# On Algorithms and Extensions of Coordination Control of Discrete-Event Systems

Jan Komenda
(with Tomáš Masopust and Jan H. van Schuppen)

Czech Academy of Sciences of the Czech Republic and CWI Amsterdam

2012 October 4,
WODES 2012
Guadalajara, Mexico

# Outline

# Outline

# The Problem

- Given a large-scale system

$$G = G_1 \| G_2 \| \dots \| G_n$$

- and a specification language $K$.

- Find supervisors $S_i$ such that

$$L( \| S_i / \| G_i) = \overline{K}$$

and

$$L_m( \| S_i / \| G_i) = K \, .$$

# Our solution: Coordination control

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K)$[1]
3. Construct a coordinator $G_k$ over $E_k \supseteq E_1 \cap E_2$
4. Compute supervisor $S_k$ for $G_k$ with respect to $P_k(K)$
5. Compute supervisors $S_{i+k}$ for $G_i \| [S_k / G_k]$ with respect to $P_{i+k}(K)$
6. Construct a nonblocking coordinator for the computed supervisors

[1] $P_{1+k} : (E_1 \cup E_2)^* \to (E_1 \cup E_k)^*$

# Our solution: Coordination control

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K)$[1]
3. Construct a coordinator $G_k$ over $E_k \supseteq E_1 \cap E_2$
4. Compute supervisor $S_k$ for $G_k$ with respect to $P_k(K)$
5. Compute supervisors $S_{i+k}$ for $G_i \| [S_k / G_k]$ with respect to $P_{i+k}(K)$
6. Construct a nonblocking coordinator for the computed supervisors

[1] $P_{1+k} : (E_1 \cup E_2)^* \to (E_1 \cup E_k)^*$

# Our solution: Coordination control

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K)$[1]
3. Construct a coordinator $G_k$ over $E_k \supseteq E_1 \cap E_2$
4. Compute supervisor $S_k$ for $G_k$ with respect to $P_k(K)$
5. Compute supervisors $S_{i+k}$ for $G_i \| [S_k / G_k]$ with respect to $P_{i+k}(K)$
6. Construct a nonblocking coordinator for the computed supervisors

[1] $P_{1+k} : (E_1 \cup E_2)^* \to (E_1 \cup E_k)^*$

# Our solution: Coordination control

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K)$[1]
3. Construct a coordinator $G_k$ over $E_k \supseteq E_1 \cap E_2$
4. Compute supervisor $S_k$ for $G_k$ with respect to $P_k(K)$
5. Compute supervisors $S_{i+k}$ for $G_i \| [S_k / G_k]$ with respect to $P_{i+k}(K)$
6. Construct a nonblocking coordinator for the computed supervisors

[1] $P_{1+k} : (E_1 \cup E_2)^* \to (E_1 \cup E_k)^*$

# Our solution: Coordination control

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K)$[1]
3. Construct a coordinator $G_k$ over $E_k \supseteq E_1 \cap E_2$
4. Compute supervisor $S_k$ for $G_k$ with respect to $P_k(K)$
5. Compute supervisors $S_{i+k}$ for $G_i \| [S_k/G_k]$ with respect to $P_{i+k}(K)$
6. Construct a nonblocking coordinator for the computed supervisors

[1] $P_{1+k} : (E_1 \cup E_2)^* \to (E_1 \cup E_k)^*$

# Our solution: Coordination control

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K)$[1]
3. Construct a coordinator $G_k$ over $E_k \supseteq E_1 \cap E_2$
4. Compute supervisor $S_k$ for $G_k$ with respect to $P_k(K)$
5. Compute supervisors $S_{i+k}$ for $G_i \| [S_k / G_k]$ with respect to $P_{i+k}(K)$
6. Construct a nonblocking coordinator for the computed supervisors

[1] $P_{1+k} : (E_1 \cup E_2)^* \to (E_1 \cup E_k)^*$

# Outline

# Conditional independence

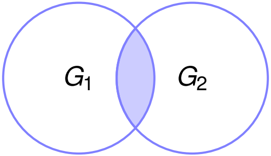
## Definition

Consider generators $G_1$, $G_2$, $G_k$. We call $G_1$ and $G_2$ conditionally independent generators given $G_k$ if

$$E_r(G_1 \| G_2) \cap E_r(G_1) \cap E_r(G_2) \subseteq E_r(G_k),$$

where $E_r(G)$ is the set of events appearing in words of $L(G)$.

Basically, we require that $E_1 \cap E_2 \subseteq E_k$.



# Conditional decomposability

**Definition**

A language $K$ is said to be conditionally decomposable with respect to event sets $E_1$, $E_2$, $E_k$ if it can be written as

$$K = P_{1+k}(K) \| P_{2+k}(K) \| P_k(K) .$$

There always exists such a set $E_k$ for which the condition is satisfied. The question which of these sets should be used (the minimal one, etc.) requires further investigation.

