

# DECENTRALIZED SUPERVISORY CONTROL WITH COALGEBRA

**Jan Komenda and Jan H. van Schuppen**

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Tel. +31 20 592 4094, Fax +31 20 592 4199,

Email: {Jan.Komenda, J.H.van.Schuppen}@cwi.nl

**Keywords:** Decentralized supervisory control, coalgebra, coinduction.

## Abstract

Coalgebraic methods provide new results and insights for the supervisory control of discrete-event systems (DES). In this paper a coalgebraic framework for the decentralized control of DES is proposed. The paper is based on the formalism developed for the supervisory control of DES in the partial observation case, the notion of bisimulation, and its generalizations (partial bisimulation, co-observability and control relation). Local indistinguishability relations are used in the relational characterizations of co-observability. Conjunctive and permissive (C&P) as well as disjunctive and antipermissive (D&A) versions of co-observability are captured by their corresponding relations. Coinduction is used to define a new operation on languages called C&P supervised product. Existence of a supervisor that achieves a given specification in the closed-loop system is equivalent to the existence of a partial bisimulation relation, which is at the same time a co-observability and control relation.

## 1. Introduction

Discrete-event (dynamical) systems (DES) can be studied using coalgebraic techniques. DES are often represented by automata viewed as a particular algebraic structure. However, they may also be viewed as partial automata, which are coalgebras of a simple functor of the category of sets. Coalgebras are categorial duals of algebras (the corresponding functor operates from a given set rather than to a given set).

This paper presents a formulation of decentralized control of DES in terms of coalgebra. The basic formalism is the one that has been developed by J.J.M.M. Rutten in [8] and extended by the first author in [4] to partial observations,

i.e. partial automata as models for DES and partial automaton of (partial) languages as the final coalgebra. The main advantage of the use of coalgebra is the naturally algorithmic character of the results, there is a canonical way how to check the properties like decomposability [6] or co-observability for different control architectures by constructing corresponding relations.

There are two control architectures for the decentralized supervisory control as proposed by T.S. Yoo and S. Lafortune in [13]: Conjunctive and permissive, referred to as C&P, and disjunctive and antipermissive, referred to as D&A. Coinduction is used to define operations on languages called C&P and D&A supervised product, which represent the languages of the closed-loop system, where the first language acts as a specification language and the second as an open-loop system language for both of the control architectures.

The paper is organized as follows. Section 2 recalls partial automata from [8] as the coalgebraic framework for DES represented by automata. In Section 3 after introducing local weak transition structures on partial automata we present an introduction to the decentralized supervisory control. An alternative definition of co-observability is presented, which is equivalent to C&P co-observability of [13] and coincides with the original definition of co-observability in the case of two local supervisors [10]. Section 4 contains the main results of the paper. First relational characterizations of both C&P and D&A co-observability are introduced. We present also a coinductive definition of the C&P supervised product that represent the closed-loop languages in the decentralized supervisory control.

## 2. Partial automata

In this section we recall partial automata as coalgebras with a special emphasis on the final coalgebra of partial automata, i.e. partial automaton of partial languages. Let

$A$  be an arbitrary set (usually finite and referred to as the set of inputs or events). The empty string will be denoted by  $\varepsilon$ . Denote by  $1 = \{\emptyset\}$  the one element set and by  $2 = \{0, 1\}$  the set of Booleans. A partial automaton is a pair  $S = (S, \langle o, t \rangle)$ , where  $S$  is a set of states, and a pair of functions  $\langle o, t \rangle : S \rightarrow 2 \times (1 + S)^A$ , consists of an output function  $o : S \rightarrow 2$  and a transition function  $S \rightarrow (1 + S)^A$ . The output function  $o$  indicates whether a state  $s \in S$  is accepting (or terminating) :  $o(s) = 1$ , denoted also by  $s \downarrow$ , or not:  $o(s) = 0$ , denoted by  $s \uparrow$ . The transition function  $t$  associates to each state  $s$  in  $S$  a function  $t(s) : A \rightarrow (1 + S)$ . The set  $1 + S$  is the disjoint union of  $S$  and  $1$ . The meaning of the state transition function is that  $t(s)(a) = \emptyset$  iff  $t(s)(a)$  is undefined, which means that there is no  $a$ -transition from the state  $s \in S$ .  $t(s)(a) \in S$  means that  $a$ -transition from  $s$  is possible and we define in this case  $t(s)(a) = s_a$ , which is denoted mostly by  $s \xrightarrow{a} s_a$ . This notation can be extended by induction to arbitrary strings in  $A^*$ . Assuming that  $s \xrightarrow{w} s_w$  has been defined, define  $s \xrightarrow{wa} s_{wa}$  iff  $t(s_w)(a) \in S$ , in which case  $s_{wa} = t(s_w)(a)$  and  $s \xrightarrow{wa} s_{wa}$ .

