A Universal Treatment of Architectures and Fusion Rules in Decentralized Discrete-Event Systems

Richard Ean  Karen Rudie

Abstract

This paper provides a universal treatment of architectures in decentralized discrete-event systems. Current approaches in discrete-event systems do not provide a direct way to compare existing architectures or proposed potentially novel architectures. Determining whether a new architecture is more general than an existing known architecture relies on producing examples \textit{ad hoc} and on individual inspiration that puts the conditions for solvability in each architecture into some form that admits comparison. From these research efforts, a method has been extracted to yield a universal approach to decentralized discrete-event system architectures and their attendant fusion rules. This treatment provides an easy and direct way to compare the fusion rules—and hence to compare the strength or generality of the corresponding architectures.

1 Introduction

Many solutions of varying level of strengths exists for decentralized supervisory control problems [Cie+88; RW92; PKK97; YL02; YL04; KT05; CK11; RM13; ER22]. The solutions are given under various architectures. The breadth of an architecture is essentially represented by the class of closed-
1 Introduction

loop behaviours (languages) synthesizable under that architecture. One architecture is more general than the other if the behaviour allowed under the former is greater than that allowed under the latter. Since a common goal in discrete-event systems (DES) is to produce (often through control) solutions that generate as large a set of behaviours as possible (or that are minimally restrictive), DES research in decentralized systems has as one of its aims to investigate novel architectures that are more general than existing ones.

Technically, an architecture is characterized by the way decentralized decisions are combined according to its fusion rule. In a few simpler settings what distinguishes one architecture from another can be interpreted as how conflicting decisions are resolved. In a broader set of settings epistemic interpretations are given [RR00; RR07; ER21; ER22]. However there does not appear to be a uniform interpretation of the differences.

The traditional approach to investigating new, perhaps superior, architectures always proceeds in a similar manner. First, a novel architecture is proposed, often by augmenting or modifying an existing one. Then, a characterization for decentralized problems to have solutions under the novel architecture is given. If it is intended to demonstrate that the novel architecture is superior to some existing architecture, one shows that the problem solvability characterization of the novel architecture logically entails that of the existing architecture. To demonstrate that the new architecture is not a superfluous complication, one provides an example: a problem solvable under the novel architecture but not under the existing one. Sometimes the novel architecture turns out to be incomparable to existing ones. An example of this would be the disjunctive architecture [PKK97], which was shown by Yoo and Lafortune [YL02] to be incomparable to the prior conjunctive architecture [RW92] first used in decentralized DES control problems.

The traditional approach is complicated. First, the problem solvability characterization is usually simply presented, with little insight provided to indicate how it was derived. The same remark applies to the presentation of the example problem. Moreover, the approach is indirect and can be laborious.

This paper gives a universal interpretation for decentralized architectures, and proposes a uniform and simple approach that lends itself easily to direct comparison of decentralized architectures.
2 Decentralized Problems

For simplicity of discussion, we shall concern ourselves with the following kind of decentralized observation problems\(^1\).

Given alphabet \(\Sigma\) and subalphabets \(\Sigma_{i,o} \subseteq \Sigma\) called the observed alphabets, natural projections \(P_i: \Sigma^* \rightarrow \Sigma_{i,o}^*\), for agents \(i \in \mathcal{N} = \{1, \ldots, n\}\), and given languages \(K \subseteq L \subseteq \Sigma^*\), the observation problem is to construct local decision functions \(f_i\) (also informally called the observers) and a fusion rule \(f\), such that

\[
\forall s \in L. \\
\begin{align*}
  s \in K &\implies f(f_1P_1(s), \ldots, f_nP_n(s)) = 1 \\
  \wedge s \in L - K &\implies f(f_1P_1(s), \ldots, f_nP_n(s)) = 0
\end{align*}
\]

That is, a fusion rule \(f\) solves the observation problem if, based on the local observers’ observations, the fused decision can distinguish between strings in \(L\) that are in \(K\) or not in \(K\).

For two strings \(s_1\) and \(s_2\) in \(L\) such that \(P_i(s_1) = P_i(s_2)\), necessarily \(f_iP_i(s_1) = f_iP_i(s_2)\) for all \(i \in \mathcal{N}\). We call this fact feasibility. From feasibility, we can directly derive the following results.

