



HAL
open science

Intelligent Case Based Decision Support System for Online Diagnosis of Automated Production System

N Ben Rabah, R Saddem, F Ben Hmida, V Carre-Menetrier, M Tagina

► **To cite this version:**

N Ben Rabah, R Saddem, F Ben Hmida, V Carre-Menetrier, M Tagina. Intelligent Case Based Decision Support System for Online Diagnosis of Automated Production System. European Workshop on Advanced Control and Diagnosis (ACD), 2016, Lille, France. pp.012009, 10.1088/1742-6596/783/1/012009 . hal-01899933

HAL Id: hal-01899933

<https://hal.science/hal-01899933>

Submitted on 20 Oct 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.

PAPER • OPEN ACCESS

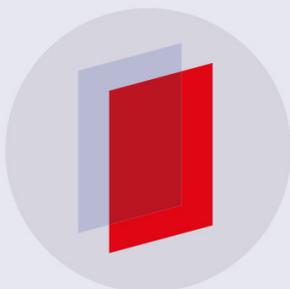
Intelligent Case Based Decision Support System for Online Diagnosis of Automated Production System

To cite this article: N Ben Rabah *et al* 2017 *J. Phys.: Conf. Ser.* **783** 012009

View the [article online](#) for updates and enhancements.

Related content

- [Methodical Approach to Developing a Decision Support System for Well Interventions Planning](#)
V A Silich, A O Savelev and A N Isaev
- [SOFish ver. 1.2 - A Decision Support System for Fishery Managers in Managing Complex Fish Stocks](#)
A K Supriatna, A Sholahuddin, A P Ramadhan *et al.*
- [Multi-Attribute Decision Making with VIKOR Method for Any Purpose Decision](#)
Dodi Siregar, Heri Nurdianto, S Sriadhi *et al.*



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Intelligent Case Based Decision Support System for Online Diagnosis of Automated Production System

N Ben Rabah^{1,2}, R Saddem¹, F Ben Hmida², V Carre-Menetrier¹ and M Tagina²

¹ Centre de Recherche en STIC (CReSTIC), Reims, France

² National School of Computer Sciences University of Manouba 2010 Manouba, Tunisia

nourhene.ben-rabah@etudiant.univ-reims.fr, ramla.saddem@univ-reims.fr,
faten.benhmida@ensi.rnu.tn, veronique.carre@univ-reims.fr, moncef.tagina@ensi.rnu.tn

Abstract. Diagnosis of Automated Production System (APS) is a decision-making process designed to detect, locate and identify a particular failure caused by the control law. In the literature, there are three major types of reasoning for industrial diagnosis: the first is model-based, the second is rule-based and the third is case-based. The common and major limitation of the first and the second reasonings is that they do not have automated learning ability. This paper presents an interactive and effective Case Based Decision Support System for online Diagnosis (CB-DSSD) of an APS. It offers a synergy between the Case Based Reasoning (CBR) and the Decision Support System (DSS) in order to support and assist Human Operator of Supervision (HOS) in his/her decision process. Indeed, the experimental evaluation performed on an Interactive Training System for PLC (ITS PLC) that allows the control of a Programmable Logic Controller (PLC), simulating sensors or/and actuators failures and validating the control algorithm through a real time interactive experience, showed the efficiency of our approach.

1. Introduction

Recent technological revolutions in the industrial field result in the appearance of more and more complex APS. This complexity is accompanied by the growing needs in terms of operation and security. Indeed, it is essential to implement systematic diagnosis approaches to detect, locate and identify possible failures in real time. In this work, our goal is to support and assist HOS in his/her decision process in order to accomplish his/her diagnosis task. To reach such a goal, it is essential to define the HOS and determine his/her tasks. So, who is a HOS?

HOS is a person responsible for the driving of an APS. He/she has to assume essentially three different tasks: acquiring knowledge from data of sensors and actuators, analyzing knowledge and making the diagnosis decision. Every time, he/she tries to remember past experiences which represent observed failures in similar situations and which can lead to similar results. In fact, it is difficult to perform these tasks, essentially the analysis of a large amount of binary data and the making of the relevant diagnosis decision in real time, notably when there is insufficient data.

