Chapter 7
An Overview of Synchronous Communication for Control of Decentralized Discrete-Event Systems

Laurie Ricker

7.1 Introduction

Communication plays a crucial role in the operation of many large decentralized and distributed systems. In the classical formulation of the decentralized discrete-event control problem, there are no communication channels between controllers. Recall that decentralized controllers have only a partial observation of system behavior, and thus, for each controller, there is some uncertainty as to the precise state of the system. When a global disablement decision must be taken, a control strategy succeeds if there is at least one controller, with the ability to take the decision that can unambiguously determine that disablement is the correct decision. If no such controller exists, then there is no control solution.[1] To ameliorate this situation, communication between controllers can occur when the communicated information, in tandem with local observations, allows all the correct control decisions to be taken.

The role of communication has been investigated within the context of synthesizing control strategies for decentralized discrete-event systems. This chapter presents a brief overview of some of the ways in which communication protocols have been incorporated into the decentralized control domain.

Some of the fundamental issues for incorporating communication into decentralized control problems were identified in [12]. In particular, the answers to these questions affect the model design and the subsequent synthesis of control and communication solutions (adapted from [12]):

Laurie Ricker
Department of Mathematics & Computer Science, Mount Allison University,
Sackville, NB Canada
e-mail: lricker@mta.ca

[1] Note that all discussions about decentralized control in this chapter are made with respect to the C&P decentralized architecture of Rudie and Wonham [11]. Analogous statements can be made for the D&A architecture of Yoo and Lafortune [17].

C. Seatzu et al. (Eds.): Control of Discrete-Event Systems, LNCIS 433, pp. 127–146.
springerlink.com

- Why should communication be introduced?
- Who should communicate with whom and when?
- What information should be communicated?
- Who should know what and when?

With inspiration from the work by Witsenhausen [15], the initial proposals for information structures with which the above questions can be addressed, focussed primarily on controllers either keeping track of estimates of sequences, based on observed strings [14, 16], or keeping track of state estimates of the system, based on observed strings [2, 10].

We will examine two approaches to the synthesis of communication protocols: state-based communication and event-occurrence communication. In each case, the communication is presumed to take place with zero delay, i.e., synchronous communication. The literature includes some preliminary examination of communication with non-zero delay [6, 13]; however, the communication protocol under consideration requires no synthesis as every observation of a controller is communicated to all other controllers. For the development of communication protocols beyond synchrony in distributed control architectures, some considerations and possible future directions are presented in [3].

7.2 Notation and Definitions

We assume that the uncontrolled system is described by a finite automaton $G_L = (Q, E, \delta, q_0)$ — with finite state set Q, finite alpabet E, partial transition function $\delta : Q \times E \rightarrow Q$, and initial state q_0 — which generates a regular language L. The corresponding specification automaton $G_K = (Q^K, E, \delta^K, q_0)$, where $Q^K \subseteq Q$ and $\delta^K \subseteq \delta$, generates a language $K \subseteq L$. Alternatively, we denote the transition set of L by T, and the transition set of K by $T^K \subseteq T$. The prefix closure of a language L is defined as follows: $\overline{L} := \{s \in E^* \mid \exists t \in E^* \text{ such that } st \in L\}$. We assume, for the rest of this chapter, that all languages are prefix-closed. To discuss decentralized control for a set of controllers $I = \{1, \ldots, n\}$, the event set E is partitioned into controllable events E_c and uncontrollable events E_{uc}. Similarly, E is partitioned into observable events E_o and unobservable events E_{uo}. To describe events that each decentralized controller $i \in I$ controls and observes, respectively, we use the notation $E_{c,i} \subseteq E_c$ and $E_{o,i} \subseteq E_o$. (The transition set T can similarly be partitioned into T_c and T_o, based on the controllable and observable properties of the transition labels.) We refer to the set of controllers that observe $e \in E_o$ by $I_o(e) := \{i \in I \mid e \in E_{o,i}\}$. Analogously, we refer to the set of controllers that control $e \in E_c$ by $I_c(e) := \{i \in I \mid e \in E_{c,i}\}$.The natural projection describing the partial view of each controller is denoted by $\pi_i : E^* \rightarrow E^*_{o,i}$, for $i \in I$.

A decentralized controller is an automaton $\mathscr{S}_i$, for $i \in I$, (see Fig. 7.1) that has only a partial view of the system behavior. Each controller issues its own local control decision based on its current view of the system and a final control decision is taken by fusing or combining all the local decisions with a particular fusion rule. The rule varies depending on the decentralized architecture in use. We can synthesize decentralized controllers that cooperate to ensure that the supervised system generates exactly the behavior in the specification K if K is controllable (Definition 7.1) and satisfies one of the versions of co-observability (Definition 7.2 or 7.3).

Definition 7.1. *A language $K \subseteq L$ is* controllable *wrt L and E_{uc} if $\overline{K}E_{uc} \cap L \subseteq \overline{K}$.*

Definition 7.2. *A language $K \subseteq L = \overline{L}$ is* unconditionally co-observable *[11] with respect to L, π_i, and E_c if*

$$\forall t \in \overline{K}, \forall \sigma \in E_c : \ t\sigma \in L \setminus \overline{K} \Rightarrow \exists i \in I_c(\sigma) : \ \pi_i^{-1}[\pi_i(t)]\sigma \cap \overline{K} = \emptyset.$$

In this scenario, decentralized controllers take local control decisions based on their partial observations and there must be at least one controller that has sufficient information from its own view of the system to take the correct control decision when the system leaves K (i.e., to disable) for each $\sigma \in E_c$. When $I = \{1\}$, this condition is called *observability*.

Definition 7.3. *A language $K \subseteq L = \overline{L}$ is* conditionally co-observable *[18] with respect to L, π_i, and E_c, if*

$$\forall t \in \overline{K}, \forall \sigma \in E_c : \ t\sigma \in L \setminus \overline{K} \Rightarrow \ \exists i \in I_c(\sigma) : \forall t'\sigma \in \pi_i^{-1}[\pi_i(t)]\sigma \cap \overline{K} \Rightarrow$$
$$\exists j \in I_c(\sigma) : \pi_j^{-1}[\pi_j(t')]\sigma \cap L \subseteq \overline{K}.$$

In this scenario, decentralized controllers that are incapable of taking the correct disable decision can *infer* that there is at least one controller that will correctly know when the system remains in K (i.e., to take an enable decision), leaving the uncertain controller with the opportunity to take a conditional control decision “disable unless another knows to enable”. That is, the enable decision, if taken by one controller, overrides conditional decisions of any of the other controllers.

When K is neither unconditionally nor conditionally co-observable but *is* observable, it may be possible to construct a communication protocol between controllers such that when communication occurs all the correct control decisions are taken.

Definition 7.4. *A language $K \subseteq L = \overline{L}$ is* articulate *wrt L, π_i and E_c if*

$$(\exists t \in \overline{K})(\exists \sigma \in E_c) t\sigma \in L \setminus \overline{K} \Rightarrow \bigcap_{i \in I_c(\sigma)} \pi_i^{-1}[\pi_i(t)]\sigma \cap \overline{K} \neq \emptyset.$$

This property corresponds to a complete absence of information that is available to be inferred from other controllers in $I_c(\sigma)$, thereby leaving communication as the only means of acquiring information from which to take the correct control decisions.

To discuss the various approaches to synthesizing communication protocols, we will refer to the following common terminology. Communicating controllers can be any controller in I. For simplicity, unless otherwise stated, we assume point-to-point communication, i.e., controller i sends a message to controller j, for $i, j \in I$.

Although the content varies from one approach to another, we refer to the finite set of messages involved in a communication protocol by $\Delta := \cup_{i \in I} \Delta_i$, where Δ_i is the set of messages that controller i sends out to other controllers as directed by its communication protocol.

Definition 7.5. *A* communication protocol *for decentralized controller i is* $\mu_{ij}^{!} : L \to \Delta_i \cup \{\varepsilon\}$ *and represents the information that controller i sends to controller j following the occurrence of some sequence* $s \in L$. *The information that controller i receives from controller j, following the occurrence of some sequence* $s \in L$ *is denoted by* $\mu_{ij}^{?} : L \to \Delta_j \cup \{\varepsilon\}$.

When $\mu_{ij}^{?}(s) \neq \varepsilon$, we can construct an alphabet of received information $E_i^{?} \subseteq E_o \setminus E_{o,i}$. To incorporate the received information into the observed information of a controller, we extend the natural projection as follows: $\pi_i^{?} : E^* \to (E_{o,i} \cup E_i^{?})^*$, for $i \in I$. As a consequence, communicating controllers take local control decisions based on $\pi_i^{?}(L)$.

