An Unifying Decision-Making Framework in Discrete-Event Systems: Application to Centralized and Decentralized Control, Diagnosis and Prognosis

Ahmed Khoumsi

Abstract: We develop a decision-making framework for DES, which is unifying in two ways. Firstly, it unifies the architectures: we develop a decision-making system for a generic well-formed architecture; and then, we specialize the framework for centralized and decentralized architectures. Secondly, it unifies supervisory control, diagnosis and prognosis: we develop a framework for a generic well-formed decision-maker; and then, we apply the framework to the three subjects. We hope that this unifying framework will contribute to a better understanding of the decision-making mechanism, and will promote its development.

Keywords: Discrete event systems, unifying decision-making, well-formed architecture, decidability, centralized architecture, decentralized inference architecture, supervisory control, diagnosis, prognosis.

1. INTRODUCTION

In this paper, we develop an unifying decision-making framework for DES, where a decision-maker observes the behavior of the plant and takes decisions that satisfy given properties. Such a framework is unifying in two aspects: unification of architectures, and unification of supervisory control, diagnosis and prognosis. Here are the contributions and the structure of the paper:

Section 2: We formulate the requirements targeted by a generic decision-maker, independently of its architecture. Those requirements consist of a Generic Decisional Requirement (GDR) and a set SDR of Specific Decisional Requirements. It is always possible to construct a decision-maker satisfying GDR. As for the capacity to satisfy SDR, it is in general influenced by the capacity to observe the plant, and sometimes by other parameters. For example, in supervisory control there is the notion of controllability, and in prognosis there is the question: does a fault become certain before its occurrence?

Section 3: We characterize a generic well-formed architecture, for which we can identify necessary and sufficient conditions to the existence of decision-makers satisfying GDR and SDR. Those conditions are formally specified by a notion of decidability. The characterization of well-formed architecture is generic, in the sense that it does not depend on SDR.

We propose a mechanism to synthesize automatically the formal expression of decidability from the formal expression of SDR, for any well-formed architecture.

Section 4: We specialize the framework of Sections 2 and 3 for a centralized architecture. The obtained centralized decision-making is a generic version of the control under partial observation of Cieslak et al. (1988), the diagnosis of Sampath et al. (1995), and the prognosis of Genc and Lafortune (2006).

Section 5: We specialize the framework of Sections 2 and 3 for a specific decentralized architecture. The obtained inference decision-making is a generic version of the inference control of Kumar and Takai (2007), the inference diagnosis of Kumar and Takai (2009), and the inference prognosis of Takai and Kumar (2008).

Sections 4 and 5: We synthesize automatically several existence theorems developed in the above references cited in Sections 4 and 5.

Sections 2, 4 and 5: All results are applied to control, diagnosis, and prognosis.

Section 6: We conclude and propose some future work.

Here are some preliminary terminology and notations. \( \mathbb{N} \) (resp. \( \mathbb{N}^+ \)) is the set of nonnegative (resp. positive) integers. \( \Sigma \) is a set of events, also called alphabet. A finite sequence of events of \( \Sigma \) is called a trace. For a trace \( \lambda \), \( |\lambda| \) denotes the length of \( \lambda \). A trace \( \mu \) is said prefix of a trace \( \lambda \), denoted by \( \mu \leq \lambda \), if there exists a trace \( \nu \) such that \( \lambda = \mu \nu \). \( \Sigma^* \) denotes the set of all traces, including the empty trace \( \epsilon \). For \( \ell \in \mathbb{N} \), \( \Sigma^\ell \) contains every trace \( \lambda \) such that \( |\lambda| \geq \ell \). A set of traces \( K \subseteq \Sigma^* \) is called a language. \( K \) denotes the set of all prefixes of traces of a language \( K \). \( K \) is said prefix-closed if \( K = K \). For two languages \( K \) and \( L \), the language \( K \setminus L \) contains every trace \( \lambda \) such that \( \lambda \in K \) and \( \lambda \notin L \).

For a property \( \pi \) defined in a language, \( \lambda \models \pi \) means that \( \pi \) is satisfied by the trace \( \lambda \).

In all the paper, we consider a plant whose behavior is modeled over an alphabet \( \Sigma \) by a prefix-closed regular language \( L \subseteq \Sigma^* \). The decisions are taken on the traces of a prefix-closed language \( S \subseteq L \).
2. DECISION-MAKING IN DES

2.1 Basic Characteristics of a Decision-Maker

One or several so-called decisional elements are associated to the plant, and decisions are made for each decisional element. In supervisory control, a decisional element is an event, because a supervisor enables or disables each event. In diagnosis and prognosis, a decisional element is a fault, because a diagnoser or prognoser gives verdicts for each fault. For simplicity, we will present the generic part of our study for a single decisional element.

We consider a decision-maker that must take decisions for every trace \( \lambda \in S \subseteq L \) executed by the plant. Let \( \text{Dec}(\lambda) \) be the decision taken after the execution of a trace \( \lambda \in S \). We consider the case of binary decision, that is, the decision-maker takes a decision among two possible decisions, noted 1, 0. The decision-maker may also choose to take no decision, which is noted \( \phi \). Therefore, a decision-maker is defined as a map \( \text{Dec} : S \mapsto \{1, 0, \phi\} \). The requirements targeted by a decision-maker consist of:

A generic decisional requirement (GDR): it has a generic form for all decision-making systems.

Specific decisional requirements (SDR): this may consist of one or several requirements that are specific to each decision-making system.

2.2 Generic decisional requirement (GDR)

To a given decisional element, are associated a pair of languages \( Y \) and \( Z \), such that \( Y \cup Z \subseteq S \) and \( Y \cap Z = \emptyset \). The decision-maker must, as much as possible, take a decision 1 (resp. 0) when the plant has executed a trace \( \lambda \in Y \) (resp. \( \lambda \in Z \)). When the decision-maker has a partial observation of the plant, it may not know the executed trace \( \lambda \). In this case, the minimal requirement, called Generic Decisional Requirement (GDR), is that the decision-maker cannot decide 0 (resp. 1) when the plant has executed a trace \( \lambda \in Y \) (resp. \( \lambda \in Z \)). Formally:

\[
\text{GDR}: \begin{cases} \forall \lambda \in Y, \text{Dec}(\lambda) \neq 0 \\ \forall \lambda \in Z, \text{Dec}(\lambda) \neq 1 \end{cases}
\]

(2.1)

2.3 Specific decisional requirements (SDR)

GDR permits to always decide \( \phi \). But a true decision-maker must decide \( \neq \phi \) in some circumstances in \( Y \cup Z \). For that purpose, we define the notion of Specific Decisional Requirement \( [\alpha \rightsquigarrow \beta] \), which is a formal expression specifying that \( \beta \) is required under a situation specified by \( \alpha \). For the consistency of our study, we consider uniquely \( \alpha \) and \( \beta \) that have the following forms, where \( \lambda \) is a symbol or an expression representing a trace:

- \( \alpha \) is an expression representing a situation in \( Y \cup Z \). It may contain none, one or more expressions of the following forms: \( \text{Dec}(\lambda) \neq 1 \), \( \text{Dec}(\lambda) \neq 0 \), \( \text{Dec}(\lambda) = \phi \).

  We consider uniquely situations in \( Y \cup Z \) just in order to simplify our framework.

- \( \beta \) is an expression that has one of the following forms: \( \text{Dec}(\lambda) = 1 \), \( \text{Dec}(\lambda) = 0 \), \( \text{Dec}(\lambda) \neq \phi \).

  We consider also so-called undecisional requirements which are constraints that do not depend explicitly on decisions, and whose domain of definition is explicit, instead of being deduced (totally or partially) from decisional requirements. Contrary to requirements \([\alpha \rightsquigarrow \beta]\) that consider uniquely situations in \( Y \cup Z \), undecisional requirements may consider any situation in \( S \). Let SDR denote the set of requirements \([\alpha \rightsquigarrow \beta]\) and undecisional requirements targeted by a specific decision-maker.