2.1 Joint Observability

When we place no restriction on \(f\), the problem is solvable iff \(K\) is jointly observable (JO) [Tri04], i.e.,

\[
\forall s_1, s_2 \in L. \\
\begin{align*}
  s_1 \in K \wedge s_2 \in L - K &\implies \exists i \in \mathcal{N}. P_i(s_1) \neq P_i(s_2).
\end{align*}
\]

That is, for two strings from \(L\), one in \(K\) and one not, there is always some observer that can tell them apart and signal the distinction to the fusion rule. Unfortunately, as shown in [Tri04], JO is undecidable for \(n \geq 2\), even when

\(^1\)It can be shown that the observation problems and the seemingly much more complicated control problems are in fact equivalent [ER]. Hence focusing on just the observation problems simplifies discussions.
both $K$ and $L$ are regular.

3 A Uniform Approach to Derive Problem Solvability Characterization from a Given Fusion Rule

We aim at deriving a uniform approach to compare decentralized architectures directly. We do so by first giving a uniform way to derive problem solvability characterization, from which we will then derive our direct approach for comparing architectures.

Suppose that a certain fusion rule $f$ is given (e.g., we could imagine it has been dictated by practice). We demonstrate how the problem solvability characterization can be derived directly from the fusion rule, by following the feasibility condition.

Without loss of generality, suppose that the local decision functions $f_i$ all have codomain $D$, for otherwise we can simply take $D = \bigcup_{i \in \mathcal{N}} \text{cdm}(f_i)$. Hence, the domain of $f$ is a subset of $D \times \cdots \times D$ ($n$ times).

We begin by considering a necessary condition for an observation problem to be solvable. If the observation problem is solvable, then there exist local decision functions $f_i$. Consider strings in $L$, we shall partition $L$ according to the following equivalence relation:

$$s \equiv t \iff (f_1 P_1(s), \ldots, f_n P_n(s)) = (f_1 P_1(t), \ldots, f_n P_n(t))$$

i.e., all local decisions, and consequently the fused decision, at $s$ and $t$ are identical such that it is then necessary that the partition $L/\equiv'$ be a refinement of the partition $\{K, L - K\}$ of $L$. Notice that $\equiv'$ is, in turn, refined by the equivalence relation $\equiv$ such that

$$s \equiv t \iff (P_1(s), \ldots, P_n(s)) = (P_1(t), \ldots, P_n(t))$$

We now have the following result.

**Theorem 3.1**

A necessary condition for problem solvability is that $\equiv$ refines the partition
\{ K, L - K \} \text{ of } L, \text{ i.e.,}

\begin{equation*}
\forall s, t \in L, i \in I . P_i(s) = P_i(t) \Rightarrow \neg(s \in K \land t \in L - K)
\end{equation*}

which is exactly Joint Observability.

\textbf{Proof.} Suppose, for the sake of contradiction, that \( \equiv \) does not refine \( \{ K, L - K \} \). Then any solution to the problem cannot be feasible. \( \square \)

Notice that the local decision functions \( f_i \) are not involved in the condition that \( \equiv \) refines \( \{ K, L - K \} \).

Historically, decentralized problems were usually expressed in ways comparable with replacing in our expressions \( f_i P_i(p) \) with \( f_i(p) \). That is, the local decision functions were defined over plant behaviours rather than actually observed behaviours. In that case, to capture partial observations of the agents, a separate constraint called feasibility was explicitly stated as part of the problem. The feasibility condition is actually equivalent to the refinement of \( \equiv' \) by \( \equiv \) (by replacing \( f_i P_i(p) \) with \( f_i(p) \) in relevant expressions as aforementioned).

Observe that, when the local decision functions are identity functions (i.e., for any string \( s \), \( f_i(s) = s \)), the relations \( \equiv' \) and \( \equiv \) coincide. In this case, \( D \) may need to be as large as \( L \) (at most \( \aleph_0 \), i.e., countably infinite), and the fusion rule \( f \) is defined on every possible combination of local decisions for strings in \( L \). In this case, the condition stated in Thm. 3.1 turns out to be also sufficient, as we will demonstrate in Thm. 3.6. That is precisely what we meant when saying that JO characterizes problem solvability when we place no restriction on \( f \).

