Do what you know: coupling knowledge with action in discrete-event systems

K. Ritsuka · K. Rudie

Received: 30 May 2022 / Accepted: 17 July 2023 / Published online: 21 August 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
An epistemic model for decentralized discrete-event systems with non-binary control is presented. This framework combines existing work on inference-based control decisions with existing work on formal reasoning about knowledge in discrete-event systems. The novelty in the epistemic formalism is in providing an approach to derive problem solvability conditions and problem solutions. The derived expressions directly encapsulate the actions that supervisors must take.

Keywords Discrete-event systems · Supervisory control · Epistemic logic · Formal reasoning about knowledge

1 Introduction
The emergence of networked systems, including smart vehicles, home automation, and wearables has increased the need for decentralized supervisory control: the concept that the control is performed by not a monolithic, but many individual entities — or agents — separated by the environment. This paper focuses on systems modelled as discrete-event systems (DES).

With control actions performed jointly, a mechanism — called a fusion rule — is needed to combine control decisions of the agents. Decentralized control of discrete-event systems under partial observations began with allowing only Boolean control decisions, and synthesis of the control policy has been studied when the fusion rule is conjunctive (Cieslak et al. 1988, Rudie and Wonham 1992), and later other fusion rules are considered (Prosser, Kam, and Kwatny 1997, Yoo and Lafortune 2002), during which time the fusion rules can be interpreted as simply an arbiter to resolve conflicting control decisions. Further work by Yoo and Lafortune (2004) extended the approach and proposed a conditional architecture to allow non-binary control decisions with a more sophisticated fusion rule, so that supervisors...
can “conditionally” turn on/off events based on the actions of other supervisors. Yoo and LaFortune gave necessary and sufficient conditions for the existence of supervisors (Yoo and LaFortune 2004) and a realization of the supervisors (Yoo and LaFortune 2005) in the conditional architecture.

In existing DES research, the conditions for solvability and supervisor synthesis (when those conditions are satisfied) typically each rely on constructions that are divorced from each other, and a similar remark applies also to their respective proofs of correctness. The constructions appear to be creations *ex nihilo*, and thus do not provide insight into how they came to be. Moreover, the formal approach used is almost always the linguistic approach — where one reasons about strings in the relevant language representing the DES. Verifying that the solvability conditions are correct or that the corresponding supervisors solve the problem requires the reader to come up with their own informal understanding and interpretation of the conditions/constructions.

With a different formalism, Ricker and Rudie (2007) gave an epistemic interpretation to the conditional architecture, where the use of the formal language of epistemic logic enabled one to discuss the supervisory control in an anthropomorphic manner, which gives a more intuitive understanding for how control decisions are made. Their epistemic modelling resolves the drawback of the aforementioned linguistic approach, namely the meaning of an epistemic expression is immediately understandable at a glance, so that an expression of the form $K_1 \phi$ means “Supervisor 1 knows $\phi$”. The interpretation is only partial as their epistemic expression only captures a weaker architecture. Moreover, in their epistemic logic formulation, there is a tenuous connection between the solvability conditions and the actions to be prescribed for supervisors in a construction that exploits the conditions.

In our earlier work (Ritsuka and Rudie 2022), we adapted Ricker and Rudie’s epistemic formalism to the interpretation of some other commonly known architectures (Cieslak et al. 1988, Rudie and Wonham 1992, Prosser, Kam, and Kwatny 1997, Yoo and LaFortune 2002). The result was fruitful, in giving concise epistemic characterizations to the various architectures in a way such that each characterization is constituted by a disjunction of epistemic terms, and each term corresponds to a specific control decision. As such, the characterizations differ by only the presence or absence of terms in the disjunction, corresponding to the presence or absence of control decisions available in an architecture. This was achieved by reformulating an architecture as a result of observing that some control decisions plays multiple distinct roles.

This paper continues our advocacy of applying epistemic formalism and interpretation as an umbrella framework to the study of decentralized problems. In the present work, we cast a standard and representative but more complex conditional architecture in epistemic logic as well. But unlike our earlier work and any other prior works, which simply present supervisor existence and realization and then establish their correctness, the novelty of this work is the direct derivation of existence and realization expressions methodologically from the fusion rule. Notably, the derivation results in a direct link between the condition that must hold for a solution to exist and the control protocol that must be followed when the condition holds. That is, the result has a line-by-line correspondence between the expressions of the knowledge the supervisors must possess and the actions they must take.

We have chosen to demonstrate an epistemic characterization only of the conditional architecture instead of over the more general inference-based architectures (Kumar and Takai 2007). This choice is made as the demonstration presented here will be sufficiently instructive for how the methodology can be routinely applied in extending the result over the inference-based architectures. Hence here we put emphasis on the process over the result.
2 Preliminaries

To lay a foundation for the discussion, we recall the definition of discrete-event systems, in particular that of the Decentralized Supervisory Control and Observation Problem, then that of epistemic logic and its use in characterizing the conditional architecture. In this section we recall the methodology we used in our previous work (Ritsuka and Rudie 2022).

2.1 Discrete event systems

The systems under consideration in this work are *discrete-event systems*. A discrete-event system is a system with a discrete state space, where actions or event occurrences may cause the system to change state. We consider the system’s behaviours to be all finite sequences of events the system can generate from its initial state.

In this work, we use the formalisms of discrete-event systems found in Wonham and Cai (2018) and Cassandras and Lafortune (2007).

**Definition 2.1** We model a plant $G$ as a finite state automaton (FSA)

$$ G = (\Sigma, Q, \delta, q_0) $$

where $\Sigma$ is a finite set of events, $Q$ a finite set of states, $\delta$ the transition function, and $q_0 \in Q$ the initial state.

When more than one automaton is under discussion, we put the name of the automaton as a superscript in the components, e.g., we use $Q^G$ to refer to the state set of $G$.

The *language generated by $G$* is defined as

$$ L(G) = \{s \in \Sigma^* | \delta(q_0, s) \} $$

We interpret $L(G)$ as the set of physically possible behaviours of $G$.

A language $L$ is *prefix-closed* whenever for all strings $s\sigma \in L$, it is always the case that $s \in L$. By definition, $L(G)$ is always prefix-closed.

2.2 Decentralized supervisory control with partial observations

When the behaviour of a plant is not desirable, we constrain its behaviours through supervisory control. We will allow an arbitrary number of supervisors to jointly perform the control, where each supervisor observes and controls a subset of events, and may issue a control decision for a given event. Decentralized control has been examined by many DES researchers. For a more extensive discussion, see the decentralized control section in Cassandras and Lafortune (2007).

Because multiple supervisors may act on any given event, we require a mechanism to combine the (potentially conflicting or differing) control decisions of the different supervisors. Prosser, Kam, and Kwatny (1997) named such mechanisms *fusion rules*. Subsequent work by Yoo and Lafortune (2002) recognized that fusion rules for each event can be chosen separately and independently.

Formally, we define the decentralized supervisory architecture as follows.

**Definition 2.2** Let $CD$ be a set of supervisory control decisions. Let $N = \{f_1, \ldots, f_n\}$ be a finite set of $n$ supervisors for plant $G$. For simplicity, we will write $i$ instead of $f_i$ when referring to the supervisor.
For each supervisor $i \in \mathcal{N}$, let $\Sigma_{i,c}, \Sigma_{i,o} \subseteq \Sigma$ be the sets of controllable and observable events for supervisor $i$, resp. Let $P_i : \Sigma^* \rightarrow \Sigma_{i,o}^*$ be the natural projection function to capture a supervisor $i$’s observation, i.e., if a plant generates a sequence of events $s$, supervisor $i$ will only see $P_i(s)$. Denote the set of events controlled by some supervisors $\Sigma_c = \bigcup_{i \in \mathcal{N}} \Sigma_{i,c}$, and the set of events not controlled by any supervisor $\Sigma_{uc} = \Sigma - \Sigma_{c} = \bigcap_{i \in \mathcal{N}} \Sigma - \Sigma_{i,c}$. The sets $\Sigma_{c}$ and $\Sigma_{uo}$ are defined similarly. Let $\mathcal{N}_\sigma = \{ i \in \mathcal{N} | \sigma \in \Sigma_{i,c} \}$ be the set of supervisors that can control $\sigma$.

With a slight abuse of notation, we use $P_i(G)$ to denote the automaton constructed by replacing all transitions labelled by an unobservable event with $\varepsilon$ and determinized, so that $P_i(G)$ recognizes the language $P_i L(G)$.