Let us illustrate GDR and SDR in control, diagnosis, and prognosis, respectively in Sections 2.4, 2.5, and 2.6.

2.4 Application to Supervisory Control

In addition to \( L \), the plant is also modeled by a marked regular language \( L_m \subseteq L \). A supervisor restricts the behavior of the plant, so that it executes only traces of a specification \( K \subseteq L_m \). Note that \( \overline{K} \subseteq L \). In this context, the decisions are taken in \( \overline{K} \), that is, \( S = \overline{K} \). The alphabet \( \Sigma \) of the plant is partitioned into \( \Sigma_c \) and \( \Sigma_{uc} \), containing the controllable and the uncontrollable events, respectively. The events of \( \Sigma_c \) can be disabled by the supervisor, while the events of \( \Sigma_{uc} \) cannot be. A decision is taken for each \( \sigma \in \Sigma \). The decision 1 (resp. 0) means that \( \sigma \) is enabled (resp. disabled), \( Y \) and \( Z \) correspond respectively to the languages \( \mathcal{E}_\sigma \) and \( D_\sigma \) defined as follows:

\[
\mathcal{E}_\sigma = \{ \lambda \in \overline{K} \mid \sigma \in K \} \quad \text{and} \quad D_\sigma = \{ \lambda \in \overline{K} \mid \sigma \notin L \}.
\]

In control, the generic notation \( \text{Dec}(\lambda) \) is replaced by \( \text{Sup}_\sigma(\lambda) \), for \( \sigma \in \Sigma \). From Eq. (2.1), GDR is:

\[
\text{GDR}: \begin{cases} \forall \lambda \in \mathcal{E}_\sigma, \text{Sup}_\sigma(\lambda) \neq 0 \\ \forall \lambda \in D_\sigma, \text{Sup}_\sigma(\lambda) \neq 1 \end{cases}
\]

(2.2)

Consider the following SDR:

\[
\text{SDR}: \begin{cases} \forall \lambda \in \mathcal{E}_\sigma \cup D_\sigma : \sigma \in \Sigma_{uc} \Rightarrow \text{Sup}_\sigma(\lambda) = 1 \\ \forall \lambda \in \mathcal{E}_\sigma \quad \text{Sup}_\sigma(\lambda) = 1 \\ \forall \lambda \in D_\sigma \quad \text{Sup}_\sigma(\lambda) = 0 \\ \forall \lambda \in \overline{K} \quad \lambda \in L_m \Rightarrow \lambda \in K \end{cases}
\]

(2.3)

Since SDR of Eqs. (2.3) guarantees GDR of Eqs. (2.2), we will not consider GDR in control. The first three eqs. in (2.3) are decisional requirements, while the fourth eq. is an undecisional requirement. The first eq. means that uncontrollable events are enabled. We consider only the case of \( \lambda \in \mathcal{E}_\sigma \cup D_\sigma \), because requirements \([\alpha \rightsquigarrow \beta]\) are specified in \( Y \cup Z \), which is \( \mathcal{E}_\sigma \cup D_\sigma \) in control. We do not lose in generality, because when \( \lambda \notin \mathcal{E}_\sigma \cup D_\sigma \), \( \sigma \) is not permitted by the plant. The second and third eqs. in (2.3) mean that any event permitted by the plant is enabled iff it is accepted by \( \overline{K} \). Therefore, the plant under control executes the traces of \( \overline{K} \). The fourth eq. in (2.3) means that every trace of \( L_m \) executed by the plant under control, is a trace of \( K \). To make a link with supervisory control as it is presented in the literature, let us define a few notions:

- Sup is said feasible if \( \forall \sigma \in \Sigma_{uc}, \forall \lambda \in \mathcal{E}_\sigma \cup D_\sigma \), we have \( \text{Sup}_\sigma(\lambda) = 1 \). That is, the supervisor enables any uncontrollable \( \sigma \) which is permitted by the plant. Usually, uncontrollable events are always enabled. But as we have explained above, we do not lose in generality when we ignore traces \( \lambda \notin \mathcal{E}_\sigma \cup D_\sigma \).

- Sup is said admissible if \( \forall \sigma \in \Sigma, \forall \lambda \in \mathcal{E}_\sigma \cup D_\sigma \), we have \( \text{Sup}_\sigma(\lambda) \neq \phi \).

Let \( G \) be an automaton accepting \( L_m \), and \( L(\text{Sup}/G) \) be the prefix-closed language of the plant under the control of
an admissible supervisor Sup. The corresponding marked language is \( L_m(\text{Sup}/G) = L(\text{Sup}/G) \cap L_m \). Sup is said nonblocking if \( L_m(\text{sup}/G) = L(\text{sup}/G) \). In the literature, the objective of supervisory control is expressed as follows: to synthesize a feasible, admissible and nonblocking supervisor that satisfies \( K = L_m(\text{Sup}/G) \). Prop. 2.1 makes the link between this objective and our framework.

**Proposition 2.1.** A supervisor Sup satisfies the SDR of Eqs. (2.3) for every \( \sigma \in \Sigma \), if and only if Sup is feasible, admissible, nonblocking and satisfies \( K = L_m(\text{Sup}/G) \).

### 2.5 Application to Diagnosis

The aim is to detect faulty and/or unfaulty executions of the plant. Consider a fault \( f \), and the decision 1 (resp. 0) which means that the presence (resp. absence) of \( f \) is detected. \( Y \) (resp. \( Z \)) corresponds to the language noted \( F \) (resp. \( H \)) containing the faulty (resp. unfaulty) traces of \( L \). Formally, \( F = \{ \lambda \mu \in L \mid \lambda, \mu \in \Sigma^* \} \) and \( H = L \setminus F \). By definition, \( H \) is prefix-closed and \( F \cup H = L \). For \( t \in \mathbb{N} \), let \( F^t = (F \Sigma^t) \cap L \). In diagnosis, the generic notation \( \text{Diag}(\lambda) \) is replaced by \( \text{Diag}(\lambda) \). From Eq. (2.1):

\[
\text{GDR}: \begin{cases} 
\forall \lambda \in F, \text{Diag}(\lambda) \neq 0 \\
\forall \lambda \in H, \text{Diag}(\lambda) \neq 1 
\end{cases}
\]

Let us consider two examples of SDR, associated to detecting the presence and the absence of fault, respectively.

**Detecting the presence of fault:** the diagnoser must detect a fault in a bounded number of events after its occurrence. This objective is used in Sampath et al. (1995); Kumar and Takai (2009), and is equivalent to detecting the absence of fault in a bounded past of Wang et al. (2005). Formally:

\[
\text{SDR}: \exists t \in \mathbb{N}^+ \text{ s.t. } \forall \lambda \in F^t, \text{Diag}(\lambda) = 1
\]

**Detecting the absence of fault:** Consider now a different objective: the diagnoser must not remain unsure during an arbitrarily long portion of unfaulty execution. This objective is used in Takai and Kumar (2006). Formally:

\[
\text{SDR}: \left\{ \begin{array}{l}
\exists t \in \mathbb{N}^+ \text{ s.t. } \forall \lambda \in H \text{ s.t. } |\mu| = t - 1 : \\
(\forall \nu < \mu, \text{Diag}(\lambda \nu) \neq 0) \Rightarrow (\text{Diag}(\lambda \mu) = 0)
\end{array} \right\}
\]

