Annual Reviews in Control 45 (2018) 257–266

Contents lists available at ScienceDirect

Annual Reviews in Control

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

On the history of diagnosability and opacity in discrete event systems☆

Stéphane Lafortune[a,*], Feng Lin[b], Christoforos N. Hadjicostis[c]

[a] Department of EECS, University of Michigan, Ann Arbor, USA
[b] Department of ECE, Wayne State University, Detroit, USA
[c] Department of ECE, University of Cyprus, Nicosia 1678, Cyprus

ARTICLE INFO

Article history:
Received 11 January 2018
Revised 12 March 2018
Accepted 3 April 2018
Available online 11 April 2018

Keywords:
Discrete event systems
Fault diagnosis
Diagnosability
Opacity
Petri nets
History of discrete event systems

ABSTRACT

This paper presents historical remarks on key projects and papers that led to the development of a theory of event diagnosis for discrete event systems modeled by finite-state automata or Petri nets in the 1990s. The goal in event diagnosis is to develop algorithmic procedures for deducing the occurrence of unobservable events, based on a formal model of the system and on-line observations of its behavior. It also presents historical remarks on the early works on the property of opacity, which occurred about ten years later. Opacity can be seen as a strong version of lack of diagnosability and it has been used to capture security and privacy requirements. Finally, diagnosability is connected with the property of observability that arises in supervisory control. This paper is part of set of papers that review the emergence of discrete event systems as an area of research in control engineering.

1. Introduction

We were invited to provide historical remarks on the emergence of the theories of diagnosability and opacity for event-driven dynamic systems, in the context of the special section in *Annual Reviews in Control* on the emergence of discrete event systems as an area of research in control engineering. The discussion herein is not meant to be a survey of these theories. The two papers Zaytoon and Lafortune (2013) and Jacob, Lesage, and Faure (2016) should be consulted in that regard. Instead, our focus is on presenting key events and papers from the 1990s that led to the definition of a formal notion of diagnosability, and later opacity, which have withstood the test of time and are still the object of current research.

We start by presenting the emergence of fault diagnosis and diagnosability using automata models of discrete event systems. We then transition to similar historical remarks on the notion of opacity and its verification when using automata models. Opacity, which is closely related but stronger than non-diagnosability, originated in the formalization of information flow security properties in computer systems in the early 2000s (cf. Mazaré, 2004a; Ryan & Peacock, 2006) and since then it has attracted considerable attention in the control engineering community. In a subsequent section, we also include brief historical remarks on parallel activities on fault diagnosis and opacity using Petri net models of discrete event systems. We conclude with a discussion on (i) the relationship between diagnosability and the property of observability that arises in supervisory control, and (ii) recent efforts on networked discrete event systems.

Our presentation is focused on events that we personally experienced or witnessed. In the discussion that follows, whenever possible, we cite the first journal paper on the topic being discussed; quite often, this first journal paper was preceded by one or more conference papers that are not cited.

This paper is an expanded version of Lafortune and Lin (2017), with additional discussions throughout, as well as the inclusion of a new section on diagnosability and opacity for Petri net models.

2. History of diagnosability

2.1. The beginning

From our own perspectives, the development of the theory of diagnosability reviewed in this paper was highly influenced by two key events: (i) the extended visit of Feng Lin at Ford Motor Co. in Dearborn, US, in the summer of 1992; and (ii) the year-long

☆ The authors' research is principally supported by the US National Science Foundation and the European Commission.

* Corresponding author.
E-mail addresses: stephane@umich.edu (S. Lafortune), flin@wayne.edu (F. Lin), chadjic@ucy.ac.cy (C.N. Hadjicostis).

https://doi.org/10.1016/j.arcontrol.2018.04.002
1367-5788/

sabbatical visit of Kasim Sinnamohideen of the research group of Johnson Controls Inc. (Milwaukee, US) at the University of Michigan in Ann Arbor in academic year 1992-93, where he collaborated with Stéphane Lafortune, Demosthenis Teneketzis, and two doctoral students at the time, Meera Sampath and Raja Sengupta.

At the time, the two main approaches for fault diagnosis were the quantitative approach in control engineering based on continuous models and the qualitative approach used in artificial intelligence based on static models.

At Ford Motor Co., there was interest in on-board diagnostics to detect and isolate (i.e., diagnose) component faults in complex processes such as the exhaust gas recirculation system, during the operation of the vehicles, i.e., "on-line". At Johnson Controls Inc., there was similar interest for Heating, Ventilation, and Air Conditioning (HVAC) systems, where faulty components are typically difficult to access. Common to these applications of interest was the fact that sensor information was limited and hence diagnosis would require inferencing from limited sensor data using a suitable model of the dynamic system under consideration. Sinnamohideen advocated that using dynamic but high-level "discrete-transition-based" models of HVAC systems, rather than detailed continuous models based on differential equations, was the right approach for diagnosing "sharp" faults, such as valves that get stuck open or closed, pumps that fail on or off, controller modules that fail on or off, and so forth. The thesis was that such discrete-event model-based inferencing would be complementary to other diagnostic approaches that would track "finer" faults, such as slow drifts of sensors for instance.

The work of Lin at Ford Motor Co. lead to a framework of diagnosis using states to model faults (Lin, 1994). The state set is divided into subsets or cells. Some cells represent normal operation, other cells represent various faults. The goal of diagnosis is to determine which cell the system is in after observation of some observable events. This approach was later used in mixed-signal circuit testing, where both digital circuits and analog circuits are modelled as discrete event systems in a uniform way (Lin, Lin, & Lin, 1997).

The group at the University of Michigan, inspired by the work of Lin (1994) and aptly guided by the practical expertise of Sinnamohideen, formulated and investigated a notion of *diagnosability* for discrete event dynamic systems modeled in the framework of regular languages and their finite-state automata representations. This effort led to the doctoral dissertation of Sampath and to the two companion journal papers (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995; 1996). We now discuss some aspects of that work.

Definition of Diagnosability: First, we state the definition of diagnosability. Consider a system modeled by an automaton, denoted by G, and where $\mathcal{L}(G)$ is the language generated by G, and where natural projection P erases the unobservable events. There are several fault events to diagnose, corresponding to the set Σ_f, which is partitioned in several fault types according to Π_f. Formally, as originally stated in Definition 1 in Sampath et al. (1995), we have:

Definition 1 (Diagnosability (Sampath et al., 1995))**.** A prefix-closed and live language L is said to be *diagnosable* with respect to the projection P and with respect to the partition Π_f on Σ_f if the following holds

$$(\forall i \in \Pi_f)(\exists n_i \in \mathbb{N})[\forall s \in \Psi(\Sigma_{fi})](\forall t \in L/s)[\ ||t|| \geq n_i \Rightarrow D\ ]\ , \tag{1}$$

where the diagnosability condition D is

$$\omega \in P_L^{-1}[P(st)] \Rightarrow \Sigma_{fi} \in \omega\ . \tag{2}$$

Explanation of notation: (i) $\Psi(\Sigma_{fi})$ is the set of strings that end with a fault event of type i; (ii) L/s is the set of all strings t such that $st \in L$; (iii) $||t||$ is the length (number of events) of t; (iv) $P_L^{-1}[P(st)]$ is the set of all strings in L whose projection under P is the same as that of st; (v) $\Sigma_{fi} \in \omega$ means that string ω contains a fault event of type i.

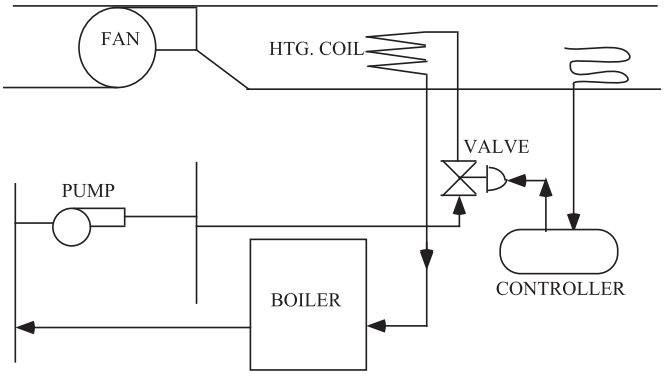

Fig. 1. Heating part of an HVAC system, as studied in Sampath et al. (1995).

