



HAL
open science

A comparative study of three model-based FDI approaches for Discrete Event Systems

Mickaël Danancher, Matthias Roth, Jean-Jacques Lesage, Lothar Litz

► **To cite this version:**

Mickaël Danancher, Matthias Roth, Jean-Jacques Lesage, Lothar Litz. A comparative study of three model-based FDI approaches for Discrete Event Systems. 3rd International Workshop on Dependable Control of Discrete Systems - DCDS 2011, Jun 2011, Saarbrücken, Germany. pp.29–34, 10.1109/DCDS.2011.5970314 . hal-00595103

HAL Id: hal-00595103

<https://hal.science/hal-00595103>

Submitted on 23 May 2011

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.

A comparative study of three model-based FDI approaches for Discrete Event Systems

Mickaël Danancher^{*,†}, Matthias Roth^{*,†}, Jean-Jacques Lesage^{*} and Lothar Litz[†]

^{*}Automated Production Research Laboratory (LURPA), ENS Cachan

61 av. du Président Wilson, 94235 Cachan, France

Email: {danancher, lesage}@lurpa.ens-cachan.fr

[†]Institute of Automatic Control, University of Kaiserslautern

P.O. Box 3049, 67653 Kaiserslautern, Germany

Email: {mroth, litz}@eit.uni-kl.de

Abstract—In this paper three model-based Fault Detection and Isolation (FDI) approaches for Discrete Event Systems (DES) are evaluated. The considered approaches are the *diagnoser* approach [1], the *templates* approach [2] and the *residual* approach [3]. The investigated methods have different characteristics like timed / non-timed behavior and fault-free / faulty system models with important impacts on the model-building process and the respective effectiveness. By applying the three methods to the same benchmark system, their respective performances are analyzed in terms of fault detection and fault isolation ability, complexity of implementation and avoidance of false alarms.

Index Terms—Discrete Event Systems, Fault Detection, Fault Diagnosis

I. INTRODUCTION

The competitiveness of production systems is a major stake. Only diminution of costs and times of production leads to a high level of productivity which has positive effects on economic issues. Fault Detection and Isolation (FDI) methods allow to detect a faulty behavior as soon as possible in order to avoid serious damages of the system or the injury of an operator. Furthermore, the aim of these methods is to reduce down time by providing a more or less accurate localization (fault isolation) of the faulty component(s) to be replaced before the production can restart.

In this paper, three model-based FDI approaches for Discrete Event Systems (DES) are evaluated. They were applied to the same case study in order to compare their possibilities and limits.

In the following section, a brief overview of the different FDI approaches is given and the three chosen approaches are presented. The system which is used as a benchmark is briefly described in section III. In the fourth section, the different models built for applying each FDI approach are described. In section V, the simulation of different scenarios of faulty behavior and the results for each approach are presented. A conclusion is given in the last section.

II. FDI OF DES: CONSIDERED APPROACHES

A. Overview of different FDI approaches

As proposed by Y. Papadopoulos and J. Mc. Dermid in [4], FDI approaches can be divided in three major groups. *Rule-based approaches* are using rules (IF-THEN-ELSE or

more complex) which allow to trigger alarms when a set of conditions is verified by the system. *Data-driven approaches* are using methods such as data mining or pattern recognition in order to detect faulty behaviors. *Model-based approaches* are comparing the real behavior of a system with its modeled behavior in order to detect the possible occurrence of a fault.

In this study, it has been chosen to focus on model-based FDI approaches. Indeed, rule-based and data-driven approaches are quite generic and can be used for a large number of system to be monitored. Model-based approaches are explicitly using the (continuous or discrete) dynamic of the system to be monitored. They are then specifically dedicated to a class of system but often allow a more precise diagnosis. A more accurate description of the general principle of model-based FDI of DES is given in Fig. 1. The system to be diagnosed is considered as a closed loop of a controller and a plant. From the point of view of the controller, the information in the loop are logical inputs and outputs (I/O). These I/Os as well as a previously built model of the system are used on line by an FDI algorithm to detect and then to isolate a fault.