### 2.6 Application to Prognosis

The aim is to predict faulty and/or unfaulty executions of the plant. Consider a fault \( f \), and the decision 1 (resp. 0) which means that the prediction that \( f \) is certain (resp. uncertain) in a bounded future. \( F \) and \( H \) are defined as in diagnosis. \( Y \) and \( Z \) correspond to the languages noted \( \Sigma \) and \( \Upsilon \) and defined as follows: \( \Sigma \) (resp. \( \Upsilon \)) contains the traces of \( H \) after which \( f \) is certain (resp. uncertain) in a bounded future. Formally, \( \Sigma = \{ \lambda \in H \mid |\lambda| \in \mathbb{N}^+, \{ \lambda \Sigma^t \cap H = \emptyset \} \} \), and \( \Upsilon = H \setminus \Sigma \). By definition, \( \Upsilon \) is prefix-closed and \( \Sigma \cup \Upsilon = H \). We define the language \( \delta \) containing the traces of \( H \) for which \( f \) is possible in the next step. Formally, \( \delta = \{ \lambda \in H \mid \{ \lambda \Sigma \cap F \neq \emptyset \} \} \). Note that \( \delta \subseteq \Sigma \cup \Upsilon = H \). In prognosis, the generic notation \( \text{Dec}(\lambda) \) is replaced by \( \text{Prog}(\lambda) \). From Eq. (2.1):

\[
\text{GDR}: \begin{cases} 
\forall \lambda \in \Sigma, \text{Prog}(\lambda) \neq 0 \\
\forall \lambda \in \Upsilon, \text{Prog}(\lambda) \neq 1 
\end{cases}
\]

We consider the prediction of a fault \( f \) before its occurrence. This objective is used in Genc and Lafonture (2006) and Takai and Kumar (2008). Formally:

\[
\text{SDR}: \forall \lambda \in \partial, \text{Prog}(\lambda) = 1
\]

### 3. WELL FORMED ARCHITECTURES

#### 3.1 Generic decision-making architecture

The architecture of a decision-maker is modeled by three functions: \( O, I \) and \( D \) defined as follows: \( O(\lambda) \) represents the information obtained from observation, after the execution of a trace \( \lambda \). \( I(O(\lambda)) \) represents intermediate decision(s) synthesized from \( O(\lambda) \). \( D(I(O(\lambda))) \) represents a decision \( \text{Dec}(\lambda) \in \{1, 0, \phi\} \) synthesized from \( I(O(\lambda)) \) and which is actually taken. An architecture \( \Psi \) is defined by fixed \( O \) and \( D \) and variable \( I \). That is, decision-makers of the same \( \Psi \) differ by their function \( I \). \( (O, I, D) \) will be illustrated by two examples, in Sects. 4 and 5, respectively.

#### 3.2 Characterization of well-formed architectures

For an architecture \( \Psi \), we denote by \( \text{DM}_\Psi \) any decision-maker of \( \Psi \). Consider the “existence question”: Does there exist a \( \text{DM}_\Psi \) satisfying given GDR and SDR? Our objective in Sect. 3 is to characterize a well-formed architecture, as an architecture for which we can answer that existence question. All SDR requirements \( [\alpha \sim \beta] \) target to obtain decisions \( \neq \phi \) in certain situations in \( Y \cup Z \). For this reason, the well-formed characterization does not depend on a particular SDR, it is rather based on GDR and on the fact that all SDR target to obtain decisions \( \neq \phi \). This brings us to define the following two notions of authority. In the sequel, \( U \subseteq Y \) and \( V \subseteq Z \), and \( \Psi \) is an architecture.

**Definition 3.1.** We say that the authority of \( \Psi \) is contained in \( (U, V) \), if the decision \( \phi \) is taken for all traces in \( (Y \cup Z) \setminus (U \cup V) \) by all \( \text{DM}_\Psi \) that satisfies GDR. Since the considered decision-makers satisfy GDR, we obtain:

- Every \( \text{DM}_\Psi \) satisfying GDR is such that: \( \text{Dec}(\lambda) = 1 \) only if \( \lambda \in U \), and \( \text{Dec}(\lambda) = 0 \) only if \( \lambda \in V \).

**Definition 3.2.** We say that \( \Psi \) has full authority in \( (U, V) \), if there exists a \( \text{DM}_\Psi \) that satisfies GDR and takes decisions \( \neq \phi \) for all traces in \( U \cup V \). Since the considered decision-makers satisfy GDR, we obtain:

- There exists a \( \text{DM}_\Psi \) satisfying GDR such that:
  \( \text{Dec}(\lambda) = 1 \) if \( \lambda \in U \), and \( \text{Dec}(\lambda) = 0 \) if \( \lambda \in V \).

**Definition 3.3.** We say that \( \Psi \) is well-formed w.r.t. \( (U, V) \) if its authority is both contained and full in \( (U, V) \).

**Proposition 3.1.** The contained authority is preserved by increasing languages, i.e., if the authority of \( \Psi \) is contained in \( (U', V') \), then it is contained in every \( (U', V') \) such that \( U \subseteq U' \) and \( V \subseteq V' \). Hence, the contained authority is preserved by union of languages, i.e., if the authority of \( \Psi \) is contained in \( (U_1, V_1) \) and \( (U_2, V_2) \), then it is contained in \( (U_1 \cup U_2, V_1 \cup V_2) \).

**Proposition 3.2.** The contained authority is preserved by intersection of languages, i.e., if the authority of \( \Psi \) is contained in \( (U_1, V_1) \) and \( (U_2, V_2) \), then it is contained in \( (U_1 \cap U_2, V_1 \cap V_2) \).

**Proposition 3.3.** The full authority is preserved by decreasing languages, i.e., if \( \Psi \) has full authority in \( (U, V) \),
then it has full authority for every \((U', V')\) such that 
\(U' \subseteq U\) and \(V' \subseteq V\). Hence, the full authority is preserved by intersection of languages.

**Proposition 3.4.** If \(\Psi\) is well-formed w.r.t. \(\langle U, V \rangle\), then the full authority is preserved by union of languages.

**Proposition 3.5.** There exists at most one pair \(\langle U, V \rangle\) s.t. \(\Psi\) is well-formed w.r.t. \(\langle U, V \rangle\). If such a pair exists, then \(\langle U, V \rangle\) is the smallest pair containing the authority of \(\Psi\), \((U, V)\) is the greatest pair where \(\Psi\) has full authority.

For certain architectures, \(\Psi\), there exists \((U, V)\) such that \(\Psi\) is well-formed w.r.t. \((U, V)\), only by uniquely decision-makers characterized as follows: \(\lambda \models \rho\) (resp. \(\lambda \models \omega\)) is a sufficient (resp. necessary) condition for \(\text{Dec}(\lambda) \neq \phi\), for some properties \(\rho\) and \(\omega\) defined in \(Y \cup Z\) and depending on \(\Psi\). Formally:

\[
\forall \lambda \in Y \cup Z : (\lambda \models \rho) \Rightarrow (\text{Dec}(\lambda) \neq \phi) \Rightarrow (\lambda \models \omega) \quad (3.1)
\]

**Definition 3.4.** Given \(\rho\) and \(\omega\), let \(\text{DM}^\rho_\omega\) denote any decision-maker of an architecture \(\Psi\) satisfying Eq. (3.1).