In this paper, we present a synergy between the CBR approach and the DSS to propose an Intelligent CB-DSSD of an APS. The proposed system includes two phases: An offline phase in which data, describing the system behavior as a set of cases, are collected and represented. Each instance of case is labeled as a normal or faulty behavior of the system to diagnose. The subset of cases describing normal behavior of the system is listed in normal case base. However, the subset of cases, describing faulty behavior of the system, is stored in abnormal case base. An online phase takes as input the diagnosis request of a HOS and as output the diagnosis response (faulty or normal behavior).



The paper is structured as follows. In section 2, we provide an overview of failures to diagnose, diagnosis definitions and diagnosis methods of APS. In section 3, we present some recent case-based systems for industrial diagnosis and we enumerate the benefits of using CB-DSSD of APS. The functional model of the proposed system is detailed in section 4. In section 5, we illustrate our proposal with an application on the sorting system of ITS PLC Collection. Conclusion and possible futures works are addressed in the last section.

2. Diagnosis of Automated Production System

Before proposing a diagnosis method for an APS, it is very important to specify first the failures to diagnose, the diagnosis definition and the diagnosis method to employ.

2.1 Failures to Diagnose

An APS is composed of two parts: Control Part (CP) and Operative Part (OP). The CP is a set of software means and information concerning the control and the conduct of the process. It sends orders to OP in order to obtain the desired effects. It allows the communication with humans and other systems. Indeed, the OP performs the commands, sent by the CP, using actuators. It also transmits the reports to the CP (information collected from sensors). The APS study can be carried out from the point of view of either the OP or the CP. Failures, on an APS, are bound to an unanticipated event which comes modifying the operation of the system. It can be either external, caused by customer, environment or raw material, or internal caused by hardware of the control system, OP or CP.

In this work, we treat only internal failures, especially those caused by the OP, such as the stuck-off to 1 or 0 of a sensor or an actuator.

2.2 Diagnosis Definition

In Discrete Event System (DES) community, there are several definitions of the diagnosis function. Some works did not consider the detection as a diagnosis entity. They rather dealt it as a monitoring entity. For example, [1] defined the diagnosis as three entities:

- The location determining the sub-system responsible for the failure.
- The identification specifying the causes behind this failure.
- The explanation which justifies the diagnosis conclusions.

Other works, like [2] and [3], considered the detection as a diagnosis entity. They defined the diagnosis as a decision-making process involving three entities:

- The detection which allows detecting any deviation of the process behavior from normal behavior and alert HOS in a failure case.
- The location which allows specifying the failure origin (area and affected component).
- The identification which allows determining the failure occurrence time, its duration and its severity.

Our work is based on the second definition of diagnosis.

2.3 Diagnosis Methods

Literature distinguishes three industrial diagnosis methods: Rule-based diagnosis, Model-based diagnosis and CBR-based diagnosis.

Rule-based diagnosis [4] requires the translation of the diagnosis problem in boolean expressions. It is easily and efficiently applied in simple industrial systems. However, it needs a lot of time in order that experts convert their knowledge in rules. Besides, it is a method having no learning ability (it cannot handle a failure not planned in its rule base).

Model-based diagnosis [2], [3] is a problem-solving technique based on a model representing the system to be diagnosed. Indeed, the model is a description of real system. It is used to detect and locate multi-failures. However, model creation is a long process and the method has no automated learning ability.

CBR-based diagnosis [5] is a method used to solve new diagnosis problems by using previous experiences stored in a case base. It is the only approach which offers a learning capacity. The knowledge

acquisition is more economical than that in the rule-based method and updating cases is much easier. For these reasons, we adopted in our work this diagnosis method.

3. Case Based Decision Support System for Industrial Diagnosis

A DSS is an interactive information system, flexible, adaptable and specifically developed to quickly provide decision makers with relevant information that will help them to make the best decisions. The human-machine cooperation is important in the decision support. It assists the human decision-maker, and therefore guarantees the sharing/distribution of tasks between human-machine [6]. According to [7] and [8], there are several types of DSS:

- *DSS guided by models*: It is based on the assumptions of analysis models type « If ... then ... else » or similar.
- *Data-driven DSS*: It analyzes large amounts of stored data to produce associations, sequences, classifications, segments and forecasts.
- *Web-based DSS*: It supports the web client decision-making process by providing more information before deciding to purchase an item or a service.
- *Communication-driven DSS*: It supports more than one person working on a shared task.
- *Knowledge-driven DSS*: It provides the expertise to solve problems formalized in terms of rules, procedures, recommendations, etc...