The sufficient condition is more complicated, especially when the fusion rule \( f \) is constrained in some way. We approach it as follows.

Suppose that \( f \) is only defined over a certain collection \( \mathcal{D} \) of combinations of local decisions\(^2\) \( (d_1, \ldots, d_n) \in D \times \cdots \times D \), i.e., \( \mathcal{D} = \text{dom}(f) \subseteq D \times \cdots \times D \). The size of \( \mathcal{D} \) roughly reflects the “capacity” of \( f \), so that if \( |\mathcal{D}| = 1 \), \( f \) is a constant 1 or 0, and that if \( \mathcal{D} \) is “large enough” for a problem at hand, \( f \) is virtually unconstrained.

\(^2\)Depending on the architecture, some combinations may be intentionally precluded.
The size of $\mathcal{D}$ alone does not fully capture the capacity of the fusion rule. What we need additionally is the following. Define symmetric relations on $\mathcal{D}$, so that, for any subset $N \subseteq \mathcal{N}$ of agents, $(d_1, \ldots, d_n) \sim_N (d'_1, \ldots, d'_n)$ when the two tuples differ by exactly the components indexed by $N$. The relations $\sim_N$ reflect that exactly the agents in $N$ have changed their decisions due to their change of observation. We may consider $\mathcal{D}$ and $\sim_N$ forming an undirected graph $(\mathcal{D}, \sim)$, which we will call a decision graph. We denote the decision graph also with $\mathcal{D}$. We consider the graph as a complete graph where edges are coloured by subsets of $\mathcal{N}$. We also colour nodes by the values of $f$ at the nodes (0 or 1). Then the decision graph $\mathcal{D}$ thus reflects the capacity of the architecture.

**Example 3.2**

The traditional way of describing the conjunctive architecture is by taking $D = \{1, 0\}$ and $f = \land$ [RW92]. For a problem with 2 agents, the decision graph can be depicted as in Fig. 1, where vertical/blue/dotted lines denote relation $\sim_1$; horizontal/red/dashed lines denote relation $\sim_2$; and diagonal/purple/solid lines denote relation $\sim_{1,2}$. The relation $\sim_\emptyset$ is the identity relation and is omitted from the graph. Red/singly-bordered nodes indicate fused decision being 0 and green/doubly-bordered nodes indicate fused decision being 1.

![Figure 1: Decision graph for the conjunctive architecture.](image)

Similarly, with a slight abuse of notation, define the analogous symmetric relations $\sim_N$ on $\mathcal{L}$, so that $s \sim_N t$ when the two tuples $(P_1(s), \ldots, P_n(s))$ and $(P_1(t), \ldots, P_n(t))$ differ by exactly the components indexed by $N$. The relations $\sim_N$ reflect that exactly the agents in $N$ have changed observation. We call the graph formed by $\mathcal{L}$ and $\sim_N$ an observation graph.

**Example 3.3**

Consider the observation problem with two agents where $L = \{a, b, ab, bb\}$, $K = b$, $\Sigma_{1,o} = \{a\}$ and $\Sigma_{2,o} = \{b\}$. This example is derived from [RW92, Fig. ...]
We depict the observation graph as in Fig. 2. Each node is labelled by strings \( s \in L \), \( P_1(s) \), and \( P_2(s) \), vertically stacked in that order. Similar to the decision graph depicted earlier, vertical/blue/dotted lines denote relation \( \sim_1 \); horizontal/red/dashed lines denote relation \( \sim_2 \); and diagonal/purple/solid lines denote relation \( \sim_{1,2} \). The relation \( \sim_\emptyset \) happens to be the identity relation for this example and is omitted from the graph. Red/singly-bordered nodes indicate strings in \( L - K \), and green/doubly-bordered nodes indicate string in \( K \).

![Observation graph for the observation problem in Example 3.3.](image)

As mentioned, we regard the decision graph as reflecting the “capacity” of the architecture. The notion of capacity originated from the fact that solving an observation problem is essentially finding a way to fold the observation graph into the decision graph. We proceed to describe such a folding formally as a graph morphism.