**Definition 3.5.** We say that \(\Psi\) is well-formed w.r.t. \(\langle U, V, \rho, \omega \rangle\) if it is well-formed w.r.t. \(\langle U, V \rangle\), but by considering uniquely decision-makers \(\text{DM}^\rho_\omega\), that is, if the next two points (a) and (b) are satisfied:

(a) If a \(\text{DM}^\rho_\omega\) satisfies GDR, then it satisfies Eqs. (3.2):

\[
\forall \lambda \in Y \cup Z : \begin{cases} (\text{Dec}(\lambda) = 1) \Rightarrow (\lambda \in U) \\ (\text{Dec}(\lambda) = 0) \Rightarrow (\lambda \in V) \end{cases} \quad (3.2)
\]

(b) There exists \(\text{DM}^\rho_\omega\) satisfying GDR and Eqs. (3.3):

\[
\begin{cases} (\lambda \in U) \Rightarrow (\text{Dec}(\lambda) = 1) \\ (\lambda \in V) \Rightarrow (\text{Dec}(\lambda) = 0) \end{cases} \quad (3.3)
\]

3.3 Existence of solutions in a well-formed architecture

**Definition 3.6.** \((Y, Z)\) is said decidable w.r.t. \((U; V; \text{SDR})\) if: (1) the undecisional requirements of SDR are satisfied; and (2) after the following replacements in every requirement \([\alpha \sim \beta]\) of SDR, the obtained expressions are satisfied.

\[\text{In } \alpha : \begin{cases} \text{every } \text{Dec}(\lambda) \neq 1 \text{ is replaced by } \lambda \notin U, \\ \text{every } \text{Dec}(\lambda) = 0 \text{ is replaced by } \lambda \notin V, \\ \text{every } \text{Dec}(\lambda) = \phi \text{ is replaced by } \lambda \notin (U \cup V) \end{cases} \]

\[\text{In } \beta : \begin{cases} \text{every } \text{Dec}(\lambda) = 1 \text{ is replaced by } \lambda \in U, \\ \text{every } \text{Dec}(\lambda) = 0 \text{ is replaced by } \lambda \in V, \\ \text{every } \text{Dec}(\lambda) = \phi \text{ is replaced by } \lambda \in (U \cup V) \end{cases} \]

**Remark 3.1.** The replacements of Def. 3.6 are consistent only if every \([\alpha \sim \beta]\) of SDR is specified in \(Y \cup Z\).

**Lemma 3.1.** For a \(\text{DM}^\rho_\omega\) satisfying Eqs. (3.2), \((Y, Z)\) is decidable w.r.t. \((U; V; \text{SDR})\) if SDR is satisfied.

** Lemma 3.2.** For a \(\text{DM}^\rho_\omega\) satisfying Eqs. (3.3), SDR is satisfied if \((Y, Z)\) is decidable w.r.t. \((U; V; \text{SDR})\).

From Lemmas 3.1 and 3.2, we obtain:

**Proposition 3.6.** For any architecture \(\Psi\) satisfying point (a) of Def. 3.5 (i.e., its authority is contained in \((U, V)\)), \((Y, Z)\) is decidable w.r.t. \((U; V; \text{SDR})\) if there exists a \(\text{DM}^\rho_\omega\) satisfying GDR and SDR.

**Proposition 3.7.** For any architecture \(\Psi\) satisfying point (b) of Def. 3.5 (i.e., it has full authority in \((U, V)\)), there exists a \(\text{DM}^\rho_\omega\) satisfying GDR and SDR if \((Y, Z)\) is decidable w.r.t. \((U; V; \text{SDR})\).

From Props. 3.6-3.7, we obtain our fundamental generic existence theorem:

**Theorem 3.1.** For any well-formed architecture \(\Psi\) w.r.t. \((U, V; \rho, \omega)\), there exists a \(\text{DM}^\rho_\omega\) satisfying GDR and SDR if and only if \((Y, Z)\) is decidable w.r.t. \((U; V; \text{SDR})\).

The proofs of Propositions 3.6-3.7 and Theorem 3.1 are quite simple. This simplicity is because we assume the knowledge of properties \(\rho, \omega\) and languages \((U, V)\) satisfying points (a) and (b) of Def. 3.5. In the literature, the existence proofs are more complex, because they integrate the proofs of points (a) and (b). By our framework, we have transformed the problem of constructing and proving a theorem of existence, if any, into a problem of constructing a well-form architecture, i.e.: identifying adequate languages \((U, V)\) and properties \((\rho, \omega)\) satisfying points (a) and (b) of Def. 3.5. Once we have constructed a well-formed architecture, we have the following fact:

- The existence condition (i.e., decidability) is synthesized automatically from SDR and \((U, V)\) by using Def. 3.6. Then, the existence theorem is automatically synthesized using Theorem 3.1.

Also, it is worth noting the following relevant fact:

- From Lemma 3.2 and Prop.3.7, any \(\text{DM}^\rho_\omega\) satisfying GDR and Eqs. (3.3) can be used, independently of SDR. The satisfaction of SDR will be guaranteed by the decidability of \((Y, Z)\).

In some cases, no properties \(\rho\) and \(\omega\) are necessary to obtain a well-formed architecture. This is the case of the centralized architecture of Sect. 4. In other cases, a single property \(\pi = \rho = \omega\) is necessary, i.e., \((\text{Dec}(\lambda) \neq \phi) \Leftrightarrow (\lambda \models \pi)\). This is the case of the inference architecture of Sect. 5.

4. CENTRALIZED DECISION-MAKING IN DES

4.1 Centralized Decision-Maker

The alphabet \(\Sigma\) is partitioned into an observable alphabet \(\Sigma_o\) and an unobservable alphabet \(\Sigma_u\). We have \(\Sigma_o \cup \Sigma_u = \Sigma\) and \(\Sigma_o \cap \Sigma_u = \emptyset\). We consider the projection \(P : \Sigma \mapsto \Sigma_o\), that is, for any trace \(\lambda \in \Sigma^*\), \(P(\lambda)\) is obtained from \(\lambda\) by removing all its unobservable events. For \(K \subseteq \Sigma^*\), \(P(K) = \{P(\lambda) \mid \lambda \in K\}\). \(P^{-1}\) denotes the inverse projection, that is, \(P^{-1}(K) = \{\lambda \in \Sigma^* \mid P(\lambda) \in K\}\).

**Definition 4.1.** A centralized (or Cent-)decision-maker is defined as a map \(I : P(S) \mapsto \{1, 0, \phi\}\). After the execution of a trace \(\lambda \in S\), the decision \(\text{Dec}(\lambda) = I(\text{P}(\lambda))\) is taken.

Here is an example of Cent-decision-maker:

\[
\forall \lambda \in S, \text{Dec}(\lambda) = \begin{cases} 1, & \text{if } P(\lambda) \in P(Y) \setminus P(Z) \\ 0, & \text{if } P(\lambda) \in P(Z) \setminus P(Y) \\ \phi, & \text{otherwise} \end{cases} \quad (4.1)
\]

With regard to the generic architecture \((O, I, D)\) of Sect. 3.1, in Cent-decision-making (w.r.t. \(P\)):

- \(O(\lambda) = P(\lambda)\).
- \(I(P(\lambda)) = \text{Dec}(\lambda)\). Eqs. (4.1) is an example of \(I\).
- \(D\) is the identity, i.e., \(D(I(P(\lambda))) = I(P(\lambda))\).

4.2 Well-Formed Architecture and Existence Theorem

Consider the languages \(U = Y \setminus P^{-1}(P(Z)), V = Z \setminus P^{-1}(P(Y))\). Let us show that the centralized (or Cent-)architecture is well-formed w.r.t. \((U, V)\), while no properties \(\rho\) and \(\omega\) are
necessary. In this case, $DM_{P}$ and $DM^{\omega}_{P}$ are equivalent and correspond to a Cent-decision-maker (Def. 4.1).

**Lemma 4.1.** For every Cent-decision-maker satisfying GDR, we have:

$$\forall \lambda \in \mathcal{Y} \cup \mathcal{Z} \left\{ \begin{array}{l} (\text{Dec}(\lambda) = 1) \Rightarrow (\lambda \in \mathcal{Y} \setminus \mathcal{P}^{-1}(\mathcal{Z})) \\ (\text{Dec}(\lambda) = 0) \Rightarrow (\lambda \in \mathcal{Z} \setminus \mathcal{P}^{-1}(\mathcal{Y})) \end{array} \right\}$$

(4.2)

**Lemma 4.2.** The Cent-decision-maker defined by Eq. (4.1) satisfies GDR.

**Lemma 4.3.** For the Cent-decision-maker defined by Eq. (4.1), we have:

$$\forall \lambda \in \mathcal{Y} \cup \mathcal{Z} \left\{ \begin{array}{l} (\lambda \in \mathcal{Y} \setminus \mathcal{P}^{-1}(\mathcal{Z})) \Rightarrow (\text{Dec}(\lambda) = 1) \\ (\lambda \in \mathcal{Z} \setminus \mathcal{P}^{-1}(\mathcal{Y})) \Rightarrow (\text{Dec}(\lambda) = 0) \end{array} \right\}$$

(4.3)

From Lemma 4.1, point (a) of Def. 3.5 is satisfied. From Lemmas 4.2-4.3, point (b) of Def. 3.5 is satisfied, a corresponding decision-maker is defined by Eqs. (4.1). Hence:

**Proposition 4.1.** The Cent-architecture is well-formed w.r.t. $\mathcal{Y} \setminus \mathcal{P}^{-1}(\mathcal{Z}), \mathcal{Z} \setminus \mathcal{P}^{-1}(\mathcal{Y})$, for any $(\mathcal{S}, \mathcal{Y}, \mathcal{Z})$ and SDR.