A *homomorphism* between partial automata  $S = (S, \langle o, t \rangle)$  and  $S' = (S', \langle o', t' \rangle)$  is a function  $f : S \rightarrow S'$  with, for all  $s \in S$  and  $a \in A$ :

$$o'(f(s)) = o(s) \text{ and } s \xrightarrow{a} s_a \text{ iff } f(s) \xrightarrow{a} f(s)_a,$$

in which case:  $f(s)_a = f(s_a)$ .

A partial automaton  $S' = (S', \langle o', t' \rangle)$  is a *subautomaton* of  $S = (S, \langle o, t \rangle)$  if  $S' \subseteq S$  and the inclusion function  $i : S' \rightarrow S$  is a homomorphism.

## 2.1. Final automaton of partial languages

Below we define partial automaton of partial languages over an alphabet (input set)  $A$ , denoted by  $\mathcal{L} = (\mathcal{L}, \langle o_{\mathcal{L}}, t_{\mathcal{L}} \rangle)$ . More formally,  $\mathcal{L} = \{\Phi : A^* \rightarrow (1 + 2) \mid \text{dom}(\Phi) \neq \emptyset \text{ is prefix-closed}\}$ . To each partial language  $\Phi$  a pair  $\langle V, W \rangle$  can be assigned:  $W = \text{dom}(\Phi)$  and  $V = \{w \in A^* \mid \Phi(w) = 1(\in 2)\}$ . Conversely, to a pair  $\langle V, W \rangle \in \mathcal{L}$ , a function  $\Phi$  can be assigned :  $\Phi(w) = 1$  if  $w \in V$ ,  $\Phi(w) = 0$  if  $w \in W$  and  $w \notin V$  and  $\Phi(w)$  is undefined if  $w \notin W$ . Therefore we can write :

$$\mathcal{L} = \{(V, W) \mid V \subseteq W \subseteq A^*, W \neq \emptyset, \text{ and } \bar{W} = W\}.$$

Transition function  $t_{\mathcal{L}} : \mathcal{L} \rightarrow (1 + \mathcal{L})^A$  is defined using the input derivatives. Recall that for any partial language  $L = (L^1, L^2) \in \mathcal{L}$ ,  $L_a = (L_a^1, L_a^2)$ , where  $L_a^i = \{w \in A^* \mid aw \in L^i\}$ ,  $i = 1, 2$ . If  $a \notin L^2$  then  $L_a$  is undefined. Given any  $L = (L^1, L^2) \in \mathcal{L}$ , the partial automaton structure of  $\mathcal{L}$  is given by:

$$o_{\mathcal{L}}(L) = \begin{cases} 1 & \text{if } \varepsilon \in L^1 \\ 0 & \text{if } \varepsilon \notin L^1 \end{cases}$$

and

$$t_{\mathcal{L}}(L)(a) = \begin{cases} L_a & \text{if } L_a \text{ is defined} \\ \emptyset & \text{otherwise} \end{cases}.$$

Notice that if  $L_a$  is defined, then  $L_a^1 \subseteq L_a^2$ ,  $L_a^2 \neq \emptyset$ , and  $L_a^2$  is prefix-closed. The following notational conventions will be used:  $L \downarrow$  iff  $\varepsilon \in L^1$  and  $L \xrightarrow{w} L_w$  iff  $L_w$  is defined iff  $w \in L^2$ .

Recall from [8] that  $\mathcal{L} = (\mathcal{L}, \langle o_{\mathcal{L}}, t_{\mathcal{L}} \rangle)$  is final among all partial automata: for any partial automaton  $S = (S, \langle o, t \rangle)$  there exists a unique homomorphism  $l : S \rightarrow \mathcal{L}$ . Recall that the unique homomorphism  $l$  given by finality of  $\mathcal{L}$  maps a state  $s \in S$  to the partial language  $l(s) = (L_s^1, L_s^2) = (\{w \in A^* \mid s \xrightarrow{w} \text{ and } s_w \downarrow\}, \{w \in A^* \mid s \xrightarrow{w} \})$ .

Denote the minimal representation of a partial language  $L$  by  $\langle L \rangle$ , i.e.  $\langle L \rangle = (DL, \langle o_{\langle L \rangle}, t_{\langle L \rangle} \rangle)$  is a subautomaton of  $\mathcal{L}$  generated by  $L$ . This means that  $o_{\langle L \rangle}$  and  $t_{\langle L \rangle}$  are uniquely determined by the corresponding structure of  $\mathcal{L}$ . The carrier set of this minimal representation of  $L$  is denoted by  $DL$ , where  $DL = \{L_u \mid u \in L^2\}$ . Let us call this set the set of derivatives of  $L$ . Inclusion of partial languages that corresponds to a simulation relation is always meant componentwise.