The local decision functions \( f_i \) can be seen as a mapping from \( P_i(L) \) to \( \mathcal{D} \), subject to the following requirement. For any strings \( s \) and \( s' \) in \( L \), let \( N \) be the (unique) set such that

\[
(P_1(s), \ldots, P_n(s)) \sim_N (P_1(s'), \ldots, P_n(s')).
\]

If

\[
(P_1(s), \ldots, P_n(s)) \xrightarrow{f_1 \times \cdots \times f_n} (d_1, \ldots, d_n)
\]

\[
(P_1(s'), \ldots, P_n(s')) \xrightarrow{f_1 \times \cdots \times f_n} (d'_1, \ldots, d'_n),
\]

then, with letting \( N' \) to be the set such that \( (d_1, \ldots, d_n) \sim_{N'} (d'_1, \ldots, d'_n) \), the mapping \( f_1 \times \cdots \times f_n \) must satisfy the following graph morphism conditions GM:
GM-1: Node-Colour Preserving

The mapping \( f_1 \times \cdots \times f_n \) preserves node colouring, that is, \( f_1 \times \cdots \times f_n \) achieves the desired fused decision.

GM-2: Edge-Colour Intensive

\( N \supseteq N' \), i.e., only agents with changed observation can change decisions, though they do not necessarily have to. In other words, the mapping \( f_1 \times \cdots \times f_n \) may drop some edge colours, but may not add any. This property captures feasibility.

Conversely, a morphism satisfying the two conditions above gives a solution to the problem.

We capture the foregoing in the following theorem.

**Theorem 3.4**

An observation problem is solvable if and only if there exists a morphism from the observation graph to the decision graph satisfying the morphism conditions GM.

**Proof.** (\( \Rightarrow \)): By the discussion preceding the theorem, which motivated the definition of GM.

(\( \Leftarrow \)): Suppose that there is a morphism \( g \) satisfying the morphism conditions GM. Then a solution can be constructed as follows. For each string \( s \in L \), let \( (d_1, \ldots, d_n) = g(P_1(s), \ldots, P_n(s)) \) and let \( f_i P_i(s) = d_i \) for \( i \in N \). Since \( g \) is edge-colour intensive (GM-2), the functions \( f_i \) are well-defined, i.e., if there were an \( s' \) such that \( P_i(s) = P_i(s') \), then \( f_i P_i(s) = f_i P_i(s') = d_i \). Since \( g \) preserves node colours (GM-1), \( f_i \) solves the problem, i.e., \( g = f_1 \times \cdots \times f_n \).

**Example 3.5**

The problem in Example 3.3 is solvable in the conjunctive architecture (Example 3.2), as we can construct the morphism depicted in Fig. 3.
Figure 3: Graph morphism from the observation graph in Fig. 2 to the decision graph in Fig. 1.

Notice how the leftmost diagonal/purple/solid edge in the graph on top loses its redness/horizontalness and become a vertical/blue/dotted edge in the graph on the bottom. All other node/edge colours do not change through the morphism.

The morphism gives a solution to the problem.

We can make the observation graphs more compact by defining the relations $\sim_N$ on $L/\equiv$ instead of directly on $L$, so that $[s] \sim_N [t]$ when the two tuples $(P_1(s), \ldots, P_n(s))$ and $(P_1(t), \ldots, P_n(t))$ differ by exactly the components indexed by $N$. Effectively we condense $\sim_{\emptyset}$ into the identity relation. That is, for two strings that are indistinguishable to any agent, nothing is lost by aggregating the nodes representing the two strings.

Notice that the relations $\sim_N$ are well-defined, as the definition does not depend on the specific elements of the equivalence classes used in the definition. Also, the observation graph $L/\equiv$ is well-defined when the node colouring is well-defined, i.e., when $\equiv$ refines $\{L, L - K\}$. Hence, whenever we discuss the graph $L/\equiv$, we implicitly assume that $\equiv$ refines $\{L, L - K\}$. Recall that $\equiv$ refining $\{L, L - K\}$ is necessary for any observation problem to be solvable.

Taking $L/\equiv$ as the observation graph instead of $L$ allows us to take advan-
A Uniform Approach to Compare Fusion Rules

The advantage of the feasibility condition: we can lift the mapping $f_1 \times \cdots \times f_n$ from $L$ to $L/\equiv$, and therefore show that the two morphism conditions, together with the implicit $\equiv$ refining $\{L, L - K\}$, are necessary and sufficient for the problem solvability in the given architecture.