Now supervisors can be prescribed by $f_i : P_i L(G) \times \Sigma_{i,c} \rightarrow \mathcal{CD}$ for all $f_i \in \mathcal{N}$. Specifying that supervisors take arguments from $P_i L(G)$ instead of $L(G)$ implicitly encodes requirements traditionally referred to as feasibility and validity, i.e., a supervisor must make consistent decisions for strings $s, s'$ that look alike to that supervisor, i.e., such that $P_i(s) = P_i(s')$. We focus only on FSA-based supervisors. That is, a supervisor $f_i$ can be realized as a Moore machine $(S_i, f_i')$ such that $f_i(s, \sigma) = f_i'(s, \delta_i(s, q_i, 0), \sigma)$, where $S_i$ is an FSA $(\Sigma, Q_i, \delta_i, q_i, 0)$, and $f_i' : Q_i \times \Sigma_{i,c} \rightarrow \mathcal{CD}$. We will refer to $f_i'$ simply as $f_i$ when convenient.

For each controllable event $\sigma$, let $cd_{\mathcal{N}_\sigma}$ denote the collection of control decisions issued by supervisors $i \in \mathcal{N}_\sigma$, hence $cd_{\mathcal{N}_\sigma}$ has exactly $|\mathcal{N}_\sigma|$ elements. Let $\mathcal{CD}_{\mathcal{N}_\sigma}$ be the collection of all such $cd_{\mathcal{N}_\sigma}$’s. Let $\mathcal{FD} = \{ \text{enable}, \text{disable} \}$ be the set of fused decisions. Let $f_\sigma : \mathcal{CD}_{\mathcal{N}_\sigma} \rightarrow \mathcal{FD}$ be the fusion functions chosen separately for each $\sigma \in \Sigma_{c}$, and the joint supervision $f_\mathcal{N} : L(G) \times \Sigma_c \rightarrow \mathcal{FD}$ be defined as $f_\mathcal{N}(s, \sigma) = f_\sigma([f_i(P_i(s), \sigma)]_{i \in \mathcal{N}_\sigma})$. Consequently, only decisions issued by supervisors $i \in \mathcal{N}_\sigma$ are fused, and decisions of supervisors not controlling event $\sigma$ are ignored.

The closed-loop behaviour of the plant under joint supervision is denoted by $L(f_\mathcal{N}/G)$, and defined inductively as the smallest set such that:

- $\varepsilon \in L(f_\mathcal{N}/G)$
- $s \in L(f_\mathcal{N}/G) \land s' \in L(G) \land \sigma \in \Sigma_{uc} \Rightarrow s\sigma \in L(f_\mathcal{N}/G)$
- $s \in L(f_\mathcal{N}/G) \land s' \in L(G) \land \sigma \in \Sigma_{c} \land f_\mathcal{N}(s, \sigma) = \text{enable} \Rightarrow s\sigma \in L(f_\mathcal{N}/G)$

The second bullet point in the definition of closed-loop behaviour encodes the requirement that a physically possible event that is not controllable by any supervisor must be allowed to occur under supervision. The third bullet point says that a physically possible event that is controllable and for which the fused decision is enable must be allowed to occur under supervision.

Whereas the fusion function $f$ can be seen as an n-ary operation on supervisory control decisions $\mathcal{CD}$, there is no operation over the fused decision set $\mathcal{FD}$, since elements in this set are to be interpreted as fused decisions and should be regarded as final. In particular, whereas we may take $\mathcal{CD} = \mathcal{FD}$ as Boolean values and $f$ as a Boolean function as existing works commonly do when it is convenient (Rudie and Wonham 1992, Yoo and Lafortune 2002), when moving to non-binary control decisions (Yoo and Lafortune 2005), we clearly separate the two sets and hence $\mathcal{FD}$ should not be considered as Boolean values (although still binary). For this reason, we also do not use the two symbols 0, 1 for elements of either set.

The sets $\mathcal{CD}$ and $\mathcal{FD}$ being disjoint also simplifies discussion: we can now refer to an element of either set without explicitly stating from which set it comes. We also refer to a particular element of either set simply as a decision when no confusion would arise.
Whereas the set $CD$ determines the number of distinct control decisions available to the supervisors, what those decisions mean — their semantics — is given by the fusion rule $f$. Although the symbols we choose for control decisions may be formally meaningless, we will still choose them with the intended fusion rule in mind. For example, in what follows, we will use the symbol on (resp., off) as an element of $CD$ with the intended meaning that some supervisor’s decision is that an event should be allowed to occur (resp., not allowed to occur).

Constructing multiple supervisors jointly restricting a plant’s behaviours will be called the **Decentralized Supervisory Control and Observation Problem** (DSCOP).

**Problem 2.3 (Decentralized Supervisory Control and Observation Problem, DSCOP)**

Given an automaton $G$ which specifies the plant behaviour as the prefix-closed language $L(G)$, an automaton $E$ which specifies the legal behaviour as the prefix-closed language $L(E)$, $n$ pairs of controllable/observable event sets, choose an appropriate set of control decisions $CD$, and a fusion rule $f$, and synthesize a set $N$ of supervisors, such that $L(f_N/G) = L(E)$.

We arrange $E$ to be a subautomaton of $G$ as we will need this assumption to construct the structure relevant to the interpretation of our epistemic expressions.

We usually study the condition for a class of DSCOP for $CD$ and $f$ that are fixed a priori. In particular, $CD$ and $f$ should be independent of any specific $G$ and $E$. In practice, one may choose whatever $CD$ and $f$ necessary to solve the problem at hand. Fixing $CD$ and $f$ allows us to classify pairs of $G$ and $E$ according to the $CD$ and $f$ sufficient for the decentralized control problem to be solvable, and thus allows comparison among pairs of $CD$ and $f$.

### 2.3 Epistemic logic

Ricker and Rudie (2000, 2007) observed that reasoning about the decision-making of decentralized supervisors could be facilitated using formal reasoning about knowledge, via epistemic logic. Although formal conditions for solving DSCOP can be described using conditions on strings in languages, and hence do not require a formal logic description, epistemic logic provides a natural modelling paradigm that parallels natural languages, thus giving better intuition into the reasoning behind the decisions that supervisors make. Specifically, the epistemic operator in the language expresses concepts such as “agent $i$ knows that a certain event must be disabled”. Imbuing supervisors with such anthropomorphic capabilities sets the stage for decision-making to be linked to “knowledge” that the agents have— much as humans base their decision-making on what they know or don’t know about a certain situation. The work in Ricker and Rudie (2000, 2007) uses epistemic logic as a way to speak about what knowledge supervisors must possess for a problem to be solvable, but it does not capitalize on the link between knowledge and action to relate a supervisor’s decision-making directly to its knowledge in an immediately apparent way.

Epistemic logic as used in distributed computing problems was first presented by Halpern and Moses (1990). See Fagin et al. (2004) for more details. We provide in the remainder of this section the concepts from epistemic logic needed to understand our work.
Definition 2.4 For a fixed set $V$ of variables, where $v$ denotes some element of $V$, and a fixed finite set $N$ of agents, where $i$ denotes some element of $N$, the set of epistemic modal formulae is defined inductively by the following grammar:

\[
S, T ::= (v) \quad \text{propositional variable } v \\
| (\neg S) \quad \text{negation of } S \\
| (S \land T) \quad \text{conjunction of } S, T \\
| (K_i S) \quad \text{agent } i \text{ knows } S
\]

Definition 2.5 It is conventional to define other connectives from the primitive ones above:

- $(\alpha \lor \beta) =_{df} \neg(\neg \alpha \land \neg \beta)$,
- $(\alpha \Rightarrow \beta) =_{df} (\neg \alpha \lor \beta)$

Where convenient, we use the connectives defined above to express ideas, but when reasoning about epistemic formulae, we assume that the connectives of Definition 2.5 have all been syntactically expanded, so that we only have to deal with primitive ones.

We omit parentheses according to the following precedence convention: unary operators $\neg, K_i$ bind tightest, then $\land, \lor, \Rightarrow$.

When an expression $S$ is short enough, we sometimes write $\neg S$ as $\overline{S}$ for compactness.

The semantics of epistemic formulae are given through the use of a structure called a Kripke structure.