Simplifying the above definition to eliminate "fault types" and considering a single event f to diagnose, we get:

Definition 2 (Diagnosability of event *f*). Event f is diagnosable in live language $L = \mathcal{L}(G)$ w.r.t. projection P if the following holds

$$(\exists n \in \mathbb{N})(\forall s : f \in s)(\forall t \in L/s)[\ ||t|| \geq n \Rightarrow D\ ]\ , \tag{3}$$

where the diagnosability condition D is

$$\omega \in P^{-1}[P(st)] \cap \mathcal{L}(G) \Rightarrow f \in \omega\ . \tag{4}$$

The set $P^{-1}[P(st)] \cap \mathcal{L}(G)$ is the best estimate of what the system could have done based on observing st. In words, the definition of diagnosability for a single event f is then:

> An unobservable (fault) event f is diagnosable in language $\mathcal{L}(G)$ if *every* occurrence of f can be detected with *certainty* in a *bounded* number of events *after* its occurrence.

This notion of diagnosability is strong and invokes the universal quantifier twice: *for every* trace of events that ends with a fault event and *for every* continuation of that trace, the fault in question must *eventually* be diagnosed. Since logical discrete event models were employed, "eventually" was quantified by counting the number of events (either observable events or all events as in the above definitions) after the fault event; the existential quantifier captures the existence of such a bound, denoted by n, over the entire system language.

This language-based definition of diagnosability proved to be adaptable to extensions to different modeling formalisms and diagnostic architectures, as the ensuing work performed in the control engineering and artificial intelligence communities shows.

Model-Building for Diagnosability: During the development of the results in Sampath et al. (1995, 1996), the authors spent significant efforts on building suitable discrete event models for the purpose of fault diagnosis, using the application area of HVAC systems as a guide. The objective was to capture all available information for nominal behavior and for faulty behavior, including information from sensor readings, in the form of traces of events generated by a finite-state automaton, i.e., in a regular language. The initial example that was analyzed was the heating part of an HVAC system, as depicted in Fig. 1, where the system components of interest were: controller, valve, pump, boiler, fan, and heating coil. The sensors available (not indicated in the figure) were a flow sensor for the heating coil and a pump pressure sensor.

Faults (not their symptoms) were explicitly modeled by *unobservable* events; this included potentially faulty sensors themselves. Faulty behavior was also modeled, including: (i) the be-

havior of the controller when it fails "on" or "off", and similarly for the pump; and (ii) the valve getting stuck in the "open" or "closed" position. High-level abstractions of component models and of closed-loop system dynamics, the latter captured by the controller model, were merged by parallel composition to obtain a "composite" model. This model was still incomplete as it did not incorporate the information provided by the sensors attached to the system, namely the flow sensor and pump pressure sensor in the system of Fig. 1. For this purpose, the readings of all the sensors attached to the system were recorded in a *sensor table* for all the reachable (discrete) physical states of the composite model. Since the goal was to have a language-based model, it was necessary to convert the information in the sensor table to events. This was done by a conversion similar to that of a Moore automaton to a standard one: the information from the sensor readings in each discrete state was embedded into the event set and transition labels into the states (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1996). Moreover, additional states were introduced when unobservable events caused observable changes in sensor readings.

The resulting model was event-based and fit into the paradigm of the theory of supervisory control of discrete event systems, developed earlier by Ramadge and Wonham (1989). Interestingly, an equivalent state-based formulation (with faulty and non-faulty system *states*, as was initially done in Lin (1994) but closer to the methodology in Sampath et al. (1995)) was later worked out by Hashtrudi Zad, Kwong, and Wonham at the University of Toronto (Hashtrudi Zad, Kwong, & Wonham, 2003). This formulation allowed the authors to take advantage of state reduction techniques to address scalability issues.

During a summer internship at Johnson Controls Inc., Sampath expanded upon the HVAC case studies reported in Sampath et al. (1996) and developed a diagnostic engine for a variable air volume terminal box application, a new type of HVAC system being considered at the time; her results appeared in the internal technical report (Sampath, 1995).

Diagnoser Automata: Diagnosability analysis was performed by building the so-called *diagnoser*, another automaton that captures diagnostic information in its state. In short, the diagnoser for a single fault type can be constructed by first forming the parallel composition of the system model with a label automaton that memorizes, in its state, the occurrence of the events in the given fault type, and second by building the *observer* of the resulting automaton with unobservable events (i.e., determinizing using subset construction). Surprisingly (perhaps), it turned out that the previously-defined notion of diagnosability was verifiable by a relatively simple cycle condition on the diagnoser. Hence, an effective test for diagnosability was born. The cycles in the diagnoser that caused a violation of diagnosability were termed *indeterminate cycles*. Their detection entailed examining not only diagnoser cycles but also their corresponding cycles in the system model. These cycles captured the existence of two traces of events, with identical projections, where one trace was fault free while the second one contained an arbitrarily long suffix after the fault event, thereby providing a counter-example to diagnosability. It is possible to make the diagnosability test "self-contained" (i.e., not requiring the examination of corresponding system cycles) by using a modified form of the diagnoser, called *extended diagnoser*, where predecessor system states are remembered in diagnoser states. Extended diagnosers were first defined in the published literature in Debouk, Lafortune, and Teneketzis (2000), where they are credited to Sampath.

Early Software Tools: The need to implement in software the construction of the diagnoser and the diagnosability test based on indeterminate cycles provided the primary impetus for the development of the software tool UMDES, originally known as UMDES-LIB, at the University of Michigan in the period 1994–1996. The tool eventually grew to include implementations of various manipulations of automata (e.g., parallel composition) and of supervisory control algorithms (e.g., relevant supremal or infimal controllable languages). UMDES is still being maintained as a stand-alone tool, as well as an embedded set of commands within the tool DESUMA, developed by the Michigan team in collaboration with Laurie Ricker of Mount Allison University, Canada (DESUMA Team, 2016).

2.2. Emergence of a comprehensive theory

The work in Sampath et al. (1995, 1996) immediately attracted considerable attention not only in the discrete event systems community, but also in the control engineering community at large as well as in the artificial intelligence community, where fault diagnosis was (and remains) a problem of strong interest. As in the initial development, practical problems motivated many of the extensions that ensued. These efforts led to many extensions or variations of the notion of diagnosability in Sampath et al. (1995). These extensions covered several aspects, such as: (i) enhanced classes of models that include time and/or stochastic information; and (ii) different diagnostic architectures that capture the modular nature of complex systems and/or the decentralization of information in distributed systems. The reader is referred to the survey paper Zaytoon and Lafortune (2013) for a detailed (but still admittedly incomplete) coverage of these efforts along with an extensive bibliography and a review of applications.

We focus the following discussion in this and the next subsection on some key aspects of the development of the comprehensive theory of (event) diagnosis for discrete event systems.

Following the completion of her dissertation, Sampath joined Xerox Corp. where she demonstrated the practical applicability of the theory in (Sampath et al., 1995) to a diagnostic problem in printing systems: diagnosis of faults in the paper feeder system (Sampath, 2001). This application required merging diagnosis techniques for continuous processes with the event-based framework of Sampath et al. (1995). This was done by capturing the results of the diagnosis of the continuous processes as a "virtual" sensor with discretized values in the event-based model.