The observation graphs are essentially a more compact alternative to the Kripke structures used in the works employing epistemic logic interpretations for decentralized problems [RR00; RR07; ER21; ER22]. The relations $\sim_N$ capture various notions of group knowledge, e.g., what is expressed by distributed knowledge and by the “everybody knows” operators in epistemic logic [Fag+04].

**Theorem 3.6**
If $\equiv$ refines $\{L, L - K\}$, then the observation problem is solvable.

**Proof.** If $\equiv$ refines $\{L, L - K\}$, one simply takes the observation graph as the decision graph and the morphism required by Thm. 3.4 is the identity function. I.e., the observers simply report to the fusion rule their observations, and the fusion rule makes decision accordingly.

However, recall that Tripakis [Tri04] demonstrated that in the most general case (i.e., when the local decision functions are the identity functions), the fusion rule may not be computable.

**4 A Uniform Approach to Compare Fusion Rules**

The traditional way to compare two fusion rules is by first obtaining a characterization of problem solvability with each of the fusion rules, and then demonstrate whether one characterization logically entails the other. In light of the discussion in the previous section, we can obtain a more direct way to compare fusion rules without deriving characterizations of problem solvability first.

Recall from the previous section, there is no formal distinction between observation graphs and decision graphs. Consequently, we have the following result.
Theorem 4.1
Given fusion rules \( f, f' \) and their respective decision graphs \( \mathcal{D}, \mathcal{D}' \), the fusion rule \( f' \) is more general than \( f \) if and only if there is a graph morphism from \( \mathcal{D} \) to \( \mathcal{D}' \) satisfying the graph morphism conditions GM.

Proof. \((\Leftarrow)\): Suppose there is a morphism \( g: \mathcal{D} \rightarrow \mathcal{D}' \) satisfying the morphism conditions GM. Consider an arbitrary observation problem solvable with the fusion rule \( f \). By Thm. 3.4, there is a morphism \( h: L/\equiv \rightarrow \mathcal{D} \) from the observation graph \( L/\equiv \) to the decision graph \( \mathcal{D} \) satisfying the morphism conditions GM. Then \( g \circ h: L/\equiv \rightarrow \mathcal{D}' \) is a morphism from the observation graph \( L/\equiv \) to the decision graph \( \mathcal{D}' \), which clearly satisfies the morphism conditions GM, and therefore, by Thm. 3.4, solves the observation problem. Thus, all problems solvable with \( f \) are also solvable with \( f' \).

\((\Rightarrow)\): Suppose that all observation problems solvable with the fusion rule \( f \) are solvable with \( f' \). A morphism from \( \mathcal{D} \) to \( \mathcal{D}' \) can be given as follows. Take \( \mathcal{D} \) as isomorphically equivalent to the observation graph of some observation problem. Clearly the observation problem is solvable as the identity morphism over \( \mathcal{D} \) satisfied GM. Then the problem is also solvable with the fusion rule \( f' \) by assumption. By Thm. 3.4, there must be a morphism \( h \) from \( \mathcal{D} \) to \( \mathcal{D}' \) satisfying the morphism conditions GM. The morphism \( h \) is what we wanted.

To see why we can consider \( \mathcal{D} \) as an observation graph, we construct an observation problem whose observation graph is isomorphic to \( \mathcal{D} \). Recall that \( \mathcal{D} \subseteq D \times \cdots \times D \). Construct the following problem. Take \( \Sigma = \{ (d, i) \mid d \in D \land i \in \mathcal{N} \} \) as our alphabet, where each symbol consists a decision \( d \), tagged by an agent \( i \), where \( (d, i) \) is alternatively written as \( d^i \). To enforce the desired observability, take \( \Sigma_{i,o} = \{ d^i \mid d \in D \} \). Associate to each node \( v = (d_1, \cdots, d_n) \) in \( \mathcal{D} \) the string \( s_v = d_1^1 \cdots d_n^i \), so that \( P_i(s_v) = d_i^i \) as desired. Let \( s_v \in K \) if \( v \) is coloured green/doubly-bordered, and \( s_v \in L - K \) if \( v \) is coloured red/singly-bordered. The association gives an isomorphism\(^3\) from the observation graph to the decision graph \( \mathcal{D} \) satisfying GM. Note that, precisely in the case when the set of available decisions \( D \) is countably infinite, the alphabet is countably infinite.