In Cent-decision-making, the generic decidability w.r.t. $\langle \mathcal{U}; \mathcal{V}; \mathcal{SDR} \rangle$ (Def. 3.6) will be called **Cent-DECID** (implicitly w.r.t. P and SDR). That is:

**Definition 4.2.** $(\mathcal{Y}, \mathcal{Z})$ is said **Cent-DECID** (for centrally decidable) if: (1) the undecisional requirements of SDR are satisfied; and (2) after the replacements of Def. 3.6 in every requirement $[a \rightarrow b]$ of SDR, using $\mathcal{U} = \mathcal{Y} \setminus \mathcal{P}^{-1}(\mathcal{Z})$ and $\mathcal{V} = \mathcal{Z} \setminus \mathcal{P}^{-1}(\mathcal{Y})$, the obtained expressions are satisfied.

From Theorem 3.1 and Prop. 4.1, we deduce:

**Theorem 4.1.** There exists a Cent-decision-maker satisfying GDR and SDR if and only if $(\mathcal{Y}, \mathcal{Z})$ is Cent-DECID.

From Prop. 4.1 and since control, diagnosis and prognosis are characterized by $(\mathcal{S}, \mathcal{Y}, \mathcal{Z})$ and SDR, we deduce:

**Proposition 4.2.** The centralized control, diagnosis and prognosis are well-formed w.r.t. their respective $(\mathcal{Y} \setminus \mathcal{P}^{-1}(\mathcal{Z}), \mathcal{Z} \setminus \mathcal{P}^{-1}(\mathcal{Y}))$.

### 4.3 Application to Supervisory Control

We apply the Cent-decision-making with $\mathcal{S} = \mathcal{K} \subseteq \mathcal{L}$, $\mathcal{Y} = \mathcal{E}_{\sigma}$, $\mathcal{Z} = \mathcal{D}_{\sigma}$, and SDR of Eqs. (2.3). For this SDR, the Cent-DECID of Def. 4.2 is:

$$\forall \lambda \in \mathcal{E}_{\sigma} \cup \mathcal{D}_{\sigma} : \sigma \in \Sigma_{uc} \Rightarrow \lambda \in \mathcal{E}_{\sigma} \setminus \mathcal{P}^{-1}(\mathcal{D}_{\sigma})$$

$$\forall \lambda \in \mathcal{E}_{\sigma} : \lambda \in \mathcal{E}_{\sigma} \setminus \mathcal{P}^{-1}(\mathcal{D}_{\sigma})$$

$$\forall \lambda \in \mathcal{D}_{\sigma} : \lambda \in \mathcal{D}_{\sigma} \setminus \mathcal{P}^{-1}(\mathcal{E}_{\sigma})$$

$$\forall \lambda \in \mathcal{K} : \lambda \in \mathcal{K}$$

(4.2)

**Proposition 4.3.** The above set of equations is equiv. to:

$$\begin{cases} \sigma \in \Sigma_{uc} \Rightarrow \mathcal{D}_{\sigma} = \emptyset \\ \mathcal{P}(\mathcal{E}_{\sigma}) \cap \mathcal{P}(\mathcal{D}_{\sigma}) = \emptyset \\ \mathcal{K} \cap \mathcal{L}_{m} \subseteq \mathcal{K} \end{cases}$$

(4.4)

From Theorem 4.1 and Proposition 4.3:

**Proposition 4.4.** There exists a centralized supervisor satisfying Eqs. (2.3) if and only if Eqs. (4.4) are satisfied.

Let us recall some notions followed by a proposition:

- $\mathcal{K}$ is said controllable if: $\mathcal{K} \cap \mathcal{L}_{uc} \subseteq \mathcal{K}$.
- $\mathcal{K}$ is said P-observable if: $\forall \lambda, \mu \in \mathcal{K}, \sigma \in \Sigma$ s.t. $\lambda, \mu \in \mathcal{L}$: $(P(\lambda) = P(\mu)) \Rightarrow (\lambda \sigma = \mathcal{K} \Leftrightarrow \mu \sigma \in \mathcal{K})$.
- $\mathcal{K}$ is said $L_m$-closed if: $\mathcal{K} \cap \mathcal{L}_{m} = \mathcal{K}$.

**Proposition 4.5.** Satisfying the three equations in (4.4) for every $\sigma \in \Sigma$, is equivalent to: $\mathcal{K}$ is controllable, P-observable and $L_m$-closed, respectively.

From Props. 2.1, 4.4 and 4.5, we obtain the classical existence theorem in centralized control under partial observation: There exists a supervisor which is feasible, admissible, nonblocking and satisfies $\mathcal{K} = \mathcal{L}_{m}(\text{Sup}/\mathcal{G})$, if and only if $\mathcal{K}$ is controllable, P-observable and $L_m$-closed. If this existence condition is satisfied, we can use the following centralized supervisor obtained from Eqs. (4.1):

$$\begin{cases} \mathcal{N} \in \mathcal{N}^{+} \text{ s.t.} \mathcal{P}(\mathcal{N}) = 0 \end{cases}$$

(4.5)

From Theorem 4.1, we obtain Prop. 4.6, which corresponds to the existence theorem of Sampath et al. (1995):

**Proposition 4.6.** There exists a centralized diagnoser satisfying Eqs. (2.4)-(2.5) if and only if Eq. (4.5) is satisfied.

As it is shown in Wang et al. (2005), F-DIAG is equivalent to NF-DIAG-3 whose objective is to detect the absence of fault in a bounded past. Intuitively, detecting the presence of fault in a bounded future is equivalent to detecting the absence of fault in a bounded past.

**Detecting the absence of fault:** We use the detection defined in Takai and Kumar (2006), whose corresponding SDR was formulated in Eq. (2.5). Recall that the objective is to detect a fault in a bounded number of events after its occurrence. Let us call F-DIAG the Cent-DECID of Def. 4.2. From Eqs. (2.5) and Def. 4.2, $(\mathcal{F}, \mathcal{H})$ is F-DIAG if: $\exists \mu \in \mathcal{N}^{+}$ s.t. $\forall \lambda \in \mathcal{F}_{\mu}, \lambda \in \mathcal{F}_{\mu} \setminus \mathcal{P}^{-1}(\mathcal{H})$. This is equivalent to Eq. (4.5), which is equivalent to the diagnosability of Sampath et al. (1995) and F-DIAG of Wang et al. (2005):

$$\exists \mu \in \mathcal{N}^{+} \text{ s.t.} \mathcal{P}(\mathcal{F}_{\mu}) \cap \mathcal{P}(\mathcal{H}) = \emptyset$$

(4.5)

From Theorem 4.1, we obtain Prop. 4.6, which corresponds to the existence theorem of Sampath et al. (1995):

**Proposition 4.7.** There exists a centralized diagnoser satisfying Eqs. (2.4)-(2.5) if and only if Prop. 4.6 is satisfied.