In our work, we choose a Knowledge-driven DSS where expertise is formalized as cases and in order to produce new knowledge, we use the CBR approach.

This section provides first an overview of CBR. Secondly, it presents some recent case-based systems for industrial diagnosis that aim at classifying our work and making our choices. Finally, the section ends with enumerating the benefits of using a CB-DSSD of APS.

3.1 Overview of CBR

CBR is a methodology for solving problems, by using past experiences to understand and deal with new problems. Psychologists and cognitive scientists proved that humans routinely use it in their decision processes [9]. Past experiences are organized and stored in a case base. Each solved problem is represented as a case. The latter can be defined as a “contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner” [10].

According to [11], a case is a set formed by Problem (P) with its solution (Sol (P)): Case= ((P, Sol (P)). It can be structured in various ways. The old cases, stored in the case base, are called Source Cases (SCs), while each solved problem is named Target Case (TC). The CBR cycle (figure 1) consists of four essential steps [12]:

- *Retrieve*: It allows searching the previous cases most likely to satisfy the objective of reasoning (new problem).
- *Reuse*: The desired solution is extracted from the most relevant case for reuse.
- *Revise*: The proposed solution is revised and adapted in order to become compatible with the current problem. Adaptation can also repair a solution causing negative impact.
- *Retain*: Confirmed solution is stored in the case base for future use.

In CBR, there are three general areas that have to be considered when creating a case base [13]: (a) The structure and representation of the cases themselves. (b) The memory model used to organize the entire case base. (c) The selection of indices used to identify each case.

3.2 Recent Case-Based Systems for Industrial Diagnosis

Many research works use the CBR approach to solve problems in the industrial systems. We study some recent case-based systems that aim at solving diagnosis, maintenance or reconfiguration problems while representing the diagnosis phase in their systems.

We notice that the efficiency of a CBR system is related to the case representation step. Indeed, it can influence all other CBR cycle phases. In the works of [14] and [15], the case representation stage was defined according to the domain-related knowledge through an ontology. In fact, in complex systems, the construction of domain ontology remains a difficult task. [16] proposed a very rich case

representation. It is based on 15 indexes showing the problem main features. Nevertheless, the choice of these indexes is done manually. Besides, [20] introduced case representation that considers the different types of knowledge (general and contextual). In their study, the case was represented by a huge number of variables (110 variables). Moreover, in their approach, there is no extraction step of the most important features modeling the problem. [17] suggested a typical case representation based on the observed symptoms, their dates and the information about vehicles. [5] is the only work that considered the temporal aspect to carry out an in-depth analysis through the extraction of sensor signals features using temporal abstraction.

Generally, in the stage of searching resource cases that are more similar to the target cases, the measure of similarity or distance depends on the type of data and information stored in the case base. In the research studies of [16], [14], [18] and [15], the authors used similarity measures that take into account the descriptor relevance degree.

The adaptation phase is a very important stage either because the systems do not treat it as it is the case in [16], [18] and [19], or because the systems merge the selected solutions [5], use the dependency relationships of a domain ontology or conduct substitutional adaptation at the case descriptors. Finally, we note that none of the studied system dealt with the maintenance step.

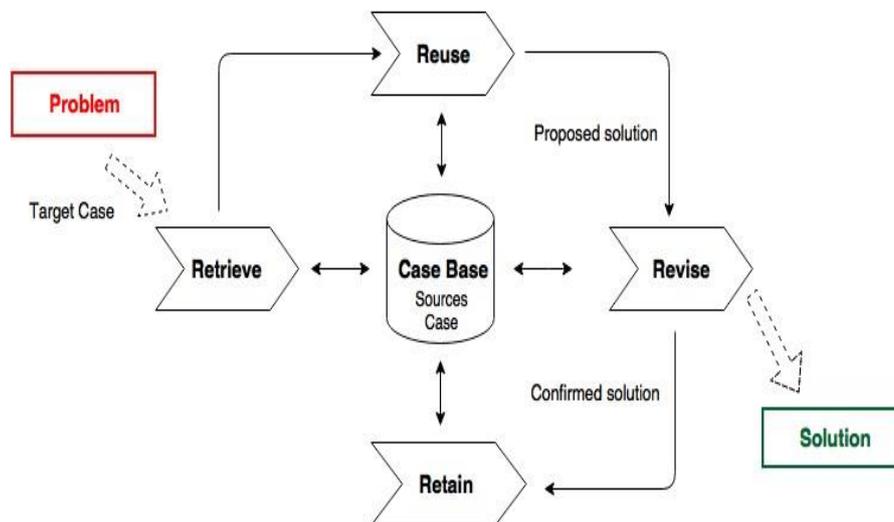


Figure 1. CBR cycle.