The decentralized architecture that we will assume for this chapter is shown in Fig. 7.1. The decentralized control problem that we will consider is described below.

Problem 7.1. Given regular languages K, L defined over a common alphabet E, controllable events $E_c \subseteq E$, observable events $E_{o,1}, \ldots, E_{o,n} \subseteq E$, and a finite set of messages Δ. We assume that $K \subseteq L \subseteq E^*$ is controllable wrt L, E_{uc}, observable wrt L, π, E_c and articulate wrt L, π_i, E_c. Construct communication protocols $\mathsf{M}_i^{!} = \langle \mu_{i,1}^{!}, \ldots, \mu_{i,j}^{!}, \ldots, \mu_{i,n}^{!} \rangle$ (for $i, j \in I$) such that either

1. K is unconditionally co-observable wrt L, $\pi_i^{?}$ (for $i \in I$), and E_c; OR
2. K is conditionally co-observable wrt L, $\pi_i^{?}$ (for $i \in I$), and E_c. ■

We will examine two main approaches to this problem: when messages are state estimates (i.e., $\Delta_i \subseteq Pwr(Q)$) and when messages are constructed from event occurrences (i.e., $\Delta_i \subseteq E_{o,i}$ or $\Delta_i \subseteq T_{o,i}$). In particular, we consider only approaches which synthesize a communication protocol, as opposed to approaches which assume that part of the input is a set of communications that must be subsequently reduced to satisfy some notion of optimality.

The motivation for introducing communication is independent of the message content or the mode of communication: in the techniques examined here, communication is introduced to eliminate *illegal configurations* from the finite state structure used to determine whether K is co-observable wrt the natural projection that has been updated to include each controller's received messages.

Example 7.1. We will use the following example (from [9]) to illustrate different ways to synthesize a decentralized communication protocol. The joint automaton for G_L and G_K is shown in Fig. 7.2. Here, we assume that $I = \{1,2\}$, $E = \{\mathsf{a},\mathsf{b},\mathsf{c},\sigma\}$ such that $E_{o,1} = \{\mathsf{a},\mathsf{c},\sigma\}$ and $E_{o,2} = \{\mathsf{b},\sigma\}$. Further, $E_c = \{\sigma\}$, where $E_{c,1} = \{\sigma\}$ and $E_{c,2} = \emptyset$. Note that K is neither unconditionally nor conditionally co-observable. Since $I_c(\sigma) = \{1\}$, we just need to check the co-observability definitions wrt controller 1. For the former case, let $t = \mathsf{ac}$ then $\pi_1^{-1}[\pi_1(t)]\sigma = \{\mathsf{ac}\sigma, \mathsf{bac}\sigma\}$. To satisfy

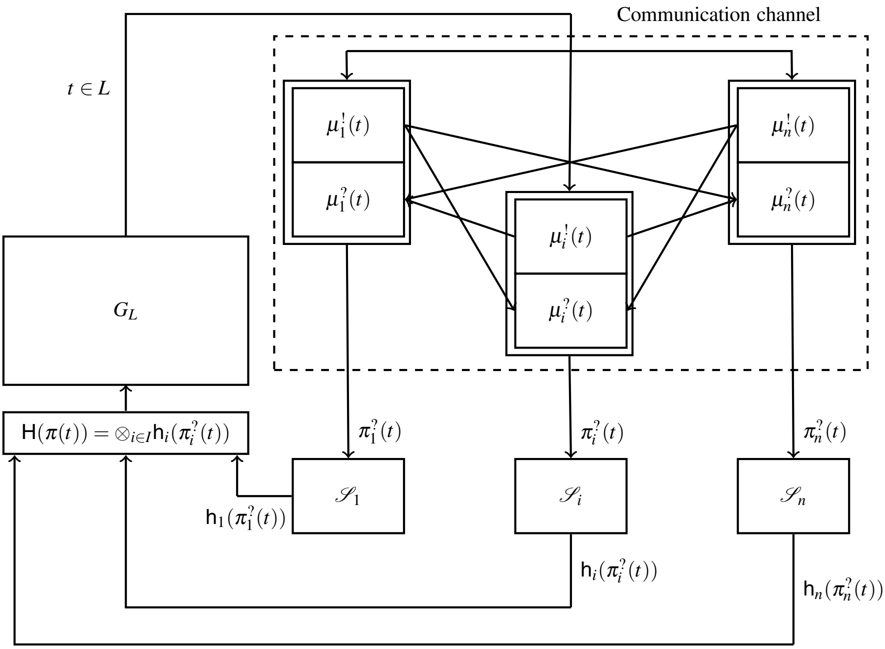

Fig. 7.1 Decentralized architecture for communication and control, where decentralized controllers $\mathscr{S}_i$ (for $i \in I$) make control decisions $\mathsf{h}_i(t)$ that are combined by a fusion rule (denoted here by $\otimes$) to produce a global control decision $\mathsf{H}(t)$ to either enable or disable events after observing sequence t generated by G_L and receiving communication from other controllers

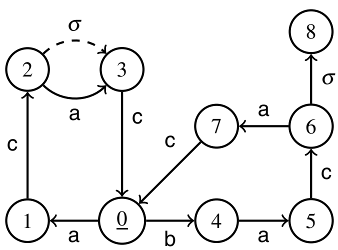

Fig. 7.2 Automata for joint G_L (all transitions) and G_K (only solid-line transitions). Initial state is underlined

unconditional co-observability, $\{\mathsf{ac}\sigma, \mathsf{bac}\sigma\} \cap \overline{K}$ must be empty; however, the intersection is $\{\mathsf{bac}\sigma\}$. It is trivial to show that K is not conditionally co-observable. It suffices to note that $I_c(\sigma) = \{1\}$ and thus there is no other $j \in I_c(\sigma)$ to take correctly the enable decisions regarding σ. It remains to show that K is articulate.

Again, this follows in a straightforward manner from K not being unconditionally co-observable. Let $t = \mathsf{ac}$. Then $\cap_{i \in I_c(\sigma)} \pi_i^{-1}[\pi_i(\mathsf{ac})]\sigma \cap \overline{K} = \{\mathsf{bac}\sigma\} \neq \emptyset$. ■

We synthesize communication protocols using this example as illustrated by two state-based approaches described in [2] and [9] and the event-occurrence approach introduced in [8].

7.3 State-Based Communication Protocols

When the communication protocol features messages that consist of local *information states* (i.e., local state estimates), there are two main synthesis approaches that have been proposed. The central idea is straightforward: build a finite structure that contains illegal configurations (i.e., states that correspond to violations of co-observability). Identify states in the structure where communication would eliminate the illegal configurations (i.e., refine the structure so that such states are no longer defined). The first approach requires an iterative update of the structure to reflect the effect of each identified communication [2, 10], ideally converging to a structure free of illegal configurations, whereas the second approach uses a much larger structure which, by construction, takes into account the effect of communication at all states where the reception of a message improves the local state estimates of the receiver and communications are chosen in such a way as to make illegal configurations unreachable [9].

We begin with the communication strategy of [2], where we have taken the liberty to adjust their notation for ease of comparison to the other models. To calculate the information state for controller i at state $q \in Q$ wrt $E_{o,i}$, we use the algorithm for subset construction [7], which is based on the notion of ε-closure.

Definition 7.6. *The* ε-closure$_i(X)$*, where* $X \subseteq Pwr(Q)$ *and* $i \in I$*, is the least set such that*

(i) $X \subseteq \varepsilon\text{-}closure_i(X)$;

(ii) $\forall x \in \varepsilon\text{-}closure_i(X), \forall \sigma \notin E_{o,i}, (\delta(x,\sigma) = x' \Rightarrow x' \in \varepsilon\text{-}closure_i(X))$.

When considering communication of information states, we build a structure $\mathscr{V}_0$ to monitor the progress of automaton G_L and each controller's state-based partial view of G_L. To describe the transition function for $\mathscr{V}_0$, we also need to calculate the set of states that can be reached in one step via a transition of an event σ from a given set of states $X \subseteq Pwr(Q)$:

$$step_\sigma(X) = \{x' \in Q \mid \exists x \in X \text{ such that } \delta(x,\sigma) = x'\}.$$

Thus, in $\mathscr{V}_0$, a transition from $(q, X_1, \ldots, X_n, \sigma')$ to $(q', X_1', \ldots, X_n', \sigma)$ via transition label $\sigma \in E$ is defined as follows: $\delta(q,\sigma) = q$; if $\sigma \in E_{o,i}$ then $X_i' = \text{step}_\sigma(\varepsilon\text{-closure}_i(X_i))$, otherwise $X_i' = X_i$.