Regarding the verification of diagnosability, it was realized independently and concurrently by Jiang et al. in Jiang, Huang, Chandra, and Kumar (2001) and Yoo et al. in Yoo and Lafortune (2002) in the early 2000s that diagnosability could be tested in polynomial time, without building the diagnoser. In hindsight, this result is not surprising. The negation of diagnosability involves two applications of the existential quantifier; roughly speaking, one must build two traces of events that have the same observable projection, where one contains the fault event of interest and the other one does not. This construction can be achieved by performing a suitable product of two copies of the automaton modeling the system to be diagnosed. This technique is referred to as the "verifier" approach or the "twin-plant" approach.

An important extension that researchers began pursuing a few years later was the adaptation of the logical approaches for fault diagnosis and diagnosability to stochastic settings. Starting with the work ofThorsley and Teneketzis (2005), which introduced and analyzed the two concepts of *A- diagnosability* and *AA- diagnosability* in stochastic automata (essentially non-deterministic finite-state automata with probabilities on the transitions), a number of researchers studied these notions in probabilistic settings: for example, without regards to diagnosability conditions, Athanasopoulou et al. developed efficient recursive (on-line) algorithms for fault diagnosis in the presence of unreliable observations (Athanasopoulou, Li, & Hadjicostis, 2010), whereas the works in Athanasopoulou and Hadjicostis (2008) and Chen and Kumar (2015) respectively were concerned with bounds on the prob-

ability of erroneous decisions in fault diagnosis applications. Also worth mentioning are more recent works byBertrand, Fabre, Haar, Haddad, and Hélouët (2014), andKeroglou and Hadjicostis (2015a), which adopt stochastic settings for diagnosis and state estimation applications, respectively.

Yet another extension that was initially considered in the 2000s by Contant et al. in Contant, Lafortune, and Teneketzis (2004) and has recently been revisited by Fabre et al. in Fabre, Hélouët, Lefaucheux, and Marchand (2018) is the case of "intermittent" or "repairable" faults, i.e., faults that can appear but be repaired. In this case, diagnosability would ideally require that each occurrence of an intermittent fault be detected before it is repaired.

2.3. Initial efforts on decentralized and distributed architectures

Fairly early on, researchers investigated diagnostic architectures where the information is *decentralized*. At the risk of omitting the noteworthy efforts of several groups, we highlight two such distinct efforts, that occurred concurrently.

(1) At IRISA-INRIA in Rennes, France, two groups led respectively by Albert Benveniste and Marie-Odile Cordier were investigating how to make sense of the flow of alarms in packet-switched telecommunication networks. Among their many publications, we mention Aghasaryan, Fabre, Benveniste, Boubour, and Jard (1998), Benveniste, Fabre, Haar, and Jard (2003), Rozé and Cordier (2002) and Pencolé and Cordier (2005). These groups considered both automata-based models, as in Sampath et al. (1996), and Petri-net-based models or variations thereof, that captured the partial ordering of events that emerges due to the distributed nature of the system. A key concept used therein was the notion of *net unfolding* for constructing global state estimates from local state estimates in the diagnostic process. (We explicitly discuss diagnosis of Petri nets later on in Section 4.1.) Related ideas in the context of modular systems consisting of interacting finite automata also appeared in the works by Fabre and collaborators in Fabre, Benveniste, and Jard (2002), Fabre and Benveniste (2007) and Su and Wonham (2005).

(2) At the University of California at Berkeley, Sengupta was interested in diagnosing the (wireless) communication network used by self-driving vehicles that traveled in platoons as a way to increase throughput on highways. Automated vehicle platooning was then a major effort in the US involving a consortium led by an organization called PATH (Program for Advanced Technology on the Highway) at Berkeley, whose director at the time was Pravin Varaiya. Sengupta first presented his results on decentralized diagnosis at WODES 1998 (Sengupta, 1998). His work was an important source of inspiration for the doctoral research of Rami Debouk at Michigan, which led to Debouk et al. (2000). At around the same time, Stavros Tripakis, working with Sengupta at UC Berkeley, proved the first undecidability results for a version of decentralized diagnosis where communication delays are not bounded (Tripakis, 2004).

In order to improve upon the decentralized solution in Debouk et al. (2000) (referred to as Protocol "3" therein) and at the same time avoid the undecidability that arises in the framework of Tripakis (2004), researchers subsequently investigated various decentralized architectures with either a coordinator (as in the protocols called Protocols "1 and 2" in Debouk et al., 2000), or by enhancing the local diagnostic modules to perform "conditional decisions" or use "inference mechanisms" or set-intersection refinements, as in the approaches initially developed in Wang, Yoo, and Lafortune (2007) and Kumar and Takai (2009), which themselves were followed by several extensions, and the subsequently developed "intersection-based" schemes in Panteli and Hadjicostis (2013) and Keroglou and Hadjicostis (2015b). A different line of work focused on performing state estimation or diagnosis while minimizing the exchange of information between a set of sensors and a centralized coordinator, as in the works in Ricker and van Schuppen (2001), and Boel and van Schuppen (2002).

Overall, the literature on decentralized and distributed approaches for diagnosis is nowadays very comprehensive and the topic is still active (see, e.g., Takai & Kumar, 2018).

3. History of opacity

While diagnosability was undergoing heavy investigation in discrete event systems, privacy and security were being investigated for computer/cyber systems. The two problems are related in the following sense: Diagnosability requires that sufficient information is provided to a diagnoser so that failures can be detected, while privacy and security require that information is concealed to an external agent so that certain secrets are not revealed.

As more and more information is exchanged in cyber systems, the task of ensuring security and privacy in information flow is increasingly becoming an important problem. Various approaches have been considered. For example, *intransitive noninterference* was investigated to ensure that no information is unintentionally being leaked from a "high level" to a "low level" in a multi-level system (Hammer, Krinke, & Nodes, 2006; Mantel & Sands, 2004; Roscoe & Goldsmith, 1999; Rushby, 1992). Intuitively, unintentional information leaking can occur, for example, in the following situation. Consider a setting in which we are given a finite-state automaton with events that are partitioned into two sets, a set of high level events and a set of low level events. Though low level events are observable to a low level entity (an entity with low level clearance), only some of the high level events are observable to the low level. Despite the fact that the low level entity does not observe the high level events, the low level entity may still be able to infer that some high level events have occurred by analyzing the sequence of events it observes (i.e., the low level events and some of the high level events). The detection of unintentional information leaking, such as information about the occurrence of high level events that are not directly observable to the low level entity, is an important problem in cyber systems. A formal method for checking intransitive noninterference was proposed in Ben Hadj-Alouane, Lafrance, Lin, Mullins, and Yeddes (2005a,b). It translates the problem of checking intransitive noninterference into a problem of checking *observability* of discrete event systems (cf. the discussion in Section 5.1).

Opacity is another important information flow property related to privacy and security. Roughly speaking, a system is opaque if some secrets are never revealed during the operation of the system. In other words, any trace (trajectory) of the system that reveals a system secret should be indistinguishable from another trace that does not reveal the system secret to an external agent or observer. Opacity can be paraphrased as "plausible deniability," i.e., the possibility (however unlikely) that the system behavior does not imply that the secret is revealed. Depending on how systems are modeled and how secrets are described, there are several formal ways to define opacity.

To our knowledge, opacity was first introduced by Mazaré in the computer systems literature (Mazaré, 2004b) to investigate security and privacy in information flow in the analysis of cryptographic protocols. Soon after, this work led to related papers that used as modeling formalisms Petri nets (Bryans, Koutny, & Ryan, 2005) and transition systems (Bryans, Koutny, Mazaré, & Ryan, 2008). At around the same time, researchers in Europe introduced the notion of enforcing "concurrent secrets" in the dynamics of discrete event systems in Badouel, Bednarczyk, Borzyszkowski, Caillaud, and Darondeau (2007). It is fair to say that this last paper, as well as the work of Mazaré (2004b), became a source of inspira-

tion for the work of several groups in the discrete event systems community.