## 3. Introduction to decentralized supervisory control

Decentralized supervisory control arises because particular engineering systems have two or more local controllers each receiving different partial observations of the system. Since communication of the local observations is either not possible or possible but costly, the partial observations of the local controllers differ. As an example consider different radio stations in a communication network such as a wireless local area network, where each station knows of itself whether it wants to send a message but does not have this information about other stations. Decentralized supervisory control consists in considering local controllers  $S_1, \dots, S_n$  and breaking the set of controllable and observable events into locally controllable and locally observable events, denoted by  $A_{c,i}$  and  $A_{o,i}$ ,  $i = 1, \dots, n$  respectively. The natural projections to locally observable events are denoted by  $P_i : A^* \rightarrow A_{o,i}^*$ . The action of  $P_i$  is simply to delete events that are not observable by  $S_i$ .

The following notation will be used:  $\{1, \dots, n\} = Z_n$ , for any  $a \in A$ :  $Z_c^a = \{i \in Z_n : a \in A_{c,i}\}$  and similarly  $Z_o^a = \{i \in Z_n : a \in A_{o,i}\}$ . Furthermore, we denote  $A_c = \cup_{i \in Z_n} A_{c,i}$ ,  $A_o = \cup_{i \in Z_n} A_{o,i}$ ,  $A_{uc} = A \setminus A_c$ , and finally  $A_{uo} = A \setminus A_o$ .

In the following definition we introduce the notion of weak derivative (transition). Roughly speaking it disre-

guards locally unobservable events.

**Definition 3.1.** (Nondeterministic weak transitions.) For an event  $a \in A$  define  $L \xrightarrow{P_i(a)} L_s$  if  $\exists s \in A^* : P_i(s) = P_i(a)$  and  $L \xrightarrow{s} L_s$ . Denote in this case  $L \xrightarrow{P_i(a)} L_s$ .

**Remark 3.1.** Let us introduce the notation for locally unobservable events  $L \xrightarrow{\varepsilon}_i$  as an abbreviation for  $\exists \tau \in A^*$  such that  $P_i(\tau) = \varepsilon$  and  $L \xrightarrow{\tau}$ . We write then  $L \xrightarrow{\varepsilon}_i L_\tau$ . We admit  $\tau = \varepsilon$ , hence  $L \xrightarrow{\varepsilon}_i$  is always true. For  $a \in A_{o,i}$  our notation means that there exist  $\tau, \tau' \in (A \setminus A_{o,i})^*$  such that  $L \xrightarrow{\tau a \tau'} L_{\tau a \tau'}$ .

### Co-observability

There are two control architectures for the decentralized supervisory control [13]. The original control architecture is called C&P (conjunctive and permissive). A local supervisor  $S_i$  is then represented as a mapping  $\gamma_{C\&P}(S_i, \cdot) : P_i(L(G)) \rightarrow \Gamma_i$ , where  $\Gamma_i = \{C \subseteq A : C \supseteq (A \setminus A_{c,i})\}$  is the set of local control patterns and  $\gamma_{C\&P}(S_i, s)$  represents the set of locally enabled events after  $S_i$  has observed string  $s \in A_{o,i}^*$ . The associated control law of the local supervisor  $S_i$  is

$$\gamma_{C\&P}(S_i, s) = (A \setminus A_{c,i}) \cup$$

$$\{a \in A_{c,i} : \exists s' \in K^2 \cap P_i^{-1}P_i(s) \text{ and } s'a \in K^2\}.$$

The control law of the conjunction of local supervisors  $S_i$ ,  $i = 1, \dots, n$  is given by :

$$\left( \bigwedge_i \gamma_{C\&P} S_i \right)(w) = \bigcap_{i=1}^n \gamma_{C\&P}(S_i, P_i(w)), \quad w \in A^*.$$

According to [10] the necessary and sufficient conditions for a given specification language  $K$  to be achieved by a joint action of local supervisors are controllability,  $L_m(G)$ -closedness, and co-observability. The definition of co-observability from [10] can be extended from two to  $n$  supervisors.

**Definition 3.2.** (Co-observability.) The language  $K \subseteq L = L(G)$  is called co-observable with respect to  $L$  and  $A_{o,i}$ ,  $i = 1, \dots, n$  if  $(\forall s \in K^2), (\forall a \in A_c : sa \in L^2) (\exists i \in \{1, \dots, n\} : a \in A_{c,i})$  such that the following implication holds true:

$$(s' \in K^2 \text{ and } P_i(s) = P_i(s') \text{ and } s'a \in K^2) \Rightarrow sa \in K^2.$$

Note that the definition of co-observability has been originally formulated for two local supervisors in [10]. An equivalent form of co-observability for two local supervisors has been given in [11], which has been used to prove that co-observability is decidable in a polynomial time. Co-observability is needed for the existence of local supervisors that jointly achieve a given language ([10]).