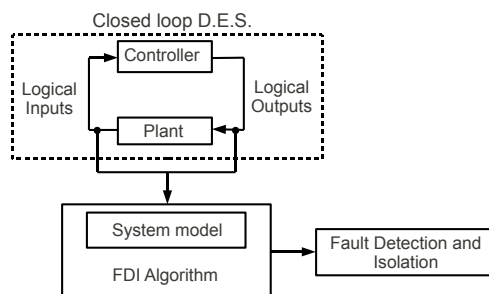


Figure 1. Principle of model-based FDI

Among the model-based FDI approaches proposed in literature in the last fifteen years, three have been selected to conduct this comparative study: the first approach which is named in this paper the *diagnoser* approach is described in [1], the second one which is named the *templates* approach is presented in [2] and the last one, the *residual* approach, is described in [3]. These three particular approaches have been chosen considering their complementarity:

- The *diagnoser* approach is based on a model containing the faults that have to be detected whereas the two other approaches are based only on a model of the fault-free behavior of the system to be monitored.
- The *diagnoser* and *residual* approaches are based on models which are Finite Automata (FA) whereas the *templates* approach is based on an event model.
- The *templates* approach is the only one among the three approaches to explicitly take into account the timed behavior.

The three approaches are briefly introduced in the following subsections. For the sake of a better understanding, a figure depicting the model on which the FDI is based is provided for each approach.

B. Diagnoser approach

The so called *diagnoser* approach has been introduced in 1996 by M. Sampath et al. [1]. This approach is based on a modeling of each system part (controller, process components,...) with a FA. Each FA has to represent on the one hand all the fault-free behavior (also called normal behavior) of the component, and on the other hand one or many faulty behavior(s) expected to be detected. By applying the algorithms proposed in [1], the *diagnoser* used for the FDI is obtained. An example of such a *diagnoser* is depicted in Fig. 2. It is a FA defined on the event set of observable events. It allows to detect faults (which are considered to be unobservable events) with the help of three different classes of states: states called "Normal" (here states 1N and 8N) in which it can be guaranteed that no fault occurred; states called "Fi-certain" (here state 7F2) in which it can be guaranteed that a fault occurred (in this particular case, the fault F2); states called "Fi-uncertain" (here states 2N 3F1 4F2 or 5N 6F1) in which no conclusion can be obtained concerning the occurrence of a fault. During on line monitoring of the system it is supposed that no fault occurred as long as the diagnoser remains in a normal or an uncertain state. As soon as the diagnoser reaches an Fi-certain state, a fault is detected. This fault is also necessarily isolated because each fault label Fi contains the necessary information to completely isolate the fault.

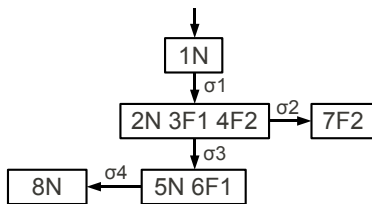


Figure 2. An example of a diagnoser

C. Templates approach

In this approach, proposed by D. Pandalai and L. Holloway in 2000 [2], the fault-free behavior of the system to be

monitored is modeled with a list of *templates*. Each one of these *templates* consists of a causal relation between a trigger event and one or many expected events. An example of such a *template* is given below:

$$(e_1, \{(\emptyset, (e_2, [t_{min_2}, t_{max_2}]), w_0)\})$$

where e_1 is the trigger event, e_2 is the expected event, $[t_{min_2}, t_{max_2}]$ a temporal window. A condition (here \emptyset) and a tag (here w_0) can also be introduced. The complete definition of a *template* as well as the FDI algorithms are presented in [2]. However, the principle can be briefly explained with the help of Fig. 3. If the event e_1 is observed at a time t_1 , then the event e_2 is expected to be observed in the temporal window $[t_1 + t_{min_2}, t_1 + t_{max_2}]$. If this event is not observed in this temporal window, then a fault is detected. The examination of the *template* involved in the fault detection helps to isolate the fault since the trigger event and the expected event(s) are the potentially faulty candidates. Usually, many *templates* working on different events are running in parallel. If one *template* detects an anomaly, the according events can be considered as related to the fault.