Here is a centralized diagnoser for diagnosing the presence (if $(\mathcal{F}, \mathcal{H})$ is F-DIAG) and the absence (if $(\mathcal{F}, \mathcal{H})$ is NF-DIAG) of fault; it is obtained from Eqs. (4.1):
4.5 Application to Prognosis

We apply the Cent-decision-making with \( S = \mathcal{H}, \mathcal{Y} = \mathcal{I}, Z = \mathcal{Y} \). The corresponding GDR is in Eqs. (2.7). We use the prediction defined in Genc and Lafortune (2006), whose corresponding SDR was formulated in Eq. (2.8). Recall that the objective is to predict a fault before its occurrence. Let us call F-PROG the Cent-DECID of Def. 4.2. From Eqs. (2.8) and Def. 4.2, (3, \( \mathcal{Y} \)) is F-PROG if: \( \forall i \in \mathcal{I}, \lambda \in \mathcal{I} \setminus \mathcal{P}^{-1}P(\mathcal{Y}) \) This is equivalent to \( (\partial \subseteq \mathcal{I}) \land (P(\partial) \cap P(\mathcal{Y}) = \emptyset) \). Since \( \partial \subseteq \mathcal{I} \setminus \mathcal{Y} \), F-PROG is equivalent to Eq. (4.7), which corresponds to the predictability of Genc and Lafortune (2006):

\[
P(\partial) \cap P(\mathcal{Y}) = \emptyset \tag{4.7}
\]

From Theorem 4.1, we obtain Prop. 4.8, which corresponds to the existence theorem of Genc and Lafortune (2006):

**Proposition 4.8.** There exists a centralized prognoser satisfying Eqs. (2.7)-(2.8) if and only if Eq. (4.7) is satisfied.

If (3, \( \mathcal{Y} \)) is F-PROG, here is a centralized prognoser for predicting a fault; it is obtained from Eqs. (4.1):

\[
\forall \lambda \in \mathcal{H}, \text{Prog}(\lambda) = \begin{cases} 
1, & \text{if } P(\lambda) \notin P(\mathcal{Y}) \\
0, & \text{if } P(\lambda) \notin P(\mathcal{I}) \\
\phi, & \text{otherwise.}
\end{cases}
\]

5. INFERENCE DECISION-MAKING IN DES

5.1 Decentralized Decision-Making

In decentralized decision-making, several (say \( n \)) local decision-makers (DM\(_i\))\(_i=1,...,n\) cooperate to take a decision. Let \( I = \{1, \ldots, n\} \) be the indexing set of all decision-makers. Each DM\(_i\) has its own set of observable events \( \Sigma_{o,i} \subseteq \Sigma \); that is, it observes the plant through the projection \( P_i : \Sigma \mapsto \Sigma_{o,i} \), which hides every event \( \sigma \notin \Sigma_{o,i} \). Based on its local observation, each DM\(_i\) takes a local decision. The local decisions of all local decision-makers are then combined in a certain way by a coordinator module C to synthesize a so-called effective decision. A decentralized decision-making is thus defined by \( ((\text{DM}_i))_{i \in I}, C \).

5.2 Inference Decision-Maker

Let us propose an Inf-architecture, which is a generic form of the inference-based frameworks developed in control, diagnosis and prognosis, in Kumar and Takai (2007, 2009); Takai and Kumar (2008). We have selected the inference framework, because it generalizes several other decentralized architectures. After the execution of a trace \( \lambda \in \mathcal{S} \), each local decision-maker DM\(_i\) has observed \( P_i(\lambda) \); it generates a local decision \( c_i(P_i(\lambda)) \in \{1, 0, \phi\} \) with an ambiguity level \( n_i(P_i(\lambda)) \). Formally, each DM\(_i\) issues:

\[
\text{Dec}_i(P_i(\lambda)) = (c_i(P_i(\lambda)), n_i(P_i(\lambda))) \tag{5.1}
\]

Let \( n(\lambda) \) be the smallest ambiguity associated to \( \lambda \in \mathcal{S} \):

\[
n(\lambda) = \min_{i \in I}(n_i(P_i(\lambda))) \tag{5.2}
\]

The effective decision is synthesized by selecting the local decision that has the smallest ambiguity level. The decision \( \phi \) is issued when distinct local decisions have the same smallest ambiguity level. Formally:

\[
\forall \lambda \in \mathcal{S}, \text{Dec}(\lambda) = \begin{cases} 
1, & \text{if } \forall i \in I \text{ s.t. } n_i(P_i(\lambda)) = n(\lambda): \\
0, & \text{if } \forall i \in I \text{ s.t. } n_i(P_i(\lambda)) = n(\lambda): \\
\phi, & \text{otherwise.}
\end{cases}
\]

**Definition 5.1.** An Inf-decision-maker is a decentralized decision-maker whose decision \( \text{Dec}(\lambda) \) is computed from \( (c_i(P_i(\lambda)), n_i(P_i(\lambda)))_{i \in I} \) by Eqs. (5.1)-(5.3)

Eqs. (5.1)-(5.3) specify how the effective decision \( \text{Dec}(\lambda) \) of any Inf-decision-maker is computed from \( (c_i(P_i(\lambda)), n_i(P_i(\lambda)))_{i \in I} \). Let us show how to compute \( (c_i(P_i(\lambda)), n_i(P_i(\lambda)))_{i \in I} \) from \( (P_i(\lambda))_{i \in I} \), by using a generic version of methods in Kumar and Takai (2007, 2009); Takai and Kumar (2008). We consider the series of pairs of languages \( (\mathcal{Y}[k], \mathcal{Z}[k])_{k \in \mathcal{N}} \) defined inductively as follows:

- Basis: \( \mathcal{Y}[0] = \mathcal{Y}, \mathcal{Z}[0] = \mathcal{Z} \).
- Inductive step: for \( k \geq 0 \),

\[
\mathcal{Y}[k+1] = \mathcal{Y}[k] \cap \bigcap_{i \in I} P_i^{-1}P_i(\mathcal{Z}[k])
\]

\[
\mathcal{Z}[k+1] = \mathcal{Z}[k] \cap \bigcap_{i \in I} P_i^{-1}P_i(\mathcal{Y}[k])
\]

Consider \( \mathcal{N} \in \mathcal{N} \) and define \( (n_i^1(P_i(\lambda)), n_i^0(P_i(\lambda)))_{i \in I} \) by:

\[
\begin{cases}
(n_i^1(P_i(\lambda))) = \min_{k \in \mathcal{N}}(\{P_i(\lambda) \notin P_i(\mathcal{Z}[k])\}) \\
(n_i^0(P_i(\lambda))) = \min_{k \in \mathcal{N}}(\{P_i(\lambda) \notin P_i(\mathcal{Y}[k])\}) \tag{5.4}
\end{cases}
\]

After the execution of \( \lambda \in \mathcal{S} \), every local decision-maker DM\(_i\) has observed \( P_i(\lambda) \) and computes \( (n_i^1(P_i(\lambda)), n_i^0(P_i(\lambda))) \) using Eqs. (5.4). The local decision \( c_i(P_i(\lambda)) \) and the ambiguity \( n_i(P_i(\lambda)) \) associated to it, are computed by:

\[
c_i(P_i(\lambda)) = \begin{cases} 
1, & \text{if } n_i^1(P_i(\lambda)) < n_i^0(P_i(\lambda)) \\
0, & \text{if } n_i^1(P_i(\lambda)) > n_i^0(P_i(\lambda)) \\
\phi, & \text{if } n_i^1(P_i(\lambda)) = n_i^0(P_i(\lambda))
\end{cases}
\]

\[
n_i(P_i(\lambda)) = \min(n_i^1(P_i(\lambda)), n_i^0(P_i(\lambda))) \tag{5.5}
\]

With regard to the generic architecture \( (\mathcal{O}, \mathcal{I}, \mathcal{D}) \) introduced in Sect. 3.1, in Inf-decision-making (w.r.t. \( (P_i)_{i \in I} \)):

- \( \mathcal{O}(\lambda) = (P_i(\lambda))_{i \in I} \)
- \( \mathcal{I}(\mathcal{O}(\lambda)) = (c_i(P_i(\lambda)), n_i(P_i(\lambda)))_{i \in I} \). Eqs. (5.4)-(5.6) define an example of function \( \mathcal{I} \)
- \( \mathcal{D}((c_i(P_i(\lambda)), n_i(P_i(\lambda)))_{i \in I}) = \text{Dec}(\lambda) \). Eqs. (5.1)-(5.3) define the function \( \mathcal{D} \).