The control law of local supervisors associated to C&P architecture is called permissive, since the default action is to enable an event whenever a local supervisor has an ambiguity what to do with this event. It should be clear that with the permissive local policy we always achieve all strings in the specification language  $K$ , i.e.  $K$  is always contained in the language of the closed-loop system. The only concern is safety, which is expressed by the following definition of C&P co-observability [13], which states that there always exists a local supervisor that is sure to disable an event resulting in an illegal string.

**Definition 3.3.** (C&P co-observability.)  $K \subseteq L$  is said to be C&P co-observable with respect to  $L$  and  $A_{o,i}$ ,  $i = 1, \dots, n$  if for all  $s \in K^2$ ,  $a \in A_c$  such that  $sa \in L^2 \setminus K^2$

$$\exists i \in \{1, \dots, n\} : a \in A_{c,i} \text{ and } (P_i^{-1}(P_i(s))a) \cap K^2 = \emptyset.$$

It has been shown in [1] that C&P-co-observability coincides for two supervisors with the 'classical' definition of co-observability introduced in [10]. We will show that the definition of co-observability above (definition 3.2) is equivalent to C&P-co-observability and can thus be considered as an extension of the definition given in [10] to an arbitrary number of local supervisors.

**Lemma 3.2.** Co-observability is equivalent to C&P co-observability.

*Proof.* Co-observability can be captured by the following condition:

$(\forall s \in K^2)$ , and  $(\forall a \in A_c : sa \in L^2) (\exists i \in \{1, \dots, n\} : a \in A_{c,i})$  such that the following implication holds true:

$$(s' \in K^2 \text{ and } P_i(s) = P_i(s') \text{ and } s'a \in K^2) \Rightarrow sa \in K^2.$$

It can be written as:

$(\forall s \in K^2), (\forall a \in A_c \text{ such that } sa \in L^2) (\exists i \in \{1, \dots, n\} : a \in A_{c,i})$  such that the following implication holds true:

$$(s' \in K^2 \text{ and } P_i(s) = P_i(s') \text{ and } sa \notin K^2) \Rightarrow s'a \notin K^2,$$

which is equivalent to

$(\forall s \in K^2), (\forall a \in A_c \text{ such that } sa \in L^2 \setminus K^2) (\exists i \in \{1, \dots, n\} : a \in A_{c,i})$

$$\forall s' \in K^2 \text{ with } P_i(s) = P_i(s') \text{ we have } s'a \notin K^2.$$

This can be written in the following equivalent form:  $\forall s \in K^2 \forall a \in A_c$  such that  $sa \in L^2 \setminus K^2$

$$\exists i \in \{1, \dots, n\} : a \in A_{c,i} \text{ and } P_i^{-1}(P_i(s))a \cap K^2 = \emptyset.$$

But this is the definition of C&P co-observability.  $\square$

There is a natural counterpart of the C&P control architecture, called D&A (disjunctive and antipermissive). The D&A control architecture is given by the following control law for local supervisors  $S_i$  and  $s \in A_{o,i}^*$

$$\gamma_{D\&A}(S_i, s) = A_{uc} \cup \{a \in A_{c,i} : \forall s' \in K^2 \text{ with}$$

$$P_i(s') = P_i(s) \text{ we have } s'a \in L^2 \Rightarrow s'a \in K^2 \},$$

Note that  $\forall i \in Z_n : \gamma_{D\&A}(S_i, s) \cap (A_c \setminus A_{c,i}) = \emptyset$ . The disjunction of local supervisors  $S_i$ ,  $i = 1, \dots, n$  is given by :

$$\left( \bigvee_i \gamma_{D\&A}(S_i, w) \right) = \bigcup_{i=1}^n \gamma_{D\&A}(S_i, P_i(w)), \quad w \in A^*.$$

The antipermissive control law of local supervisors (local decision rule) means that the default action is to disable an event whenever a local supervisor has an ambiguity what to do with this event or if he can not control a globally controllable event. The advantage of this architecture is that we are sure to synthesize a safe behavior. The only concern is that we do not necessarily achieve the whole language  $K$ , but in general only a sublanguage. The corresponding condition that ensures together with controllability and  $L_m(G)$ -closedness that language  $K$  is achieved is called D&A co-observability [13].