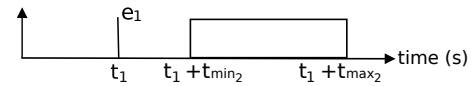


Figure 3. Illustration of a template

D. Residual approach

In this approach, more recently proposed by M. Roth et al. in 2009 [3], a FA model is used. This model has to represent only the expected fault-free behavior of the system to be monitored. An example of such a FA is depicted in Fig. 4. The *residual* method leads to the detection of a fault as soon as the observed behavior is not reproducible by the model [3]. For example, if after state 3 an event $s3_0$ is observed, this is not reproducible by the model. Thus, it leads to the detection of a fault. Moreover, this fault can be isolated by computing the *residuals*. They allow to characterize the difference between the real observed behavior of the system and the expected behavior of the model. For the considered example, using an informal representation of the *residuals*: $Res1 = \text{observed behavior} \setminus \text{expected behavior} = \{s3_0\} \setminus \{s1_0, s2_1\} = \{s3_0\}$ and $Res4 = \text{expected behavior} \setminus \text{observed behavior} = \{s1_0, s2_1\} \setminus \{s3_0\} = \{s1_0, s2_1\}$. The formal representation of these *residuals* is given in [3]. The union of all the *residuals* represents the set of the potentially faulty candidates.

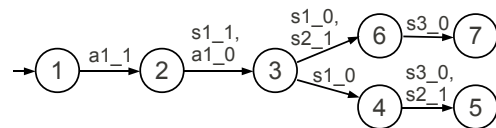


Figure 4. An example of a FA representing a fault-free behavior

III. PRESENTATION OF THE CASE STUDY

A. The studied system

A Pick and Place station (Fig. 5) has been proposed as a benchmark system [5]. One special feature of this system is that its plant is simulated with a computer software named "ITS PLC" (see [6]) whereas the controller is a Programmable Logic Controller (PLC) linked with the simulated plant through a data acquisition box. The functioning of this system is the following: parcels are brought by the conveyor A0 to the pick station. At the same time, empty boxes are brought by the conveyor A1 to the place station. The two horizontal double-acting pneumatic cylinders (A2,A3 and A4,A5) as well as the vertical single-acting pneumatic cylinder (A6) are used to move the gripper between the two stations. With the help of these cylinders and a magnetic gripper (A7), the parcels are gripped at the pick station and placed in the box. When the box is full, it is evacuated. This system is also equipped with 9 sensors (S0 to S8).

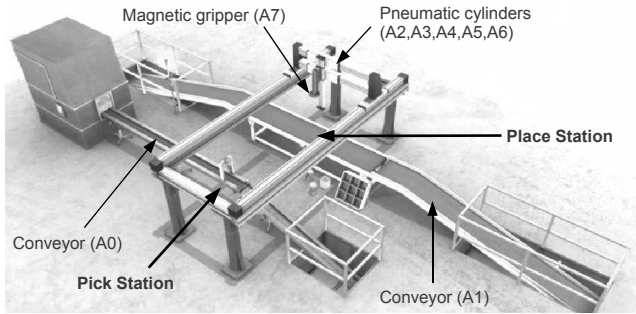


Figure 5. The Pick and Place station

B. Fault classification

In order to simulate different scenarios of malfunctioning, the following classification of the faults to be detected is proposed:

- *Actuator faults* are faults involving either an actuator (for instance a cylinder or a motor), a pre-actuator (for instance a single or double solenoid valve), a connection (linking an actuator to its pre-actuator or a pre-actuator to the controller) or the output unit of the controller.
- *Sensor faults* are faults involving either a sensor (for instance a position sensor), a connection (linking a sensor to the controller) or the input unit of the controller.
- *Process faults* are faults involving the process in the plant (for instance a parcel falling down a conveyor or an operator dropping off an additional parcel on the conveyor).

This classification is based on the representation of a DES as a closed loop depicted in Fig. 1. The controller is assumed to have a fault-free operation (this assumption is required to apply the *residual* approach).

Even if the size of this system is not so large (17 I/Os) it has been too difficult to build a model for the *diagnoser* approach

(even though all the necessary algorithms have been implemented in order to minimize this difficulty). The construction of a FA for each component (cylinders and conveyors) is quite easy. On the contrary, the construction of a FA for the controller is hard because this FA corresponds to the graph of the accessible states of the PLC program as this program is executed by the controller. Starting from these FA, their parallel composition can easily automatically be done. But the most complicated task is the mapping of this composition with a sensormap (see [1]) because for each state of the composition the expected value of each chosen sensor has to be defined by hand. Furthermore, the introduction in the model of a large number of faults to be detected quickly increases its size and complexity. Although there exist methods which allow to apply the *diagnoser* and the *residual* approaches to large systems (particularly distributed approaches [7] or [8]), it has been chosen to focus only on the monolithic ones that seemed to us more suitable to the size of this benchmark. For all these reasons, it has been chosen to restrict the study, in a first time, to a subsystem so that the three approaches can be applied. This subsystem is depicted in Fig. 6. A6 represents the subset of the vertical movement with the solenoid valve and the single-acting cylinder. A7 represents the magnetic gripper and its electric contactor. S6 and S7 are position sensors, they are the limit switches of the cylinder and S8 is the gripper detector (it sends a signal equal to 1 if a parcel is gripped and equal to 0 otherwise).