There are several characteristics of $\mathscr{V}_0$ that set it apart from subsequent structures described in this chapter. First, because communication is synchronous, for a given state in the state set of $\mathscr{V}_0$, say $(q, X_1, \ldots, X_n, \sigma)$, the local state estimates for controller i, namely X_i, do not include the ε-closure$_i$ of the incoming observation σ. As a consequence, the initial state is always $(0, \{0\}, ..., \{0\}, \varepsilon)$. Second, the states include the incoming transition σ to make clear which event observation triggers a communication.

Formally, $\mathscr{V}_0 = (X, E, \delta_{\mathscr{V}}, x_0, B_{uncond}, B_{cond})$, where the finite state set $X \subseteq Q \times Pwr(Q)^n \times E$; the transition function is $\delta_{\mathscr{V}} : X \times E \to X$; the initial state is $x_0 = (q_0, \{q_0\}, \ldots, \{q_0\}, \varepsilon)$; $B_{uncond} \subseteq X$ is a set of *illegal configurations* that correspond to violations of unconditional co-observability, where $B_{uncond} = \{(q, X_1, \ldots, X_n, \gamma) \in X \mid \exists \sigma \in E_c$ such that $\delta(q, \sigma) \in Q \setminus Q_K$ and for all $i \in I_c(\sigma)$ $\exists q' \in X_i$ such that $\delta(q', \sigma) \in Q_K\}$; and $B_{cond} \subseteq X$ is an additional set of illegal configurations that correspond to violations of conditional co-observability, where $B_{cond} = \{(q, X_1, \ldots, X_n, \gamma) \in X \mid \exists \sigma \in E_c$ such that $\delta(q, \sigma) \in Q_K$ and for all $i \in I_c(\sigma)$ $\exists q' \in X_i$ such that $\delta(q', \sigma) \in Q \setminus Q_K\}$.

For all three communication protocol synthesis approaches discussed here, the common goal is to eliminate illegal configurations so that the resulting system satisfies one of the notions of co-observability. Proofs of these theorems (in various forms) can be found in the original papers [2, 8, 9].

Theorem 7.1. *$B_{uncond} = \emptyset \Leftrightarrow K$ is unconditionally co-observable wrt L, $\pi_i^?$, and E_c.*

This theorem can be extended to include conditional co-observability, even though only unconditional co-observability is considered in [2, 8, 9].

Theorem 7.2. *$B_{uncond} \neq \emptyset$ and $B_{cond} = \emptyset \Leftrightarrow K$ is conditionally co-observable wrt L, $\pi_i^?$, and E_c.*

The structure $\mathscr{V}_0$ for Example 7.1 is shown in Fig. 7.3. Although $B_{cond} = \emptyset$, there are two illegal configurations: $B_{uncond} = \{(2, \{2,6\}, \{0\}, \mathsf{c}), ((2, \{2,6\}, \{4\}, \mathsf{c}))\}$. These states are identified by their double box outline in Fig. 7.3.

The process for transforming $\mathscr{V}_0$ into a structure that contains no illegal configurations begins by identifying *communication states* that will lead to the elimination of illegal configurations by refining a controller's set of local state estimates after taking into account communicated information. We first define the set of states $\Omega \subseteq Q$ from which states in $Q \setminus Q_K$ are reachable. Let $\Omega = \{q' \in Q \mid \exists s_1, s_2 \in E^*$ where $\delta(q_0, s_1) = q'$ and $\delta(q', s_2) \in Q \setminus Q_K\}$.

Definition 7.7. *A state $x = (q, X_1, \ldots, X_n, \sigma)$ is a* communication state *if*

$$\exists i \in I_o(\sigma) \text{ s.t. } (X_i \cap (\cap_{j \in I \setminus \{i\}} \varepsilon\text{-}closure_j(X_j))) \setminus \Omega = \emptyset.$$

In [2] communication occurs "as late as possible" and thus the search for a communication state begins at each $b \in B_{uncond}$. (Note that conditional control decisions came about after [2] appeared; however, it is straightforward to extend the model to detect violations of conditional co-observability.) If b is not suitable, then a backwards reachability is performed until a communication state is found. The proof of

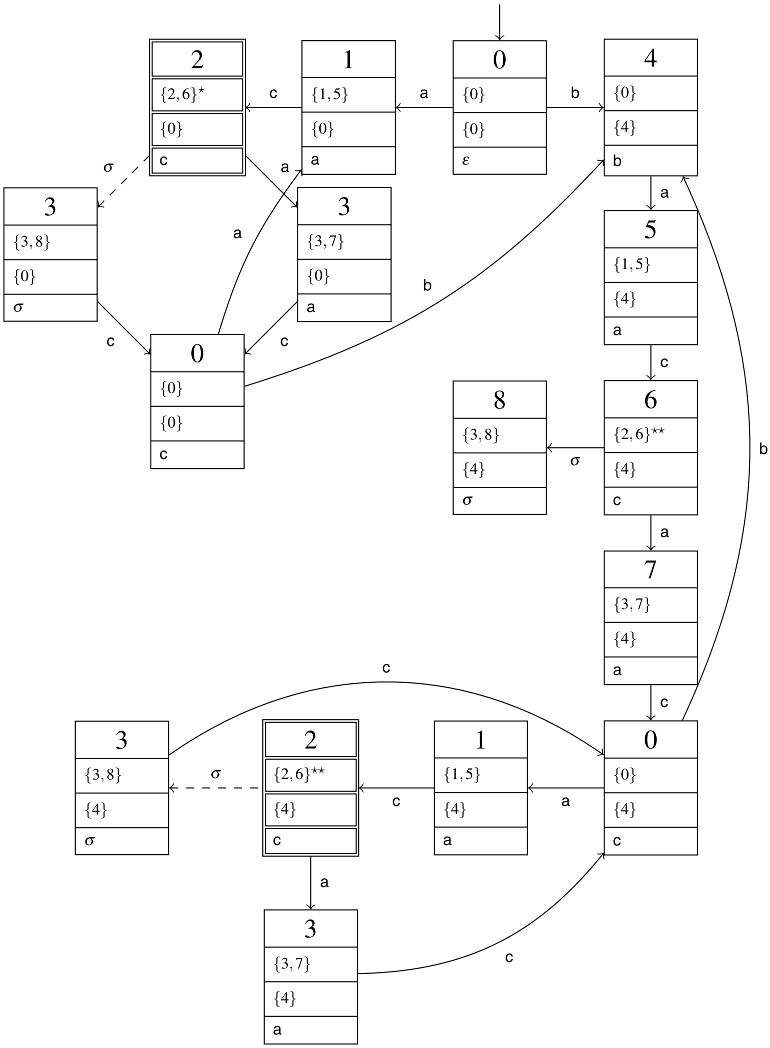

Fig. 7.3 The automaton $\mathscr{V}_0$ constructed from G_L and G_K in Fig. 7.2. Illegal configurations are indicated by states with a double box. State where communication is initiated to resolve illegal configuration $\langle 2, \{2,6\}, \{0\}, \mathsf{c}\rangle$ according to [2] is indicated by a $\star$. States where communication occurs to satisfy feasibility are indicated by a $\star\star$

guaranteed existence of a communication state for each $b \in B_{uncond}$ can be found in [2].

At a communication state, communication is initiated by a controller that has just observed the most recently occurred event γ, as indicated by the event component of the communication state. For simplicity, the initiator is fixed to be one of the controllers in $I_o(\gamma)$. The initiator then broadcasts its information state to the others, who respond by sending the initiator their information state. The communication protocol for the sending and receiving of messages at a communication state $x \in X$, for $s\gamma \in L$ such that $\delta_{\mathscr{V}}(x_0, s\gamma) = x$, is defined as follows:

$$\begin{aligned} \text{for initiator } i \in I_o(\gamma), \forall j \in I \setminus \{i\}, \quad & \mu^!_{ij}(s\gamma) = \mu^?_{ji}(s\gamma) = X_i, \\ & \mu^!_{ji}(s\gamma) = \mu^?_{ij}(s\gamma) = \varepsilon\text{-closure}_j(X_j). \end{aligned}$$

The communication state in $\mathscr{V}_0$ wrt illegal configuration $(2, \{2,6\}, \{0\}, \mathsf{c})$ for initiator controller 1 is $(2, \{2,6\}, \{0\}, \mathsf{c})$ itself. We can now specify the communication protocol for $\mathscr{V}_0$: $\mu^!_{12}(\mathsf{ac(acac)^*}) = \{2,6\}$, $\mu^!_{21}(\mathsf{ac(acac)^*}) = \{0,1,2,3\}$.