In 2006, Christoforos Hadjicostis spent two weeks at INRIA, Rennes, France for a project jointly funded by the University of Illinois at Urbana-Champaign and INRIA. The idea of state-based opacity was born during that visit from discussions with several INRIA researchers, most notably Philippe Darondeau, Benoît Caillaud, Eric Fabre, Thierry Jéron, Hervé Marchand, and doctoral student Jérémy Dubreil. After returning to Illinois, Hadjicostis and his student Anooshiravan Saboori formalized state-based opacity as follows. A system is opaque if the evolution of its true state through a set of secret states remains opaque to an external agent. In other words, the external agent is never certain that the current state of the system is within the set of secret states (Saboori & Hadjicostis, 2007). The external agent observes the system via a projection, that is, only the occurrences of some observable events can be seen by the external agent.

The formal definition of current state opacity for a finite-state automaton under a natural projection map is described below.

Definition of Current State Opacity: Consider a system modeled by a deterministic finite-state automaton, denoted by $G = (X, \Sigma, \delta, x_0)$ where X is the set of states, Σ is the set of events, δ: $X \times \Sigma \to X$ is the (possibly partially defined) state transition function, and x_0 is the initial state. Formally, as originally stated in Definition 1 of Saboori and Hadjicostis (2007), we have the following definition:

Definition 3 (Current-State Opacity (Saboori & Hadjicostis, 2007))**.** Given a deterministic finite-state automaton $G = (X, \Sigma, \delta, x_0)$ and a projection map P with respect to the set of observable events Σ_{obs}, G is opaque with respect to a set of secret states $S \subseteq X$ and the projection map P, or (S, P)-opaque, if

$$\forall t \in \mathcal{L}(G), \exists s \in \mathcal{L}(G) \setminus \{t\} : \delta(x_0, t) \in S \Rightarrow \{\delta(x_0, s) \notin S \text{ and } P(s) = P(t)\}. \quad (5)$$

In words, the above definition requires that every possible execution in the system results in a sequence of observations that allows the observer to deduce that the state of the system is either exclusively in the set of non-secret states $X \setminus S$, or in some states in S but also in at least one state in $X \setminus S$. Stated in a simpler way, each sequence of events results in a sequence of observations, such that at least one non-secret current state is possible.

The observer automaton can be used to determine the "state estimate" of the system at any time, which is defined as the set of all possible states that the system may be in. If a state estimate contains only secret states, then opacity is violated. Hence, the observer can be used to check opacity (Saboori & Hadjicostis, 2011). The construction of the observer is worst-case exponential with respect to the number of states in the system.

Meanwhile, at INRIA in Rennes, while investigating the problem of detection of intrusion in a system using fault diagnosis techniques, Marchand's group also encountered the notion of opacity. Their work led to supervisory control for opacity (Dubreil, Darondeau, & Marchand, 2010). They considered the situation where the given system is not opaque and the task of the supervisor is to restrict the system's behavior by disabling some events so that the supervised system will not enter states where the secret is revealed. Since the supervisor is internal, it can potentially observe more events than the external agent. An effective algorithm for computing the most permissive supervisor was developed for this purpose. Enforcing opacity using supervisory control techniques was also investigated by other groups, such as Saboori and Hadjicostis (2008).

Since diagnosability refers to the ability of distinguishing certain things (occurrences of faults vs. no occurrence of faults), while opacity refers to the inability of distinguishing certain things (secret states vs. non-secret states), intuitively, it seems that opacity is related to diagnosability (or rather lack of diagnosability) in some ways. This relation was formally established by Lin using language-based opacity (Lin, 2011). Two languages are used in this definition, rather than one language and its complement as in the previous definitions. The two-language definition is more flexible in the applications of opacity and is given as follows. A language is *strongly opaque* with respect to a second language if *all* traces in the first language are indistinguishable from some traces in the second language, where two traces are indistinguishable if their projections (observations by an external agent) are the same. A language is *weakly opaque* with respect to a second language if *some* traces in the first language are indistinguishable from some traces in the second language. A language is not opaque with respect to a second language if it is not weakly opaque with respect to the second language. The notion of opacity mentioned previously is strong opacity. It was shown in Lin (2011) that weak opacity (and its negation, non-opacity) is related to other properties of discrete event systems, including diagnosability. More precisely, let the first language be the set of all traces of the system in which no fault has occurred and the second language be the set of all traces of the system in which at least one fault has occurred and some positive number of events have occurred since the last fault. Then a discrete event system is diagnosable if and only if the first language is not opaque with respect to the second language. Since diagnosability corresponds to the negation of a weak version of opacity, in a sense, opacity is a strong version of lack of diagnosability.

Since weak opacity is the negation of diagnosability, it is not surprising that verifying certain versions of weak opacity can be done with polynomial time complexity. On the other hand, verifying strong opacity cannot be done with polynomial time complexity, as demonstrated in Cassez, Dubreil, and Marchand (2012). Intuitively, this is due to the fact that the negation of strong opacity requires the existential quantifier followed by the universal quantifier: there must exist a secret string for which none of the non-secret strings are observationally-equivalent.

Logical opacity does not provide a measure of how opaque a system is; it simply considers that behavior that is not secret (however unlikely) can be matched to a given sequence of observations. Thus, in an effort to better characterize the "degree" of opacity of a given system, researchers have also considered quantitative (as opposed to qualitative) opacity via extensions to probabilistic settings. These include probabilistic finite-state automata (Bérard, Mullins, & Sassolas, 2015; Saboori & Hadjicostis, 2014), Markov decision processes (Bérard, Chatterjee, & Sznajder, 2015a; Bérard, Haddad, & Lefaucheux, 2017), and hidden Markov models (Keroglou & Hadjicostis, 2018) (see also the special issue in Andres, Palamidessi, & Smith, 2015). It is worth pointing out that some of the earlier works on opacity originated in probabilistic settings (e.g., Lakhnech & Mazaré, 2005).

As was mentioned earlier, decidability and complexity for a variety of (qualitative) opacity problems in finite-state automata has been relatively well-characterized (see the detailed discussions on this topic in the survey paper (Jacob et al., 2016)). Quantitative notions of opacity have been introduced more recently, and their decidability and complexity are still the subject of ongoing research. For example, certain quantitative opacity properties have been shown to lead to undecidable problems (Jacob et al., 2016; Saboori & Hadjicostis, 2014); however, other formulations can be verified even with polynomial complexity (Bérard, Mullins et al., 2015; Saboori & Hadjicostis, 2014).

4. Diagnosability and opacity for Petri nets

Following the developments on diagnosability and opacity using finite-state automata models, many different approaches for tackling these problems using Petri net models were developed. In this section we review some of the key events and contributions in these areas of research. The reader is also referred to Zaytoon and Lafortune (2013) and Jacob et al. (2016) for brief surveys of research in fault diagnosis and opacity, respectively. The reader may also refer to a companion history paper by Giua and Silva on the use of Petri net models for the analysis and control of discrete event systems which is included in the same special section of Annual Reviews in Control (Giua & Silva, 2018).

4.1. Diagnosability of Petri nets

The introduction of Petri nets in Petri (1962) was followed by substantial research efforts on analyzing their properties and structure, and led to their adoption by the control community as an important modeling tool for discrete event systems in a variety of applications (ranging from manufacturing and process engineering to computer systems and network/traffic protocols (Giua & Silva, 2018)). Naturally, the control community was interested not only in pure mathematical analysis of Petri nets and their properties, but also in their modeling power and in mechanisms for proper monitoring and supervision of the underlying systems.

