AN UML MODELLING OF A NEURO-FUZZY MONITORING SYSTEM

Nicolas Palluat, Daniel Racoceanu, Noureddine Zerhouni

Laboratoire d'Automatique de Besançon, UMR CNRS 6596, 24 rue Alain Savary, 25000 Besançon, FRANCE
Tel: +33 (0)381 402 794 Fax: +33 (0)381 402 809
E-mail: {npalluat,daniel.racoceanu,zerhouni}@ens2m.fr

Abstract: The complexity of real production systems implies more difficulties to make an efficient monitoring and especially fault diagnosis. We propose a new method supporting the operator to find the cause and the origin of a fault. To obtain a diagnosis aid system that is both reactive and easy to configure, we define a set of artificial intelligence tools using neuro-fuzzy techniques. The interest of these techniques is to combine the neural networks learning capabilities and the natural language formalism modelling capabilities of the fuzzy logic. Our approach follows the UML approach with the description of the seven use cases of our method.

Keywords: UML, neural network, neuro-fuzzy, diagnosis, monitoring, maintenance, SCADA, CMMS, FMECA, fault tree.

1. GENERAL CONCEPT

The improvement of the complexity of real production systems in a hard concurrent marketing context encourages the managers to give more importance to the maintenance functions. The industrial monitoring, which is one of the most significant of them, is divided into two tasks: the failure detection, and the failure diagnosis (failure localization and identification of the cause) (How *et al.*, 1999; Pencolé, 2002; Tromp, 2000). But, more complex the system is, more difficult monitoring will be. So, an efficient monitoring system must be able to learn all changes (reconfiguration, etc) and to take into account the experts and operators human experiences. The proposed method concern all phases of the monitoring function: the fault detection with a dynamic neural network and the fault diagnosis with a neuro-fuzzy tool. (figure 1)

The fault detection uses a recurrent neural network (Zemouri *et al.*, 2003). The output of this kind of real time detection system is given by the failure mode or failure symptom. This symptom will be used in the diagnosis tool. The fault diagnosis uses a fuzzy neural network based on the fault tree of the supervised system. This neural network uses information from the CMMS historic and will be able to learn any new maintenance situations (symptom-mode, localization, origin). In this way, we have a reactive and dynamic diagnosis system.

Our method is divided into several parts (Palluat *et al.*, 2004):

- Acquisition of relevant information of the system: Using studies carried out on the system (FMECA, fault tree, functional analysis ...), and with the help of the production and maintenance operators and managers,

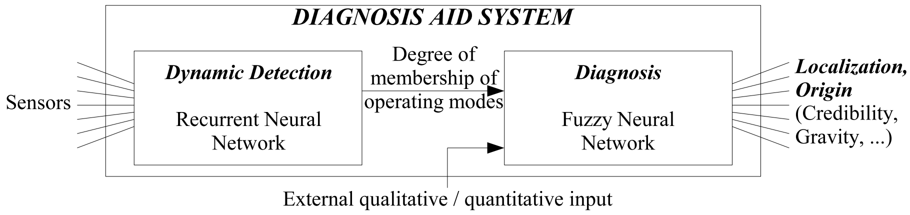

Fig. 1. Overview of the system

it is necessary to extract critical zones to monitor, as well as information available on these zones: static (fault tree, functional analyzes . . .) and dynamic information (CMMS, historic, given sensors, SCADA . . .).

- Fault detection system based on the dynamic neural networks: The input of the detection system is given by sensors data. In output, we obtain the operating mode (symptom) of the supervised element. The use of neural networks is justified by their training and parallel computation capabilities, their capacity to solve problems inherent to the system non-linearity and their computation speed when implemented in an integrated circuit.
- Diagnosis aid system based on a fuzzy neural network: The input of the diagnosis system will be the degree of membership of each operating mode given by the detection system. We find also external qualitative or quantitative inputs like information given by operators to improve diagnosis. In output, we find a list of possible causes ordered by degree of credibility, and as complementary information: the degree of severity. These degrees help the maintenance manager to evaluate and plan the maintenance actions.

During the process, the detection system scans continuously the critical element. When a failure is detected or predicted, an alarm is raised and the diagnosis aid system starts. Detection keeps working. According to the information provided by the detection system, the diagnosis aid system proposes to the operator the possible causes of the problem as well as the fuzzy interpretation of these causes.