It must be the case that communication occurs at all states that are indistinguishable to the initiator of the communication.

Definition 7.8. *Two states are* indistinguishable *to initiator i if the incoming event is identical and the local state estimate is the same:*

$$(q, X_1, \ldots, X_i, \ldots X_n, \gamma) \sim_i (q', X'_1, \ldots, X'_i, \ldots X'_n, \gamma') \Leftrightarrow X_i = X'_i \text{ and } \gamma = \gamma' \in E_{o,i}.$$

Thus, in addition to incorporating the effect of communication at a communication state, one must also add the effect of communication at states that the initiator finds indistinguishable from the communication state. This is called a *feasible* communication state.

There are two feasible communication states in $\mathscr{V}_0$ wrt communication state $(2, \{2,6\}, \{0\}, \mathsf{c})$, namely $(6, \{2,6\}, \{4\}, \mathsf{c})$ and $(2, \{2,6\}, \{4\}, \mathsf{c})$. Extending the communication protocol for $\mathscr{V}_0$, we have $\mu^!_{12}(\mathsf{(acac)^*bac((acbac)^*(acacbac)^*)^*}) = \{2,6\}$ and $\mu^!_{12}(\mathsf{ac(acac)^+}) = \{2,6\}$; $\mu^!_{21}(\mathsf{(acac)^*bac((acbac)^*(acacbac)^*)^*}) = \{0..8\}$ and $\mu^!_{21}(\mathsf{ac(acac)^+}) = \{0..8\}$.

When the controllers receive information after the occurrence of $s\gamma \in L$, they update their local state estimates according to

$$(\forall i \in I) \quad X_i = X_i \cap \mu^?_{i,1}(s\gamma) \cap \mu^?_{i,2}(s\gamma) \cap \ldots\ \mu^?_{i,i-1}(s\gamma) \cap \mu^?_{i,i+1}(s\gamma) \cap \ldots \mu^?_{i,n}(s\gamma).$$

This gives rise to the construction of a new version of $\mathscr{V}_0$, which we denote by $\mathscr{V}_1$, where the effect of the communication is calculated and then propagated through the calculation of a new state set and transition function, as well as an updated B_{uncond}.

Taking into account the communication performed at the communication states identified in $\mathscr{V}_0$, the next iteration $\mathscr{V}_1$ is shown in Fig. 7.4. In keeping with the notational conventions in [2], the communication state and the transformed state after

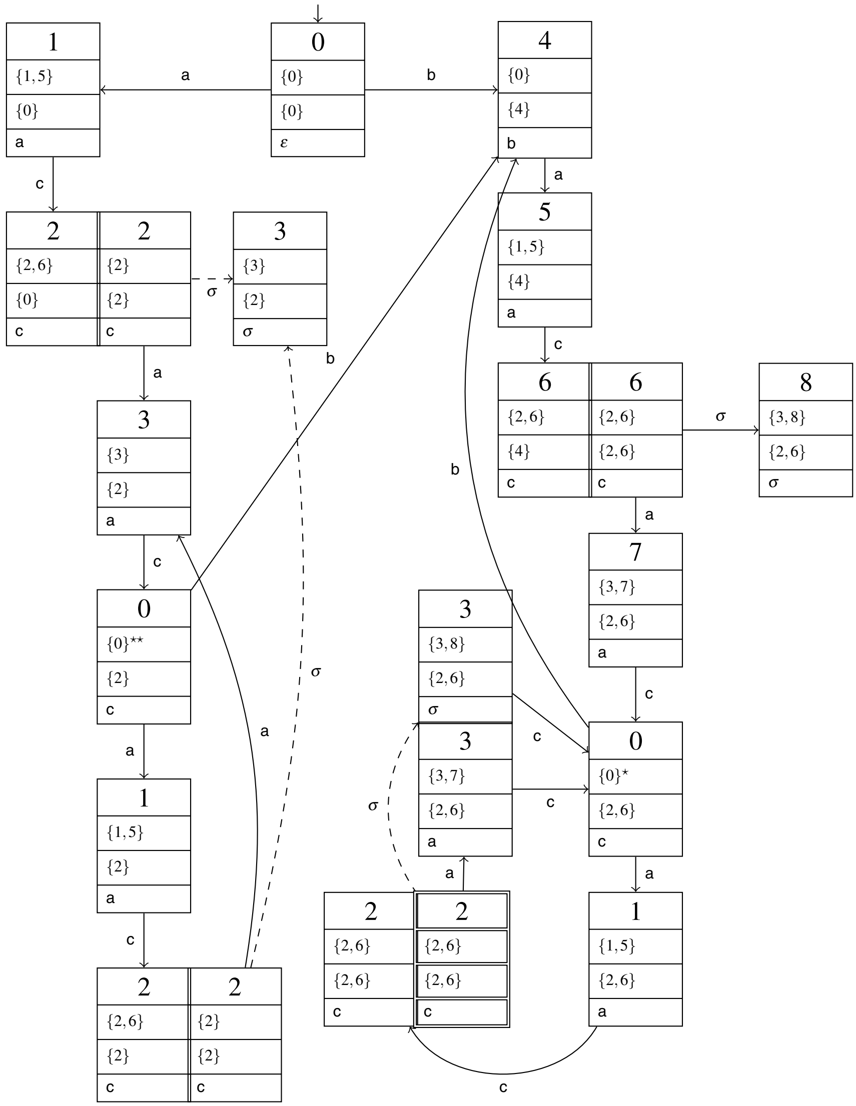

Fig. 7.4 The automaton $\mathscr{V}_1$ after the effects of the communication to eliminate illegal configuration $\langle 2, \{2,6\}, \{0\}, \mathsf{c}\rangle$ is taken into consideration. Additional communication states identified during this iteration are noted by a $\star$ whereas feasible communication states are indicated by $\star\star$

communication are shown as double states. Subsequent states in $\mathscr{V}_1$ are calculated based on the post-communication state (the rightmost box of a double state). Note that $\mathscr{V}_1$ still contains an illegal configuration. Thus, we must identify an additional communication state by examining, states that precede the illegal configuration. It can be verified that it is necessary to backtrack two states, to state $(0,\{0\},\{2,6\},\mathsf{c})$, to identify another communication state, where the initiator is controller 1. The corresponding feasible communication state is $(0,\{0\},\{2\},\mathsf{c})$.

The final iteration, $\mathscr{V}_2$, is shown in Fig. 7.5. Communication has resolved all occurrences of illegal configurations. The communication protocol for the initiator controller 1 wrt $\mathscr{V}_2$ is $\mu^{!}_{12}(\mathsf{ac}) = \{2,6\}$, $\mu^{!}_{12}(((\mathsf{bacac})^*(\mathsf{acac})^+)^*) = \{0\}$, $\mu^{!}_{12}((\mathsf{acac})^*\mathsf{bac}((\mathsf{acbac})^*(\mathsf{acacbac})^*)^*) = \{6\}$, $\mu^{!}_{12}(((\mathsf{acac})^*(\mathsf{bacac})^+)^*) = \{0\}$. The proof of convergence of this algorithm (i.e., that the iteration of $\mathscr{V}_0$ will terminate in a finite number of steps) is presented in [2].

Synthesizing communication protocols using the results of [2] assumes that communication to eliminate an illegal configuration occurs "as late as possible" and only along sequences that eventually leave K. To explore a wider range of communication opportunities, a different model was proposed in [9]. This model, denoted by $\mathscr{W}$, is more complex because, by construction, it explicitly contains communication and subsequent effect of communication on the receiver's information state, whenever communication leads to the introduction of new information for a controller. As a result, $\mathscr{W}$ is built only once and requires no further iterations; however, in the worst case, it is significantly larger than $\mathscr{V}_0$. Other differences between the two models include the definition of an information state (the trailing incoming event is no longer needed in states of $\mathscr{W}$) and communication occurs between a single sender and a single receiver as a point-to-point communication and not as a two-way broadcast between the initiator and the other controllers.