Towards this end, in the late 1980s and 1990s several researchers exploited structural properties of Petri nets in order to obtain enhanced Petri nets that are fault-tolerant, i.e., capable of overcoming one or more faults that might occur during their operation. Early works in this regard include the works by Sifakis in Sifakis (1979) followed by the works in Silva and Velilla (1985), and Hadjicostis and Verghese (1999), which systematically constructed Petri net embeddings that can handle combinations (up to a given maximum number) of so-called *transition* and *place* faults. Effectively, these works were attempting to re-design a given Petri net so that the analysis of its marking (state) at any given time would allow the detection/identification of faults. A typical assumption would be that the firing of transitions is not observable, but the number of tokens at each place (i.e., the marking of the Petri net) is observable; thus, based on the marking of the Petri net at a given time instant, one could exploit the (generalized) Hamming distance properties that were systematically enforced, in order to make inferences about "faults" (e.g., the firing of certain transitions of interest) that might have taken place during the operation of the Petri net (Wu & Hadjicostis, 2005).

In the early 1990s, there were also works on fault diagnosis for Petri nets that did not amount to re-designing an enhanced Petri net. For example, the work in Prock (1991) considered the detection of faults by monitoring the number of tokens in P-invariants of the given Petri net (in contrast to the constructed P-invariants of the above mentioned works). Similarly, the work in Srinivasan and Jafari (1993) attempted to backtrack transition firing in order to determine valid behavior and eventually detect/identify faults.

Following the work on diagnosability using automata in Sampath et al. (1995), researchers started considering the same problem for Petri net models. One of the earliest examples is work in Ushio, Onishi, and Okuda (1998), which (following the observability assumptions in the works mentioned above) assumed unobservable transitions with partial marking observation and constructed a diagnoser as in Sampath et al. (1995) to perform online fault detection and verify diagnosability. Soon after, Chung considered a similar model in Chung (2005) but allowed some of the transitions to be observable; this was perhaps the first work where a language-based approach with unobservable transitions was adopted for diagnosing Petri nets.

It soon became evident that fault diagnosis in a given Petri net could be studied under more general observation viewpoints that would involve partial marking observations (i.e., the number of tokens in some of the places) and some information about the firing of transitions (e.g., labels associated with the firing of some of the transitions, including the empty label for the firing of unobservable transitions). State estimation and fault diagnosis is such *interpreted* Petri net settings were pursued by several researchers; see, for example, the works by (Ramírez-Treviño, Ruiz-Beltrán, Rivera-Rangel, & Lopez-Mellado, 2007), (Lefebvre & Delherm, 2007), and (Ru & Hadjicostis, 2009).

The work in Ru and Hadjicostis (2009) established that each labeled Petri net under partial marking observation can be transformed to an observationally equivalent labeled Petri net (under no marking observation). Furthermore, a direct translation of the fault diagnosis setting in Sampath et al. (1995) from finite-state automata to Petri nets would imply the existence of a set of observable transitions, with some of them perhaps sharing the same label, and a set of unobservable transitions, some of which constitute faults whose occurrence needs to be inferred after a bounded number of observations. These observations prompted several researchers to focus on the study of state estimation and fault diagnosis in labeled Petri nets or timed Petri nets (e.g., Basile, Cabasino, & Seatzu, 2015; Cabasino, Giua, Pocci, & Seatzu, 2011; Cabasino, Giua, & Seatzu, 2010; Cabasino, Giua, & Seatzu, 2013; Dotoli, Fanti, Mangini, & Ukovich, 2009).

While fault diagnosis using Petri net models shares many similarities with fault diagnosis in automata, there are also several important differences. For example, Petri nets can have an unbounded number of states, which implies that the construction of a diagnoser or a verifier (for checking diagnosability) may not be straightforward (Cabasino, Giua, Lafortune, & Seatzu, 2012). In any case (even when one deals with bounded Petri nets, which implies that direct translation of the techniques used in automata is possible), it might be desirable to exploit the structure of a given Petri net to obtain more efficient and concise representations of possible sets of markings or faults. Starting in 2003, René Boel and his collaborators investigated such approaches in Boel and Jiroveanu (2003), Jiroveanu, Boel, and Bordbar (2008) and Jiroveanu and Boel (2010). In parallel, during the period 2007–2010, Maria Paola Cabasino (then a doctoral student at the University of Cagliari, under the co-supervision of Alessandro Giua and Carla Seatzu) visited Hadjicostis' group at the University of Illinois and Lafortune's group at the University of Michigan. Methodologies that have built on the work by Giua and Seatzu in Giua and Seatzu (2005) and the intuition developed in these visits have appeared in Cabasino et al. (2011, 2010) and obtain *minimal explanations* and *basis markings* that match a given sequence of observations. An explanation of an observed transition is a (possibly empty) sequence of unobservable transitions that need to fire in order to explain (i.e., enable of the firing of) this observed transition. Assuming that there are no cycles of unobservable transitions, tracking of explanations can be reduced to tracking of firing vectors (i.e., tracking the number of times each unobservable transition has fired, but not the order in which firings take place). In fact, one only needs to track the minimal such firing vectors (explanations) and the corresponding markings they lead to (called basis markings). Faults are modeled as unobservable transitions and are detected (identified) when all minimal explanations include the firing of a fault event (or a specific type of fault event).

Other works have also exploited the structure of Petri nets, but in different ways. Benveniste et al. in Benveniste et al. (2003) (which was mentioned earlier) used net unfoldings to efficiently determine if a given fault occurs without having to reconstruct all possible reachable markings. Genc and Lafortune in Genc and Lafortune (2007) exploited the inherent modularity of Petri net models and used marking information about certain *bordering* places (assumed to be observable) in order to distributively perform fault detection and identification by analyzing information in (smaller) local modules. Finally, Cabasino et al. in Cabasino, Giua, Hadjicostis, and Seatzu (2015) assumed knowledge of a nominal Petri model and tried to identify the structure of the faulty system (if present) using integer linear programming techniques.

While the works mentioned above are mostly concerned with logical state estimation and fault diagnosis, some attempts to better characterize the likelihood of a state or a fault using probabilistic or other measures can be found in Aghasaryan et al. (1998) which was mentioned earlier, and Ru and Hadjicostis (2009) and Cabasino, Hadjicostis, and Seatzu (2015). Parallel efforts have also been pursued in the context of fault diagnosis of discrete event systems using continuous Petri net models by combining logical fault diagnosis techniques with fluid approximations; see, for example, the work by Mahulea et al. in Mahulea, Seatzu, Cabasino, and Silva (2012), which materialized following several exchange visits between the Zaragoza and Cagliari groups.

4.2. Opacity in Petri nets

As was mentioned in Section 3, some of the very first opacity works in dynamic systems (such as Bryans et al., 2005; Bryans et al., 2008) involved Petri net formalisms, but aimed primarily at modeling information flow properties of various protocols and establishing associated (un)decidability results. The attention on language-based and state-based opacity in finite-state automata initiated by Badouel et al. (2007) and Saboori and Hadjicostis (2007), and the subsequent characterization of the verification complexity for several opacity properties of interest (e.g., using observers for verifying current-state opacity (Saboori & Hadjicostis, 2011)), revived research interest on opacity in Petri net models.

Following a visit by Hadjicostis at the University of Cagliari in June 2012, during which he presented a seminar on "State-Based Notions of Opacity in Security Applications of DES," Alessandro Giua, Carla Seatzu, and collaborators started working on opacity problems in Petri nets. These efforts materialized in a few directions, such as the work led by Yin Tong in Tong, Li, Seatzu, and Giua (2017b) that (i) efficiently verifies current-state opacity in bounded Petri nets by exploiting the notion of basis markings; and (ii) proposes an efficient approach for verifying initial-state opacity based on the notion of basis reachability graph. More recently, works by Tong and collaborators have dealt with (un)decidability of certain opacity verification problems in Petri nets (Tong, Li, Seatzu, & Giua, 2017a). Complexity considerations pertaining to opacity (as well as diagnosis) problems of interest have also been addressed by Bérard and collaborators for certain classes of Petri nets in Bérard, Haar, Schmitz, and Schwoon (2017).

The above developments have only touched the tip of the iceberg concerning opacity formalisms in specific Petri net settings; this direction (meaningful opacity formulations in different classes of Petri nets, and characterization of their decidablity and complexity) will be a fertile research area in the years to come.

