Engineering Applications of Artificial Intelligence 36 (2014) 99–113

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Interactive diagnosis for a grid network of rain gauges using fuzzy reasoning

Quang Huy Giap [a], Stephane Ploix [a], Olivier Adrot [a], Christian Depraetere [b]

[a] G-SCOP Lab, Grenoble Institute of Technology, 46 avenue Félix Viallet, 38 031 Grenoble Cedex, France
[b] Maison de la télé-detection, UMR ESPACE-DEV, 500 rue Jean-François Breton, 34093 Montpellier Cedex 5, France

ARTICLE INFO

Article history:
Received 14 January 2013
Received in revised form
11 April 2014
Accepted 4 July 2014
Available online 12 August 2014

Keywords:
Fault diagnosis
Diagnosis analysis
Detection test design
Symptom generation
Fuzzy logic

ABSTRACT

This paper aims at designing a diagnosis tool to support experts for detecting and localizing faults in a network of rain gauges. This problem is presented in the context of human–machine cooperation. In this problem, it is impossible to model completely the whole expert knowledge about misbehavior. Diagnostic becomes a process where only a part of the expert knowledge is formalized, the remaining is kept implicit and is exploited gradually during the diagnostic process thanks to interactions with experts. At each step, the proposed diagnosis tool supports the expert by presenting selected data to be analyzed, i.e. rainfall hyetographs for a cluster of rain gauges for which the expert has to identify possible discrepancies. A fuzzy logic based diagnostic reasoning is then used because it proves to be more relevant to express the expert conclusions, which may be dubious. The way of handling such diagnosis processes is presented in this paper.

1. Introduction

Performing diagnoses rely on a complex process, which can be decomposed into a design process followed by a running process. The following design tasks may be distinguished: system modeling, sensor placement, detection test design and isolation algorithm design.

The *system modeling* task aims at formalizing the reference behaviors. Generally speaking, reference behaviors may be modeled by constraints establishing links between data coming from observations and unknown variables. Each constraint models one or several behavioral modes. In AI community, a system modeling task is detailed in Reiter (1987), De Kleer and Williams (1992), Struss (1992), and Chittaro and Ranon (2004). Each element of a system may behave according to some modes. The *ok* mode, denoted *ok*(*component*) in De Kleer and Williams (1992), stands for the normal expected behavior of a component. Constraints may model faulty behaviors like *leak*(*pipe*). Finally, the complementary fault mode *cfm*(*component*) gathers all the behaviors that are not related to *ok* mode and to modeled fault modes. It is named unknown fault mode in De Kleer and Williams (1992) and Struss (1992). The modeling task consists in formalizing the constraints modeling the different modes of the system to be diagnosed.

The *detection test design* task is usually not explicit in AI approaches: consistency tests are usually done directly in checking the consistency between the constraints and the data. However, results coming from FDI community (Patton et al., 2000; Blanke et al., 2006) point out that for many systems, a detection test may not be given from a simple consistency test between the elements of the system description and observations. Therefore, works gathering researchers from FDI and AI appeared with the aim of making a bridge between FDI and AI results (Cordier et al., 2000; Nyberg and Krysander, 2003; Ploix et al., 2003). As a result, *detection test design* tasks can be decomposed into the two following sub-tasks: testable subsystem generation, which consists in selecting subsets of constraints that may lead to detection tests (Blanke et al., 2006; Krysander et al., 2005; Ploix et al., 2005), and detection algorithm design that consists in designing a detection test, often named Analytical Redundancy Relation, corresponding to each testable subset of constraints.

The *isolation algorithm design* task consists in selecting the most relevant diagnostic analysis approach. Different kinds of approaches may be used to analyze the symptoms provided by detection tests: decision tree based approach (Quinlan, 1986; Nakasuka and Koishi, 1995; Pomorski and Perche, 2001), case based reasoning approach (Xia and Rao, 1999; Goker et al., 2005) or the signature based approach (Patton et al., 2000; Blanke et al., 2006). In this paper, the bridge approach is considered (Nyberg and Krysander, 2003; Ploix et al., 2003). It has been shown that consistency based reasoning can be used to analyze symptoms coming from detection tests depicted by the involved modes.

E-mail addresses: gqhuy@dut.udn.vn (Q.H. Giap), stephane.ploix@grenoble-inp.fr (S. Ploix), olivier.adrot@grenoble-inp.fr (O. Adrot), christian.depraetere@teledetection.fr (C. Depraetere).

http://dx.doi.org/10.1016/j.engappai.2014.07.002
0952-1976/

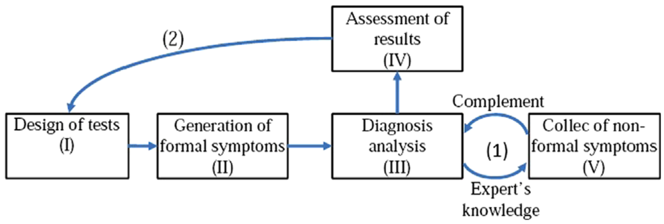

Fig. 1. Interactive diagnosis process.

