Industrial Automation

(Automação de Processos Industriais)

Supervised Control of Discrete Event Systems

http://users.isr.ist.utl.pt/~jag/courses/api1415/api1415.html

Slides 2010/2011 Prof. Paulo Jorge Oliveira
Rev. 2011-2015 Prof. José Gaspar

Syllabus:

...

Chap. 8 - SEDs and Industrial Automation [2 weeks]

Chap. 9 – Supervised Control of SEDs [1 week]

* SCADA
* Methodologies for the Synthesis of Supervision Controllers
* Failure detection

Some jokes available in http://members.iinet.net.au/~ianw/cartoon.html

The End.

Some pointers on Supervised Control of DES

History: The SCADA Web, http://members.iinet.net.au/~ianw/
Monitoring and Control of Discrete Event Systems, Stéphane Lafortune, http://www.ece.northwestern.edu/~ahaddad/ifac96/introductory_workshops.html

Tutorial: http://vita.bu.edu/cgc/MIDEDS/
http://www.daimi.au.dk/PetriNets/

Analysers, and Simulators: http://www.nd.edu/~isis/techreports/isis-2002-003.pdf (Users Manual)
http://www.nd.edu/~isis/techreports/spnbox/ (Software)

Bibliography:
* SCADA books http://www.sss-mag.com/scada.html
* Moody J. e Antsaklis P., **"Supervisory Control of Discrete Event Systems using Petri Nets,"** Kluwer Academic Publishers, 1998.
* Cassandras, Christos G., **"Discrete Event Systems - Modeling and Performance Analysis,"** Aksen Associates, 1993.
* Yamalidou K., Moody J., Lemmon M. and Antsaklis P. **Feedback Control of Petri Nets Based on Place Invariants** http://www.nd.edu/~lemmon/isis-94-002.pdf

Supervision of DES: SCADA

Supervisory

Control

And

Data

Acquisition

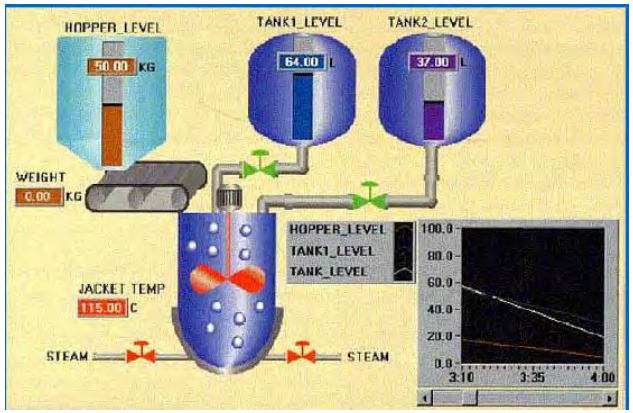
Supervision of DES

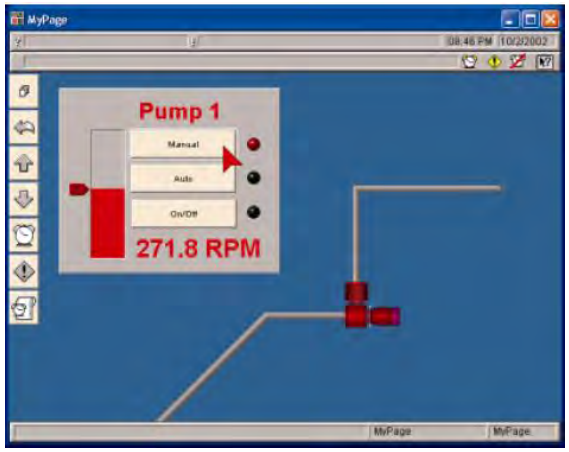
SCADA topics

- Remote monitoring of the state of automation systems
- Logging capacity (resorting to specialized Databases)
- Able to access to *historical* information (plots along time, with selectable periodicity)
- Advanced tools to design Human-Machine interfaces
- Failure Detection and Isolation capacity (*threshold* and/or logical functions) on supervised quantities
- Access control