5. Discussion

5.1. Diagnosability and observability

We describe in this section the relationship between the notions of *diagnosability* and *observability*. We mentioned the property of observability earlier on in Section 3. This discrete-event system-theoretic property was introduced by Lin in his PhD research with his advisor, Murray Wonham, for supervisory control under partial observation (Lin & Wonham, 1988).

A history of supervisory control of discrete event systems is given in the paper by Wonham, Cai, and Rudie included in the same special section of Annual Reviews in Control (Wonham, Cai, & Rudie, 2018). In supervisory control (Lin & Wonham, 1988; Ramadge & Wonham, 1987), the supervisor may disable some controllable events based on its observation of the system trajectory. The task of the supervisor is to ensure that the supervised (or closed-loop) system generates a certain language called specification language. To ensure that a supervisor exists for a given specification language, two important concepts were introduced in supervisory control: controllability (Ramadge & Wonham, 1987) and observability (Lin & Wonham, 1988). A specification language is said to be controllable if all events that need to be disabled at some instances in the system trajectory are controllable. A specification language is said to be observable if two traces in the language that look the same to the supervisor require the same control actions, i.e., no event needs to be disabled after one trace and enabled after the other. In terms of distinguishability, observability ensures that if the control actions required after two traces are inconsistent, then the two traces must be distinguishable. It was shown in Lin and Wonham (1988) that a supervisor synthesizing a given specification language exists if and only if the language is controllable and observable.

In decentralized supervisory control, several local supervisors rather than one central supervisor are used to control a given discrete event system. Each local supervisor has its own set of controllable and observable events. The notion of observability was extended to *coobservability* in the first two papers on decentralized supervisory control, (Cieslak, Desclaux, Fawaz, & Varaiya, 1988; Rudie & Wonham, 1992). Essentially, a specification language is said to be coobservable if the following is true: whenever the control actions on a given controllable event required after two traces in the language are inconsistent, then the two traces are distinguishable to at least one local supervisor that can execute the required control action. Since coobservability is an extension of observability, observability is a special case of coobservability when there is only one (local) supervisor whose event set is the entire event set.

Diagnosability was also extended to codiagnosability for decentralized diagnosis. If several local diagnosers are used to diagnose a system, then a fault can be diagnosed if and only if the system is codiagnosable. Codiagnosability is the same as diagnosability under Protocol 3 in Debouk et al. (2000), which was mentioned in Section 2.2.

Interestingly, it turns out that coobservablilty and codiagnosability are closely related in the sense that the *verification* of each property can be transformed to the *verification* of the other one. The idea is to map a violation of coobservability to a violation of codiagnosability, and vice-versa, by suitably altering the automaton representing the language of interest. The reader can find the transformation of the verification of coobservability to that of codiagnosability in Wang, Girard, Lafortune, and Lin (2011), while the transformation for the other direction is presented in Yin and Lafortune (2015) (the reader is referred to Figure 6 in that paper for a summary of the known transformations). Since observability is a special case of coobservability and diagnosability is a special

case of codiagnosability, observability and diagnosability are also transformable to each other.

5.2. Networked discrete event systems

Turning our attention to recent developments, we wish to mention current research efforts on networked discrete event systems, which explicitly account for communication delays and losses in distributed architectures. Without communication delays and losses, the observer automaton can be used to determine the state estimate of the system. When event observation is delayed or lost in a networked discrete event system, the state estimate obtained using the observer may not be correct. State estimation is an important problem in networked discrete event systems since it is a key task for diagnosability, opacity, and supervisory control. One way to estimate states is proposed in Lin (2014). It is assumed in Lin (2014) that some transitions in G may be lost in the communication channels and communication may be delayed up to M steps (events). A *networked observer* is then constructed to obtain state estimates. The networked observer takes into account both communication delays and losses in a networked discrete event system. State estimation using networked observer has been used in supervisory control of networked discrete event systems (Lin, 2014; Shu & Lin, 2014; 2017), where the existence of a networked supervisor is characterized by network controllability and network observability. Diagnosability of networked discrete event systems with communication delays and losses was investigated in Carvalho, Basilio, and Moreira (2012), Debouk, Lafortune, and Teneketzis (2003) and Takai and Kumar (2012). Since diagnosability does not require specific times for diagnosis, communication delays are less critical than communication losses. Hence, robust diagnosability against intermittent loss of observations was investigated in Carvalho et al. (2012). A necessary and sufficient condition and tests for robust diagnosability were derived. Opacity of networked discrete event systems has yet to be investigated.

6. Conclusion

The objective of this paper was to present historical remarks about the development of the theories of diagnosability and opacity for dynamic systems modeled in the framework of discrete event systems, using either automata or Petri nets. Our recounting was influenced by events that we personally witnessed, and this paper should be read in that context.

Diagnosability and opacity are properties about model-based inferencing that are naturally captured and analyzed in event-driven dynamic systems. This explains why the original works reviewed in this paper have led to an ever-growing literature on these topics. We surmise that interest in these two properties will remain strong in control engineering. Indeed, the theories of diagnosability and opacity have important roles to play in solving current security and privacy problems in advanced control systems that operate in an open environment subject to cyber-attacks.

Acknowledgements

It is a pleasure to acknowledge the efforts of Manuel Silva in organizing the special section on the history of discrete event systems, in which this paper is included. We are indebted to several colleagues for providing comments that were embedded in this article: Alessandro Giua, Hervé Marchand, Meera Sampath, Carla Seatzu, Raja Sengupta, and Demosthenis Teneketzis. We also thank the reviewers for their pertinent comments.

References

Aghasaryan, A., Fabre, E., Benveniste, A., Boubour, R., & Jard, C. (1998). Fault detection and diagnosis in distributed systems: An approach by partially stochastic Petri nets. *Discrete Event Dynamic Systems: Theory and Applications, 82*(2), 203–231.

Andres, M. E., Palamidessi, C., & Smith, G. (2015). Preface to the special issue on quantitative information flow. *Mathematical Structures in Computer Science, 25*(2), 203–206.

Athanasopoulou, E., & Hadjicostis, C. N. (2008). Probability of error bounds for failure diagnosis and classification in hidden Markov models. In *Proceedings of 47th IEEE conference on decision and control (CDC)* (pp. 1477–1482).

Athanasopoulou, E., Li, L., & Hadjicostis, C. N. (2010). Maximum likelihood failure diagnosis in finite state machines under unreliable observations. *IEEE Transactions on Automatic Control, 55*(3), 579–593.

Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., & Darondeau, P. (2007). Concurrent secrets. *Discrete Event Dynamic Systems: Theory and Applications, 17*(4), 425–446.

Basile, F., Cabasino, M. P., & Seatzu, C. (2015). State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions. *IEEE Transactions on Automatic Control, 60*(4), 997–1009.

Ben Hadj-Alouane, N., Lafrance, S., Lin, F., Mullins, J., & Yeddes, M. (2005a). Characterizing intransitive noninterference for 3-domain security policies with observability. *IEEE Transactions on Automatic Control, 50*(6), 920–925.

Ben Hadj-Alouane, N., Lafrance, S., Lin, F., Mullins, J., & Yeddes, M. (2005b). On the verification of intransitive noninterference in mulitlevel security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*(5), 948–958.

Benveniste, A., Fabre, E., Haar, S., & Jard, C. (2003). Diagnosis of asynchronous discrete-event systems: A net unfolding approach. *IEEE Transactions on Automatic Control, 48*(5), 714–727.

Bérard, B., Chatterjee, K., & Sznajder, N. (2015). Probabilistic opacity for Markov decision processes. *Information Processing Letters, 115*(1), 52–59.

Bérard, B., Haar, S., Schmitz, S., & Schwoon, S. (2017). The complexity of diagnosability and opacity verification for Petri nets. In *Proceedings of international conference on applications and theory of petri nets and concurrency* (pp. 200–220).