Definition 2.6 For some $V$ and $N$, a Kripke structure, or simply a frame $I$ is

\[
(W, \pi, \{\sim_i\}_{i \in N})
\]

where

- $W$ is a finite set of possible worlds, or states.
- $\pi : W \times V \rightarrow \{\text{true, false}\}$ evaluates each propositional variable in $V$ at each possible world in $W$ to either true, or false.
- For each $i \in N$, $\sim_i \subseteq W \times W$ is the accessibility relation over possible worlds, and we say world $w'$ is considered by agent $i$ as an epistemic alternative if $w' \sim_i w$.

Whereas the accessibility relations are commonly required to be equivalence relations over $W$, the formal construction we will present uses relations that are not reflexive, and are thus partial equivalence relations.

In our formalism, the propositional connectives (the second and third items in Definition 2.7 below) are to be understood as usual. The semantics of the epistemic operator (the last item in Definition 2.7) reflect that, upon observing a sequence of events generated by the plant, a supervisor can only know something (i.e., be certain that it is true), if it is always true after any sequence (generated by the plant) that looks the same to the supervisor as the sequence of events it has observed.

To reflect the discussion above, we thus adopt the following formal definition of the semantics of epistemic formulae as the relation $\models$ of pairs of Kripke structures and worlds, and epistemic modal formulae, given inductively over the structure of the formulae.

---

1 The term “states” is not at odds with the DES notion of “state” since the worlds in the frames we construct in this work happen to be states of some FSA.
3 Epistemic approach to the conditional architecture

We now present an epistemic approach to describe the conditional architecture by Yoo and Lafortune (2004). The approach is adapted from Ricker and Rudie (2007) with some necessary modifications.

3.1 Epistemic model for decentralized problems

Consider a plant \( G \), with legal behaviour prescribed by a subautomaton \( E \) of \( G \), and \( n \) pairs of sets of controllable/observable events. For each supervisor \( i \) (\( i \in \mathcal{N} \)), we construct \( G_i^{obs} = P_i(\mathcal{G}) \), which is effectively an observer of the plant from supervisor \( i \)'s point of view. The state set, \( Q_i^{obs} \), of each observer is \( \{\Sigma_{i,uo}\text{-closure of } q | q \in Q\} \). In other words, the supervisor cannot distinguish \( G \) and \( G_i^{obs} \) by only observing sequences of events generated by these two FSA.

Next we construct a composite structure that will allow us to keep track of plant behaviour and each supervisor’s view of the corresponding plant behaviour. We do this through the construction \( G' = G \times G_1^{obs} \times \cdots \times G_n^{obs} = (\Sigma, Q', \delta', q'_0) \), where \( Q' \subseteq Q \times Q_1^{obs} \times \cdots \times Q_n^{obs} \subseteq Q \times \bigwedge Q \times \cdots \times \bigwedge Q \), \( \delta' \) is component-wise application of \( \delta \) and \( \delta_i^{obs} \) for \( i \in \mathcal{N} \), \( q'_0 = (q_0, q_{0,1}^{obs}, \ldots, q_{0,n}^{obs}) \) where \( q_{0,i}^{obs} \in Q_i^{obs} \) and thus \( q_{0,i}^{obs} \subseteq Q \) for \( i \in \mathcal{N} \).

Our composite structure \( G' \) generates the same language as \( G \) does, however the Cartesian product of states forming \( Q' \) allows us to track more information than that available by simply tracking the sequence of states in \( Q \) visited by some sequence of events in the plant language. Namely, \( (q, q_1^{obs}, \ldots, q_n^{obs}) \in Q \) records not only the current state \( q \) of \( G \), but also each supervisor’s estimate \( q_i^{obs} \) of the set of states the plant could possibly be in based on supervisor \( i \)’s observation, for each \( i \in \mathcal{N} \). For the accessible part of \( G' \), we always have that \( q \in q_i^{obs} \) for all \( i \in \mathcal{N} \), which is to be expected since the actual plant state should always be a state that any observer thinks the plant could be in.

Although the automaton \( G' \) is not necessarily isomorphic to \( G^2 \), its behaviour is identical to that of \( \mathcal{G} \), namely, it is always the case that \( L(G') = L(G) \). In other words, from a behavioural standpoint, \( G \) and \( G' \) cannot be distinguished by observations of generated events. Moreover, while \( G_i^{obs} \) and \( G \) are not distinguishable by the particular supervisor \( i \), \( G' \) and \( G \) are not distinguishable by any observer (even one observing \( \Sigma \)). Consequently, if \( G \) specifies a plant, one can think of that plant as also being modelled by \( G' \). Automaton \( G' \) can be computed off-line, and therefore its information is available to all supervisors.

Now we are ready to construct the Kripke structure against which the expression of conditional co-observability is interpreted.

---

\(2\) This fact has been known since the centralized setting (Tong et al. 2017).
We will start by letting $W = Q'$. To avoid multiple arguments with both subscripts and superscripts, we will write $(w, q_1^w, \ldots, q_n^w)$ for an element of $W$ instead of $(q, q_1^ob, \ldots, q_n^ob)$.

Next we construct accessibility relations $\sim_i$ such that $w \sim_i w'$ whenever $w e Q E \wedge w_i = w'$. Particularly note that $\sim_i$ is an equivalence relation on $\{w \in W | w e Q E\}$, and for all $w$ such that $w e Q E$, $w$ has no referent nor relatum (participating $\sim_i$). Hence, the relations $\sim_i$ are partial equivalence relations on $W$. For $w$ such that $w e Q E$, it is reasonable to consider the equivalence class $[w]_{\sim_i}$, or simply, $[w]_i$. With an abuse of notation, if $w e Q E$, let $[w]_i = \emptyset$. Informally, one may interpret $[w]_i$ as containing exactly the worlds that are epistemic alternatives to $w$ as perceived by supervisor $i$.

We need the following atomic propositions, taken from (Ricker and Rudie 2007), to capture the presence/absence of transitions of an event $\sigma$:

\[
\pi(w, \sigma_G) := \delta^G(\sigma, w)!
\]

\[
\pi(w, \sigma_E) := \delta^E(\sigma, w)!
\]

The intended meaning of $\pi(w, \sigma_G) = true$ is that $\sigma$ can physically occur at state $w$, as specified by $G$; whereas $\pi(w, \sigma_E) = true$ indicates that $\sigma$ is legal and should be allowed to happen. It follows that $\pi(w, \sigma_E) = true$ implies that $\pi(w, \sigma_G) = true$, which reflects the fact that $E$ is a subautoman of $G$. That is, we take the set of all atomic propositions to be $V = \bigcup_{\sigma e \Sigma_i} \{\sigma_G, \sigma_E\}$.

With the arguments all defined, we can now let the Kripke structure be $I = (W, \pi, \{\sim_i\})_{i \in \mathbb{N}}$.

Although the constructed Kripke structure $I$ should be parameterized over specific $G$, $\{P_i\}_{i \in \mathbb{N}}$, and $E$, since in our discussions we will not need to simultaneously consider multiple sets of these entities, but assume an indefinite one, we write simply $I$, rather than $I(G, P_1, \ldots, P_n, E)$.

For convenience, we need the following shorthand defined in terms of $\sigma_G$ and $\sigma_E$. We use over-bar instead of the standard symbol for logical negative in expressions when it makes it clearer to see at a glance which propositions in a compound proposition are or are not negated.

1. $\sigma_{el} := \sigma_E = \sigma_G \wedge \sigma_E$, reads “$\sigma$ must be enabled to satisfy the control requirement”;
2. $\sigma_{dl} := \sigma_G \wedge \sigma_E$, reads “$\sigma$ must be disabled to satisfy the control requirement”;
3. $\sigma_e := \overline{\sigma_G} \vee \sigma_E = \sigma_G \Rightarrow \sigma_E$, reads “$\sigma$ can be enabled without violating the control requirement” or alternatively, “if $\sigma$ is even possible then it ought to be enabled; otherwise, it does not matter”;
4. $\sigma_d := \overline{\sigma_E}$, reads “$\sigma$ can be disabled without violating the control requirement”, or alternatively, “if $\sigma$ is even possible then it ought to be disabled; otherwise, it does not matter”.

Note that $\sigma_d$ could have been defined as $\overline{\sigma_G} \vee \overline{\sigma_E}$ to parallel our definition of $\sigma_e$, but since an event that is not possible legal is also not legal, $\overline{\sigma_G}$ implies $\overline{\sigma_E}$ already.