We presented in (Palluat *et al.*, 2004) an overview of this system. In this paper, we present how our tool can be used and his link with maintenance actors. Our design approach follows the UML standard (OMG, 2003; Larman, 2002; Rumbaugh *et al.*, 1998) and starts by the specification of the use cases, from the point of view of the user of the monitoring tool. We thus propose seven use cases, corresponding to a marketing study implying a large consortium of maintenance partners [1]

[1] ITEA European Project – PROTEUS – A E-maintenance Platform (web site : `http://www.proteus-iteaproject.com/`).

2. USE CASES

First, it is necessary to identify the actors. We have the human actors and the non-human actors.

- the CMMS – Computerized Maintenance Management System (non-human actor)
- the FMECA – Failure Modes Effects and Critical Analysis (non-human actor)
- the FT – Fault Tree (non-human actor)
- the Maintenance Manager (human actor)
- the Maintenance Operator (human actor)
- the SCADA – Supervisory Control And Data Acquisition (non-human actor)
- the Tool Expert (human actor)

For each human actor, we define a set of use cases. We obtain seven use cases:

- to create a new tool,
- to configure the tool,
- to initialize the tool,
- to raise an alert,
- to perform a diagnosis,
- to update a configuration,
- to update the model of the tool.

In the figure 2 and 3, we show connections between the use cases and the actors. We organize the use cases and group them in two coherent functional packages:

- Off-line package for use cases used when the tool doesn't work,
- On-line package for use cases used when the tool works.

We develop by next the textual description of the use cases. This method of description is not normalized in UML so we use a formalization inspired by the work of Alistair Cockburn (Cockburn, 2000).

2.1 To create a new tool

The Maintenance Manager needs a new monitoring tool. We assume that the hardware and the software are installed and working.

Main Actor: Maintenance Manager.
Secondary Actor: Tool Expert.
Goal: Creation of a new tool.
Precondition: The Maintenance Manager want a new monitoring tool. He knows the critical

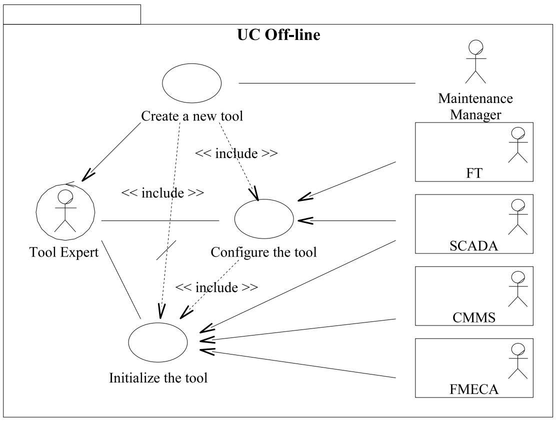

Fig. 2. Off-Line Package

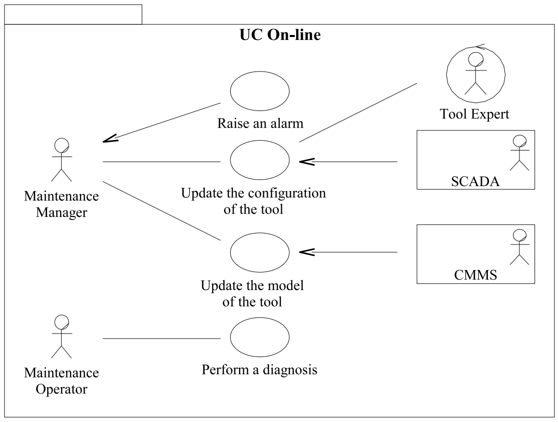

Fig. 3. On-Line Package

sub-equipment to monitor and all the accessible data (sensors, SOA – Symptom, Origin, Action) with the maintenance tools (GMAO, SCADA, FT, FMECA)

Postcondition: A new tool is in use.

Main scenario:

(1) Maintenance Manager asks to the Tool Expert to create a new tool;
(2) Tool Expert launchs the use case "To configure the tool";
(3) Tool Expert launchs the use case "To initialize the tool";
(4) Tool Expert informs the Maintenance Manager of the setting up of the tool.

2.2 To configure the tool

Main Scenario of the neuro-fuzzy system configuration starts with the request of the Fault Tree. The Tool Expert translates the Fault Tree into a Fuzzy Neural Network. The architecture of the tool will also take into account the sensors availability trough the SCADA.

Main Actor: Tool Expert.