The presented study allows us to classify our work and make choices while trying to highlight some limitations of the afore-mentioned research studies. We choose first to diagnose the APS. The diagnosis is carried out from the point of view of OP. Second, our proposed system is based on a case representation according to the structural model. Such representation considers the temporal aspect in order to carry out a thorough analysis through the extraction of sensor and actuator signals to diagnose its current and previous PLC cycles. In addition, the APS sensors and actuators to be diagnosed are of AON types (ALL or Nothing). For this reason, our attributes values are binary (0 or 1), and the applied similarity metrics is a symmetric binary measure. Finally, our introduced system allows automatically formulating a new case when a new diagnosis request is made by the HOS. The new case inherits the solution from the selected cases.

3.3 The benefits of the proposed system

The proposed CB-DSSD of APS has many benefits:

- Simple and easy implementation.
- Achievable in real time: It is able to provide help and assistance to the HOS in order to perform his/her diagnosis task in real time.
- Incremental learning: The system can be created with a small or limited amount of incrementally-developed experiences, adding more cases to the case base.

- Offers several solutions for the same problem: The system can provide more than one alternative for a similar problem, and the HOS can choose the right solution that seems most suitable.
- The system does not need any rules, complete knowledge of the domain or any models: It does not need a complete knowledge of the domain or rules to reproduce new knowledge. In other words, it requires less knowledge acquisition. In fact, it consists in collecting a set of past experiences without extracting a formal domain model from these cases.

4. Functional model of the proposed system

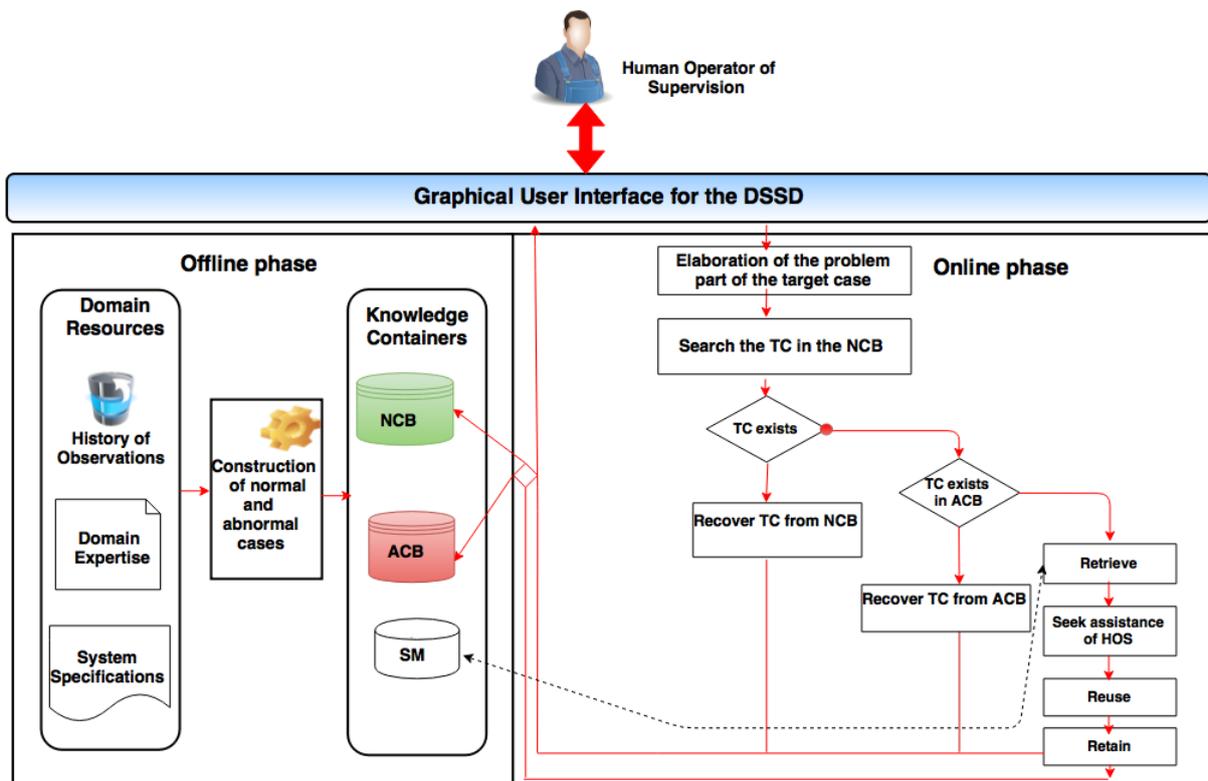
The generic functional model of our proposal (figure 2) consists essentially of a graphical user interface that allows the HOS to interact with the CB-DSSD and express his/her choices and decisions, an offline phase to collect and represent data which describes the APS behaviors as a set of cases and an online phase to extract for each new case the similar or nearest case using a similarity metric. Both phases are developed in the following subsections.