**Definition 3.4.** (D&A co-observability.)  $K \subseteq L$  is said to be D&A co-observable with respect to  $L$  and  $A_{o,i}$ ,  $i = 1, \dots, n$  if for all  $s \in K^2$ ,  $a \in A_c$  such that  $sa \in K^2$

$$\exists i \in Z_c^a \text{ such that } P_i^{-1}(P_i(s) \cap K^2) a \cap L^2 \subseteq K^2.$$

## 4 Decentralized supervisory control and coalgebra

We have presented in [4] a coalgebraic approach to the supervisory control of DES with partial observations. It is possible to formulate basic concepts of the decentralized supervisory control using coalgebra. First observe that the concept of observational indistinguishability relation can be easily extended to the family of observation indistinguishability relations associated to local observers. For partial automaton  $S$  with initial state  $s_o$  we define for  $i \in Z_n$ :

**Definition 4.1.** (Observational indistinguishability relation on  $S$ .) A binary relation  $Aux_i(S)$  on  $S$  is called an observational indistinguishability relation if the following three conditions hold:

- (i)  $\langle s_o, s_o \rangle \in Aux_i(S)$
- (ii) If  $\langle s, t \rangle \in Aux_i(S)$  then :  $(s \xrightarrow{\xi_i} s' \text{ for some } s' \in S \text{ and } t \xrightarrow{\xi_i} t' \text{ for some } t' \in S) \Rightarrow \langle s', t' \rangle \in Aux_i(S)$

- (iii) If  $\langle s, t \rangle \in Aux_i(S)$  then  $\forall a \in A_{o,i} : (s \xrightarrow{a} s_a \text{ and } t \xrightarrow{a} t_a) \Rightarrow \langle s_a, t_a \rangle \in Aux_i(S)$ .

Since we work with the final automaton of partial languages, and for  $K \subseteq L$ , it is not in general true that  $\langle K \rangle$  is a subautomaton of  $\langle L \rangle$ , the following concept is needed.

**Definition 4.2.** A binary relation  $Aux_i(K, L) \subseteq (DK \times DL)^2$ ,  $i \in \{1, \dots, n\}$  is called an observational indistinguishability relation with respect to  $P_i$  if the following three conditions hold :

- (i)  $\langle (K, L), (K, L) \rangle \in Aux_i(K, L)$
- (ii) If  $\langle (M, N), (Q, R) \rangle \in Aux_i(K, L)$  then :  $[ (M, N) \xrightarrow{\xi_i} (M', N') \text{ for some } (M', N') \in DK \times DL \text{ and } (Q, R) \xrightarrow{\xi_i} (Q', R') \text{ for some } (Q', R') \in DK \times DL ] \Rightarrow \langle (M', N'), (Q', R') \rangle \in Aux_i(K, L)$
- (iii) If  $\langle (M, N), (Q, R) \rangle \in Aux_i(K, L)$  then  $\forall a \in A_{o,i} : [(M, N) \xrightarrow{a} (M_a, N_a) \text{ and } (Q, R) \xrightarrow{a} (Q_a, R_a)] \Rightarrow \langle (M_a, N_a), (Q_a, R_a) \rangle \in Aux_i(K, L)$ .

For  $\langle (M, N), (Q, R) \rangle \in DK \times DL$  we write  $(M, N) \approx_{Aux_i(K, L)}^{K, L} (Q, R)$  whenever  $\langle (M, N), (Q, R) \rangle \in Aux_i(K, L)$ . Similarly as for the centralized  $Aux(K)$  we have:

**Lemma 4.1.** For given partial languages  $K, L$ :  $\langle (M, N), (Q, R) \rangle \in Aux_i(K, L)$  iff there exist two strings  $s, s' \in K^2$  such that  $P_i(s) = P_i(s')$  and  $M = K_s$ ,  $N = L_s$ ,  $Q = K_{s'}$ , and  $R = L_{s'}$ .

Now the C&P co-observability can be formulated within the coalgebraic framework.

**Definition 4.3.** (C&P Co-observability relation.) Given two (partial) languages  $K$  and  $L$ , a binary relation  $CO(K, L) \subseteq DK \times DL$  is called a C&P co-observability relation if for any  $\langle M, N \rangle \in CO(K, L)$  the following items hold:

- (i)  $\forall a \in A : M \xrightarrow{a} \Rightarrow N \xrightarrow{a} \text{ and } \langle M_a, N_a \rangle \in CO(K, L)$
- (ii)  $\forall a \in A_c : N \xrightarrow{a} \Rightarrow [(\exists i \in \{1, \dots, n\} : a \in A_{c,i}) \text{ such that } (M' \in DK, N' \in DL : (M', N') \approx_{Aux_i(K, L)}^{K, L} (M, N) \text{ and } M' \xrightarrow{a} \Rightarrow M \xrightarrow{a})]$ .