Secondary Actors: The two systems FT and SCADA.

Goal: Configuration of the tool.

Precondition: The Maintenance Manager asks to the tool expert to create a new tool.

Postcondition: The tool is configured.

Main scenario:

(1) System FT provides the fault tree to the tool expert;
(2) Tool Expert configure the diagnosis tool with the fault tree;
(3) Tool Expert configure the detection tool so that it is subscribed to the SCADA in order to benefits to the availability of the useful sensors.

2.3 To initialize the tool

The initialization of a Neurofuzzy system consists in setting values for the diagnosis that will be inspired from the existent FMECA and from the CMMS. This process will take into account the criticity (frequency, severity) of the failures and the associated Symptom, Origin and Actions given by the CMMS.

Main Actor: Tool Expert.

Secondary Actors: The three systems FMECA, CMMS and SCADA.

Goal: Initialization of the tool.

Precondition: The tool is configured.

Postcondition: The tool is initialized.

Main scenario:

(1) System FMECA provides the FMECA to the tool expert;
(2) Tool Expert analyzes the FMECA and extracts useful data (Operating mode, causes, frequency and severity);
(3) Tool Expert initialize the detection tool and the diagnosis tool with the extracted data;
(4) Tool Expert initialize the diagnosis tool so that it is subscribed with the events maintenance of CMMS system.
(5) Detection tool receives the sensors value at the appropriate frequency of the SCADA system.

2.4 To raise an alert

A failure has been detected or predicted by the neuro-fuzzy system and the alert is emitted to the maintenance manager in order to prevent him that a failure has occurred or has a great possibility to occur in the near future evolution of the system.

Main Actor: Maintenance Manager.

Goal: Raising an alert due to a fault or to a prediction of a fault.

Precondition: The tool is in use.

Postcondition: An alert is sent to the Maintenance Manager.

Main scenario:

(1) Detection tool has one or more of their operating mode which the degree exceed a threshold;
(2) An alert is sent to the Maintenance Manager.

2.5 To perform a diagnosis

A failure has been detected or predicted and the Maintenance Operator needs help to diagnose. First a request of information is emitted. Afterward, the Maintenance Operator fills the information form in. The system will suggest then a diagnosis.Finqlly, the Maintenance Operator validates the diagnosis.

Main Actor: Maintenance Operator.
Goal: The maintenance operator needs the assistance of the tool.
Precondition: The tool is in use. A failure was detected or predicted. The Maintenance Operator needs the assistance of the tool.
Main scenario:
(1) Maintenance Operator request an assistance for diagnosis;
(2) tool provides a set of possible causes classified per degree of credibility and degree of severity;
(3) maintenance operator validates the diagnosis.
Exceptions:
1a. Maintenance operator chooses a detailed diagnosis.
(1) Maintenance operator reaches a specialized form allowing him to add information which cannot be given by the detection tool (smoked, odors, ...) and the use case continue to the step 2 of the main scenario.
3a. Maintenance operator is not satisfied by the results.
(1) Maintenance operator returns to the step 1 of the main scenario to launch a new request.
(1) Maintenance operator gives up the request. the use case finishes (failure).

2.6 To update a configuration

The Maintenance Manager wants to add a new sensor to the tool. The Tool Expert add the associated input to the system and request SCADA to send the values at the suitable frequency acquisition.

Main Actor: Maintenance Manager.
Secondary Actors: Tool Expert and the system SCADA.
Goal: The Maintenance Manager wants to add a new sensor in the tool.
Precondition: The tool is in use.
Postcondition: The tool is updated.
Main scenario:
(1) Maintenance Manager gives the designation of the sensor to the Tool Expert;
(2) Tool Expert adds a new input to the detection tool;
(3) Tool Expert updates the detection tool so that it is subscribed to the SCADA in order to benefits to the availability of the new sensor;
(4) Detection tool receives the sensors value at the appropriate frequency of the SCADA system.
Exceptions:
2a. Updating the configuration creates a new operating mode.
(1) Maintenance Manager gives the link between this new mode and a step in the Fault Tree;
(2) Tool Expert updates the diagnosis tool and go to the step 3 of the main scenario.

2.7 To update the model of the tool

The main actor of the model update (on line training) is the CMMS. First, we check the case in the database (where a case is a pair (symptom, origin) where a symptom is a detectable failure and the origin is the primary cause). Secondly, we update parameters for this case. If the case is a new case, the Maintenance Manager gives the severity of the failure and the new case is integrated into the tool.