The need for an infinite alphabet can be eliminated, as we can encode symbols in an infinite alphabet in terms of a finite alphabet. Using a finite al-

\(^3\)The term “isomorphism” refers to the fact that the morphism is bijective and preserves edge colouring.
A Uniform Approach to Compare Fusion Rules

... instead however requires more sophistication in specifying the desired observability. Since we have assumed that the decision set is enumerable, let function \( \cdot : D \to \mathbb{N} \) be the enumeration of decisions in natural numbers. This enumeration function allows us to speak of the “\( j \)-th” decision in the set \( D \). First take \( \Sigma = \bigcup_{i \in \mathbb{N}} \{ 0_i, 1_i \} \) and \( \Sigma_{i,o} = \{ 0_i, 1_i \} \). Then associate to each node \( v = (d_1, \ldots, d_n) \) in \( D \) the string \( s_v = 0_i^{d_1} 1_1 \cdots 0_i^{d_n} 1_n \), so that \( P_i(s_v) = 0_i^{d_1} 1_i \). The intention of the encoding is that \( 0 \) enumerates decisions in unary notation, \( 1 \) marks the endings of code words, and subscripts impose observabilities. In other words, \( 0_i^{d_1} \cdot 1_i \) means a string of \( 0 \)'s of length \( d_i \). The idea is that if \( d_i \) is the \( j \)’th decision in the set \( D \), then it gets encoded by \( j \) \( 0 \)'s followed by \( 1 \).

Since the enumeration \( \cdot \) is injective, the encoding \( d_i \mapsto 0_i^{d_1} 1_i \) is also injective. Moreover, the encoding is prefix-free and hence instantaneously and uniquely decodable, i.e., the association to \( v \) of \( s_v \) is one-to-one. \( \square \)

We illustrate the methodology in the proof of Thm. 4.1 on the following example. The example uses a finite decision set, so that we can display the observation graph of our example however, the methodology is the same for infinite but countable sets \( D \).

Example 4.2
The decision graph of the conjunctive architecture in Example 3.2 can be seen as the observation graph of the following problem. Let the enumeration function \( \cdot \) send the symbol 0 to the number 0 and the symbol 1 to the number 1. With \( \mathcal{N} = \{ 1, 2 \} \), let \( \Sigma = \{ 0_1, 1_1, 0_2, 1_2 \} \), \( \Sigma_{1,o} = \{ 0_1, 1_1 \} \), and \( \Sigma_{2,o} = \{ 0_2, 1_2 \} \). Let \( L = \{ 1_1 1_2, 0_1 1_1 1_2, 1_1 0_2 1_2, 0_1 1_1 0_2 1_2 \} \) and \( K = \{ 0_1 1_1 0_2 1_2 \} \). Then the observation graph is depicted in Fig. 4.
We now illustrate how two architectures can be directly compared with our approach. We first show how one architecture can be determined to be strictly more general than another.

**Example 4.3**
In the C&P $\land$ D&A architecture [ER22], the local decisions available are $\{0, 1, dk\}$, where $dk$ stands for “don’t know”. The associated fusion rule outputs 0 whenever a 0 local decision is present, 1 whenever a 1 local decision is present, and is undefined otherwise: either when there are conflicting local decisions (both 0 and 1 are present), or when all supervisors don’t know (all supervisors are confused). The decision graph is depicted in Fig. 5, where grey/dash-bordered nodes indicate disallowed combination of local decisions.

![Decision graph for the C&P $\land$ D&A architecture.](image)

The C&P $\land$ D&A architecture is known to be weaker than the C&P architecture (which we have been calling the “conjunctive architecture”). This fact can be readily demonstrated by giving a decision graph morphism. Fig. 6
depicts such a morphism, where for representation purpose we no longer make use of horizontalness/verticalness to denote edge colouring.

Figure 6: Graph morphism from the decision graph for the C&P∧D&A architecture (bottom) to the decision graph for the C&P architecture (top).

One can also see that there can be no morphism going in the other direction, as there is no green/doubly-bordered node in the bottom graph having both red/dashed and blue/dotted edges to red/singly-bordered nodes, which is necessary for the node \((1, 1)\) in the top graph.