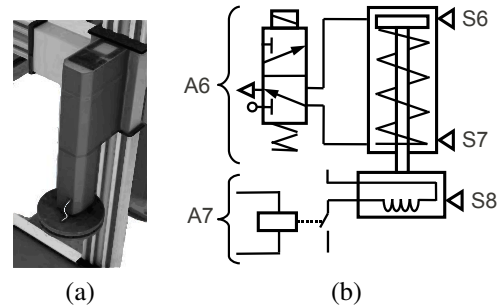


Figure 6. (a) and (b): The considered subsystem

Moreover, the following convention is adopted in this paper: the event "rising edge" of the signal associated to a sensor S_i (respectively to an actuator A_j) is denoted S_{i_1} (resp. A_{j_1}), the events "falling edge" are denoted S_{i_0} and A_{j_0} .

IV. MODEL BUILDING

A. Construction of the diagnoser

The first step of the modeling using the *diagnoser* approach is to build a FA model for each component (A6, A7) of the plant and for the controller (C). Two of these models are depicted in Fig. 7 (because of a lack of space, the FA representing A7 is not shown here). These FA are representing the fault-free behavior as well as actuator faults (in dotted line) that are introduced. Once the parallel composition of these three FA has been done, a sensormap has also been defined. Only the functional sensors already installed in the system and

used to control the process have been considered to build the sensormap and thus, the *diagnoser*. No additional sensor has been added (it would be possible, but not reasonable from our point of view, to add for example a pressure sensor for the cylinder, only for a FDI purpose, as proposed in [1]).

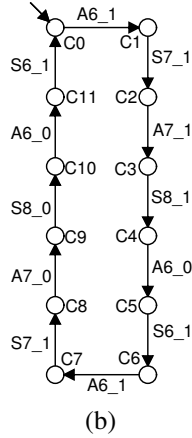


Figure 7. (a): Model of the cylinder. (b): Model of the controller

The algorithms from [1] have then be applied to create the *diagnoser*. It has 48 states and 75 transitions. A part of it is shown in Fig. 8.

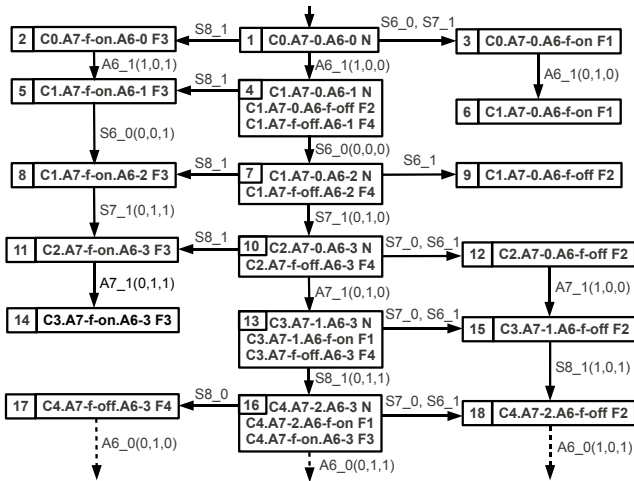


Figure 8. A part of the diagnoser

It is important to notice that some choices have to be made, particularly concerning the faults to be modeled. In this case, only actuator faults have been introduced (for the reasons of complexity described in section III-B). In addition, other way to model these faults can be chosen (for instance: only one state representing the two faults or one state for each fault from each normal state, i.e. 12 faulty states in the model Fig. 7 (a)).

B. Construction of a list of templates

The list of *templates* (table I) has been built with the following method: the trigger event is always a controllable

event (i.e. an order sent by the controller which is supposed to always have a fault-free behavior) and the expected event(s) is (are) uncontrollable event(s) (i.e. the change of value of a (several) sensor(s)). These pairs of events have been defined with the help of the causal relations that can be extracted from the FA previously built (considering only the fault-free behavior). For instance, in Fig. 7 (a), the controllable event A6_1 is followed by two uncontrollable events S6_0 and S7_1. This relation has been modeled in the *templates* (1) and (2) in table I. Finally, 20 cycles of functioning (supposed to be fault-free) of the system have been observed in order to define the temporal window associated to each one of the *templates* by measuring the values t_{min} and t_{max} .