One of the most significant differences between the two models is the introduction of three different state types: $\circ$ represents an *update* state, $\square$ represents a *configuration* state, and $\diamond$ represents a *communication* state. An update state reflects the changes to information states as a result of a communication from sender i to receiver j, thus avoiding the need for subsequent iterations of $\mathscr{W}$. A configuration state is equivalent to a state of $\mathscr{V}_0$ without the trailing incoming event. A communication state in the context of $\mathscr{W}$ encapsulates the information states just prior to a message being sent from sender i to receiver j.

The second significant difference is the introduction of three different kinds of transition labels: an *update mapping*, an event occurrence, and a *communication directive*. An update mapping Υ provides details of the mechanics of communication. In particular, given the event triggering the communication, the identity of the sender, and the sender's message (i.e., its local information state without ε-closure), the update mapping indicates the identity of the receiver. For example, a transition label of $\Upsilon(b,\mathsf{2},\{4\}) = \mathsf{1}$ means that at its information state $\{4\}$, the sender, controller $\mathsf{2}$, will send information regarding the occurrence of event b (as encoded by its information state $\{4\}$) to the receiver, controller $\mathsf{1}$. A communication directive

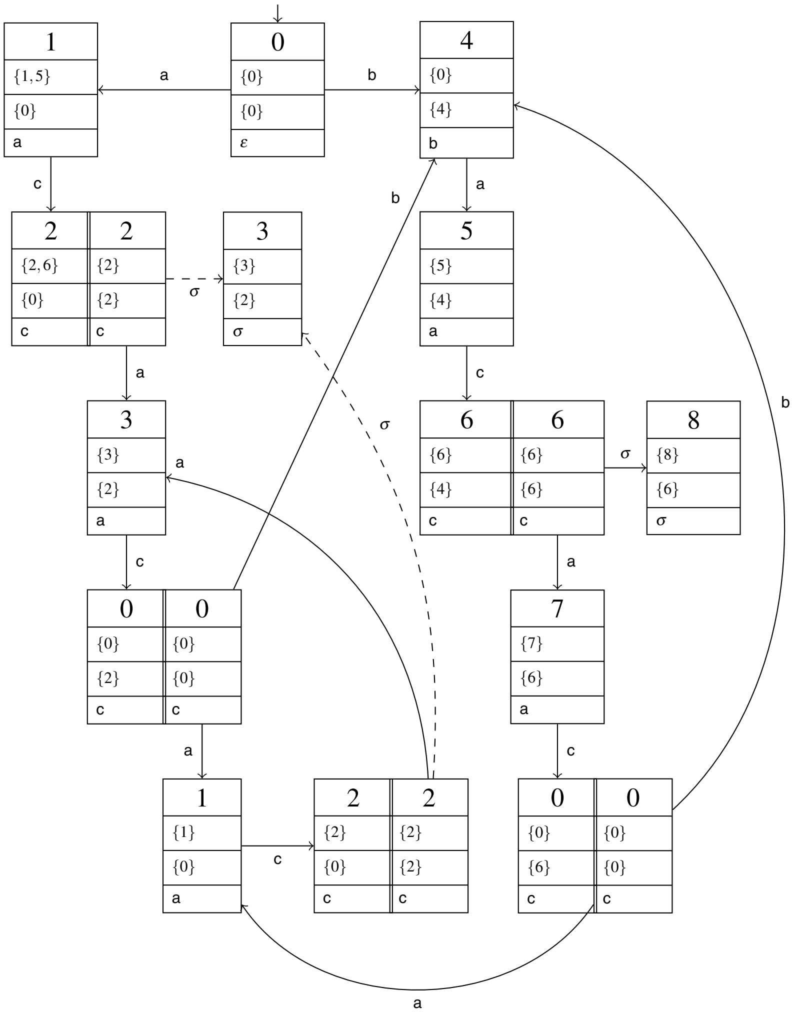

Fig. 7.5 The automaton $\mathscr{V}_2$ after the effects of the communication to eliminate illegal configuration $\langle 2, \{2,6\}, \{2,6\}, \mathsf{c} \rangle$ is taken into consideration

$\varPhi$ simply indicates the identity of the sender and the receiver and is used to update the information states after a communication occurs.

Formally, $\mathscr{W} = (R, E, \delta_{\mathscr{W}}, r_0, \varPhi, \varUpsilon, B_{uncond}, B_{cond})$, where the finite state set $R \subseteq (\{\circ, \square\} \cup (\{\diamond\} \times E)) \times Q \times Pwr(Q)^n$; the transition set is $\delta_{\mathscr{W}} \subseteq R \times (E \cup \varPhi \cup \varUpsilon) \times R$; the initial state is $r_0 = \langle \circ, q_0, \{q_0\}, \ldots, \{q_0\} \rangle \in R$; the communication directive is $\varPhi \subseteq (I \cup \emptyset) \times (I \cup \emptyset)$; the update mapping is $\varUpsilon : E \times I \times Pwr(Q) \to I \cup \emptyset$; $B_{uncond} \subseteq X$ and $B_{cond} \subseteq X$ are sets of illegal configurations, as defined previously for structure $\mathscr{V}_0$.

Before continuing with a closer examination of the different transition and state types, we first update the definition of ε-closure in light of received information. We need to calculate the set of states that are reachable via unobservable events, with the exception of those unobservable events that were just received in a communication.

Definition 7.9. *The* $\boldsymbol{\varepsilon}$***-closure***$_{i,u}(\boldsymbol{X})$*, where* $X \subseteq Pwr(Q)$*,* $i \in I$ *and* $u \in I \cup \emptyset$ *is the least set such that*

(i) $X \subseteq \varepsilon\text{-}closure_{i,u}(X)$;

(ii) $\forall x \in \varepsilon\text{-}closure_{i,u}(X), \forall \sigma \notin E_{o,i}, (\forall j \in I, \exists y \subseteq Pwr(Q), i \notin u = \varUpsilon(\sigma, j, y))$ *and* $(\delta(x, \sigma) = x' \Rightarrow x' \in \varepsilon\text{-}closure_{i,u}(X))$.

The three types of transitions—communications, updates, and a move of the system—are now described in more detail.

1. An update transition goes from an update state to a configuration state:
$$\langle \circ, q, X_1, \ldots, X_n \rangle \xrightarrow{u} \langle \square, q, X_1', \ldots, X_n' \rangle,$$
where $u \in I \cup \emptyset, \forall i \in I \; X_i' = \varepsilon\text{-closure}_{i,u}(X_i)$.
2. A system transition goes from a configuration state to a communication state:
$$\langle \square, q, X_1, \ldots, X_n \rangle \xrightarrow{\sigma} \langle (\diamond, \sigma), q', X_1', \ldots, X_n' \rangle,$$
where $\delta(q, \sigma) = q'$, and $\forall i \in I \; (\sigma \in E_{o,i} \Rightarrow X_i' = step_\sigma(X_i))$ and $(\sigma \notin E_{o,i} \Rightarrow X_i' = X_i)$.
3. A communication transition goes from a communication state to an update state:
$$\langle (\diamond, \sigma), q, X_1, \ldots, X_n \rangle \xrightarrow{\phi} \langle \circ, q, X_1', \ldots, X_n' \rangle,$$
where $\phi = (i, j) \in \varPhi$ such that $\sigma \notin E_{o,j}$ $(X_j' = step_\sigma(X_j))$ and $(\forall i \in I \setminus \{j\}, X_i' = X_i)$.

Update transitions are unobservable to all $i \in I$. An update is merely an automatic consequence of a communication. A communication transition is observable to the sender and to the receiver j. We abuse notation and define the set of communications observable to controller i by $\Phi_i := \{\phi \in \Phi \mid (\phi = (i,j),\ \text{for}\ i \neq j)\ \text{and}\ (\phi = (k,i),\ \text{for}\ i \neq k)\}$.

To build a communication protocol for $\mathscr{W}$ we choose an update mapping at an update state such that illegal configurations are unreachable. Having chosen a specific update mapping transition (i.e., either a specific sender or no communication at all), it is necessary to propagate this choice to the relevant communication state, making the feasible choice for the communication directive. In lieu of pruning the transitions not chosen at update states and at communication states, it may be simpler to think of the update mapping and communication directives as being controllable events only for the senders involved. Then by taking a choice of a particular update mapping transition, the sender enables the chosen transition and simply disables all others at that particular update state. The sender then follows a similar strategy at communication states. Subsequently, the communication protocol $\mathsf{M}_i^!$ for each $i \in I$ consists of all enabled communication directives.