We had claimed that the epistemic logic is much closer to natural language than predicate logic. This claim can now be seen through the following argument. Consider, for example, the epistemic formula $K_i \sigma_e$, which can be loosely read as “agent $i$ knows that if the event $e$ is physically possible, it is legal”. Rendering the same expression without epistemic formula results in the following expression. Let $b$ be such that $\delta'(b, q_i^0) = w$, then $(I, w) \models K_i \sigma_e$ is equivalent to

$$\forall s'. P_i(s') = P_i(s) \land s' \in L(E) \Rightarrow s' \sigma \in L(G) \lor s' \sigma \in L(E).$$
Sometimes this also appears in the literature as
\[ \forall s' \in P_i^{-1} P_i(s) \cap L(E) s' \sigma \not\in L(G) \lor s' \sigma \in L(E). \]

One may then argue that the epistemic expression is more concise.

### 3.2 Direct derivation of supervisor existence and realization for conditional architecture

The conditional architecture of Yoo and Lafortune (2004) admits five possible local decisions: “enable”, “disable”, “enable if nobody disables”, “disable if nobody enables”, and “no decision”. As argued in (Ritsuka and Rudie 2022), to remove the potential confusion of a local “enable” (resp., “disable”) and a global “enable” (resp., “disable”) decisions, we renamed the former to on (resp., off). We also give more compact names to the conditional decisions “enable if nobody disables” and “disable if nobody enables” and called them weak on and weak off, respectively. Finally, we consider “no decision” as a decision and hence call it abstain.

We now present the conditional architecture from Yoo and Lafortune (2004) as follows.

The set of control decisions is
\[ \mathcal{CD} = \{ \text{on, off, weak on, weak off, abstain} \}. \]

For each \( \sigma \in \Sigma_c \), a default action \( \text{dft} \in \{ \text{enable, disable} \} \) must be chosen as part of the solution. By letting the collection of local decisions for \( \sigma \) after string \( s \) be \( \text{cd} = \{ f_i(P_i(s), \sigma) \}_{i \in N_\sigma} \) for short, the fusion rule \( f^\text{dft}_{\sigma} \) for \( \sigma \) is defined as

\[
\begin{align*}
f^\text{dft}_{\sigma}(\text{cd}) &= \begin{cases}
\text{enable} \text{ if on } \in \text{cd}, & \text{off } \not\in \text{cd} \quad (1.1) \\
\text{disable} \text{ if on } \not\in \text{cd}, & \text{off } \in \text{cd} \quad (1.2) \\
\text{enable} \text{ if on } \not\in \text{cd}, & \text{off } \not\in \text{cd}, \quad (1.3) \\
\text{weak on } \in \text{cd}, & \text{weak off } \not\in \text{cd} \\
\text{disable} \text{ if on } \not\in \text{cd}, & \text{off } \not\in \text{cd}, \quad (1.4) \\
\text{weak on } \not\in \text{cd}, & \text{weak off } \in \text{cd} \\
\text{dft} \text{ if on } \not\in \text{cd}, & \text{off } \not\in \text{cd}, \quad (1.5) \\
\text{weak on } \not\in \text{cd}, & \text{weak off } \not\in \text{cd}
\end{cases}
\end{align*}
\]

Hence, a solution to the DSCOP, in addition to constructing the supervisors, also requires that for each \( \sigma \in \Sigma_c \) one chooses either the fusion rules \( f^\text{enable} \) or the fusion rule \( f^\text{disable} \).

Ideally, we would like to gradually derive the expressions of the problem solvability condition and derive construction of the supervisors directly from the fusion rule. However, as the written form of communication prevents us from doing so, we have to give both of them a priori. Nonetheless, the development process will still be apparent from the proof. We stress that the expressions are not creations ex nihilo, but are obtained from the fusion rule. This process is quite methodologically, as in every step there is only one sensible choice. Hence our methodology contrasts with the traditional approaches, which generally involve some human cleverness.

For ease of understanding and compactness, we define the following shorthand notation for epistemic formulae, all implicitly parameterized by an event \( \sigma \) known from the context. The notation originated in our previous work (Ritsuka and Rudie 2022).

Define the modal operator “someone knows…”:
\[
S \phi := \bigvee_{i \in N_\sigma} K_i \phi
\]
With a supervisor \( i \) known from the context, define a variant of the modal operator “someone knows” as “some other supervisor (other than \( i \)) knows...”:

\[
O \phi := \bigvee_{j \in \mathbb{N}, j \neq i} K_i \phi
\]

Finally, we need the following shorthand for some frequently needed epistemic expressions.

\[
\begin{align*}
K_i^0 \sigma_e &:= K_i \sigma_e \\
K_i^0 \sigma_d &:= K_i \sigma_d \\
K_i^1 \sigma_e &:= K_i (\sigma_d! \Rightarrow O \sigma_d) \\
K_i^1 \sigma_d &:= K_i (\sigma_e! \Rightarrow O \sigma_e)
\end{align*}
\]

Although we plan to derive the solvability condition and a control policy, it is nonetheless beneficial to first consider a tentative, but tangible proposal. Consider a tentative knowledge-based control policy \((G^{obs}_i, \mathcal{KP}_i)\), where \( \mathcal{KP}_i \) is defined according to Eq. 2:

\[
\mathcal{KP}_i(w, \sigma) = \begin{cases} 
\text{on} & \text{if } (I, w) \models K_i^0 \sigma_e \land \overline{K_i^0 \sigma_d} \quad (2.1) \\
\text{off} & \text{if } (I, w) \models K_i^0 \sigma_e \land K_i^0 \sigma_d \quad (2.2) \\
\text{weak on} & \text{if } (I, w) \models K_i^0 \sigma_e \land K_i^0 \sigma_d \land K_i^1 \sigma_e \land \overline{K_i^1 \sigma_d} \quad (2.3) \\
\text{weak off} & \text{if } (I, w) \models K_i^0 \sigma_e \land K_i^0 \sigma_d \land K_i^1 \sigma_e \land K_i^1 \sigma_d \quad (2.4) \\
\text{abstain} & \text{otherwise} \quad (2.5)
\end{cases}
\]

This construction appears to be quite natural. Even without formally deriving it from the fusion rule Eq. 1, one may still be able to instinctively come up with it, as the epistemic expressions directly capture what decisions are desirable. The point can be made stronger, if one temporarily ignores the semantics of the epistemic formulae and focus on how the form of Eq. 2 parallels with that of the fusion rule Eq. 1.

Clearly, the cases defining Eq. 2 are mutually exclusive and exhaustive. Intuitively, one can see that the correctness of cases Eqs. 2.1 to 2.4 is guaranteed by the epistemic formulae, but since the case Eq. 2.5 does not involve any epistemic expressions, a condition to ensure its correctness is needed. This condition is our conditional-co-observability.

**Definition 3.1** The Kripke structure \( I \) is said to be conditional-co-observable whenever for each \( \sigma \in \Sigma_e \), there is a choice of \( s \) from \( e \) and \( d \) for this \( \sigma \), such that for any string \( s \in L(E) \), where \( w \) is the world \( s \) leads to, it must be that

\[
(I, w) \models \left[ \bigwedge_{i \in \mathbb{N}_e} \overline{K_i^0 \sigma_d} \land K_i^0 \sigma_e \land \overline{K_i^1 \sigma_d} \land K_i^1 \sigma_e \right] \Rightarrow \sigma_s.
\]
As it will turn out, whenever Definition 3.1 holds, a solution to the DSCOP exists and can be expressed as Eq. 2. While Eq. 3.1 resembles the expressions Ricker and Rudie (2007) had, it is not ideal as its last disjunction is not an epistemic formula, and hence not very illuminating, since it doesn’t describe the knowledge that an agent must possess. Ultimately we will replace Eq. 3.5 with an epistemic formula.

The last two ingredients we need before showing that conditional-co-observable is necessary and sufficient to solve DSCOP are two characterizations of “solving” DSCOP. The first characterization (used in the proof of necessity) is expressed in the following lemma.

Lemma 3.2 A joint supervision \( f_N \) solves DSCOP iff

\[
\begin{align*}
\text{l} \in L(E) \land s \sigma \in L(G) \land \sigma \in \Sigma_{uc} & \Rightarrow s \sigma \in L(E) \\
\text{l} \in L(E) \land s \sigma \in L(G) \land \sigma \in \Sigma_c & \Rightarrow f_N(s, \sigma) = \text{enable} \Rightarrow s \sigma \in L(E) \\
\land f_N(s, \sigma) = \text{disable} \Rightarrow s \sigma \notin L(E)
\end{align*}
\]

The lemma expresses that a solution to DSCOP must ensure that uncontrollable events do not lead to illegality Eq. 4.1, and that if a controllable event is allowed to happen Eq. 4.2, it leads to a legal string; and if it is prevented from happening Eq. 4.3, it leads to an illegal string.