Table I
THE TEMPLATES DESCRIBING THE SUBSYSTEM

(1)	$(A6_1, \{ (\emptyset, (S6_0, (0, 0.2]), w_0),$
(2)	$(\emptyset, (S7_1, [1.16, 1.25]), w_0) \})$
(3)	$(A6_0, \{ (\emptyset, (S7_0, (0, 0.2]), w_0),$
(4)	$(\emptyset, (S6_1, [1.12, 1.28]), w_0) \})$
(5)	$(A7_1, \{ (\emptyset, (S8_1, (0, 0.2]), w_0) \})$
(6)	$(A7_0, \{ (\emptyset, (S8_0, (0, 0.2]), w_0) \})$

C. Construction of a fault-free FA of the closed-loop system

The *residual* approach requires a model which is a FA representing the whole fault-free behavior of the system. It has been chosen to refer to the already built models of the *diagnoser* approach (Fig. 7). By doing so, it has to be noticed that the models for the three FDI approaches are based on the same semantics. States and events representing faulty behaviors have been removed from the models and the parallel composition of the three FA (C, A7 and A6) has been done. The result is shown in Fig. 9.

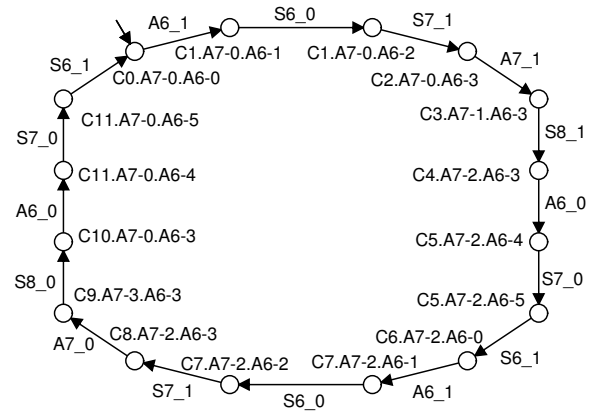


Figure 9. FA representing the fault-free behavior

It is also possible to get such a FA by using an identification method such as the one proposed in [9]. This method has also been applied to this subsystem and a FA having 13 states and 14 transitions has been obtained. It is not shown here because of a lack of space. However, this model has also been evaluated while simulating different fault scenarios (cf. section V).

V. SIMULATION OF DIFFERENT SCENARIOS OF FAULT

In order to evaluate the performances of each FDI approach, many scenarios have been simulated. Four among these scenarios are presented in detail here. They are based on the previously introduced fault classification (section III-B).

The first scenario consists in an *actuator fault*. The single solenoid valve of cylinder A6 burns while the cylinder is extending itself. This makes the solenoid valve return in its initial position. Even if the controller still orders the cylinder to extend itself ($A6 = 1$), the cylinder is moving back to its retracted state and will not move anymore.

The second scenario consists in a *sensor fault*. The position sensor S8 is disconnected at the very beginning of a cycle (the cable linking this sensor to the controller is inopportunistically snatched). The controller gets no more information from this sensor, the logical value associated to it stays equal to 0.

The third scenario represents a *process fault*. A parcel falls inopportunistically down while being gripped and transported from the place station to the pick station (it can happen due to the acceleration of the cylinder).

The last scenario consists in a *fault-free evolution* of the system. The system is normally functioning without any fault occurrence but a lack of stability of the gripped parcel leads to an oscillation of the value of the sensor S8 (the parcel is shaking but not falling down). This scenario, not artificially created but really observed, allows to point out the possible false alarms generated by the three FDI algorithms. This behavior stays acceptable as long as the parcel does not fall down. To consider it as a fault is excessive.

The following subsections present the results of the fault simulations for each approach.

A. Simulations with the *diagnoser* approach

As expected, since the actuator fault of scenario 1 has been modeled in the *diagnoser* (Fig. 7 (a)), it is detected and isolated (the diagnosability theorem is proved in [1]).