The following example shows how two seemingly different architectures can be determined to be equivalent.

**Example 4.4**
An alternative way to describe the conjunctive architecture is by using three decisions \(\{0, 1, cd\}\), where \(cd\) is to be interpreted as a conditional decision, so that the fusion rule outputs 1 when only the conditional decision is present, and otherwise behaves identically to the fusion rule in the C&P∧D&A archi-
A Uniform Approach to Compare Fusion Rules

tecture (although we renamed the decision \( d_k \) to \( cd \)). The decision graph is depicted in Fig. 7 without edges for compactness.

Figure 7: Alternative decision graph for the conjunctive architecture.

It is easy to check that this architecture is indeed equivalent to the conjunctive architecture. Since the graph would be too complex to draw, we describe the morphisms verbally. The morphism \( h \) to the conjunctive architecture is like the morphism from the C&P∧D&A architecture to the conjunctive architecture as given in the previous example, where all green/doubly-bordered nodes are sent to \((1, 1)\). Unlike the C&P∧D&A architecture, now we have a morphism \( g \) from the conjunctive architecture: red/singly-bordered nodes are mapped by reversing \( h \), where the only green/doubly-bordered node \((1, 1)\) is mapped to \((cd, cd)\). That is, in the C&P architecture, the decision \( 1 \) can be interpreted as a conditional decision, which aligns with the interpretation in [ER22], and contrasts to that in [RM13].

Notice that the composition \( g \circ h \) is the identity morphism over the decision graph of the conjunctive architecture, and \( h \circ g \) collapses the decision graph of the alternative description. The latter, in graph theoretical terms, can be called a retraction of the graph, and in fact gives a core of the graph. The presence of a retraction indicates redundancy in the alternative description; it may nonetheless provide a different perspective to comprehend the conjunctive architecture, as presented in [ER22].

The following example shows how two architectures can be determined to be incomparable.

**Example 4.5**
Recall the decision graph for the conjunctive architecture in the left part of Fig. 8. Compare it with the disjunctive architecture [PKK97], also known as
the D&A architecture, whose decision graph is depicted in the right part of Fig. 8.

![Decision graph for the conjunctive architecture recalled in the left, with the decision graph for the disjunctive architecture in the right.](image)

Figure 8: Decision graph for the conjunctive architecture recalled in the left, with the decision graph for the disjunctive architecture in the right.

One can see that there can be no morphism from left to right, as there is no green/doubly-bordered node in the right graph having both red/dashed and blue/dotted edges to red/singly-bordered nodes, which is necessary for the node $(1, 1)$ in the left graph. By a similar argument over the node $(0, 0)$ in the right graph, one can see that there can be no morphism from right to left either. This is sufficient to determine that the conjunctive architecture and the disjunctive architecture are incomparable. This confirms the result by Yoo and Lafortune [YL02].

5 Conclusion

We proposed two useful tools in studying decentralized observation problems: observation graphs and decision graphs. The decision graphs alone provide a systematic approach to directly compare decentralized architectures. Together with observation graphs, we have systematic approaches to derive problem solvabilities and solutions.

As we can see in the development of decentralized observation problems, the earlier works propose verifiable characterizations for problem solvability and computable algorithms to construct solutions [Cie+88; RW92; PKK97; YL02; YL04], but subsequent works can no longer provide computable solvability characterizations, let alone algorithms to construct solutions [KT05; CK11]. Said differently, finding a graph morphism may be hard, but verifying a witness could be easier. Specifically, when the graphs involved are finite, the problem can be solved in nondeterministic polynomial time, but has proofs verifiable in polynomial time. When the graphs are infinite, the
problem can be undecidable, but proofs can be verified. This suggests that in a situation where the solvability characterization becomes undecidable, one should attempt to prove the characterization instead. Moreover, when a solution is “finite” in some sense, e.g., the solution is described by finite state automata, it remains verifiable.

In summary, fusion rules with unbounded numbers of decisions present challenges for finding graph morphisms. Nonetheless, for fusion rules with finite, bounded numbers of decisions, our work provides a direct and easy approach to compare the corresponding architectures.