For  $M \in DK$  and  $N \in DL$  we write  $M \approx_{CO(K, L)} N$  whenever there exists a C&P co-observability relation  $CO(K, L)$  on  $DK \times DL$  such that  $\langle M, N \rangle \in CO(K, L)$ . In order to check whether for a given pair of (partial) languages  $(K$  and  $L)$ ,  $K$  is C&P co-observable with respect

to  $L$  and  $A_{o,i}, i = 1, \dots, n$ , it is sufficient to establish a C&P co-observability relation  $O(K, L)$  on  $DK \times DL$  such that  $\langle K, L \rangle \in O(K, L)$ . Indeed, we have:

**Theorem 4.2.** *A (partial) language  $K$  is C&P co-observable with respect to  $L$  (where  $K \subseteq L$ ) and  $A_{o,i}, i = 1, \dots, n$  iff  $K \approx_{CO(K,L)} L$ .*

*Proof.* ( $\Rightarrow$ ) Let  $K$  be C&P co-observable with respect to  $L$ . Denote

$$CO(K, L) = \{ \langle K_u, L_u \rangle \in DK \times DL \mid u \in K^2 \}.$$

Let us show that  $CO(K, L)$  is a C&P co-observability relation.

Let  $\langle M, N \rangle \in CO(K, L)$ . We can assume that  $M = K_s$  and  $N = L_s$  for  $s \in K^2$ . We must show that conditions (i) and (ii) are satisfied.

(i) Let  $M \xrightarrow{a}$  for  $a \in A$ . Notice that  $K \subseteq L$  implies that for any  $u \in K^2$ ,  $K_u \subseteq L_u$ . In particular  $N \xrightarrow{a}$ , because  $M = K_s \subseteq L_s = N$  and it follows from the definition of  $CO(K, L)$  that  $\langle M_a, N_a \rangle \in CO(K, L)$ .

(ii) Let  $N \xrightarrow{a}$  for  $a \in A_c$ . Then we have  $sa \in L^2$  and recall that  $s \in K^2$ . Then by C&P co-observability of  $K$  with respect to  $L$  there exists  $i \in Z_c^a$  such that whenever there is a string  $s'a \in K^2$  with  $P_i(s') = P_i(s)$ , then also  $sa \in K^2$ . Using Lemma 4.1 this means that there exists  $i \in Z_c^a$  such that whenever  $\langle M', N' \rangle \approx_{Aux(i)}^{K,L} \langle M, N \rangle : M' \xrightarrow{a}$ , then  $M \xrightarrow{a}$ . Indeed this means that there exist  $s', s'' \in A^*$ :  $M' = K_{s'}, N' = L_{s'}, M = K_{s''} = K_s, N = L_{s''} = L_s$ , and  $P_i(s'') = P_i(s')$ . Note that it can be that  $s = s''$ . We have  $s'' \in K^2$  and  $s''a \in L^2$ . By applying the co-observability of  $K$ , where  $s''$  plays the role of  $s$ , it follows that  $s''a \in K$ , i.e.  $K_{s''} = M \xrightarrow{a}$ . Hence  $CO(K, L)$  is a C&P co-observability relation.

( $\Leftarrow$ ) Let  $K \approx_{CO(K,L)} L$ . Let us show that  $K$  is C&P co-observable with respect to  $L$ . For this purpose, let  $s \in K^2$  and  $a \in A_c$  such that  $sa \in L^2$ . Then  $s \in K^2 \cap L^2$ , i.e.  $L \xrightarrow{s}$  and  $K \xrightarrow{s}$ , whence from (i) of definition 4.3 inductively applied  $K_s \approx_{CO(K,L)} L_s$ . Now,  $sa \in L^2$  means that  $L_s \xrightarrow{a}$ , hence by definition of 4.3 there exists  $i \in Z_c^a$  such that whenever  $\langle M', N' \rangle \approx_{Aux(i)}^{K,L} \langle K_s, L_s \rangle : M' \xrightarrow{a}$ , then  $K_s \xrightarrow{a}$ . According to Lemma 4.1 we have for  $P_i(s') = P_i(s)$  that  $\langle K_{s'}, L_{s'} \rangle \approx_{Aux(i)}^{K,L} \langle K_s, L_s \rangle$ . Also notice that  $s'a \in K^2$  is equivalent to  $K_{s'} \xrightarrow{a}$ . But this means that there exists  $i \in Z_c^a$  such that whenever there is a string  $s'a \in K^2$  with  $P_i(s') = P_i(s)$ , then also  $sa \in K^2$ , i.e.  $K$  is C&P co-observable with respect to  $L$  and  $A_{o,i}, i = 1, \dots, n$ .  $\square$