A *diagnoser* is built for a set of defined faults. Nevertheless, when it is implemented and used on line for a FDI purpose, there inevitably exist non-modeled faults that can occur. Thus, the reactions of the *diagnoser* towards these non-modeled faults have to be evaluated. Concerning the particular sensor fault of scenario 2, an uncertain state (state 13 in Fig. 8) is reached from which no more evolution is possible because the process is stopped.

However, other scenarios of sensor faults have been tested and sometimes a sensor fault is detected. It could never be isolated as a sensor fault but only as an actuator fault. Generally speaking, the *diagnoser* approach is not suited to the detection of sensor faults. Plant components are modeled through the actions of the actuators (whose faults can be taken into account) and observed through the sensors (eventually through additional sensors). These sensors are also used, through the sensormap, to build the *diagnoser*. Considering the faults of the sensors whereas they are used in order to detect the faults presents a quite unavoidable difficulty.

The process fault of scenario 3 (which was not modeled in the *diagnoser*) is detected but isolated as an actuator fault of the gripper blocked on its "stuck open" position, which is not the case here. Moreover, the detection of such a fault can not be guaranteed, it can not be considered as diagnosable.

Concerning the last scenario, whereas the particular observed behavior is acceptable, it was not modeled in the fault-free part of the *diagnoser*. The simulation of this scenario leads the *diagnoser* to reach a F4-certain state which means the gripper is blocked in a "stuck open" position. A fault is detected and isolated, it is a false alarm.

B. Simulations with the *templates* approach

The actuator fault of scenario 1 is detected because the *template* (2) of table I is violated. The fault is isolated, the potentially faulty candidates are A6 and S7. It can be noticed that the isolation is quite accurate since it was a fault involving A6.

Contrary to the *diagnoser* approach, the *templates* approach leads to the detection and the isolation of the sensor fault (scenario 2). Since no new event is observed, taking into account the time leads to the violation of the *template* (5) in table I. Two candidates are given: A7 and S8. For this scenario again, the isolation is quite good since it was a fault involving S8.

The process fault (scenario 3) is detected (*template* (2) of table I) and 2 candidates are obtained (A6 and S7). These candidates are very inaccurate because they are not directly linked to the fault. Manually checking these 2 candidates will probably not lead to the correct localization of the fault.

Concerning the last scenario, it is not detected as a fault by the *templates* approach because this particular behavior does not lead to the observation of any trigger event (that would make new events be expected) or any expected event (that might be observed too early and thus lead to the violation of a *template*).

Concerning other possible false alarms, it happens sometimes that the vertical cylinder retracts itself slower than when it was modeled. Consequently, the position sensor S6 is triggered too late, the *template* (4) in table I is violated and a fault is detected. Since this behavior is still acceptable, it has to be considered as a false alarm. This specific case illustrates the difficulty to set the values of the temporal windows and furthermore to take into account the deviation of these values with the age of the system.

C. Simulations with the *residual* approach

The actuator fault (scenario 1) is detected (using a model built by expert knowledge or an identified model). A non reproducible behavior has been observed (see Fig. 10). The fault is then isolated by computing the *residuals*. Two candidates (S6 and S7) are obtained using a model built by expert knowledge, three candidates (S6, S7 and A7) are obtained using an identified model. The candidates are not directly linked to the fault. By checking S6, the fault might be correctly identified.

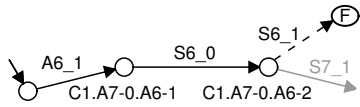


Figure 10. Evolution of the FA for the actuator fault

Concerning the sensor fault (scenario 2), it is not detected (either with a model built by expert knowledge or an identified model). The state C3.A7-1.A6-3 (see Fig. 9) is reached and just as with the *diagnoser* approach, no new observed event makes the FA further evolve.

However, other scenarios of sensor faults have been tested and they have shown some promising abilities to detect and accurately isolate these faults. The candidate set reduction algorithm proposed in [3] can be very helpful to reduce the number of candidates (leading to a very accurate isolation in some cases).

The process fault (scenario 3) is detected and isolated. Two candidates are given (A6 and S8). The candidates set reduction algorithm leads to the elimination of the candidate A6 and thus to a more accurate isolation.

Concerning the fault-free scenario, using the *residual* approach with a model built by expert knowledge leads to the detection of a fault which is a false alarm. Nevertheless, using an identified model shows some interesting performances because this particular acceptable behavior has been observed during the step of identification and is thus included in the model of the fault-free behavior of the system. Consequently, since no fault is detected, there is no false alarm in this case.