Bérard, B., Haddad, S., & Lefaucheux, E. (2017). Probabilistic disclosure: Maximisation vs. minimisation. In *Proceedings of 37th iarcs annual conference on foundations of software technology and theoretical computer science (fsttcs 2017). Kanpur, India.*

Bérard, B., Mullins, J., & Sassolas, M. (2015). Quantifying opacity. *Mathematical Structures in Computer Science, 25*(2), 361–403.

Bertrand, N., Fabre, E., Haar, S., Haddad, S., & Hélouët, L. (2014). Active diagnosis for probabilistic systems.. In *Proceedings of international conference on foundations of software science and computation structures (FOSSACS): 14* (pp. 29–42).

Boel, R., & Jiroveanu, G. (2003). Petri nets model based fault section detection and diagnosis in electrical power networks. In *Proceedings of the 6th international power engineering conference* (pp. 24–29). Nanyang Technological University.

Boel, R. K., & van Schuppen, J. H. (2002). Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Proceedings of 6th international workshop on discrete event systems (WODES)* (pp. 175–181).

Bryans, J., Koutny, M., & Ryan, P. (2005). Modeling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science, 121*, 101–115.

Bryans, J. W., Koutny, M., Mazaré, L., & Ryan, P. Y. (2008). Opacity generalised to transition systems. *International Journal of Information Security, 7*(6), 421–435.

Cabasino, M. P., Giua, A., Hadjicostis, C. N., & Seatzu, C. (2015). Fault model identification and synthesis in Petri nets. *Discrete Event Dynamic Systems: Theory and Applications, 25*(3), 419–440.

Cabasino, M. P., Giua, A., Lafortune, S., & Seatzu, C. (2012). A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Transactions on Automatic Control, 57*(12), 3104–3117.

Cabasino, M. P., Giua, A., Pocci, M., & Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets: An application to manufacturing systems. *Control Engineering Practice, 19*(9), 989–1001.

Cabasino, M. P., Giua, A., & Seatzu, C. (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica, 46*(9), 1531–1539.

Cabasino, M. P., Giua, A., & Seatzu, C. (2013). Diagnosis using labeled Petri nets with silent or undistinguishable fault events. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43*(2), 345–355.

Cabasino, M. P., Hadjicostis, C. N., & Seatzu, C. (2015). Probabilistic marking estimation in labeled Petri nets. *IEEE Transactions on Automatic Control, 60*(2), 528–533.

Carvalho, L. K., Basilio, J. C., & Moreira, M. V. (2012). Robust diagnosis of discrete event systems against intermittent loss of observations. *Automatica, 48*(9), 2068–2078.

Cassez, F., Dubreil, J., & Marchand, H. (2012). Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System Design, 40*(1), 88–115.

Chen, J., & Kumar, R. (2015). Failure detection framework for stochastic discrete event systems with guaranteed error bounds. *IEEE Transactions on Automatic Control, 60*(6), 1542–1553.

Chung, S.-L. (2005). Diagnosing PN-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing, 18*(2–3), 158–169.

Cieslak, R., Desclaux, C., Fawaz, A., & Varaiya, P. (1988). Supervisory control of discrete-event processes with partial observations. *IEEE Transactions on Automatic Control, 33*(3), 249–260.

Contant, O., Lafortune, S., & Teneketzis, D. (2004). Diagnosis of intermittent faults. *Discrete Event Dynamic Systems: Theory and Applications, 14*(2), 171–202.

Debouk, R., Lafortune, S., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications, 10*(1–2), 33–86.

Debouk, R., Lafortune, S., & Teneketzis, D. (2003). On the effect of communication delays in failure diagnosis of decentralized discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications, 13*(3), 263–289.

DESUMA Team (2016). DESUMA software tool. https://wiki.eecs.umich.edu/desuma/.

Dotoli, M., Fanti, M. P., Mangini, A. M., & Ukovich, W. (2009). On-line fault detection in discrete event systems by Petri nets and integer linear programming. *Automatica, 45*(11), 2665–2672.

Dubreil, J., Darondeau, P., & Marchand, H. (2010). Supervisory control for opacity. *IEEE Transactions on Automatic Control, 55*(5), 1089–1100.

Fabre, E., & Benveniste, A. (2007). Partial order techniques for distributed discrete event systems: Why you cannot avoid using them. *Discrete Event Dynamic Systems: Theory and Applications, 17*(3), 355–403.

Fabre, E., Benveniste, A., & Jard, C. (2002). Distributed diagnosis for large discrete event dynamic systems. *IFAC Proceedings Volumes, 35*(1), 1–6.

Fabre, E., Hélouët, L., Lefaucheux, E., & Marchand, H. (2018). Diagnosability of repairable faults. *Discrete Event Dynamic Systems: Theory and Applications*.

Genc, S., & Lafortune, S. (2007). Distributed diagnosis of place-bordered Petri nets. *IEEE Transactions on Automation Science and Engineering, 4*(2), 206–219.

Giua, A., & Seatzu, C. (2005). Fault detection for discrete event systems using Petri nets with unobservable transitions. In *Proceedings of IEEE conference on decision and control and european control conference (CDC-ECC)* (pp. 6323–6328).

Giua, A., & Silva, M. (2018). Petri nets and automatic control: An historical perspective. *Annual Reviews in Control*.

Hadjicostis, C. N., & Verghese, G. (1999). Monitoring discrete event systems using Petri net embeddings. *Proceedings of 20th International Conference on Application and Theory of Petri Nets (ICATPN)*. 689–689

Hammer, C., Krinke, J., & Nodes, F. (2006). Intransitive noninterference in dependence graphs. In *Proceedings of 2nd international symposium on leveraging applications of formal methods, verification and validation (ISOLA)* (pp. 119–128).

Hashtrudi Zad, S., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: Framework and model reduction. *IEEE Transactions on Automatic Control, 48*(7), 1199–1212.

Jacob, R., Lesage, J.-J., & Faure, J.-M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control, 41*, 135–146.

Jiang, S., Huang, Z., Chandra, V., & Kumar, R. (2001). A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control, 46*(8), 1318–1321.

Jiroveanu, G., & Boel, R. K. (2010). The diagnosability of Petri net models using minimal explanations. *IEEE Transactions on Automatic Control, 55*(7), 1663–1668.

Jiroveanu, G., Boel, R. K., & Bordbar, B. (2008). On-line monitoring of large Petri net models under partial observation. *Discrete Event Dynamic Systems: Theory and Applications, 18*(3), 323–354.

Keroglou, C., & Hadjicostis, C. N. (2015a). Detectability in stochastic discrete event systems. *Systems & Control Letters, 84*, 21–26.

Keroglou, C., & Hadjicostis, C. N. (2015b). Distributed diagnosis using predetermined synchronization strategies in the presence of communication constraints. In *Proceedings of IEEE international conference on automation science and engineering (CASE)* (pp. 831–836).

Keroglou, C., & Hadjicostis, C. N. (2018). Probabilistic system opacity in discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*.

Kumar, R., & Takai, S. (2009). Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete-event systems. *IEEE Transactions on Automation Science and Engineering, 6*(3), 479–491.

Lafortune, S., & Lin, F. (2017). From diagnosability to opacity: A brief history of diagnosability or lack thereof. In *Proceedings of 20th IFAC world congress* (pp. 3022–3027).

Lakhnech, Y., & Mazaré, L. (2005). Probabilistic opacity for a passive adversary and its application to Chaum's voting scheme.. *IACR Cryptology ePrint Archive, 2005*, 98.

Lefebvre, D., & Delherm, C. (2007). Diagnosis of DES with Petri net models. *IEEE Transactions on Automation Science and Engineering, 4*(1), 114–118.

Lin, F. (1994). Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems: Theory and Applications, 4*(2), 197–212.