Proof Directly from the definition of \( L(f_N/G) \).

The second characterization of what it means to solve DSCOP (used in the proof of sufficiency) is expressed in the following lemma.

Lemma 3.3 A joint supervision \( f_N \) solves DSCOP iff

\[
\begin{align*}
\text{l} \in L(E) \land s \sigma \in L(G) \land \sigma \in \Sigma_{uc} & \Rightarrow s \sigma \in L(E) \\
\text{l} \in L(E) \land s \sigma \in L(G) \land \sigma \in \Sigma_c & \Rightarrow s \sigma \in L(E) \Rightarrow f_N(s, \sigma) = \text{enable} \\
\land s \sigma \notin L(E) \Rightarrow f_N(s, \sigma) = \text{disable}
\end{align*}
\]

The lemma expresses that a solution to DSCOP must ensure that uncontrollable events do not lead to illegality Eq. 5.1, and that if a controllable event is legal, it is allowed to happen Eq. 5.2, and if it is illegal, it is prevented from happening Eq. 5.3.

Proof Directly from the definition of \( L(f_N/G) \).
We can now proceed to the development of our main result.

**Theorem 3.4** In the conditional architecture, there exists a set \( N \) of \( n \) supervisors that solves the DSCOP iff \( I \) is controllable and conditional-co-observable.

Moreover, whenever a solution exists, the knowledge-based control policy in Eq. 2 is a solution.

Again, we emphasize that for the sake of the statement, we take Eq. 2 and Definition 3.1 as given. But we will actually derive them during the proof.

Informally, for the necessity part of the proof, we will perform a case analysis on all possible combinations of local control decisions after a string, i.e., on Eq. 1. The fused decision will imply some global properties of the string, i.e., whether it can be followed by a legal/illegal event, as for this direction, we are assuming the supervisors solve DSCOP. From the local decisions, we can derive epistemic characterizations of the supervisors’ knowledge in the following way.

By feasibility, we know that a supervisor has to issue an identical decision for other strings indistinguishable from the actual string. Then it is possible to extrapolate the possible global decisions at those strings, and hence their global properties. The common global properties of all these strings is then the supervisor’s knowledge. After we exhaust all combinations of local control decisions, we will obtain the proposed epistemic expression of conditional-co-observability.

We now provide our formal proof.

**Proof** Conditional-coobservability is necessary (\( \Rightarrow \))

For the proof of necessity, we use Eq. (4) as the characterization of what it means to solve DSCOP. Condition Eq. 4.1 directly implies to controllability. What is left is to show that Eqs. 4.2 and 4.3 imply conditional-co-observability.

Suppose there exists such a set \( N = (f_1, \ldots, f_n) \) of \( n \) supervisors, such that Eqs. 4.2 and 4.3 hold.

Although the proof would be much easier by showing that if otherwise conditional-co-observability fails, then a contradiction arises, for our purpose, we explicitly derive conditional-co-observability as a necessity.

Consider some \( s \in L(E), \sigma \in \Sigma_c \) such that \( s\sigma \in L(G) \). Let \( w \) be the world \( s \) leads to; since \( s \in L(E), w_c \in Q^E \).

Now we perform a case analysis on the possible combinations of local decisions, in the same order as specified in the fusion rule Eq. 1.

In each case there will be a specific supervisor \( i \) that we will focus our attention on. For this supervisor we will consider strings \( s' \in L(E) \) such that \( P_i(s') = P_i(s) \) and consider the global decision \( f_{N}(s', \sigma) \). Either the global decision at \( s' \) is the same as that at \( s \), or they differ. We will further assume that \( s'\sigma \in L(G) \) so that the difference, if present, is material by Eq. (4). Since the string \( s' \) is arbitrary, if we can conclude proposition \( \phi \) for \( s' \), we can conclude it for all \( w' \in [w]_i \), and hence conclude \( K_i\phi \) for \( w \).

Since the control policy for each controllable event is designed individually (Ritsuka and Rudie 2022), for brevity, going forward when we speak of supervisors \( i, j \), we implicitly mean \( i, j \in N_\sigma \), i.e., supervisor \( i \) and supervisor \( j \) each controls \( \sigma \).

**Case A.** Suppose that for some \( i \), the local decision is \( f_i(P_i(s), \sigma) = on \), and consequently by Eq. 1.1 the fused decision must be \( f_{\sigma}(s) = enable \).

By feasibility, it must be that \( f_i(P_i(s'), \sigma) = on \) as well, and consequently \( f_{\sigma}(s') = enable \) as well. That is, in this case the global decision at \( s' \) must
be the same as at \( s \). Because we assumed the supervisors solve the DSCOP, it must be that \( s'\sigma \in L(G) \Rightarrow s'\sigma \in L(E) \) by Eq. 4.2, which is equivalent to \( s'\sigma \notin L(G) \lor s'\sigma \in L(E) \). Hence, for this particular agent \( i \), at world \( w \), we have

\[
K_i\sigma_e = K_i^0\sigma_e
\]

and

\[
\neg K_i\sigma_d = K_i^0\sigma_d
\]

holds. Hence

\[
f_i(P_i(s), \sigma) = \text{on} \Rightarrow K_i^0\sigma_e \land K_i^0\sigma_d
\]

(6)

Also, from \( K_i^0\sigma_e \) we have \( S^0\sigma_e \), which is exactly Eq. 3.1.

**Case B.** The case in which for some \( i, f_i(P_i(s), \sigma) = \text{off} \) is reasoned analogously to Case A, from which it follows that

\[
f_i(P_i(s), \sigma) = \text{off} \Rightarrow K_i^0\sigma_e \land K_i^1\sigma_d
\]

(7)

and Eq. 3.2.

Note that in Case A and Case B, when defining the control protocol Eq. 2 we explicitly excluded the situation where \( K_i^0\sigma_e \land K_i^0\sigma_d \) holds at world \( w \), which is equivalent to \( K_i(\sigma_e \land \sigma_d) \) and implies that \( \sigma_e \land \sigma_d \), i.e., \( K_i(\sigma_e \land \sigma_d) \), which, by disjunctive syllogism (modus tollendo ponens), is in turn equivalent to \( \neg \sigma_G \). Although this contradicts the fact that \( s\sigma \in L(G) \) anyway and thus is redundant, we nonetheless choose to preclude \( K_i^0\sigma_e \land K_i^0\sigma_d \) explicitly in Eqs. 2.1 and 2.2.

**Case C.** Suppose that for some \( i, f_i(P_i(s), \sigma) = \text{weak on}, \text{but for no } j, f_j(P_j(s), \sigma) = \text{on, off, weak off} \). Consequently by Eq. 1.3 the fused decision must be \( f_{\sigma}(s) = \text{enable} \).

Suppose that the global decision at \( s' \) differs from that at \( s \), i.e., suppose \( f_{\mathcal{N}'}(s', \sigma) = \text{disable} \), which is equivalent to \( s'\sigma \notin L(E) \) by Eq. 4.3. Moreover, suppose that the difference is material, i.e., suppose \( s'\sigma \in L(G) \). That is, we have assumed that \( \sigma_d! \). Then, since supervisor \( i \)'s decision for \( s' \) cannot be different from that supervisor’s decision for \( s \) — by feasibility, there must be some supervisor \( j \) other than \( i \) such that supervisor \( j \)'s decision for \( s' \) differs from that supervisor’s decision for \( s \), i.e., \( f_j(P_j(s'), \sigma) = \text{off} \). By the argument in case B applied to \( s' \), we have \( K_j^0\sigma_d \) (which is \( O^0\sigma_d \) because \( j \neq i \)) hold at \( w' \), provided the assumption that \( \sigma_d! \) holds at \( w' \), i.e., \( \sigma_d! \Rightarrow O^0\sigma_d \). Hence, for this particular agent \( i \), at world \( w \), we have \( K_i(\sigma_d!) \Rightarrow O^0\sigma_d \) = \( K_i^1\sigma_e \), i.e.,

\[
f_i(P_i(s), \sigma) = \text{weak on} \Rightarrow K_i^1\sigma_e.
\]

(8)

Further, from \( K_i^1\sigma_e \) we have

\[
S^1\sigma_e
\]

(i.e., Eq. 3.3) holds at \( w \).