D. Evaluation of each approach and comparison

In order to evaluate the performances of each approach applied on this particular case study, four criteria are considered:

- Detectability (ability to detect a fault) and diagnosability (guaranteed detectability)
- Precision of isolation (certainty or ambiguity)
- Generation of false alarms
- Complexity of implementation of the approach (based on our experience)

The *diagnoser* approach shows the best performance in terms of detection and isolation because of the diagnosability of the modeled faults. However, fault modeling is not easy and when the number of introduced faults becomes too high, the mapping of the parallel composition with a sensormap becomes impossible to do. Another disadvantage consists in the fact that all the faults can not be modeled and thus, the occurrence of a non-modeled fault can lead to a malfunctioning of the *diagnoser* and an inaccurate isolation of the fault.

In terms of detectability of the faults, there also exists the problem of deadlock states for both the *diagnoser* and *residual* approaches. This problem can often be solved by taking into account the timed behavior (using the *templates*).

In terms of precision of isolation, the *templates* and *residual* approaches are not so efficient. The isolation is not accurate. Using the *templates* approach, it can not be decided whether

the actuator (trigger event) or the sensor (expected event) is faulty. Using the *residual* approach, it can not be decided which one of the *residuals* is relevant. To get a better isolation with the *residual* approach, the candidates set reduction algorithm of [3] can be used. It reduced the number of potentially faulty candidates and thus leads to a more accurate isolation but it can also be dangerous to use this algorithm since it requires the system to continue functioning even though a fault has been detected.

The three approaches have shown poor results in terms of avoidance of false alarms because of the non completeness of the models.

Concerning the complexity of each approach, the three FDI algorithms are not complex and have been quite easily implemented and performed on line. The difficulty consists in building the model the FDI is based on. Even though all the algorithms have been implemented, expert knowledge is still required at some point. For the *residual* approach based on an identified model, it is required to decide if an observed behavior is acceptable and thus has to be part of the model or not. This is quite easy. For the *diagnoser* approach, the sensormap has to be done by hand by analyzing the parallel composition of all the FA (Fig. 7) and setting the value of each sensor for each state of the composition (for this case study: 64 states and 3 sensors mean 192 values to be set).

VI. CONCLUSION

In this paper, we presented a comparative study of three model-based FDI approaches applied on a benchmark system. The algorithms have been implemented and the models have been built for each approach. We proposed 4 criteria to evaluate the approaches and gave a comparison of them.

REFERENCES

- [1] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Trans. on Control Systems Technology*, vol. 4 (2), pp. 105–124, 1996.
- [2] D. Pandalai and L. Holloway, "Templates languages for fault monitoring of timed discrete event processes," *IEEE Trans. on Automatic Control*, vol. 45, pp. 868–882, 2000.
- [3] M. Roth, J.-J. Lesage, and L. Litz, "A residual inspired approach for fault localization in DES," *Proc. of the 2nd IFAC Workshop on Dependable Control of Discrete Event Systems, DCDS'09, Bari, Italy*, pp. 347–352, 2009.
- [4] Y. Papadopoulos and J. McDermid, "Automated safety monitoring: A review and classification of methods," *Int. Journal of Condition Monitoring and Diagnostic Engineering Management*, vol. 4 (4), pp. 14–32, 2001.
- [5] A. Philippot, "Survey on diagnosis of a pick and place benchmark," *Proc. of the 3rd International Workshop on Dependable Control of Discrete Event Systems, DCDS'11, Saarbrücken, Germany*, 2011.
- [6] B. Riera, P. Marange, F. Gellot, O. Nocent, A. Magalhaes, and B. Vigário, "Complementary usage of real and virtual training manufacturing systems for safe PLC training," *8th IFAC Symposium on Advances in Control Education, ACE2009, Wellington, New Zealand*, 2009.
- [7] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynamic Systems: Theory and applications*, vol. 10, no. 1–2, pp. 33–86, 2000.
- [8] M. Roth, "Identification and fault diagnosis of industrial closed-loop discrete event systems," Ph.D. dissertation, 2010.
- [9] M. Roth, J.-J. Lesage, and L. Litz, "An FDI method for manufacturing systems based on an identified model," *Proc. of the 13th IFAC Symposium on Information Control Problem in Manufacturing, INCOM'09, Moscow, Russia*, pp. 1389–1394, 2009.