# Outline

# Problem formulation

- Consider generators $G_1$, $G_2$, $G_k$ over $E_1$, $E_2$, $E_k$, respectively.
- Let $K \subseteq L_m(G_1 \| G_2 \| G_k)$ be a specification language.
- $G_k$ makes $G_1$ and $G_2$ conditionally independent
- $K$ and $\overline{K}$ are conditionally decomposable wrt $E_1$, $E_2$, $E_k$.
- Aim: determine supervisors $S_1$, $S_2$, $S_k$ so that the closed-loop system with the coordinator satisfies

$$L(S_1/[G_1 \| (S_k/G_k)]) \parallel L(S_2/[G_2 \| (S_k/G_k)]) = \overline{K}$$

and

$$L_m(S_1/[G_1 \| (S_k/G_k)]) \parallel L_m(S_2/[G_2 \| (S_k/G_k)]) = K\,.$$

# Coordinator

## Algorithm (Construction of a coordinator)

Let $G_1$ and $G_2$ be two subsystems over the event sets $E_1$ and $E_2$, respectively. Construct the event set $E_k$ and the coordinator $G_k$ as follows:

1. Set $E_k = E_1 \cap E_2$
2. Extend $E_k$ so that $K$ and $\overline{K}$ are conditional decomposable
3. Extend $E_k$ so that $P_k$ is $L(G_i)$-observer
4. Define $G_k = P_k(G_1) \parallel P_k(G_2)$

# Existence of the solution

## Theorem (Existence)

*There exist supervisors $S_1$, $S_2$, $S_k$ such that*

$$L(S_1/[G_1\|(S_k/G_k)]) \parallel L(S_2/[G_2\|(S_k/G_k)]) = \overline{K}$$

*and*

$$L_m(S_1/[G_1\|(S_k/G_k)]) \parallel L_m(S_2/[G_2\|(S_k/G_k)]) = K$$

*if and only if $K$ is both*

- *conditionally controllable with respect to generators $G_1$, $G_2$, $G_k$ and event sets $E_{1,u}$, $E_{2,u}$, $E_{k,u}$, and*
- *conditionally closed with respect to $G_1$, $G_2$, $G_k$.*

# Conditional controllability

## Definition (Definition)

We call a language $K$ conditionally controllable for generators $G_1$, $G_2$, $G_k$ and uncontrollable event sets $E_{1,u}$, $E_{2,u}$, $E_{k,u}$ if

1. $P_k(K)$ is controllable with respect to $L(G_k)$ and $E_{k,u}$,
2. $P_{1+k}(K)$ is controllable with respect to $L(G_1)\|\overline{P_k(K)}$ and $E_{1+k,u}$,
3. $P_{2+k}(K)$ is controllable with respect to $L(G_2)\|\overline{P_k(K)}$ and $E_{2+k,u}$.

# Conditional closedness

## Definition

Call a language $K \neq \emptyset$ conditionally closed for generators $G_1$, $G_2$, $G_k$ if

1. $P_k(K)$ is $L_m(G_k)$-closed,
2. $P_{1+k}(K)$ is $L_m(G_1) \| P_k(K)$-closed, and
3. $P_{2+k}(K)$ is $L_m(G_2) \| P_k(K)$-closed.

# Local control consistency

## Definition (LCC)

Let $L = \overline{L} \subseteq \Sigma^*$, and let $\Sigma_0 \subseteq \Sigma$. Projection $P_0 : \Sigma^* \to \Sigma_0^*$ is *locally control consistent* (LCC) wrt $s \in L$ if for all words $\sigma_u$ of $\Sigma_0 \cap \Sigma_u$ s. t. $P_0(s)\sigma_u$ in $P_0(L)$, either there does not exist any $u \in (\Sigma \setminus \Sigma_0)^*$ s. t. $su\sigma_u$ is in $L$, or there exists a word $u$ in $(\Sigma_u \setminus \Sigma_0)^*$ s. t. $su\sigma_u \in L$. $P_0$ is LCC wrt $L$ if $P_0$ is LCC $\forall s \in L$.

## Definition (Observer)

$P_k : \Sigma^* \to \Sigma_k^*$, where $\Sigma_k \subseteq \Sigma$, is an *L-observer* for $L \subseteq \Sigma^*$ if, $\forall t \in P_k(L)$ and $s \in \overline{L}$, $P_k(s) \leq t$ implies that there exists a word $u \in \Sigma^*$ s. t. $su \in L$ and $P_k(su) = t$.