References

[Cie+88] R. Cieslak et al. “Supervisory control of discrete-event processes with partial observations”. In: IEEE Transactions on Automatic Control 33.3 (Mar. 1988), pp. 249–260. DOI: 10.1109/9.402 (cit. on pp. 1, 16).

[CK11] H. Chakib and A. Khoumsi. “Multi-Decision Supervisory Control: Parallel Decentralized Architectures Cooperating for Controlling Discrete Event Systems”. In: IEEE Transactions on Automatic Control 56.11 (Nov. 2011), pp. 2608–2622. DOI: 10.1109/tac.2011.2128730 (cit. on pp. 1, 16).

[ER] Richard Ean and Karen Rudie. A correspondence between control and observation problems in decentralized discrete-event systems. Submitted for publication. Preprint available as arXiv:2204.10792 (cit. on p. 3).

[ER21] Richard Ean and Karen Rudie. “A Visualization of Inference-Based Supervisory Control in Discrete-Event Systems”. In: 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, Dec. 2021. DOI: 10.1109/cdc45484.2021.9683210 (cit. on pp. 2, 10).

[ER22] Richard Ean and Karen Rudie. “Epistemic interpretations of decentralized discrete-event system problems”. In: Discrete Event Dynamic Systems (June 2022). DOI: 10.1007/s10626-022-00363-7 (cit. on pp. 1, 2, 10, 13, 15).
References

[Fag+04] Ronald Fagin et al. *Reasoning About Knowledge*. The MIT Press, 2004. DOI: 10.7551/mitpress/5803.001.0001 (cit. on p. 10).

[KT05] R. Kumar and S. Takai. “Inference-based Ambiguity Management in Decentralized Decision-Making: Decentralized Control of Discrete Event Systems”. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005. DOI: 10.1109/CDC.2005.1582701 (cit. on pp. 1, 16).

[PKK97] J. H. Prosser, M. Kam, and H. G. Kwatny. “Decision fusion and supervisor synthesis in decentralized discrete-event systems”. In: *Proceedings of the American Control Conference*. IEEE, 1997. DOI: 10.1109/ACC.1997.608978 (cit. on pp. 1, 2, 15, 16).

[RM13] S. L. Ricker and H. Marchand. “A parity-based architecture for decentralized discrete-event control”. In: *2013 American Control Conference*. IEEE, June 2013. DOI: 10.1109/acc.2013.6580727 (cit. on pp. 1, 15).

[RR00] S. L. Ricker and K. Rudie. “Know means no: Incorporating knowledge into discrete-event control systems”. In: *IEEE Transactions on Automatic Control* 45.9 (2000), pp. 1656–1668. DOI: 10.1109/9.880616 (cit. on pp. 2, 10).

[RR07] S. L. Ricker and K. Rudie. “Knowledge Is a Terrible Thing to Waste: Using Inference in Discrete-Event Control Problems”. In: *IEEE Transactions on Automatic Control* 52.3 (Mar. 2007), pp. 428–441. DOI: 10.1109/TAC.2007.892371 (cit. on pp. 2, 10).

[RW92] K. Rudie and W. M. Wonham. “Think globally, act locally: decentralized supervisory control”. In: *IEEE Transactions on Automatic Control* 37.11 (1992), pp. 1692–1708. DOI: 10.1109/9.173140 (cit. on pp. 1, 2, 6, 16).

[Tri04] Stavros Tripakis. “Undecidable problems of decentralized observation and control on regular languages”. In: *Information Processing Letters* 90.1 (Apr. 2004), pp. 21–28. DOI: 10.1016/j.ipl.2004.01.004 (cit. on pp. 3, 10).
[YL02] T.-S. Yoo and Stéphane Lafortune. “A General Architecture for Decentralized Supervisory Control of Discrete-Event Systems”. In: Discrete Event Dynamic Systems 12.3 (2002), pp. 335–377. DOI: 10.1023/a:1015625600613 (cit. on pp. 1, 2, 16).

[YL04] T.-S. Yoo and S. Lafortune. “Decentralized Supervisory Control With Conditional Decisions: Supervisor Existence”. In: IEEE Transactions on Automatic Control 49.11 (Nov. 2004), pp. 1886–1904. DOI: 10.1109/tac.2004.837595 (cit. on pp. 1, 16).