1,6), (LF3,9), (V1,12), (LT4,14),
(LF1,16), (HT1,22), (V1,24), (V2,25), (HT1,27),
(h_{F1},30), (LF2,32), (HT4,36), (HF2,41), (l_{F1},43),
(h_{T1},45), (HT4,48), (hf2,50), (V1,68), (LF1,71))$

This scenario can be interpreted from the standard
procedure. For the startup stage, the initial conditions are
that the oven (OV) 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 ($F1, F2, F3, T1, T2, T3, T4$).
Then the scenario starts with the opening of the valve V3
that is to say the occurrence of an event of type V3. After this event type occurrence, the system is in a mode of operation where only the valves V1 and V2 are closed. The variable T1 increases and and
event of type LT1 must occur indicating that the internal
oven temperature has passed the limit of low. Then the
flow of the fuel gas reaches its low limit and an event
of type LF3 occurs. So, the ordered sequence of event
types that has occurred is V3, LT1, LF3. Passing the low
limit of F3 is the condition for continuing the procedure
by the action Open V1 (V1). When the operator opens
the valve V1, the system evolves to a mode of operation
where the internal flow in the vacuum oven starts. In
this situation, the flow F1 and the outflow temperature
T4 increase(event of type LT4 followed by an event of
type LF1). The next event that occurs is of type HF1
indicating that the flow F1 has passed its high level. At
At this stage, the ordered sequence of event types is given by: \(V3, L_{T1}, L_{F3}, V1, L_{T4}, L_{F1}, H_{F1}\). The next procedural action is the closing of the valve \(V1\) (\(v1\)) followed by the opening of the valve \(V2\) (\(V2\)). Then, the high limit of the temperature \(T1\) is reached and an event of type \(H_{F1}\) occurs. The flow \(F1\) decreases from its high limit (event type \(h_{F1}\)). An event of type \(L_{F2}\) occurs because the flow in \(V2\) increases. The high limit in the temperature \(T4\) induced and event of type \(H_{F4}\). Following up with the procedure, due to the high limit of \(F2\) an event of type \(H_{F2}\). At this time point, the ordered sequence of event types that has occurred is \(V3, L_{T1}, L_{F3}, V1, L_{T4}, L_{F1}, H_{F1}, V1, L_{F2}, H_{F4}, H_{F2}\). In this situation, an unobservable event of type \(M2A\) and the control loops are closed, carrying the system to a steady state. Note that in this study, we assume that the control loops are closed immediately after and event of type \(H_{F4}\). When \(H_{F4}\) occurs because \(V1\) is opened between 2 and 8 time units after that the valve \(V3\) is opened. \(H_{F4}\) is composed of \(\{Ar1, Ar2, ..., Arm\}\) where each area \(Ar_m\), \(m = 1, ..., n\) has \(K\) in \(N\) operational modes (e.g. startup, shutdown..) noted \(O_i\), \(i = 1, ..., K\). The process behavior in each operating mode can be either normal or faulty. We define the set of failure labels \(\Delta_f = f_1, f_2, ..., f_r\) and the complete set of possible labels is \(\Delta = N \cup \Delta_f\), here \(N\) means normal.

To monitor the process and to recognize the different situations (normal or faulty) of the operational modes, we propose to build a chronicle base for each area. For a given area, a learned chronicle \(C_{ij}^m\) is associated to each couple \((O_i, l_j)\) where \(l_j \in \Delta\):

\[
CAr_m = \begin{bmatrix}
O_1 & N & f_1 & f_2 & \cdots & f_r \\
O_2 & \begin{bmatrix}
C_{10}^m & C_{11}^m & C_{12}^m & \cdots & C_{1r}^m \\
C_{20}^m & C_{21}^m & C_{22}^m & \cdots & C_{2r}^m \\
\vdots & \cdots & \cdots & \cdots & \cdots \\
C_{k0}^m & C_{k1}^m & C_{k2}^m & \cdots & C_{kr}^m 
\end{bmatrix}
\end{bmatrix}
\]

(2)

When \(l_j = N\), the chronicle is a model of the normal behavior of the considered system, otherwise (\(l_j = f_i\)) the chronicle is a model of the behavior of the system under the occurrence of the fault \(f_i\).

For the vacuum oven we have considered a normal startup, a normal shutdown and several faulty cases. For instance, a startup stage during which a fault occurs on the control valve \(V3\). We present in this section only the chronicle \(C_{ij}^0\) learned from the input sequences \(S_1, S_2\) and \(S_3\) capturing a normal startup operating mode of the vacuum oven (area number 2 of the Cartagena Refinery).

In this case of normal startup stage, expert knowledge is available and integrated in our proposal by three temporal runs:

\[
TR_{V3, L_{F3}} = V3[6,8]L_{F3},
\]

this temporal run 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 run indicates that the lower limit of the flow in \(F1\) can occur 76 time units before that the valve \(V1\) is opened or 82 time units after that.

\[
TR_{L_{F1}, V2} = L_{F2}[2,8]V2,
\]

this temporal run 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 output of the extended HCDAM in this case is a chronicle that represents the behavior of the system taking as reference the representative event sequences obtained by simulation and the temporal runs obtained from the expertise knowledge. The directed graph associated to the chronicle \(C_{ij}^0\) is given Figure 5. As we can see in the chronicle resulted \(C_{ij}^0\), this chronicle expresses the temporal pattern for a normal start up of this system. Although the three representative event sequences in this scenario are similar, this temporal pattern can represent other cases where the order of the event types could be different between the learning event sequences. The frequency of occurrence of the event types \(V1\) and \(L_{F1}\) are \(f_{V1}=2\), \(f_{L_{F1}}=2\) and for the others event types it is 1. With this information and the use of temporal runs, the resulted chronicle is only one for this scenario.

5. CONCLUSION AND FUTURE WORK

A new methodology for an alarm management based on a diagnosis process has been proposed. The proposal is based on a hybrid causal model of the system and a chronicle based approach for diagnosis. An illustrative example of
the vacuum oven has been considered to introduce the main concepts of the approach. The algorithm HCDAM is a tool for the automatic generation of the chronicles from the representative event sequences and partial temporal runs. Transposing this methodology to large-scale systems would benefit from a decentralized approach in which local chronicles would be learned and then integrated thanks to shared events.

**Future work**

The following work will be related to the testing and validation of the model of chronicles. A new approach to carry into account negative examples and forgetting capabilities in the chronicle learning algorithm will be proposed.

**ACKNOWLEDGEMENTS**

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

**REFERENCES**