Although, the construction of $\mathscr{W}$ assumes that there is a general pattern of update state, followed by configuration state, followed by communication state, for every transition in G_K, we can substantially reduce the size of the model as follows. For a given violation of conditional or unconditional co-observability, we will consider update transitions and communication states for only those transitions that correspond to events in $E_{o,j} \setminus E_{o,i}$, for $j \in I \setminus \{i\}$ and $i \in I_c(\sigma)$.

Figure 7.6 shows $\mathscr{W}$ for Example 7.1. Here, $B_{uncond} = \{(\square, 2,\ \{2,6\},\ \{0,1,2,3\}), (\square, 2,\ \{2,6\},\ \{0..8\})\}$, and, as before, are indicated by a double box. By enabling the update mapping transition $\Upsilon(\mathsf{b}, \mathsf{2}, \{4\}) = \mathsf{1}$ at update state $(\circ, 0, \{0\}, \{0\})$, the illegal configuration $(\square, 2,\ \{2,6\},\ \{0,1,2,3\})$ is no longer reachable. As a consequence, at communication state $(\diamond, \mathbf{b}, 4, \{0\}, \{4\})$, controller 2 must choose to communicate to controller 1, as indicated by the mapping transition, thereby enabling communication directive $(2,1)$.

To satisfy feasibility, controller 2 must also choose to communicate to controller 1 at communication state $(\diamond, \mathbf{b}, 4, \{0,4\}, \{4\})$. To ensure that all other communication directives are consistent with these two communication directives, i.e., when controller 2 has a local state of $\{4\}$ after the occurrence of event b. Thus, controller 2 must enable any transition $\Upsilon(\mathsf{b}, \mathsf{2}, \{4\}) = \mathsf{1}$ at any other update state that has such an outgoing transition label. Hence, $\Upsilon(\mathsf{b}, \mathsf{2}, \{4\}) = \mathsf{1}$ is enabled at update states $(\circ, 4, \{4\}, \{4\})$, $(\circ, 0, \{0\}, \{0..8\})$, $(\circ, 4, \{0\}, \{4\})$ and $(\circ, 4, \{0,4\}, \{4\})$; however, by enabling transition $(2,1)$ at each of the communication states, the update states $(\circ, 4, \{0\}, \{4\})$ and $(\circ, 4, \{0,4\}, \{4\})$ become unreachable, even though their outgoing transition $\Upsilon(\mathsf{b}, \mathsf{2}, \{4\}) = \mathsf{1}$ is enabled.

Thus, for Example 7.1, $\mu_{12}^!(L) = \emptyset$ and $\mu_{21}^!(((\mathsf{acac})^*\mathsf{b})^+) = \{4\}$, and for all $s \in L \setminus ((\mathsf{acac})^*\mathsf{b})^+$, $\mu_{21}^!(s) = \emptyset$, where we interpret the message $\emptyset$ to correspond to silence. The behavior of $\mathscr{W}$ operating under communication protocol $\mathsf{M}^!$ is shown in Fig. 7.6 by the collection of transitions in bold.

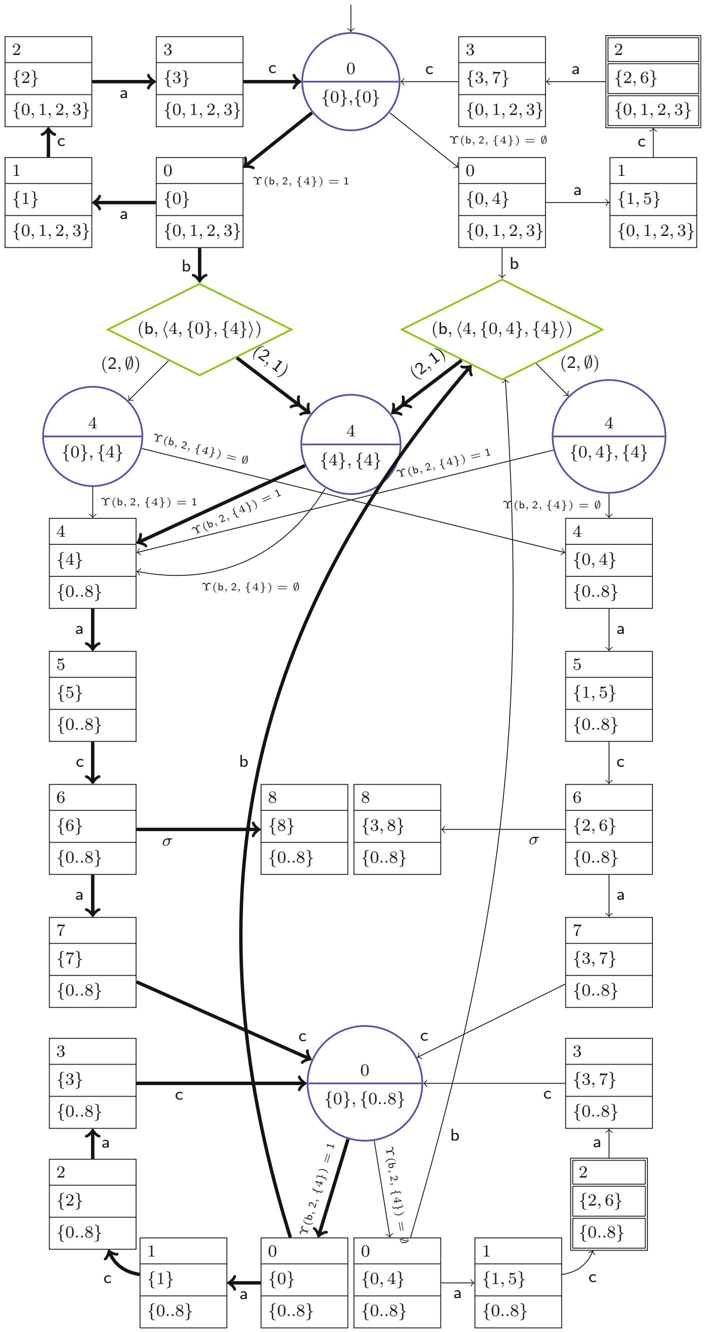

Fig. 7.6 The structure $\mathscr{W}$ (all transitions) for the automaton in Fig. 7.2. The collection of transitions in bold represent $\mathscr{W}$ operating under communication protocol $\mathsf{M}^!$. The communication protocol $\mathsf{M}^!$ is indicated by the collection of transitions with double arrowheads

7.4 Event-Based Communication Protocols

The strategy for synthesizing communication protocols based on the communication of event occurrences to distinguish sequences in $L \setminus K$ from those in K is significantly different from the information state strategy. Like $\mathscr{V}_0$ and $\mathscr{W}$, the structure $\mathscr{U}$ described below is finite-state. Like $\mathscr{V}_0$ there is only one type of state and one type of transition label; however, it is this set of transition labels that distinguish $\mathscr{U}$ from the other models.

The alphabet of $\mathscr{U}$ is based on vector labels of [1]. To begin the construction of the alphabet, we use an augmented set of controllers $I_0 = \{0\} \cup I$, where 0 represents the system. As well, we use an augmented alphabet $E^{\varepsilon} = \{\varepsilon\} \cup E$. A label $\ell : I_0 \longrightarrow E^{\varepsilon}$ is a mapping from each controller to either an event from E or ε. We will sometimes refer to the ith element of label $\ell = \langle \ell(0), \ell(1), \ldots, \ell(n) \rangle$ by $\ell(i)$, where $i \in I_0$. The empty label is $\langle \varepsilon, \ldots, \varepsilon \rangle$, i.e., for all $i \in I_0, \ell(i) = \varepsilon$.

Labels are generated from a given finite set of *atoms* denoted by A. The set of atoms is defined as the union of the following sets of labels, based on the observability of events in E:

- $\sigma \in E_{uo,i} \Rightarrow \ell(i) = \sigma$ and $\forall j \in I_0 \setminus \{i\}, \ell(j) = \varepsilon$; and
- $\forall i \in I_0$ s.t. $\sigma \in E_{o,i} \Rightarrow \ell(i) = \sigma$, otherwise $\ell(i) = \varepsilon$.

The set of atoms for Example 7.1 is $A = \{\langle \mathsf{a}, \mathsf{a}, \varepsilon \rangle, \langle \varepsilon, \varepsilon, \mathsf{a} \rangle, \langle \mathsf{b}, \varepsilon, \mathsf{b} \rangle, \langle \varepsilon, \mathsf{b}, \varepsilon \rangle, \langle \mathsf{c}, \mathsf{c}, \varepsilon \rangle, \langle \varepsilon, \varepsilon, \mathsf{c} \rangle\}$. We define $A^{\varepsilon} := A \cup \{\langle \varepsilon, \ldots, \varepsilon \rangle\}$.