4.1 Offline phase

The objective of this phase is to extract knowledge necessary as cases. Indeed, to achieve such goal, the offline phase is based on the expert knowledge, observations and specifications of the system to be diagnosed. But, firstly, it is necessary to choose the good structure of case representation. We have already mentioned in the third section of this work that the choice of this structure influence the CBR cycle.

Figure 2. Generic functional model of our proposal.

Our proposed case is composed of two parts: problem representation part and solution representation part (see figure 3).



4.1.1 Problem Representation

During this part, the system state is defined to understand the situation to be diagnosed. The problem representation relies on the information obtained from sensors (inputs) and actuators (outputs) for the current PLC cycle (t) and the previous PLC cycle (t-1). This information is modeled by a set of descriptors. Each descriptor is a triplet:

$$\langle Co_i, V_t, V_{t-1} \rangle$$

Where Co_i is a sensor or a system actuator, V_t is the value of Co_i at t , and V_{t-1} is its value at $t-1$, with t : the current PLC cycle, and $t-1$: the previous PLC cycle.

Problem representation		Solution representation
Representation of the current situation for each I/O	Representation of the previous situation for each I/O	Case Type

Figure 3. Case generic format.

4.1.2 Solution Representation

This part describes the problem solution. It also indicates the case type (normal or abnormal behavior) (figure 3). It is composed of three sub-parts: detection, location and identification. The normal cases, representing normal behaviors of APS, are listed in Normal Case Base (NCB). However, a set of abnormal cases, representing faulty behaviors of APS, is stored in Abnormal Case Base (ACB). Figure 4 shows the case detailed format for an APS composed of 2 actuators and 4 sensors.

Actuator 0	Actuator 1	Sensor 0	Sensor 1	Sensor 2	Sensor 3	Normal or Abnormal
Val _t ; Val _{t-1}						

Figure 4. Case detailed format.

4.2 Online phase

The online phase of the CB-DSSD is triggered by a diagnosis request of the HOS. It corresponds to a new problem which is formalized as a problem part of a new case called Target Case (TC). The formalization is based on sensors and actuators data at the current PLC cycle time t (the time of request) and at the previous PLC cycle time $t-1$ (the time preceding the request).

After this step, we choose the most appropriate Similarity Metric (SM) among a set of similarity metrics [20]. Our choice depends on data representation and data type to compare (nominal, ordinal, continuous or binary). In this work, we use APS with sensors and actuators having only two possible values: 0 or 1 (All or nothing). For this reason, we employ a binary symmetric similarity (binary attributes are symmetrical, which implies that there is no preference that the attribute is coded 0 or 1) called the simple matching coefficient [21].

After applying the metric selection, we search the TC in the NCB. If it exists in NCB, we are in a normal situation. So, the CB-DSSD shows that there is no problem in the process. If it is not located in NCB, we search it in ACB. If it is found, then we are in an abnormal situation known as the CB-DSSD. Otherwise, CB-DSSD calculates the distance between TC, all NCB source cases and all ACB source cases. After that, it asks for the HOS help in order to choose the most appropriate solution. Once HOS intervenes, the new case inherits the solution from the selected cases. Finally, it will be stored in the NCB or ACB for future use. These treatments show that the CB-DSSD is able to use the obtained experience to promote learning.