# Outline

# Supremal coordination control synthesis

We want to compute

$$supCC(K, L(G_1 \| G_2 \| G_k), (E_{1,u}, E_{2,u}, E_{k,u}))$$

To this end, define

$$\begin{aligned} supC_k &= supC(P_k(K), L(G_k), E_{k,u}) \\ supC_{1+k} &= supC(P_{1+k}(K), L(G_1) \| \overline{supC_k}, E_{1+k,u}) \\ supC_{2+k} &= supC(P_{2+k}(K), L(G_2) \| \overline{supC_k}, E_{2+k,u}) \end{aligned}$$

# Prefix-closed specifications

# If $K$ is prefix-closed

## Theorem

*Let the projection $P_k^{i+k}$ be an $(P_i^{i+k})^{-1}(L_i)$-observer and OCC for $(P_i^{i+k})^{-1}(L_i)$, for $i = 1, 2$. Then,*

$$supC_k \| supC_{1+k} \| supC_{2+k} = supCC(K, L(G_1 \| G_2 \| G_k), (E_{1,u}, E_{2,u}, E_{k,u}))\,.$$

Then $supC_x$ corresponding supervisors; $L_i$ denotes $L(G_i)$

# Non-prefix-closed specifications

# Properties

**Lemma**

*$P_k(supC_{i+k}) \subseteq supC_k$, for $i = 1, 2$.*

If also the opposite inclusion holds, then we immediately have the supremal conditionally-controllable sublanguage.

**Theorem**

*If $supC_k \subseteq P_k(supC_{i+k})$, for $i = 1, 2$, then*

$$supC_{1+k} \| supC_{2+k} = supCC(K, L, (E_{1,u}, E_{2,u}, E_{k,u})) .$$

# Example I

## Example (Concurrent access to a database)

Consider three users with events $r_i, a_i, e_i$. All possible schedules are given by the language of $G = G_1 \| G_2 \| G_3$, where $G_1$, $G_2$, $G_3$ are defined as in the figure, and the set of controllable events is $E_c = \{a_1, a_2, a_3\}$.

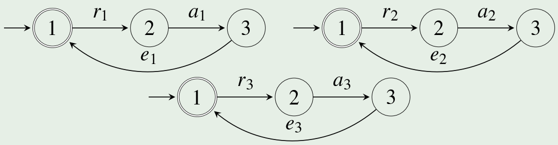



**Figure:** Generators $G_i$, $i = 1, 2, 3$.

# Example II

## Example (Specification)

The specification language $K$, depicted in the figure,

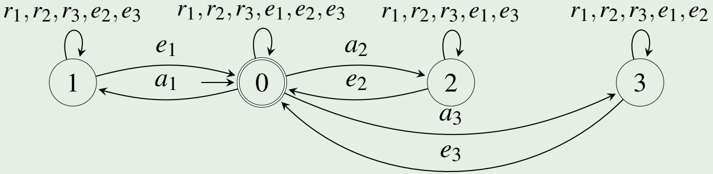



**Figure:** The specification $K$.

describes the correct behavior consisting in finishing the transaction in the exit stage before another transaction can proceed to the exit phase.

# Example III

## Example (Coordinator)

For $E_k = \{a_1, a_2, a_3\}$, the coordinator $G_k = P_k(G_1) \| P_k(G_2) \| P_k(G_3)$.

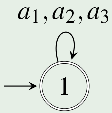



**Figure:** The coordinator $G_k$, where $supC_k = G_k$.

# Example IV

## Example (Supervisors)

$supC_k = G_k$, and compute

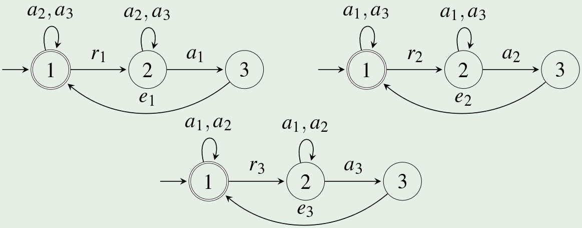



**Figure:** Supervisors $supC_{1+k}$, $supC_{2+k}$, and $supC_{3+k}$.

The solution is optimal: the supremal conditionally-controllable sublanguage of $K$ coincides with the supremal controllable sublanguage of $K$.

# Outline

# Coordinator for nonblockingness

## Theorem

*Consider generators $G_1$ and $G_2$, and assume that $P_k$ is an $L(G_i)$-observer, for $i = 1, 2$. Define $G_k$ as a trim of $P_k(G_1) \| P_k(G_2)$. Then the system $G_1 \| G_2 \| G_k$ is nonblocking.*

## Algorithm (Computation of a Nonblocking Coordinator)

1. Compute $supC_{1+k}$ and $supC_{2+k}$
2. Extend $E_k$ so that $P_k$ is a $supC_{1+k}$- and $supC_{2+k}$-observer
3. Define the nonblocking coord. $C = P_k(supC_{1+k}) \| P_k(supC_{2+k})$

# Conclusion

- Simplification of conditional controllability, use of LCC instead of OCC
- Supremal conditionally controllable sublanguages of general non-prefix-closed languages
- Computation of a coordinator for nonblockingness
- multi-level hierarchy of coordinators
- Applications to decentralized control with communication

# Thank You