**Case D.** The case in which for some \( i, f_i(P_i(s), \sigma) = \text{weak off} \), but for no \( j, f_j(P_j(s), \sigma) = \text{on, off, weak on} \) is reasoned analogously as in Case C. We can derive that

\[
f_i(P_i(s), \sigma) = \text{weak off} \Rightarrow K_i^1\sigma_d
\]

(9)

and Eq. 3.4.
Case E. Finally, suppose that $f_i(P_i(s), \sigma) = \text{abstain}$ for all $i$. If $\text{dft} = \text{enable} = f_N(s, \sigma)$, it must be $s \sigma \notin L(G) \lor s \sigma \in L(E)$ by Eq. 4.2. If $\text{dft} = \text{disable} = f_e(s)$, it must be $s \sigma \notin L(E)$ by Eq. 4.3, which gives $\phi = \sigma_d$. Thus we derive Eq. 3.5.

Conditional-coobservability is sufficient ($\Leftarrow$)

For the proof of sufficiency, we use Eq. (5) as the characterization of what it means to solve DSCOP. Controllability directly implies condition Eq. 5.1. What is left is to show that whenever conditional-co-observability holds, our knowledge-based control policy Eq. 2 is a solution to the problem under the required fusion rule Eq. 1, i.e., it satisfies Eqs. 5.2 and 5.3.

Before proceeding to the proof, as we have promised, we need to show how the knowledge-based control policy was derived.

Recall the proof for the necessity part. Gathering Eqs. 6 to 9, we have that for any solution $f_i$, it is necessary that

$$f_i(P_i(s), \sigma) = \begin{cases} 
\text{on} & \Rightarrow K_i^0 \sigma_e \land K_i^0 \sigma_d \\
\text{off} & \Rightarrow K_i^0 \sigma_e \land K_i^0 \sigma_d \\
\text{weak on} & \Rightarrow K_i^1 \sigma_e \\
\text{weak off} & \Rightarrow K_i^1 \sigma_d 
\end{cases}$$

To recover a design of the agents, we essentially need to establish the implication in the other direction, with the additional requirement that the cases of the definition must be exhaustive and mutually exclusive. We attempt to establish the mutual exclusiveness in the most obvious way: i.e., define the control protocol as Eq. 2. It is clearly fully defined due to the “otherwise” clause.

Then note that conditional-co-observability is equivalent to

$$(I, w) \models \bigvee_{i \in \mathcal{N}_\pi} K_i^0 \sigma_e \lor K_i^0 \sigma_d \lor K_i^1 \sigma_e \lor K_i^1 \sigma_d \lor \sigma_s.$$ (10)

So, when Eq. 10 holds, according to Eq. 2, $f_i(P_i(s, \sigma)) = \text{abstain}$ if and only if

$$(I, wa) \models K_i^0 \sigma_e \land K_i^0 \sigma_d$$ (11.1)

$$\lor K_i^0 \sigma_e \land K_i^0 \sigma_d \land K_i^1 \sigma_e \land K_i^1 \sigma_d$$ (11.2)

$$\lor K_i^0 \sigma_e \land K_i^0 \sigma_d \land K_i^1 \sigma_e \land K_i^1 \sigma_d \land \sigma_s.$$ (11.3)

hence we can replace the “otherwise” clause in Eq. 2 by Eq. (11). We now reproduce the knowledge-based protocol as follows:

$$\mathcal{K}\mathcal{P}_i(w, \sigma) = \begin{cases} 
\text{on} & \text{if } (I, w) \models K_i^0 \sigma_e \land K_i^0 \sigma_d \\
\text{off} & \text{if } (I, w) \models \overline{K_i^0 \sigma_e} \land K_i^0 \sigma_d \\
\text{weak on} & \text{if } (I, w) \models \overline{K_i^0 \sigma_e} \land \overline{K_i^0 \sigma_d} \land K_i^1 \sigma_e \land \overline{K_i^1 \sigma_d} \\
\text{weak off} & \text{if } (I, w) \models \overline{K_i^0 \sigma_e} \land \overline{K_i^0 \sigma_d} \land \overline{K_i^1 \sigma_e} \land K_i^1 \sigma_d \\
\text{abstain} & \text{if } (I, w) \models K_i^0 \sigma_e \land K_i^0 \sigma_d \\
& \lor \overline{K_i^0 \sigma_e} \land \overline{K_i^0 \sigma_d} \land K_i^1 \sigma_e \land K_i^1 \sigma_d \\
& \lor \overline{K_i^0 \sigma_e} \land \overline{K_i^0 \sigma_d} \land K_i^1 \sigma_e \land \overline{K_i^1 \sigma_d} \land \sigma_s. 
\end{cases}$$ (12.1) - (12.7)
By intentionally putting Eq. (11) into disjunctive normal form, we reveal two distinct roles of the abstain decision. First, as discussed, the situation Eq. 11.1 cannot happen unless \( s \sigma \notin L(G) \). Then, Eq. 11.2 is the true abstaining decision, since regardless of the legality of \( \sigma \), there is always some other supervisor that can make a correct decision. In the case of Eq. 11.3, since all epistemic formulae are negated, we take, for now, that Eq. 11.3 expresses the situation that the supervisor is in a “doesn’t know” situation.

To verify the correctness of our knowledge-based control protocol Eq. (12), we perform a case analysis over conditional-co-observability. The trick is how to split the cases so that in each case we can infer the local decisions. Then a natural way to proceed is to split the cases according to the lines defining Eq. (12). Recall that the cases are exhaustive given conditional-co-observability, since that is how Eq. (11) was obtained.

To establish Eqs. 5.2 and 5.3, fix an \( s, L(E) \), and \( \sigma \in \Sigma_e \) such that \( s \sigma \in L(G) \). Let \( w \) be the world \( s \) leads to. Since conditional-co-observability Eq. 10 holds at \( w \), there is a supervisor \( i \) for which \( K^0_i \sigma_e \vee K^0_i \sigma_d \vee K^1_i \sigma_e \vee K^1_i \sigma_d \) holds. Consider the following cases, which, as argued above, are exhaustive. We will show that in each case, Eqs. 5.2 and 5.3 hold.

Case 1. If \( K^0_i \sigma_e \land K^0_i \sigma_d \) (i.e., Eq. 12.1), then locally we have that \( i \) issues on by Eq. 12.1, and by \( K^1_i \sigma_e \), globally we have \( s \sigma \in L(E) \). So Eq. 5.3 holds vacuously. To show Eq. 5.2, it suffices to show that the fused decision is enable. We argue that it is impossible for there to be some supervisor \( j \) that issues off. If that were possible, we’d have \( K^0_j \sigma_d \) which, together with \( K^0_i \sigma_e \), would imply \( s \sigma \notin L(G) \), contradicting the assumption.

Case 2. The case \( K^0_i \sigma_e \land K^0_i \sigma_d \) (i.e., Eq. 12.2) is reasoned analogously to Case 1.

Case 3. If \( K^0_i \sigma_e \land K^0_i \sigma_d \land K^1_i \sigma_e \land K^1_i \sigma_d \) (i.e., Eq. 12.3), then locally we have that \( i \) issues weak on by Eq. 12.3.

(a) If \( s \sigma \in L(E) \), we desire that the fused decision be enable.

(i) If some supervisor \( j \) issues the decision on, then as argued in Case 1, there cannot be a third supervisor \( k \) issuing the decision off, thus the fused decision must be enable by Eq. 1.1.

(ii) If some \( j \) issues the decision off, then, by Eq. 12.2, we’d have \( K^0_j \sigma_e \land K^0_i \sigma_d \), and the argument can be established by letting \( j \) play the role of \( i \) in Case 2.

(iii) If some \( j \) issues the decision weak off, then by Eq. 12.4, we have \( K^1_j \sigma_d \), which implies that \( \sigma_e \Rightarrow \\bigvee_{k \neq j} K_k \sigma_e \). Since we do have \( \sigma e \) by \( s \sigma \in L(E) \), there is some \( k \) such that \( K_k \sigma_e \). Moreover, we have that \( K_k \sigma_d \) since \( s \sigma \in L(G) \). Then the argument can be established by letting \( k \) play the role of \( i \) in Case 1.

(iv) Otherwise, the fused decision must be enable, as desired.

(b) If \( s \sigma \notin L(E) \), we desire that the fused decision be disable. The argument is analogous to Case 3(a).

Case 4. The case \( K^0_i \sigma_e \land K^0_i \sigma_d \land K^1_i \sigma_e \land K^1_i \sigma_d \) (i.e., Eq. 12.4) is reasoned analogously to Case 3.