Main Actor: CMMS.
Secondary Actor: Maintenance Manager.
Precondition: The tool is in use.
Postcondition: The tool is updated.
Main scenario:
(1) The new maintenance task is done and is converting into a pair (symptom, origin);
(2) Parameters of the diagnosis tool for this case are updated;
(3) The tool is updated.
Exceptions:
2a. This is a new case.
(1) The tool send a request to the Maintenance Manager to have the severity of this cause;
(2) The Maintenance Manager sends the information.
(3) The tool is updated.

2.8 conclusion

In this part, we saw the different textual description of the tool. UML provides a number of diagrams that we don't show in this paper. These diagrams could be sequence diagrams, collaboration diagrams, interaction diagram, ... Other thinks can be shown in a UML method like classes. The table 1 shows the package, the different use cases, the actors links by use case and classes.

Table 1. Synthesis analysis of use cases

Package	Use cases	Actors	Classes
Off-line package	Create a new tool	Maintenance Manager	Send a request for a new tool
		Tool Expert	Configure the tool
			Initialize the tool
			Validate the tool
	Configure a tool	Tool Expert	Transform FT
		FT	Provide the FT
		SCADA	Subscribe to SCADA
	Initialize the tool	Tool Expert	Modify parameter of the tool with FMECA
		FMECA	Provide the FMECA
		CMMS	Subscribe to CMMS
		SCADA	Send value of the sensors
On-line Package	Raise an Alert	Maintenance Manager	Verify Alert
	Perform a Diagnosis	Maintenance Operator	Diagnose
			Detailed Diagnose
	Update a configuration	Maintenance Manager	Add sensor
		Tool Expert	Add a new mode
			Update the tool
		SCADA	Subscribe to SCADA
	Update the model	CMMS	Send the new maintenance event
		Maintenance Manager	Send the severity

3. SYNTHESIS AND PROTOTYPING OF THE MONITORING SYSTEM

Our system is divided into two parts: a dynamic neural network detection system and a neuro-fuzzy diagnosis system. Before using the neuro-fuzzy tool, two steps are necessary: the first one is the configuration where data are collected and extracted to create the tools and the second one is the initialization where data extracted are learned by the tools. In use, the tools are in detection state where the dynamic neural network determines in which mode the system is with an associated possibility. When there is a problem on the system, the detection tool emits an alert. When a diagnosis is requested, the diagnosis tool uses data of the detection tool to give possible localizations and origins of the problem, classified by degree of credibility and severity. During the monitoring, the maintenance manager can improve the tools by configuration and/or model updating.

These tools are applied on industrial system, a flexible platform, available to the Besançon "Institut de Productique" [2] (France). This platform is equipped with five PLC communicating between them through a local industrial network. The flexible system permits to move pallets which can receive components to assembly.

The network permits to exchange the information received by the PLC concerning the changes of states of the sensors, the sequences of the control program, and operations made on the pallets. The platform is divided into five stations. Each station has its own PLC. They work independently.

[2] Institut de Productique, Besanon, France `http://www.institutdeproductique.com/`

3.1 Overall vision of the platform

The platform consists of two lines¿

On the primary line, the pallets circulate between all the stations. This line must always be free to let circulate the pallets.

The secondary line is dedicated to the treatment of the tasks, which will be made on the pallets. It is on this line that the robots and the manipulator are installed. It is possible to forward a pallet of a station to the next one without returning to the internal line.

On the platform, the pallets are belt-driven. These belts are driven by an electric motor integrated in each station.

The pallets are provided with a magnetic label, which is there "embarked memory". These memories can be read in each station thanks to the RFID [3] tags read/write magnetic heads (BALOGH). These labels make it possible to memorize the assembly range of the products, to know by which station(s), the pallets must pass.

3.2 Application of our method

For this example, we limit the study to the input of one station. We can find below the extracted fault tree (figure 4) and the extracted FMECA (table 2).

We test the system by blocking a pallet. The pallet should not go to the external line and the blocker S1 jams up so the pallet is blocked on primary line. A scheme of this situation is drawing in figure 5.

The detection tool gives the alert to the maintenance manager who decides to launch a diagnosis;

[3] Radio Frequency Identification

Table 2. Extracted FMECA.