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica, 47*(3), 496–503.

Lin, F. (2014). Control of networked discrete event systems: Dealing with communication delays and losses. *SIAM Journal on Control and Optimization, 52*(2), 1276–1298.

Lin, F., Lin, Z. H., & Lin, T. W. (1997). A uniform approach to mixed-signal circuit test. *International journal of circuit theory and applications, 25*(2), 81–93.

Lin, F., & Wonham, W. M. (1988). On observability of discrete-event systems. *Information Sciences, 44*(3), 173–198.

Mahulea, C., Seatzu, C., Cabasino, M. P., & Silva, M. (2012). Fault diagnosis of discrete-event systems using continuous Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 42*(4), 970–984.

Mantel, H., & Sands, D. (2004). Controlled declassification based on intransitive noninterference. In *Proceedings of asian symposium on programming languages and systems* (pp. 129–145). Springer.

Mazaré, L. (2004a). Using unification for opacity properties. *Proceedings of the 4th IFIP WG1, 7*, 165–176.

Mazaré, L. (2004b). Using Unification For Opacity Properties. *Technical Report*. Verimag Technical Report.

Panteli, M., & Hadjicostis, C. N. (2013). Intersection based decentralized diagnosis: Implementation and verification. In *Proceedings of 52nd IEEE conference on decision and control (CDC)* (pp. 6311–6316).

Pencolé, Y., & Cordier, M.-O. (2005). A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence, 164*(1–2), 121–170.

Petri, C. A. (1962). *Kommunikation mit Automaten*. Technischen Hoschule Darmstadt Ph.D. thesis..

Prock, J. (1991). A new technique for fault detection using Petri nets. *Automatica, 27*(2), 239–245.

Ramadge, P. J., & Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization, 25*(1), 206–230.

Ramadge, P. J., & Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of the IEEE, 77*(1), 81–98.

Ramírez-Treviño, A., Ruiz-Beltrán, E., Rivera-Rangel, I., & Lopez-Mellado, E. (2007). Online fault diagnosis of discrete event systems: A Petri net-based approach. *IEEE Transactions on Automation Science and Engineering, 4*(1), 31–39.

Ricker, S. L., & van Schuppen, J. H. (2001). Decentralized failure diagnosis with asynchronous communication between supervisors. In *Proceedings of European control conference (ECC)* (pp. 1002–1006).

Roscoe, B., & Goldsmith, M. (1999). What is intransitive noninterference?. In *Proceedings of computer security foundations workshop: 50*. IEEE Press.

Rozé, L., & Cordier, M.-O. (2002). Diagnosis discrete-event systems: Extending the diagnoser approach to deal with telecommunication networks. *Discrete Event Dynamic Systems: Theory and Applications, 12*, 43–81.

Ru, Y., & Hadjicostis, C. N. (2009). Fault diagnosis in discrete event systems modeled by partially observed Petri nets. *Discrete Event Dynamic Systems: Theory and Applications, 19*(4), 551–575.

Rudie, K., & Wonham, W. M. (1992). Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control, 37*(11), 1692–1708.

Rushby, J. (1992). *Noninterference, transitivity, and channel-control security policies*. SRI International, Computer Science Laboratory.

Ryan, P. Y., & Peacock, T. (2006). Opacity-further insights on an information flow property. *Technical Report Series-University of Newcastle Upon Tyne Computing Science, 958*.

Saboori, A., & Hadjicostis, C. N. (2007). Notions of security and opacity in discrete event systems. In *Proceedings of the 46th IEEE conference on decision and control (CDC)* (pp. 5056–5061).

Saboori, A., & Hadjicostis, C. N. (2008). Opacity-enforcing supervisory strategies for secure discrete event systems. In *Proceedings of the 47th IEEE conference on decision and control (CDC)* (pp. 889–894).

Saboori, A., & Hadjicostis, C. N. (2011). Verification of K-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering, 8*(3), 549–559.

Saboori, A., & Hadjicostis, C. N. (2014). Current-state opacity formulations in probabilistic finite automata. *IEEE Transactions on Automatic Control, 59*(1), 120–133.

Sampath, M. (1995). Discrete Event Systems Based Diagnostics for a Variable Air Volume Terminal Box Application. *Technical Report*. Advanced Development Team, Johnson Controls, Inc..

Sampath, M. (2001). A hybrid approach to failure diagnosis of industrial systems. In *Proceedings of American control conference (ACC)* (pp. 2077–2082).

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control, 40*(9), 1555–1575.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1996). Failure diagnosis using discrete event models. *IEEE Transactions on Control Systems Technology, 4*(2), 105–124.

Sengupta, R. (1998). Diagnosis and communication in distributed systems. In *Proceedings of 4th international workshop on discrete event systems (WODES)* (pp. 144–151).

Shu, S., & Lin, F. (2014). Decentralized control of networked discrete event systems with communication delays. *Automatica, 50*(8), 2108–2112.

Shu, S., & Lin, F. (2017). Predictive networked control of discrete event systems. *IEEE Transactions on Automatic Control, 62*(9), 4698–4705.

Sifakis, J. (1979). Realization of fault tolerant systems by coding Petri nets. *Journal of Design Automation and Fault-Tolerant Computing, 3*(2), 93–107.

Silva, M., & Velilla, S. (1985). Error detection and correction on Petri net models of discrete events control systems. In *Proceedings of IEEE international symposium on circuits and systems (ISCAS)*.

Srinivasan, V. S., & Jafari, M. A. (1993). Fault detection/monitoring using time Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, 23*(4), 1155–1162.

Su, R., & Wonham, W. M. (2005). Global and local consistencies in distributed fault diagnosis for discrete-event systems. *IEEE Transactions on Automatic Control, 50*(12), 1923–1935.

Takai, S., & Kumar, R. (2012). Distributed failure prognosis of discrete event systems with bounded-delay communications. *IEEE Transactions on Automatic Control, 57*(5), 1259–1265.

Takai, S., & Kumar, R. (2018). Implementation of inference-based diagnosis: computing delay bound and ambiguity levels. *Discrete Event Dynamic Systems: Theory and Applications*.

Thorsley, D., & Teneketzis, D. (2005). Diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control, 50*(4), 476–492.

Tong, Y., Li, Z., Seatzu, C., & Giua, A. (2017a). Decidability of opacity verification problems in labeled Petri net systems. *Automatica, 80*, 48–53.

Tong, Y., Li, Z., Seatzu, C., & Giua, A. (2017b). Verification of state-based opacity using Petri nets. *IEEE Transactions on Automatic Control, 62*(6), 2823–2837.

Tripakis, S. (2004). Undecidable problems of decentralized observation and control on regular languages. *Information Processing Letters, 90*(1), 21–28.

Ushio, T., Onishi, I., & Okuda, K. (1998). Fault detection based on Petri net models with faulty behaviors. In *Proceedings of IEEE international conference on systems, man, and cybernetics* (pp. 113–118).

Wang, W., Girard, A. R., Lafortune, S., & Lin, F. (2011). On codiagnosability and coobservability with dynamic observations. *IEEE Transactions on Automatic Control, 56*(7), 1551–1566.

Wang, Y., Yoo, T.-S., & Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems: Theory and Applications, 17*(2), 233–263.

Wonham, W. M., Cai, K., & Rudie, K. (2018). Supervisory control of discrete-event systems: A brief history. *Annual Reviews in Control*.

Wu, Y., & Hadjicostis, C. N. (2005). Algebraic approaches for fault identification in discrete-event systems. *IEEE Transactions on Automatic Control, 50*(12), 2048–2055.

Yin, X., & Lafortune, S. (2015). Codiagnosability and coobservability under dynamic observations: Transformation and verification. *Automatica, 61*, 241–252.

Yoo, T.-S., & Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on Automatic Control, 47*(9), 1491–1495.

Zaytoon, J., & Lafortune, S. (2013). Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control, 37*(2), 308–320.