**Remark 5.1.** An Inf-decision-maker is by definition based on Eqs. (5.1)-(5.3). Therefore, when we refer to an Inf-decision-maker defined by Eqs. (5.4)-(5.6), we mean a decentralized decision-maker defined by Eqs. (5.1)-(5.3).

**Definition 5.2.** For \( N \in \mathcal{N} \), an Inf-decision-maker is said \( N \)-inferring w.r.t. \( (\mathcal{Y}, \mathcal{Z}) \) if:

\[
\forall \lambda \in \mathcal{Y} \cup \mathcal{Z} : \text{Dec}(\lambda) \neq \phi \Leftrightarrow n(\lambda) \leq N \tag{5.7}
\]
5.3 Well-Formed Architecture and Existence Theorem

Consider the languages $\mathcal{U} = \mathcal{Y} \setminus \mathcal{Y}[N+1]$, $\mathcal{V} = \mathcal{Z} \setminus \mathcal{Z}[N+1]$, and the property $\pi = \rho = \omega$ defined by: $n(\lambda) \leq N$. In this case, Eq. (3.1) corresponds to the definition of N-inferring. Let us show that the Inf-architecture defined by Eqs. (5.1)-(5.3) is well-formed w.r.t. $\{\mathcal{U}, \mathcal{V}; \text{N-inferring}\}$. In this case, $\text{DM}_\Psi$ is an Inf-decision-maker, and $\text{DM}_\Psi^\omega$ is an N-inferring Inf-decision-maker.

**Lemma 5.1.** For every N-inferring Inf-decision-maker satisfying GDR, we have:

$$\forall \lambda \in \mathcal{Y} \cup \mathcal{Z} \left\{ \begin{array}{l}
(\text{Dec}(\lambda) = 1) \Rightarrow (\lambda \in \mathcal{Y} \setminus \mathcal{Y}[N+1]) \\
(\text{Dec}(\lambda) = 0) \Rightarrow (\lambda \in \mathcal{Z} \setminus \mathcal{Z}[N+1])
\end{array} \right\}$$

(5.8)

**Lemma 5.2.** For the specific Inf-decision-maker defined by Eqs. (5.4)-(5.6), we have:

$$\left\{ \begin{array}{l}
(\lambda \in \mathcal{Y}) \Rightarrow (\lambda \in \mathcal{Y}[n(\lambda)]) \\
(\lambda \in \mathcal{Z}) \Rightarrow (\lambda \in \mathcal{Z}[n(\lambda)])
\end{array} \right\}$$

(5.9)

**Lemma 5.3.** The specific Inf-decision-maker defined by Eqs. (5.4)-(5.6) is N-inferring.

**Lemma 5.4.** The specific Inf-decision-maker defined by Eqs. (5.4)-(5.6) satisfies GDR.

**Lemma 5.5.** For the specific Inf-decision-maker defined by Eqs. (5.4)-(5.6), we have:

$$\left\{ \begin{array}{l}
(\lambda \in \mathcal{Y} \setminus \mathcal{Y}[N+1]) \Rightarrow (n(\lambda) \leq N) \\
(\lambda \in \mathcal{Z} \setminus \mathcal{Z}[N+1]) \Rightarrow (n(\lambda) \leq N)
\end{array} \right\}$$

(5.10)

**Lemma 5.6.** For the specific Inf-decision-maker defined by Eqs. (5.4)-(5.6), we have:

$$\forall \lambda \in \mathcal{Y} \cup \mathcal{Z} \left\{ \begin{array}{l}
(\lambda \in \mathcal{Y} \setminus \mathcal{Y}[N+1]) \Rightarrow (\text{Dec}(\lambda) = 1) \\
(\lambda \in \mathcal{Z} \setminus \mathcal{Z}[N+1]) \Rightarrow (\text{Dec}(\lambda) = 0)
\end{array} \right\}$$

(5.11)

From **Lemma 5.1**, point (a) of Def. 3.5 is satisfied. From **Lemmas 5.4 and 5.6**, point (b) of Def. 3.5 is satisfied, and a corresponding specific decision-maker is the one defined by Eqs. (5.1)-(5.6). Therefore:

**Proposition 5.1.** The Inf-architecture is well-formed w.r.t. $\langle \mathcal{Y} \setminus \mathcal{Y}[N+1], \mathcal{Z} \setminus \mathcal{Z}[N+1]; \text{N-inferring} \rangle$, for any $(S, \mathcal{Y}, \mathcal{Z})$ and SDR.

In Inf-decision-making, the generic decidability w.r.t. $\langle \mathcal{U}; \mathcal{V}; \text{SDR} \rangle$ (Def. 3.6) will be called Inf$_N$-DECID (implicitly w.r.t. $\{P_i\}_{i \in I}$ and SDR) . That is:

**Definition 5.3.** $(\mathcal{Y}, \mathcal{Z})$ is said Inf$_N$-DECID if: (1) the undecisional requirements of SDR are satisfied; and (2) after the replacements of Def. 3.6 in every requirement $\alpha \sim 3$ of SDR, using $\mathcal{U} = \mathcal{Y} \setminus \mathcal{Y}[N+1]$ and $\mathcal{V} = \mathcal{Z} \setminus \mathcal{Z}[N+1]$, the obtained expressions are satisfied.

From **Lemma 5.6** and the generic **Lemma 3.2**, we deduce:

**Lemma 5.7.** If $(\mathcal{Y}, \mathcal{Z})$ is Inf$_N$-DECID, then SDR is satisfied by the Inf-decision-maker defined by Eqs. (5.4)-(5.6). From **Theorem 3.1** and **Proposition 5.1**, we deduce:

**Theorem 5.1.** There exists an N-inferring Inf-decision-maker satisfying GDR and SDR if and only if $(\mathcal{Y}, \mathcal{Z})$ is Inf$_N$-DECID.

From Prop. 5.1, and since control, diagnosis and prognosis are characterized by $(S, \mathcal{Y}, \mathcal{Z})$ and SDR, we deduce:

**Proposition 5.2.** The influence control, diagnosis and prognosis are well-formed w.r.t. their respective $(\mathcal{Y} \setminus \mathcal{Y}[N+1], \mathcal{Z} \setminus \mathcal{Z}[N+1]; \text{N-inferring})$.

Lemmas 5.2-5.4, 5.7 and **Theorem 5.1** correspond respectively to Lemmas 1-4 and Theorem 1 of Kumar and Takai (2009). We have added Lemmas 5.1, 5.5 and 5.6, which present interesting properties of the inference framework.

5.4 Application to Supervisory Control

We obtain an Inf-control by applying the Inf-decision-making with $S = K \subseteq L$, $\mathcal{Y} = \mathcal{E}_\sigma$, $\mathcal{Z} = \mathcal{D}_\sigma$, and SDR of Eqs. (2.3). For this SDR, the Inf$_N$-DECID of Def. 5.3 is:

$$\left\{ \begin{array}{l}
\forall \lambda \in \mathcal{E}_\sigma \cup \mathcal{D}_\sigma : \sigma \in \Sigma \Rightarrow \lambda \in \mathcal{E}_\sigma \cup \mathcal{D}_\sigma [N+1] \\
\forall \lambda \in \mathcal{E}_\sigma : \lambda \in \mathcal{E}_\sigma [N+1] \\
\forall \lambda \in \mathcal{D}_\sigma : \lambda \in \mathcal{D}_\sigma [N+1] \\
\forall \lambda \in K : \lambda \in L_m \Rightarrow \lambda \in K
\end{array} \right\}$$

(5.12)

From **Theorem 5.1** and **Proposition 5.3**:

**Proposition 5.4.** There exists a N-inferring inference supervisor satisfying Eqs. (2.3) if and only if Eqs. (5.12) are satisfied.