D&A co-observability can be characterized by relations as well. We define

**Definition 4.4.** *(D&A co-observability relation.) Given two (partial) languages  $K$  and  $L$ , a binary relation*

$DC(K, L) \subseteq DK \times DL$  is called a D&A co-observability relation if for any  $\langle M, N \rangle \in DC(K, L)$  the following items hold:

- (i)  $\forall a \in A : M \xrightarrow{a} \Rightarrow N \xrightarrow{a}$  and  $\langle M_a, N_a \rangle \in DC(K, L)$
- (ii)  $\forall a \in A_c : M \xrightarrow{a} \Rightarrow \{ \exists i \in Z_c^a \text{ such that } (M' \in DK, N' \in DL : \langle M', N' \rangle \approx_{Aux(i)}^{K,L} \langle M, N \rangle \text{ and } N' \xrightarrow{a}) \Rightarrow M' \xrightarrow{a} \}$ .

D&A co-observability relation corresponds to D&A co-observability in the sense of definition 3.5.

**Theorem 4.3.** *A (partial) language  $K$  is D&A co-observable with respect to  $L$  (where  $K \subseteq L$ ) and  $A_{o,i}, i = 1, \dots, n$  iff there exists a D&A co-observability relation  $D(K, L) \subseteq DK \times DL$  such that  $\langle K, L \rangle \in D(K, L)$ .*

*Proof.* Similar to that of the preceding theorem, namely it relies on Lemma 4.1.  $\square$

The definition of supervised product for DES with partial observations [5] extends to the decentralized control in the case of C&P control architecture. Local  $Aux_i(K, L)$  are involved therein. The corresponding co-observability and controllability theorem can then be easily formulated in the coalgebraic framework.

In the sequel we need also another type of auxiliary relation  $Aux_i(S)$  for the special case  $S = \langle K \rangle$ . We will write  $Aux_i(K)$  instead of  $Aux_i(\langle K \rangle)$ . Now we are prepared to give the coinductive definition of the C&P supervised product. However, the coinductive definition considers arguments from  $\text{Pwr}(\text{suffix}(K))$  rather than from  $DK$ . In fact we will work with unions of the form  $\bigcup_{j \in Z_c^a} \bigcup_{m=1}^{M_j} K_{s_{j,m}} \in \text{Pwr}(\text{suffix}(K))$  that will be associated to  $L_s \in DL$  where  $P_j(s_{j,m}) = P_j(s)$  for all  $j \in Z_c^a$  and  $m = 1, \dots, M_j$ . In order to keep the notation simple, we will use an extension of  $Aux_i(K)$  to such unions of derivatives and denote the extended relation in the same way. A natural extension of Lemma 4.1 holds. First we give a definition of local supervised products denoted by  $K/iL$ .

**Definition 4.5.** *(Local supervised products.) Define first for all  $M \in \text{Pwr}(\text{suffix}(K))$  and  $N \in DL$ :*

$$(M/iN)_a =$$

- (1)  $M_a/iN_a$  if  $M \xrightarrow{a}$  and  $N \xrightarrow{a}$ ;
- (2)  $(\bigcup_{\{M' : M' \approx_{Aux(i)}^K M\}} M'_a) / iN_a$  if  $M \not\xrightarrow{a}$  and  $\exists (M' \in DK : M' \approx_{Aux(i)}^K M \text{ such that } M' \xrightarrow{a} \text{ and } N \xrightarrow{a} \text{ and } a \in A_{c,i} \cup A_{o,i})$ ;

(3)  $0/iN_a$  if  $M \not\rightarrow$  and  $\forall M' \in DK : M' \approx_{Aux(i)}^K M$  :  
 $M' \not\rightarrow$  and  $N \xrightarrow{a}$  and  $a \in (A \setminus A_{c,i}) \cap A_{o,i}$ ;

(4)  $M/iN_a$  if  $M \rightarrow$  and  $N \xrightarrow{a}$  and  $a \in (A \setminus A_{c,i}) \cap (A \setminus A_{o,i})$ ;

(5)  $\emptyset$  otherwise

and  $(M/iN) \downarrow$  iff  $N \downarrow$ .

Now we define C&P decentralized supervised product:

$$K/\overset{C\&P}{Dec}L = \bigcap_{i \in Z_n} (K/iL).$$

Decentralized version of partial bisimulation corresponds to the necessary and sufficient conditions for a given language to be achieved by the joint action of local supervisors using the C&P control architecture.