5. Application on the sorting system of ITS PLC

To test the effectiveness of our proposal and describe its performance, we implemented our approach under the beta version of the Interactive Training System for PLC (ITS PLC) proposed by the Portuguese company Real Games [22]. In fact, it is a set of simulation software dedicated to automation training, writing and using scripts written in IronPython [23]. It allows notably simulating failures in sensors and actuators.

With the failures panel (figure. 5), the user is able to induce malfunctions in the system, presenting new challenges and increasing the realism of the simulation.

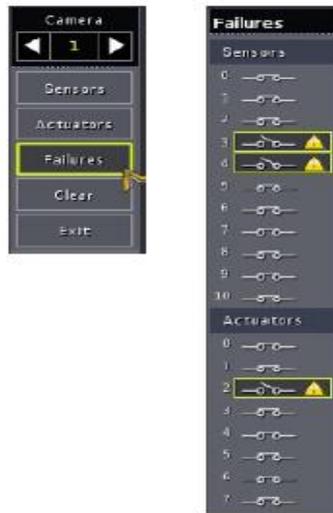


Figure 5. Failures panel of ITS PLC.

In our work, we developed two scripts. The first one allows controlling the APS without the need for a real API. However, in the second script, we implement our approach. In the following section, we operate the virtual system "Sorting System". Then, we present the offline and online phases of the CB-DSSD.

5.1 Description of the sorting system

The objective of the sorting system (figure 6) is to bring boxes of entry conveyor to exit conveyor by sorting them according to their heights. The system has eleven sensors to determine boxes size (small or large) and the box entry or exit in different conveyors (feeding, intermediate, and evacuation) or turntable. It has also seven actuators to activate the various conveyors and the turntable.



Figure 6. The sorting system from ITS PLC.

5.2 Offline phase and online phase of CB-DSSD

From a set of cyclically- recorded raw data (cycle time is 16 ms), representing signals of various sensors and actuators of the sorting system, a regrouping of these data is made, so that each 2 instances will represent a case. Then, an expert accompanies the construction phase of initial cases and expresses his/her opinion with regard to each case (normal behavior or failing behavior). In this stage, we obtain 52 normal cases and 10 abnormal cases.

During online phase and at every 60 ms, the CB-DSSD elaborates the problem part of the TC (new problem). After that, it searches the TC first at NCB. If the TC is already in NCB, then diagnosis result will be normal operation (case type=N). An example of this situation is shown in figure 7.



Figure 7. Diagnosis of normal behavior.

Then, we trigger a failure which corresponds to an existing case in ACB. The CB-DSSD is able to detect and locate this failure (figure 8).

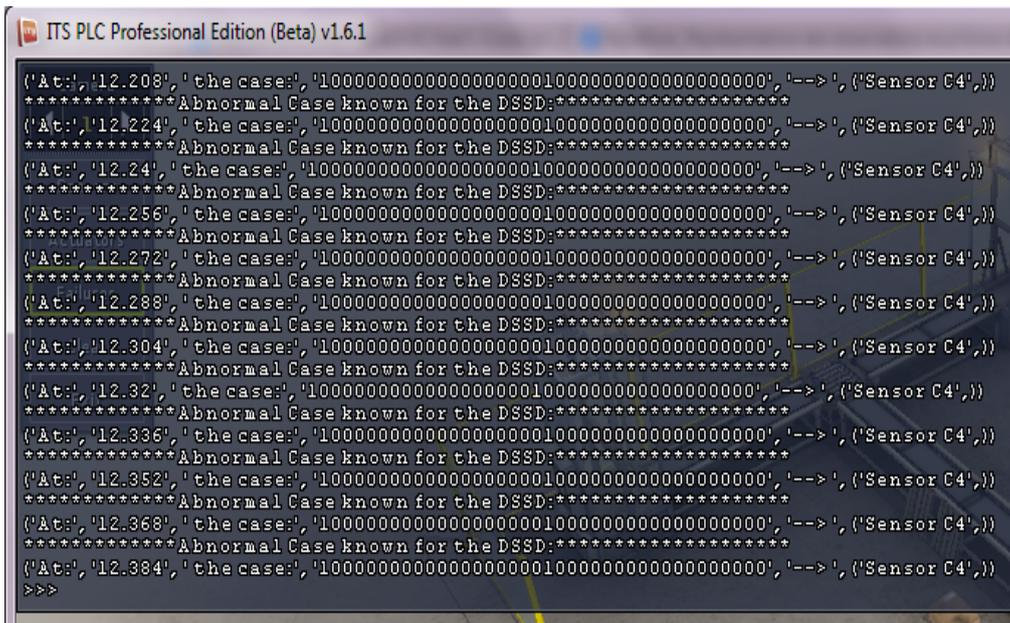


Figure 8. Diagnosis of faulty operation.