We require the following three properties of labels.

Definition 7.10. *Two labels ℓ_1, ℓ_2 are* compatible*, denoted by $\ell_1 \uparrow \ell_2$, iff $\forall i \in I_0$, $\ell_1(i) = \varepsilon$ or $\ell_2(i) = \varepsilon$ or $\ell_1(i) = \ell_2(i)$.*

Definition 7.11. *The* least upper bound *of two compatible labels, denoted by $\ell_1 \vee \ell_2$, is computed as follows.*

$$\forall i \in I_0, (\ell_1 \vee \ell_2)(i) = \begin{cases} \ell_1(i), & \text{if } \ell_1(i) \neq \varepsilon; \\ \ell_2(i), & \text{if } \ell_2(i) \neq \varepsilon; \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Definition 7.12. *Two labels, ℓ_1 and ℓ_2, are* independent*, denoted $\ell_1 | \ell_2$, iff $\forall i \in I_0$ $\ell_1(i) = \varepsilon$ or $\ell_2(i) = \varepsilon$.*

The alphabet for $\mathscr{U}$ is the least upper bound of compatible elements in A^{ε}:

$$\mathbb{A} := \{a \vee \ell \mid a \in A^{\varepsilon},\ \ell \in \mathbb{A} \text{ and } a \uparrow \ell\}.$$

To construct $\mathscr{U}$, we build an augmented version of G_L and G_K as follows. Update their alphabets to be $E \cup \{\varepsilon\}$ and add a self-loop of ε at each state of Q_L and Q_K. We refer to the augmented automaton as G_L^{ε} and G_K^{ε}. We replace δ and δ_K with T and T^K, the transition sets for G_L and G_K, respectively. Finally, we add a set of special transitions that correspond to whether or not the transition is part of $L \setminus K$ or K: in G_K

we add $F_K := \{(q,e,q') \in T \mid \exists s \in \overline{K}$ such that $se \in \overline{K}$ and $(q_0,s,q)(q,e,q') \in T_K^*\}$ and in G_L we add $F_L := T \setminus F_K$. We continue by composing G_L^{ε} with n copies of G_K^{ε}, one for each decentralized controller:

$$\mathscr{U} = G_L^{\varepsilon} \times \prod_{i=1}^{n} G_K^{\varepsilon} = (X, \mathbb{A}, T^{\mathscr{U}}, x_0, B_{uncond}, B_{cond}),$$

where $(x(0),x(1),\ldots,x(n)) \in X \subseteq (Q)^{n+1}$; $\mathbb{A}$ is a finite alphabet of labels; the transition relation $T^{\mathscr{U}}$ is defined according to: $x \xrightarrow{\ell} x' \in T^{\mathscr{U}}$ iff $\ell \in \mathbb{A}$ and $\forall i \in I_0$, $x(i) \xrightarrow{\ell(i)} x'(i) \in T$; the initial state $x_0 = (q_0)^{n+1}$; and $B_{uncond}, B_{cond} \subseteq T^{\mathscr{U}}$, where the set of illegal configurations wrt transitions that correspond to violations of unconditional co-observability is $B_{uncond} := \{x \xrightarrow{\ell} x' \in T^{\mathscr{U}} \mid x(0) \xrightarrow{\ell(0)} x'(0) \in F_L$ and $(\forall i \in I_c(\ell(0)))x(i) \xrightarrow{\ell(i)} x'(i) \in F_K\}$ and the set of illegal configurations wrt transitions that correspond to violations of conditional co-observability is $B_{cond} := \{x \xrightarrow{\ell} x' \in T^{\mathscr{U}} \mid |I_c(\ell(0))| > 1$ and $x(0) \xrightarrow{\ell(0)} x'(0) \in F_K$ and $(\forall i \in I_c(\ell(0)))x(i) \xrightarrow{\ell(i)} x'(i) \in F_L\}$.

The resulting $\mathscr{U}$ structure for Example 7.1 has 1513 states, 54 labels, and 3929 transitions. As was the case for the corresponding $\mathscr{V}_0$ and $\mathscr{W}$, in $\mathscr{U}$, $B_{cond} = \emptyset$; however, $B_{uncond} = \{\langle(1,5,2),\langle\sigma,\sigma,\sigma\rangle,(2,6,2)\rangle,\langle(1,5,6),\langle\sigma,\sigma,\sigma\rangle,(2,2,6)\rangle\}$. The portion of $\mathscr{U}$ containing the transitions in B_{uncond} is shown in Fig. 7.7.

A communication protocol $M^!$ is synthesized using $\mathscr{U}$ by choosing transitions representing potential communications. We rely on an architectural property of $\mathscr{U}$ that provides a straightforward means of identifying communication transitions:

Definition 7.13. (Adapted from [4].) *The* diamond/step property *holds at* $x_1 \in X$ *if there exist labels* $\ell_1, \ell_2 \in \mathbb{A}$ *that satisfy the following axioms:*

(i) $x_1 \xrightarrow{\ell_1} x_2, x_1 \xrightarrow{\ell_2} x_3 \in T^{\mathscr{U}}$ and $\ell_1|\ell_2 \Rightarrow x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T^{\mathscr{U}}$ [Forward step];

(ii) $x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T^{\mathscr{U}}$ and $\ell_1|\ell_2 \Rightarrow x_1 \xrightarrow{\ell_1} x_2, x_2 \xrightarrow{\ell_2} x_4 \in T^{\mathscr{U}}$ [Step decomposition];

(iii) $x_1 \xrightarrow{\ell_1} x_2, x_2 \xrightarrow{\ell_2} x_4 \in T^{\mathscr{U}}$ and $\ell_1|\ell_2 \Rightarrow x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T^{\mathscr{U}}$ [Independent step].

Definition 7.14. *A* communication transition *for* $(b,\ell,b') \in B_{uncond} \cup B_{cond}$ *wrt* $i \in I_c(\ell(0))$ *is a transition* $x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T^{\mathscr{U}}$ *such that* x_1, ℓ_1, *and* ℓ_2 *satisfy the forward step axiom (axiom (i) of Definition 7.13) where* $\ell_1(0), \ell_2(i) \in E_o \setminus E_{o,i}$, *and* $\exists s \in \mathbb{A}^*$ *such that* $x_4 \xrightarrow{s} b$.

At a communication transition for some illegal configuration $b \in B_{uncond} \cup B_{cond}$, we interpret label ℓ_1 as the occurrence and observation of event $\ell_1(0)$, an event that is not observed by controller i, and ℓ_2 controller i's "guess" that $\ell_1(0)$ has occurred. A label for an unobservable event for controller i acts merely as a placeholder. Then $\ell_1 \vee \ell_2$ represents the synchronous communication to controller i that $\ell_1(0)$ has just occurred. Thus, by choosing this communication transition, the two other transitions

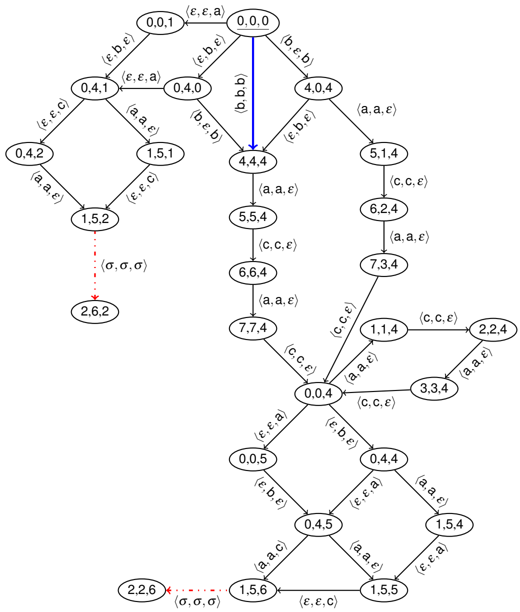

Fig. 7.7 A portion of $\mathscr{U}$ that contains all the illegal configurations for the example from Fig. 7.2. Initial state is underlined. Transitions in B_{uncond} are denoted by a dashed/dotted line (red). Potential communication transitions are indicated in bold (blue)