**Definition 4.6.** (C&P partial bisimulation.) A binary relation  $R(K, L) \subseteq DK \times DL$  is called a C&P partial bisimulation if for all  $\langle M, N \rangle \in R(K, L)$ :

- (i)  $o(M) = o(N)$  ( $M \downarrow$  iff  $N \downarrow$ )
- (ii)  $\forall a \in A : M \xrightarrow{a} \Rightarrow N \xrightarrow{a}$  and  $\langle M_a, N_a \rangle \in R(K, L)$
- (iii)  $\forall u \in A_{uc} : N \xrightarrow{u} \Rightarrow M \xrightarrow{u}$
- (iv)  $\forall a \in A_c : N \xrightarrow{a} \Rightarrow [(\exists i \in Z_c^a) \text{ such that } (M' \in DK, N' \in DL : (M', N') \approx_{Aux(i)}^{K,L} (M, N) \text{ and } M' \xrightarrow{a}) \Rightarrow M \xrightarrow{a}]$ .

Denote  $K \approx_{C\&P} L$  if there exists a C&P partial bisimulation  $R$  such that  $\langle K, L \rangle \in R$ :

**Theorem 4.4.**  $K = K/\overset{C\&P}{Dec}L$  iff  $K \approx_{C\&P} L$  iff (i)  $K$  is  $L^1$ -closed ( $K^1 = K^2 \cap L^1$ ), (ii) controllable with respect to  $L$  and  $A_{uc}$ , and (iii) C&P co-observable with respect to  $L$  and  $A_{o,i}$ ,  $i = 1, \dots, n$ .

We have shown in [3] a difficulty while trying to define an antipermissive (centralized) supervised product. The same difficulty appears in the decentralized D&A control architecture. Note however that using a suitable automaton representation closed-loop languages under D&A decentralized control architecture can be computed.

## 5. Conclusion

Decentralized supervisory control of DES has been treated by coalgebraic techniques. Both types of co-observability known from the literature have been characterized by appropriate relations in this framework. Another approach, based on the finality of the automaton of partial languages, consists in using coinductive definitions for describing closed-loop language under C&P control architecture. It

gives at the same time algorithms for the computation of closed-loop languages. It has been shown in [12] that unlike the exact matching problem, the inclusion problem of the decentralized control is undecidable. Therefore it is of interest to look for special subproblems that are decidable.

## References

- [1] G.Barrett, S. Lafortune."Decentralized supervisory control with communicating controllers", *IEEE Trans. on Automatic Control*, **45**, pp. 1620-1638, (2000).
- [2] S.G. Cassandras, S. Lafortune."Introduction to Discrete Event Systems", *Kluwer Academic Publishers*, Dordrecht, (1999).
- [3] J. Komenda."Coalgebra and coinduction in discrete-event control", To appear as *research report CWI*, Amsterdam, (2003).
- [4] J. Komenda."Coalgebra and Supervisory Control of Discrete-Event Systems with Partial Observations," *Proceedings of MTNS 2002*, Notre Dame (IN), August 15-18, (2002).
- [5] J. Komenda."Computation of Supermal Sublanguages of Supervisory Control Using Coalgebra", *Proceedings of the 6th International Workshop on Discrete Event systems (WODES'02)*, IEEE Computer Society Press, IEEE, Los Alamitos, CA, U.S.A., pp. 26-33, (2002).
- [6] F. Lin, W.M. Wonham."Decentralized Supervisory Control of Discrete-Event Systems", *Information Sciences* **44**, 199-224, (1988).
- [7] A. Overkamp, J.H. van Schuppen."Maximal solutions in decentralized supervisory control", *SIAM Journal on Control and Optimization*, **39**, No.2, pp. 492-511, (2000).
- [8] J.J.M.M. Rutten."Coalgebra, Concurrency, and Control", *Research Report CWI SEN-R9921*, Amsterdam, (1999).
- [9] J.J.M.M. Rutten."Universal coalgebra: a theory of systems", *Theoretical Computer Science*, **249(1)**, pp. 3-80, (2000).
- [10] K. Rudie, W.M. Wonham."Think Globally, Act Locally: Decentralized Supervisory Control", *IEEE Trans. on Automatic Control*, **37**, N 11, pp. 1692-1708, (1992).
- [11] K. Rudie, J.C. Willems."The Computational Complexity of Decentralized Discrete-Event Control Problems", *IEEE Trans. on Automatic Control*, **40**, N 7, pp. 1313-1319, (1995).
- [12] S. Tripakis."Undecidable problems of decentralized observation and control", *Proceedings of the IEEE Conference on Decision and Control (CDC)*, (2001).
- [13] T.S. Yoo, S. Lafortune."General Architecture for Decentralized Supervisory Control of Discrete-Event Systems", *Discrete-Event Dynamic Systems: Theory and Applications*, **12**, pp. 335-377, (2002).