Figure 9 shows a new case which corresponds to an inexistent case in NCB and ACB. The CB-DSSD begins by calculating similarity between the new case, all abnormal sources cases and all normal sources cases. Then, it asks the help of HOS to choose the most appropriate solution. The new case inherits the behavior (normal or failed) of the selected case and will be stored in the NCB or ACB.

```

ITS PLC Professional Edition (Beta) v1.6.1
('Distance[TC,Source Case:', 41, '=', 0.31578947368421051)
('Distance[TC,Source Case:', 42, '=', 0.21052631578947367)
('Distance[TC,Source Case:', 43, '=', 0.18421052631578949)
('Distance[TC,Source Case:', 44, '=', 0.15789473684210531)
('Distance[TC,Source Case:', 45, '=', 0.13157894736842102)
('Distance[TC,Source Case:', 46, '=', 0.39473684210526316)
('Distance[TC,Source Case:', 47, '=', 0.39473684210526316)
('Distance[TC,Source Case:', 48, '=', 0.42105263157894735)
('Distance[TC,Source Case:', 49, '=', 0.31578947368421051)
('Distance[TC,Source Case:', 50, '=', 0.18421052631578949)
('Distance[TC,Source Case:', 51, '=', 0.18421052631578949)
***Abnormal Sources Cases:***
('Distance[TC,Source Case:', 0, '=', 0.13157894736842102)
('Distance[TC,Source Case:', 1, '=', 0.10526315789473684)
('Distance[TC,Source Case:', 2, '=', 0.13157894736842102)
('Distance[TC,Source Case:', 3, '=', 0.10526315789473684)
('Distance[TC,Source Case:', 4, '=', 0.13157894736842102)
('Distance[TC,Source Case:', 5, '=', 0.18421052631578949)
('Distance[TC,Source Case:', 6, '=', 0.21052631578947367)
('Distance[TC,Source Case:', 7, '=', 0.21052631578947367)
('Distance[TC,Source Case:', 8, '=', 0.21052631578947367)
('Distance[TC,Source Case:', 9, '=', 0.21052631578947367)
-----Please Select your most appropriate Case:
>>> _

```

Figure 9. Calculating distance between the new case and all other cases stored in NCB and ACB.

After the learning of this experience, we trigger the same failure. Thus, the DSSD is able to detect and locate it.

5.3 Difficulties:

The negative point of our proposal is described through this example:

Example:

At the triggering of stuck-off to 1 of actuator 6, the CB-DSSD forms the TC and firstly seeks it in the NCB. The TC exists in the NCB.

According to our CB-DSSD, the TC represents a normal behavior of the sorting system based on sensors and actuators data at the current PLC cycle time t and at the previous PLC cycle time $t-1$. Hence, to describe a stuck-off to 1 of an actuator or a sensor, these data are not sufficient. They only help to diagnose unexpected move of actuators or sensors from 0 to 1 or from 1 to 0.

To solve this problem, we should add temporal constraints of the occurrences of the observables events to our case representation.

6. Conclusion

Conventional CBR systems for industrial diagnosis were designed as automated problem solvers. Most of them were not efficiently used to diagnose real industrial systems. This paper outlined a case-based decision support system which resulted from a synergy between CBR and DSS. Its main purpose is to support and assist the HOS in his decision process in order to accomplish its diagnosis task.

From a diagnosis request of the HOS, the CB-DSSD begins by formalizing it as a problem part of a target case based on sensors and actuators signals at the current and the previous PLC cycle. Second, it searches the TC in the NCB. If it exists in NCB, the process will be normal. If it is not located in NCB, it will search it in the ACB. If it is found, the process will be faulty. Otherwise, the CB-DSSD asks the HOS to choose the appropriate similarity metric and it calculates distance between the TC, all source cases of NCB and all source cases of ACB. Finally, it asks for the HOS help to choose the appropriate solution. The new case inherits the solution from the selected cases and it is stored in the NCB or ACB for future use.