Let “$K$ is Inf$_N$-OBS” mean: $\forall \sigma \in \Sigma, \mathcal{E}_\sigma [N+1] = \mathcal{D}_\sigma [N+1] = \emptyset$. Prop. 5.5 makes a link with Kumar and Takai (2007):

**Proposition 5.5.** Satisfying the three equations in (5.12) for every $\sigma \in \Sigma$, is equivalent to: $K$ is controllable, Inf$_N$-OBS and $L_m$-closed, respectively.

From Props. 2.1, 5.4 and 5.5, we obtain the existence theorem in Inf-control: There exists an N-inferring Inf-supervisor which is feasible, admissible, nonblocking and satisfies $\mathcal{K} = \mathcal{L}_m(\text{Sup} / G)$, if and only if $K$ is controllable, Inf$_N$-OBS and $L_m$-closed. We do not obtain exactly the Inf-control of Kumar and Takai (2007). Indeed:

1. Equations (5.4) are used in Kumar and Takai (2007) without the portion “$\forall [k = N+1]$”.
2. The N-inferring condition of Kumar and Takai (2007) is weaker than the definition obtained from Eq. (5.7). Indeed, the N-inferring condition of Kumar and Takai (2007) means that $n(\lambda) \leq N$ must be satisfied in one of the following two languages:

$$\{ \lambda \in \mathcal{E}_\sigma \cup \mathcal{D}_\sigma : \text{Dec}(\lambda) = 1 \} \cup \{ \lambda \in \mathcal{E}_\sigma \cup \mathcal{D}_\sigma : \text{Dec}(\lambda) = 0 \}$$

With our framework, from Eq. (5.7), $n(\lambda) \leq N$ must be satisfied in both languages.
3. Consequently to point (2), the N-inferring observability of Kumar and Takai (2007) is satisfied when $\mathcal{E}_\sigma [N+1] \cup \mathcal{D}_\sigma [N+1]$ is empty, while our Inf$_N$-OBS is satisfied when $\mathcal{E}_\sigma [N+1]$ and $\mathcal{D}_\sigma [N+1]$ are empty.

This difference is not really important, because if $\mathcal{E}_\sigma [k]$ or $\mathcal{D}_\sigma [k]$ is empty, then $\mathcal{E}_\sigma [k+1]$ and $\mathcal{D}_\sigma [k+1]$ are empty.

5.5 Application to Diagnosis

We apply Inf-decision-making with $S = L$, $\mathcal{Y} = \mathcal{F}$, $\mathcal{Z} = \mathcal{H}$. The corresponding GDR is in Eqs. (2.4). Let us consider the two types of detection of Sect. 2.5.
Detecting the presence of fault: We obtain the Inf-diagnosis of Kumar and Takai (2009) by using SDR of Eq. (2.5). The $\text{Inf}_N$-DECID of Def. 4.2 becomes: \(\exists \ell \in \mathbb{N}^+\) s.t. \(\forall \lambda \in \mathcal{F}^\ell, \lambda \in \mathcal{F} \setminus \mathcal{F}[N+1]\). This is equivalent to Eq. (5.13), which corresponds to the \(N\)-inference $F$-diagnosability of Kumar and Takai (2009):

\[
\exists \ell \in \mathbb{N}^+\text{ s.t. } \mathcal{F}[N+1] \cap \mathcal{F}^\ell \neq \emptyset \tag{5.13}
\]

From Theorem 5.1, we obtain Prop. 5.6, which corresponds to the existence theorem of Kumar and Takai (2009):

**Proposition 5.6.** There exists an $N$-inferring Inf-diagnoser satisfying Eqs. (2.4)-(2.5) if and only if Eq. (5.13) is satisfied.

If $N = 0$, Eq. (5.13) is equivalent to \(\mathcal{F}^\ell \cap \bigcap_{i \in \ell} P_i^{-1} P_i(\mathcal{H})\), which corresponds to F-CODIAG of Wang et al. (2005).

Detecting the absence of fault: We obtain the Inf-diagnosis of Takai and Kumar (2006) by using SDR of Eq. (2.6). The $\text{Inf}_N$-DECID of Def. 4.2 becomes: \(\exists \ell \in \mathbb{N}^+\) s.t. \(\forall \mu \in \mathcal{H} \text{ s.t. } |\mu| = \ell-1, (\forall \nu < \mu, \lambda \nu \notin \mathcal{H} \setminus \mathcal{H}[N+1]) \Rightarrow (\lambda \mu \in \mathcal{H} \setminus \mathcal{H}[N+1])\). This is equivalent to Eq. (5.14), which corresponds to $N$-inference $NF$-diagnosability of Takai and Kumar (2006).

\[
\begin{align*}
\exists \ell \in \mathbb{N}^+\text{ s.t. } &\forall \lambda \mu \in \mathcal{H} \text{ s.t. } |\mu| = \ell-1, \\
&\exists \nu \leq \mu \text{ s.t. } \lambda \nu \notin \mathcal{H}[N+1] \tag{5.14}
\end{align*}
\]

From Theorem 5.1, we obtain Prop. 5.7, which corresponds to the existence theorem of Takai and Kumar (2006):

**Proposition 5.7.** There exists an $N$-inferring Inf-diagnoser satisfying Eqs. (2.4)&(2.6) if and only if Eq. (5.14) is satisfied.

5.6 Application to Prognosis

We apply the Inf-decision-making with $S = \mathcal{H}, \mathcal{Y} = \mathfrak{S}, \mathcal{Z} = \mathfrak{Y}$. The corresponding GDR is in Eqs. (2.7). We obtain the Inf-prognosis of Takai and Kumar (2008) by using SDR of Eq. (2.8). Recall that the objective is to predict a fault before its occurrence. The $\text{Inf}_N$-DECID of Def. 4.2 becomes: \(\forall \lambda \in \mathcal{E}, \lambda \in \mathfrak{S} \setminus \mathfrak{S}[N+1]\). This is equivalent to Eq. (5.15), which corresponds to $N$-inference $NF$-prognosability of Takai and Kumar (2008).

\[
\partial \subseteq \mathfrak{S} \setminus \mathfrak{S}[N+1] \tag{5.15}
\]

From Theorem 5.1, we obtain Prop. 5.8, which corresponds to the existence theorem of Takai and Kumar (2008):

**Proposition 5.8.** There exists an $N$-inferring Inf-prognoser satisfying Eqs. (2.7)-(2.8) if and only if Eq. (5.15) is satisfied.

6. CONCLUSION

6.1 Contributions, Comparison with some frameworks

We have proposed a decision-making framework for DES, which is doubly unifying: unification of architectures, and unification of control, diagnosis and prognosis. The contributions are indicated in the introduction. Our framework may contribute to better understand the decision-making mechanism, and promote the discovery of new examples of decision-making and architectures.

The diagnosis frameworks of Sampath et al. (1995); Wang et al. (2005) may seem incompatible with our framework, because they permit to issue the decision 0 for faulty traces. For example, with F-DIAG and NF-DIAG-3 of Wang et al. (2005), the decision 0 can be issued for faulty traces which were unfaulty in a bounded past. In fact, these frameworks generate the decision 0 in the situations where our framework generates 0 or $\phi$. Our use of $\phi$ may be relevant, because it permits to distinguish the situations where the diagnoser is certain that no fault has occurred until now (0), from the situations where the diagnoser is certain that no fault has occurred until a bounded past ($\phi$).

Total compatibility of our framework with F-DIAG and NF-DIAG-3 is obtained if we replace every decision $\phi$ by 0. The same type of comparison can be made in prognosis with the framework of Genc and Lafortune (2006).

6.2 Future work

- Study the case of more general SDR.
- Study the case where the decision-maker selects a decision among more than two decisions.
- Apply our framework to architectures with dynamic observability, as in Cassez and Tripakis (2008).
- Adapt our framework to real-time decision-making.

REFERENCES