Case 5. The case \( K^0_i \sigma_e \land K^0_i \sigma_d \) (i.e., Eq. 12.5) contradicts \( s \sigma \in L(G) \) as argued.

Case 6. If \( K^0_i \sigma_e \land K^0_i \sigma_d \land K^1_i \sigma_e \land K^1_i \sigma_d \) (i.e., Eq. 12.6), then locally we have that \( i \) issues abstain by Eq. 12.6. Recall that by our analysis of the abstain decision, this represents the true abstaining decision. The argument is established analogously to Case 3 and Case 4.
Case 7. If $K_i^0 \sigma_e \land \overline{K}_i^0 \sigma_d \land K_i^1 \sigma_e \land \overline{K}_i^1 \sigma_d \land \sigma_*$ (i.e., Eq. 12.7), then locally we have that $i$ issues abstain by Eq. 12.7. Recall that by our analysis of the abstain decision, this represents the “doesn’t know” decision.

(a) If $* = e$, i.e., $s \sigma \in L(E)$, we desire that the fused decision be enable. If there is some supervisor $j$ that issues the decision on, off, weak off, then the fused decision is enable by an argument exactly the same as Case 3(a). If there is some supervisor $j$ that issues the decision weak on, then the fused decision is enable by letting $j$ play the role of $i$ in Case 3. In the last case where all supervisors issue the decision abstain, the desired fused decision can be achieved by setting dft to enable in Eq. 1.5.

(b) The case that $* = d$, i.e., $s \sigma \notin L(E)$, is argued analogously. \hfill \Box

We have thus completed the derivation of the problem solvability condition and a knowledge-based control protocol from the fusion rule. Additionally, the process is entirely methodologically and at no point requires human cleverness. We hence propose this process as a prototypical example of a more uniform, formal approach to study other decentralized architectures.

One drawback of the expressions Eqs. 3.5 and 12.7 is that they contain the non-epistemic term $\sigma_*$ (which becomes either $\sigma_e$ or $\sigma_d$), and therefore does not provide any insight into what knowledge an agent must possess to support the agent’s actions. Consequently, we interpreted the situation Eq. 12.7 as that the supervisor possesses no knowledge. However, we will demonstrate that it does, in fact, possess some knowledge.

We resume the proof of Case E and show that further progression will lead to an epistemic expression in place of Eq. 3.5. We start by aggregating local properties of strings $s'$ indistinguishable from $s$ to a specific supervisor $i$ as we have done for all other cases, so that we can obtain an epistemic expression.

Similar to the argument in Case C, suppose the global decision at $s'$ differs from that at $s$, i.e., suppose $f_N(s', \sigma) = \text{disable}$, which is equivalent to $s' \sigma \notin L(E)$ by Eq. 5.3. Moreover, suppose that the difference is material, i.e., suppose $s' \sigma \in L(G)$. That is, we have assumed $\sigma_d$. Now let $j$ be the supervisor such that $f_j(P_j(s'), \sigma) = \text{off}$ or weak off (these are the only two possibilities to get $f_N(s \sigma) = \text{disable}$ by Eq. 1). Applying the argument in Case B or Case C to $s'$ with $j$ playing the role of $i$, we have $K_j \sigma_d$ (which is $O \sigma_d$ because $j \neq i$) or $K_j(\sigma_d! \Rightarrow O \sigma_d)$ (which is $O(\sigma_d! \Rightarrow O \sigma_d)$).

Hence at $w$, if dft = enable, we have

$$K_i^2 \sigma_e := K_i(\sigma_d! \Rightarrow O \sigma_d \lor O(\sigma_d! \Rightarrow O \sigma_d)).$$

and by a similar argument, if dft = disable, we have

$$K_i^2 \sigma_d := K_i(\sigma_e! \Rightarrow O \sigma_e \lor O(\sigma_e! \Rightarrow O \sigma_e)).$$

I.e.,

$$f_i(P_i(s), \sigma) = \text{abstain} \Rightarrow K_i^2 \sigma_*$$

(13)

Also, we have $S^2 \sigma_*$. That is, Eq. 3.1 can now be formally replaced with

$$(I, w) \models S^0 \sigma_e \lor S^0 \sigma_d$$

$$\lor S^1 \sigma_e \lor S^1 \sigma_e$$

$$\lor S^2 \sigma_*,$$

(14)
where $\ast$ is either $e$ or $d$.

With the reformulated expression of conditional-co-observability, in exactly the same way we obtained Eq. (11) in the original proof, now we can further establish that $f_i(P_i(s, \sigma)) =$ abstain if and only if

$$
(I, w) \models K^0_i \sigma_e \land K^0_i \sigma_d \\
\lor K^0_i \sigma_e \land \overline{K^0_i \sigma_d} \land K^1_i \sigma_e \land K^1_i \sigma_d \\
\lor K^0_i \sigma_e \land \overline{K^0_i \sigma_d} \land \overline{K^1_i \sigma_e} \land \overline{K^1_i \sigma_d} \land K^2_i \sigma_\ast.
$$

We have discussed the meaning of the first two disjuncts in the proof. In the last case, abstain is instead used as an even weaker version of weak on or weak off, as indicated by the expression $K^2_i \sigma_\ast$. But in any case, it is not entirely illustrative to say that the supervisor “doesn’t know”, which is what Ricker and Rudie (2007) called the “abstain” decision.

Finally, while we only demonstrated the epistemic formalism on the conditional architecture, the approach can be systematically extended over the more general inference-based architectures (Kumar and Takai 2007). One note is that as the level of inference increases, the recursive structure of epistemic expressions $K^M_i$ and $K^N_i$ explodes in size quickly. Hence it is not advised to explicitly expand out the expression for evaluation, but to employ dynamic programming.

### 3.3 A visualization to aid in the revision of problem requirements

In this section, we demonstrate how to visually represent the epistemic logic formalism for modestly-sized problems on an example of a non-conditional-co-observable language. We will see how the epistemic logic expression can provide a more meaningful explanation on how the language fails to be conditional-co-observable. This information aids in understanding the approach of Takai and Kumar (2008) to revise the problem to synthesizing a sublanguage, and can also be extended to revising the problem to synthesizing a superlanguage or even an incomparable language. We refer the reader to (Ritsuka and Rudie 2021) for an even more compact visualization, underlying which is nonetheless the epistemic interpretation.

Consider the following example. The set of possible events is $\Sigma = \{\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \mu\}$, observable event sets are $\Sigma_{1,o} = \{\mu\}$, $\Sigma_{2,o} = \{\beta_1, \beta_2\}$, and controllable event sets are $\Sigma_{1,c} = \Sigma_{2,c} = \{\gamma\}$. The plant $G$ and legal language specification $E$ are captured in the automaton $G' = G \times P_1(G) \times P_2(G)$ depicted in Fig. 1. The language $L(E)$ is marked by states with double borders. Since the states of $G'$ are the worlds of its Kripke structure, we can embed the Kripke structure in the representation of $G'$ as shown in Fig. 1. The problem is to determine whether there exist two supervisors with the observable and controllable event sets given above, such that $L(f_N/G) = L(E)$.

Let us focus on $\gamma$ since it is the only controllable event. Hence we focus on states 0, 2, 3, 4, 5, since these are the states where $\gamma$ can happen.

In state 4 (resp. 5), supervisor 2 can enable (resp. disable) $\gamma$. In state 0, supervisor 1 can enable $\gamma$. But in states 2, 3, which are indistinguishable to both supervisors, since they are both in the same equivalence classes ($M_1$ for supervisor 1 and $B_0$ for supervisor 2), neither supervisor 1 nor 2 can control $\gamma$ unambiguously. Hence the language $L(E)$ is not conditional-co-observable.
The representation of $G'$ and the epistemic interpretation of conditional control decisions provides guidance for how to modify the control requirement to obtain a conditional-co-observable language.

If we are looking for a sublanguage, we can only make legal states illegal but not vice versa. By our previous analysis, at least one supervisor is able to make a correct control decision unambiguously in states $S = \{0, 1, 4, 5, 7, 8, 10\}$, hence all we need to worry about are the states in the set $M_1 - S = B_0 - S = \{2, 3\}$. To resolve the conflict that $\gamma$ is legal at state 3 but illegal at state 2, we can make state 7 illegal.