An application on the virtual sorting system from ITS PLC Collection showed the feasibility of our approach created with a small amount of incrementally-developed experiences, which adds more cases to the two case bases based on HOS decisions.

In future works, we will improve our case representation by taking into account the temporal constraints of the occurrences of the observables events. To diagnose complex system, we will use multi-expert

knowledge in an aggregate approach in order to facilitate collaborative decision making. In fact, the approach will be based on the multi-agent systems for their adaptation to the efficient management of complex systems.

7. References

- [1] Combacau M, Berruet P, Charbonnaud K A and Zamai E 2000 Supervision and monitoring of production systems *IFAC 2nd Conference on Management and Control of Production and Logistics* Grenoble France
- [2] Darkhovski B and Staroswiecki M 2003 Theoretic Approach to Decision in FDI *Transactions On Automatic Control* IEEE 48(5) pp 292-297
- [3] Phillippot A, Sayed-Mouchaweh M, Carré-Ménétrier V, Riera B 2006 Decentralized approach to diagnose manufacturing systems *Computational Engineering in Systems Applications* IEEE Vol 1 pp 912-918
- [4] Saddem R and Philippot A 2014 A Causal Temporal Signature from diagnoser model for online diagnosis of Discrete Event Systems *Control, Decision and Information Technologies* IEEE pp 551-556
- [5] Xiong N, Funk P and Olsson T 2012 Case-based reasoning supports fault diagnosis using sensor information *2nd International Workshop and Congress on eMaintenance*
- [6] Millot P and Lemoine M P 1998 An attempt for generic concepts toward human-machine cooperation *Man and Cybernetics* IEEE pp 1044-1049
- [7] Turban E, Aronson J, Liang T P 2005 *Decision Support Systems and Intelligent Systems* Pearson Prentice Hall
- [8] Power D J 2008 Decision support systems: a historical overview *Decision Support Systems* Vol 1 pp 121-140
- [9] Kolodner J L 1991 Improving human decision making through case-based decision aiding *AI Magazin* 12(2) pp 52-68
- [10] Kolodner J L 1992 An introduction to case-based reasoning *Artificial Intelligence Review* 6(1) pp 3-34
- [11] Napoli A, Lieber J and Curien R 1996 Classification-based problem-solving in case based reasoning *European Workshop on Advances in Case-Based Reasoning* Springer Berlin Heidelberg pp 295-308
- [12] Aamodt A and Plaza E 1994 Case-based reasoning: Foundational issues, methodological variations, and system approaches *AI communications* 7(1) pp 39-59
- [13] Main J, Dillon T S and Shiu S C 2001 A tutorial on case based reasoning *Soft computing in case based reasoning* Springer London pp1-28
- [14] Dendani N, Khadir M and Guessoum S 2012 Hybrid approach for fault diagnosis based on CBR and ontology: using jCOLIBRI framework *Int. Conf. on Complex Systems* IEEE pp 1-8
- [15] Ruiz P A P, Kamsu-Foguem B and Noyes D 2013 Knowledge reuse integrating the collaboration from experts in industrial maintenance management *Knowledge-Based Systems* Vol 50 pp 171-186
- [16] Behbahani M, Saghaee A and Noorossana R 2012 A case-based reasoning system development for statistical process control: Case representation and retrieval *Computers and Industrial Engineering* 63(4) pp 1107-1117
- [17] Chougule R, Rajpathak D and Bandyopadhyay P 2011 An integrated framework for effective service and repair in the automotive domain: An application of association mining and case-based-reasoning *Computers in Industry* 62(7) pp 742-754
- [18] Lejri O and Tagina M 2012 Representation in case-based reasoning applied to control reconfiguration *Advances in Data Mining* pp 113-120
- [19] Vareilles E, Aldanondo M, De Boisse A C, Coudert T, Gaborit P and Geneste L 2012 How to take into account general and contextual knowledge for interactive aiding design: Towards the coupling of CSP and CBR approaches *Engineering Applications of Artificial Intelligence* 25(1) pp 31-47
- [20] Cha S H 2007 *Comprehensive survey on distance/similarity measures between probability density*

functions City 1(2) pp 1

- [21] Legendre P and Legendre L 1998 *Numerical Ecology* 2nd English edition Elsevier Science BV Amsterdam 852p
- [22] www.realgames.pt
- [23] <http://ironpython.net>