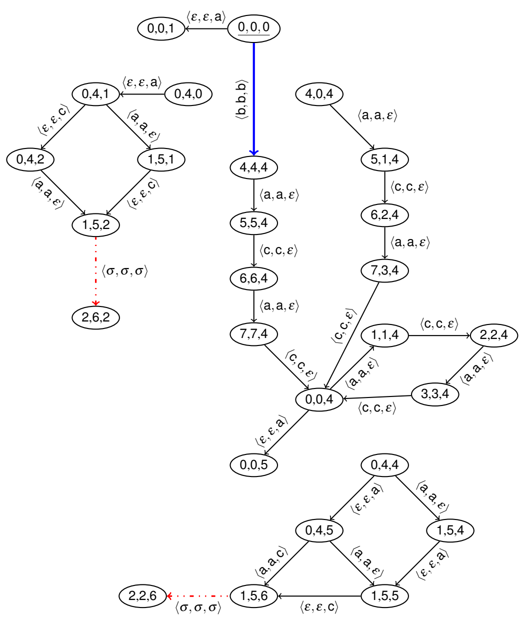

Fig. 7.8 Result of pruning the portion of $\mathscr{U}$ from Fig. 7.7. The transitions in B_{uncond} are no longer reachable

must be pruned from $\mathscr{U}$. That is, when we prune $x_1 \xrightarrow{\ell_1} x_2$ and $x_1 \xrightarrow{\ell_2} x_3$, we must prune *all* transitions $x' \xrightarrow{\ell_1} x''$ such that $x'(i) = x_1(i)$, $x''(i) = x_2(i)$ and all transitions $x' \xrightarrow{\ell_2} x'''$ such that $x'(i) = x_1(i)$, $x'''(i) = x_3(i)$.

There is only one communication transition in $\mathscr{U}$, namely $(0,0,0) \xrightarrow{\langle \mathsf{b},\mathsf{b},\mathsf{b}\rangle} (4,4,4)$. We prune $(0,0,0) \xrightarrow{\langle \varepsilon,\mathsf{b},\varepsilon\rangle} (0,4,0)$ and $(0,0,0) \xrightarrow{\langle \mathsf{b},\varepsilon,\mathsf{b}\rangle} (4,0,4)$. It is now the case that illegal configuration $(1,5,2) \xrightarrow{\langle \sigma,\sigma,\sigma\rangle} (2,6,2)$ is unreachable. Pruning of $(0,0,4) \xrightarrow{\langle \mathsf{b},\mathsf{b},\mathsf{b}\rangle} (0,4,4)$ and $(0,0,5) \xrightarrow{\langle \mathsf{b},\mathsf{b},\mathsf{b}\rangle} (0,4,5)$ makes the other illegal configuration $(1,5,6) \xrightarrow{\langle \sigma,\sigma,\sigma\rangle} (2,2,6)$ unreachable.

To ensure that communication in $\mathscr{U}$ is feasible, we must also choose communication transitions at states of $\mathscr{U}$ that are indistinguishable to the sender.

Definition 7.15. *Two states $x = (x(0),\ldots,x(n))$, $x' = (x'(0),\ldots,x'(n)) \in X$ are* indistinguishable *to controller i, denoted $x \sim_i x'$, where $\sim_i$ is the least equivalence relation such that*
i. $x \xrightarrow{\langle \ell(0),\ldots,\ell(i)=\varepsilon,\ldots,\ell(n)\rangle} x' \Rightarrow x \sim_i x'$;
ii. $x \xrightarrow{\langle \varepsilon,\ldots,\varepsilon,\ell(i)\neq\varepsilon,\varepsilon,\ldots,\varepsilon\rangle} x' \Rightarrow x \sim_i x'$;
iii. if $x \sim_i x'$ and $(x,\ell,x''),(x',\ell,x''') \in T^{\mathscr{U}} \Rightarrow x'' \sim_i x'''$.

As there is only one potential communication transition in $\mathscr{U}$ in Fig. 7.7, there are no additional communications that must be identified to satisfy feasibility. The final communication protocol is $\mu^!_{12}(L) = \varepsilon$ and $\mu^!_{21}(((\mathsf{acac})^*\mathsf{b})^+) = \mathsf{b}$ whereas $\mu^!_{21}(L \setminus ((\mathsf{acac})^*\mathsf{b})^+) = \varepsilon$.

7.5 Further Reading

Although this chapter has focussed on communication protocol synthesis for control, there are additional strategies to calculate optimal communication sets from a given set of communications. This literature focuses on state disambiguation, where the analysis is performed on the original state space (in contrast to the synthesis techniques presented in this chapter). The problem of dynamic sensing is also closely related to the synthesis of decentralized communication protocols, where one can think of turning a sensor on and off as equivalent to communicating an event occurrence. Finally, communication has been examined in the context of decentralized diagnosis. Some representative papers are noted below.

- K. Rudie, S. Lafortune and F. Lin, ”Minimal Communication in a Distributed Discrete-Event System”, *IEEE Trans. Autom. Control*, vol. 48, no. 6, 957–975, 2003.
- W. Wang, S. Lafortune and F. Lin, ”Minimization of Communication of Event Occurrences in Acyclic DES,” *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2197–2202, 2008.
- F. Cassez and Tripakis, S., “Fault Diagnosis with Static and Dynamic Observers,” *Fundamenta Informaticae*, vol. 88, no. 4, pp. 497–540, 2008.

- R. Debouk, S. Lafortune and D. Teneketzis, "Coordinated Decentralized Protocols for Failure Diagnosis of DES," *Discrete Event Dyn. S.*, vol. 10, no. 1/2, pp. 33–86, 2000.
- Qiu, W. and Kumar, R., "Distributed Diagnosis Under Bounded-Delay Communication of Immediately Forwarded Local Observations," *IEEE Trans. Sys. Man Cyber Part A*, vol. 38, no. 3, pp 628–643, 2008.

References

1. Arnold, A.: Finite Transition Systems. Prentice-Hall (1994)
2. Barrett, G., Lafortune, S.: Decentralized supervisory control with communicating controllers. IEEE Transactions on Automatic Control 45(9), 1620–1638 (2000)
3. Darondeau, P., Ricker, L.: Towards distributed control of discrete-event systems. In: Proc. Workshop Applications of Region Theory, Newcastle upon Tyne, UK (2011)
4. Diekert, V., Rozenberg, G. (eds.): The Book of Traces. World Scientific (1995)
5. Mannani, A., Gohari, P.: Decentralized supervisory control of DES over communication networks. IEEE Transactions on Automatic Control 53(2), 547–559 (2008)
6. Hiraishi, K.: On solvability of a decentralized supervisory control problem with communication. IEEE Transactions on Automatic Control 54(3), 468–480 (2009)
7. Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM Journal of Research and Development 3(2), 114–125 (1959)
8. Ricker, S.L.: Asymptotic minimal communication for decentralized discrete-event control. In: Proc. 9th Workshop on Discrete Event Systems, Göotegorg, Sweden (2008)
9. Ricker, S.L., Caillaud, B.: Revisiting state-based models for synthesizing optimal communicating decentralized discrete-event controllers. In: Proc. European Control Conference, Budapest, Hungary (2009)
10. Ricker, S.L., Rudie, K.: Incorporating communication and knowledge into decentralized discrete-event systems. In: Proc. 38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA (1999)
11. Rudie, K., Wonham, W.M.: Think globally, act locally: decentralized supervisory control. IEEE Transactions on Automatic Control 37(11), 1692–1708 (1992)
12. Teneketzis, D.: On information structures and nonsequential stochastic control. CWI Quarterly 10(2), 179–199 (1997)
13. Tripakis, S.: Decentralized control of discrete event systems with bounded or unbounded delay communication. IEEE Transactions on Automatic Control 49(9), 1489–1501 (2004)
14. van Schuppen, J.H.: Decentralized supervisory control with information structures. In: Proc. 4th Workshop on Discrete Event Systems, Cagliari, Italy (1998)
15. Witsenhausen, H.: Separation of estimation and control for discrete time systems. Proceedings of the IEEE 59, 1557–1566 (1971)
16. Wong, K.C., van Schuppen, J.H.: Decentralized supervisory control of DES with communication. In: Proc. 3rd Workshop on Discrete Event Systems, Edinburgh, Scotland (1996)
17. Yoo, T.S., Lafortune, S.: A general architecture for decentralized supervisory control of discrete-event systems. Discrete Event Dynamic Systems 12(3), 335–377 (2002)
18. Yoo, T.S., Lafortune, S.: Decentralized supervisory control with conditional decisions: supervisor existence. IEEE Transactions on Automatic Control 49(11), 1886–1904 (2004)