Another task may be mentioned: the *sensor placement*. It consists in defining where to place sensors in order to satisfy diagnosability, discriminability and detectability properties (Yassine et al., 2008; Frisk and Krysander, 2007).

The running process is closely related to the design task: symptom generation, diagnostic analysis and possibly backward analysis. In scientific literature, most of the contributions aim at automating the running process assuming that models can be completely established before the diagnosis analysis. But Ploix and Chazot (2006) point out that, in many practical contexts, this prerequisite cannot be satisfied. Consequently, new problems arise because:

- the system is too complex to carry out detailed modeling of the whole system;
- some knowledge cannot be easily formalized.

In order to tackle these difficulties, interactions between experts and computer-aided diagnosis systems are obviously needed during the running process, and also in the modeling task. This paper focuses on the diagnosis problem with human–machine interactions in the diagnostic analysis phase to exploit the implicit expertise. The diagnostic process is illustrated in Fig. 1. (1) represents the human–machine interactions in the diagnosis phase. The interaction (2), called retro-analysis, will also be discussed in this work.

Some expert knowledge can be formalized as reference models and detection tests but some other knowledge is difficult to formalize and should rather remain implicit for the expert. The main idea is to gradually present data to the expert during the diagnosis process in order that he can analyze data based on his tacit knowledge. Hence, the problematic is: what should be presented to the expert at each interaction?

This problem has been studied in the project Hydrodiag in collaboration with Christian Depraetere, a researcher at the Institute of Development Research. The objective is to diagnose faults on a network of rain gauges. The tools used to solve the problem will be detailed later. Let summarize the main solving steps that lead to the considered problem. The first step aims at designing the detection tests. A set of detection tests is generated: they are based on correlations between quantities of water received from rain gauges. It contains the expert explicit knowledge. Multiple faults may occur in this problem. In order to tackle this difficulty, the Hitting-Set Tree algorithm (HS-Tree) (Reiter, 1987) has been used.

From the diagnosis framework based on the crisp logic, the results of fuzzy logic have been integrated in the diagnostic analysis (Touaf and Ploix, 2004b) that allows to avoid false alarms on the one hand and to minimize non-detections on the other hand.

Next, in many cases, a numerous set of diagnostic assumptions may be given by the algorithm, and it is difficult for an operator to use them. For example, in the Hydrodiag problem, 16 different diagnoses can be found with 5 or 6 simultaneous faults for each diagnosis. Since it is not reasonable to present all the computed diagnostics, human–machine interactions appear necessary in this case to locate faults without upsetting the expert with a large set of assumptions that he is not able to analyze. For this reason, a solution is proposed to guide the expert to establish a diagnostic. The idea is that only a part of the expert's knowledge can be formalized by detection tests automatically generated from system behavioral models. On the one hand, there is a tool-aided diagnosis with mathematical models and reasoning tools that can tackle complexity without difficulty, and on the other hand, an expert with complementary tacit knowledge that makes it possible to determine whether a sensor is faulty or not by looking at its rain hyetograph and those of its neighbors. It cannot be detected automatically by a diagnosis system.

The paper is organized as follows: the diagnosis problem for a network of rain gauges is stated and a framework fault detection and diagnosis based on crisp logic is presented in Section 2. Section 3 shows how to integrate fuzzy diagnostic reasoning to avoid false alarms and to minimize non-detections. Then, an interactive diagnosis matrix is presented in Section 4: it proposes a solution to guide the expert during the interactive diagnosis process. This interactive diagnosis matrix allows better exploitation of the implicit expert knowledge.

2. Problem statement and crisp logic reasoning

The diagnosis problem for a network of rain gauges is stated in this section.

2.1. Problem under study

The purpose is to set up a computer-aided diagnosis process to determine the faults in a network of rain gauge sensors set up in the Upper Oueme Valley in Benin, with an area of 47 536 km^2. 46 bucket rain gauges, that switch whenever 5 mm of water is received, are available. Their latitude and longitude coordinates are known. The objective is to diagnose the faults of the rain gauges.

2.2. Design of detection tests

The first task (task I in Fig. 1) of a diagnostic process is generally the design of detection tests. The main idea for testing the rain gauges is to compare data from two nearby sensors. Therefore, each consistency test is built by a couple of sensors (rain gauges). Each couple of sensors can be tested by using the average correlation level (residues value) within each month, provided that the distance between two sensors is less than 10 km.[1] This value has been obtained thanks to experience feedback. If a bigger distance is chosen, the interpretation of the generated residues becomes risky because sensors are no longer sufficiently correlated. Conversely, if a smaller value is chosen, many useful correlated test may be lost.

Because of the low density of the sensor network, only 25 out of 46 rain gauges can be tested. These 25 sensors yield 44 tests involving 2 sensors each, as shown in Table 1. Each test is represented by a segment in Fig. 2.

2.3. Symptom generation

The second task (task II in Fig. 1) of a diagnostic process is the symptom generation. Symptoms are generated thanks to a threshold of residuals (also called decision threshold).

[1] Hydrologists estimate that, for the considered problem of rainfall, if the distance between two sensors is more than 10 km, the rainfall amounts received by these two sensors are independent (Depraetere et al., 2009).