Failure Modes	Cause	Frequency	Severity
Pallet jam to the inner jack	D1 sensor failure	4	2
Pallet jam to the inner jack	S1 actuator jam up	3	4
Pallet jam to the inner jack	Jack actuator failure	1	2

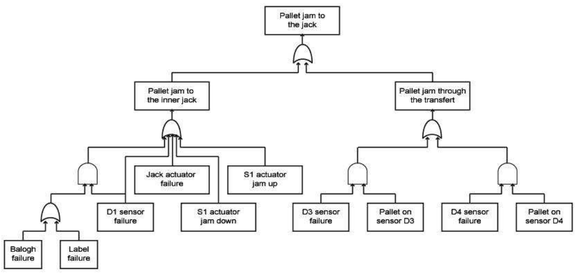

Fig. 4. Extracted fault tree

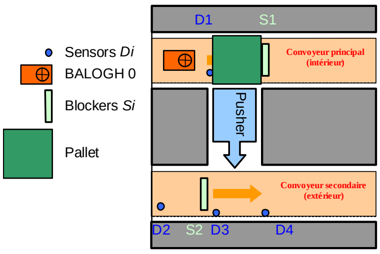

Fig. 5. Overview of a pallet jams to the inner jack

the diagnostic tool gives the following results (table 3).

Table 3. Diagnosis results.

Possibility	Cause	Severity
83,77	D1 sensor failure	2/5
61,22	S1 actuator jam down	4/5
35,44	Jack actuator failure	2/5
32,31	Label failure	0/5
32,31	Balogh failure	0/5
22,41	S1 actuator jam up	0/5
0	Pallet on sensor D4	0/5
0	Pallet on sensor D3	0/5
0	D4 sensor failure	0/5
0	D3 sensor failure	0/5

The system gives "D1 sensor failure" as the best possible cause of the problem. In fact, there is the good answer. For this configuration, there is another possible response which is S1 actuator jam up. The diagnosis doesn't give it in the second place because its never happened before. So, in the future, the maintenance manager can decide to improve the diagnosis by introduce this possibility.

4. CONCLUSION

In this paper, we present a new neuro-fuzzy tool following an UML approach. Our system is divided into two parts: a dynamic neural network detection system and a neuro-fuzzy diagnosis system. Before using the neuro-fuzzy tool, two steps are necessary: the first one is the configuration where data are collected and extracted to create the tools and the second one is the initialization where data extracted are learned by the tools. In use, the tools are in detection state where the dynamic neural network determines in which mode the system is with an associated possibility. When there is a problem on the system, the detection tool raises an alert.

When a diagnosis is requested, the diagnosis tool uses data of the detection tool to give possible localizations and origins of the problem, classified by degree of credibility and severity. During the monitoring, the maintenance manager can improve the tools by configuration and/or model updating.

We illustrate the use of our diagnosis aid tool on an industrial flexible platform.

Further work will investigate methods to improve the online learning of the aid diagnosis tool.

REFERENCES

Cockburn, Alistair (2000). *Writing Effective Use Cases*. Addison-Wesley Pub Co.

How, Wan Yat, Marzuki Khalid and Syed Ahmad Fuad Syed Zain (1999). Transformer fault diagnosis using fuzzy logic interpretations. In: *Instrument Asia Technical Symposium'99*. Singapore.

Larman, Craig (2002). *Applying UML and patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process.*. Prentice Hall.

OMG (2003). *Unified Modeling Language Specification. Version 1.5*. Object Management Group.

Palluat, Nicolas, Daniel Racoceanu and Noureddine Zerhouni (2004). Diagnosis aid system using a neuro-fuzzy approach. In: *Advances in Maintenance and Modeling, Simulation and Intelligent Monitoring of Degradation, IMS'2004*. Arles, France. CDROM.

Pencolé, Yannick (2002). Diagnostic décentralisé de systèmes événements discrets : application aux réseaux de télécommunications. PhD thesis. Université de Rennes I.

Rumbaugh, James, Ivar Jacobson and Grady Booch (1998). *The unified modeling language reference manual*. Addison-Wesley.

Tromp, Laurent (2000). Surveillance et diagnostic de systèmes industriels complexes: une approche hybride numérique/symbolique. PhD thesis. Université de Rennes I.

Zemouri, Ryad, Daniel Racoceanu and Noureddine Zerhouni (2003). Recurrent radial basis function network for time-series prediction. *Engineering Applications of Artificial Intelligence* **16**, 453–463.