To see how making state 7 illegal gives a conditional-co-observable sublanguage, let’s look at states in $M_1$ and $B_0$. At states in $M_1$, $\gamma$ is illegal at states 2, 3, 5 but is legal at state 4. With only binary control decisions, supervisor 1 cannot possibly make an unambiguous decision. We can see that supervisor 2 is in a similar situation by examining states in $B_0$.

However, with the ability to infer the knowledge of other supervisors and the conditional decisions at their disposal, the desired control requirement can be achieved. Suppose that supervisor 1 is a intelligent being, and let’s imagine how the intelligent being may attempt to solve the dilemma. Consider what if supervisor 1 were to try to guess the legality of $\gamma$ after it sees $\mu$. Clearly this guess is not always correct, i.e., it is false at exactly state 4. But knowing that the other supervisor can unambiguously enable $\gamma$ if the plant is indeed at state 4 supervisor 1 is then able to focus on only the rest of the states in $M_1$, and fortunately, its guess is correct in all of them. Hence supervisor 1 can confidently disable $\gamma$ at states in $M_1$ unambiguously, knowing its mistake would be corrected by the other supervisor. Similar reasoning is also carried out by supervisor 2.
The design of the fusion rule is exactly to allow the correction of mistakes. A weak off is issued by a supervisor knowing that if disabling the event is incorrect then another supervisor can correct the first supervisor by a definite on decision.

Formally, with state 7 made illegal, states \{2, 3\} are unambiguous. However, since the set \{2, 3\} is a proper subset of both \(M_1\) and \(B_0\), and states in both sets \(M_1\) and \(B_0\) remain ambiguous, the conditional decision, i.e., weak off has to be issued at states in the set \(M_1\) (resp. \(B_0\)) by supervisor 1 (resp. supervisor 2).

If it is reasonable for the problem at hand to admit a solution that is not necessarily a sublanguage, we can also make state 6 legal too. By similar reasoning as we just did, supervisor 1 should issue decision weak on at states 2, 3; and supervisor 2 can issue decision on at states in the set \(B_1\), since this set is no longer ambiguous.

### 4 Conclusion

In this paper, we discuss how decentralized control problems can benefit from the use of epistemic logic.

We point out that epistemic logic can be used to discuss not only some specific classes of DSCOP (Ricker and Rudie 2000, Ricker and Rudie 2007), but also it can be used more broadly to describe other classes of decentralized supervisory control problems. The use of epistemic formalism provides a formal approach towards describing decentralized problems, and consequently allows mechanical derivation of problem solvability conditions and solution constructions. The derivation also results in direct coupling between the expression of problem solvability condition and the expression describing the control policies. This line-by-line coupling allows us to use the same expression throughout the discussions of proving necessary and sufficient conditions, of describing the algorithm to construct the supervisors, and of verifying the correctness of the algorithm.

From the foregoing discussions, we would expect other decentralized control or diagnosis conditions could be treated in a comparable fashion. For instance, consider the work of Kumar and Takai (2007), which is more general than that of Yoo and Lafortune (2005). We developed our epistemic expressions based on Yoo and Lafortune (2005) because it is simpler and thus we are able to demonstrate our key ideas without more complex (yet not conceptually different) technical development. The same principles demonstrated here could apply to Kumar and Takai (2007) as well. The only technical difference is that one would need to use a finer, (possibly infinite) string-based Kripke structure as described by Ricker and Rudie (2000), along with a corresponding definition of relations \(\sim_i\).

Casting the decentralized problem the way we did makes it easier to understand the reasoning behind various control decisions. We believe that one advantage of our framework is that in trying to come up with solutions to future DES problems, this framework can aid in going directly from a working supervisor solution to the necessary and sufficient conditions that would match such a solution. Moreover, if the constraints of some given problem are not met (and hence that problem is not solvable as is using decentralized control), our model makes it more apparent how to alter the constraints in a way that is meaningful for the application at hand.
Acknowledgements The research described in this paper was undertaken at Queen’s University, which is situated on traditional Anishinaabe and Haudenosaunee territory. The research was inspired by and supported through an NSERC CRD-DND project with General Dynamics Land Systems–Canada and Defence Research and Development Canada.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

Cieslak R et al (1988) Supervisory control of discrete-event processes with partial observations. IEEE Trans Autom Control 33(3):249–260. https://doi.org/10.1109/9.402
Cassandras CG, Lafortune S (2007) Introduction to Discrete Event Systems. Springer-Verlag GmbH, Second, p 772
Fagin R et al (2004) Reasoning About Knowledge. The MIT Press. https://doi.org/10.7551/mitpress/5803.001.0001
Halpern JY, Moses Y. (1990) Knowledge and common knowledge in a distributed environment. J ACM 37(3):549–587. https://doi.org/10.1145/79147.79161
Kumar R, Takai S (2007) Inference-Based Ambiguity Management in Decentralized Decision-Making: Decentralized Control of Discrete Event Systems. IEEE Trans Autom Control 52(10):1783–1794. https://doi.org/10.1109/TAC.2007.906158
Prosser JH, Kam M, Kwatny HG (1997) “Decision fusion and supervisor synthesis in decentralized discrete-event systems”. In: Proceedings of the American Control Conference. IEEE. https://doi.org/10.1109/ACC.1997.608978
Ricker SL, Rudie K (2000) Know means no: Incorporating knowledge into discrete-event control systems. IEEE Trans Autom Control 45(9):1656–1668. https://doi.org/10.1109/9.880616
Ricker SL, Rudie K (2007) Knowledge Is a Terrible Thing to Waste: Using Inference in Discrete-Event Control Problems. IEEE Trans Autom Control 52(3):428–441. https://doi.org/10.1109/TAC.2007.892371
Ritsuka K, Rudie K (2021) “A Visualization of Inference-Based Supervisory Control in Discrete-Event Systems”. In: 2021 60th IEEE Conference on Decision and Control (CDC). IEEE. https://doi.org/10.1109/cdc5484.2021.9683210
Ritsuka K, Rudie K (2022) A correspondence between control and observation problems in decentralized discrete-event systems. arXiv:2204.10792 [eess.SY]
Ritsuka K, Rudie K (2022) Epistemic interpretations of decentralized discrete-event system problems. Discrete Event Dyn Syst 32(3):359–398. https://doi.org/10.1007/s10626-022-00363-7
Rudie K, Wonham WM (1992) Think globally, act locally: decentralized supervisory control. IEEE Trans Autom Control 37(11):1692–1708. https://doi.org/10.1109/9.173140
Takai S, Kumar R (2008) Synthesis of Inference Based Decentralized Control for Discrete Event Systems. IEEE Trans Autom Control 53(2):522–534. https://doi.org/10.1109/tac.2007.915171
Tong Y et al (2017) Current-state opacity enforcement in discrete event systems under incomparable observations. Discrete Event Dyn Syst 28(2):161–182. https://doi.org/10.1007/s10626-017-0264-7
Wonham WM, Cai K (2018) Supervisory Control of Discrete Event Systems. Springer-Verlag GmbH. 487 pp
Yoo TS, Lafortune S (2002) A General Architecture for Decentralized Supervisory Control of Discrete-Event Systems. Discrete Event Dyn Syst 12(3):335–377. https://doi.org/10.1023/a:1015625600613
Yoo TS, Lafortune S (2004) Decentralized Supervisory Control With Conditional Decisions: Supervisor Existence. IEEE Trans Autom Control 49(11):1866–1904. https://doi.org/10.1109/tac.2004.837595
Yoo TS, Lafortune S (2005) Decentralized supervisory control with conditional decisions: supervisor realization. IEEE Trans Autom Control 50(8):1205–1211. https://doi.org/10.1109/tac.2005.852556

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.
K. Ritsuka received their Ph.D. in 2023 from Queen’s University. Their research interests include the control of discrete-event system.

K. Rudie received her Ph.D. in 1992 from the University of Toronto, in the Systems Control Group. In 1992-93 she was a postdoctoral researcher at the Institute for Mathematics and its Applications, Minnesota. Since 1993 she has been at Queen’s University (Canada) where she is a Professor of Electrical and Computer Engineering, with a cross-appointment in the School of Computing. From 2004-2006 she was an IEEE Control Systems Society Distinguished Lecturer. She has served on the editorial board of the Journal of Discrete Event Dynamic Systems (since 2000), where she is currently a Department Editor, and as an Associate Editor for IEEE Transactions on Control Systems Technology (since 2015), IEEE Transactions on Automatic Control (1996-1999), and IEEE Control Systems Magazine (2003). She is a Fellow of the IEEE. Her research focuses on the control of discrete-event systems.