Supervision of DES

Examples of SCADA





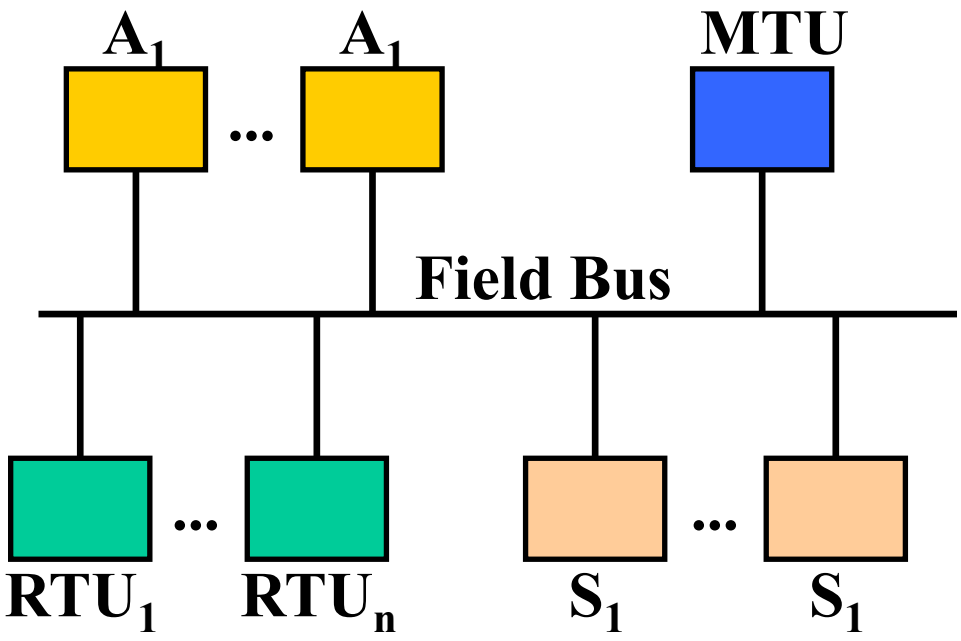
Supervision of DES

Examples of software packages including SCADA solutions

- **Aimax**, de Desin Instruments S.A.
- **CUBE**, Orsi España S.A.
- **FIX**, de Intellution.
- **Lookout**, National Instruments.
- **Monitor Pro**, de Schneider Electric.
- **SCADA InTouch**, de LOGITEK.
- **SYSMAC SCS**, de Omron.
- **Scatt Graph 5000**, de ABB.
- **WinCC**, de Siemens.

Supervision of DES

Hardware Support Architecture of SCADA



Legend:

MTU - Main Terminal Unit

RTU - Remote Term. Unit

S – Sensor

A - Actuator

More terminology: Field Bus (IEC 61158) / PROFIBUS (Field Bus type, Siemens), MODBUS (Schneider)

Supervision of DES

And

Now

Something

Completely

Different

Supervision of DES

Objectives of the Supervised Control

- Supervise and bound the work of the supervised DES
- Reinforce that some properties are verified
- Assure that some states are not reached
- Performance criteria are verified
- Prevent deadlocks in DES
- Constrain on the use of resources (e.g. mutual exclusion)

Supervision of DES

Some history on Supervised Control

- Methods for finite automata [Ramadge et *al.*], 1989
  - some are based on brute-force search (!)
  - or may require simulation (!)

- Formal verification of *software* in Computer Science (since the 60s) and on *hardware* (90, ...)

- Supervisory Control Method of Petri Nets, method based on *monitors* [Giua et *al.*], 1992.

- Supervisory Control of Petri Nets based on Place Invariants [Moody, Antsaklis et *al.*], 1994 (shares some similarities with the previous one, but deduced independently!...).

Supervision of DES

Advantages of the Supervisory Control of Petri Nets

- Mathematical representation is clear (and easy)
- Resorts only to linear algebra (matrices)
- More compact then automata
- Straightforward the representation of infinity state spaces
- Intuitive graphical representation available

The representation of the controller as a Petri Net leads to simplified Analysis and Synthesis tasks

Supervision of DES

Place Invariants

Place invariants are sets of places whose token count remains always constant.

Place invariants can be computed from integer solutions of $x^T D = 0$. Non-zero entries of x correspond to the places that belong to the particular invariant.

Supervisor Synthesis using Place Invariants [ISIS docs]:

What type of relations can be represented in the method of Place Invariants?

- Sets of linear constraints in the state space
- Representation of convex regions (there are extensions for non-convex regions)
- Constraints to guarantee liveness and to avoid deadlocks *(that can be expressed, in general, as linear constraints)*
- Constraints on the events and timings *(that can be expressed, in general, as linear constraints)*

Methods of Analysis/Synthesis

Method of the Matrix Equations (*just to remind*)

The dynamics of the Petri net state can be written in compact form as:

$$\mu(k+1) = \mu(k) + Dq(k)$$

where:

- $\mu(k+1)$ - marking to be reached
- $\mu(k)$ - initial marking
- $q(k)$ - firing vector (transitions)
- D - incidence matrix. Accounts the balance of tokens, giving the transitions fired.

Methods of Analysis/Synthesis

How to build the Incidence Matrix? (*just to remind*)

For a Petri net with *n* places and *m* transitions

$$\mu \in N_0^{\,n}$$

$$q \in N_0^{\,m}$$

$$\boxed{D = D^+ - D^-}, \quad D \in Z^{n\times m}, \quad D^+ \in N_0^{n\times m}, \quad D^- \in N_0^{n\times m}$$

The enabling firing rule is $\boxed{\mu \geq D^- q}$

Can also be written in compact form as the inequality $\mu + Dq \geq 0$, interpreted element-by-element.

Note: unless otherwise stated in this course all vector and matrix inequalities are read element-by-element.

Methods of Synthesis

Some notation for the method

- The supervised system is modelled as a Petri net with n places and m transitions, and incidence matrix

$$D_P \in Z^{n \times m}.$$

- The supervisor is modelled as a Petri net with n_C places and m transitions, and incidence matrix

$$D_C \in Z^{n_C \times m}.$$

- The resulting total system has an incidence matrix

$$D \in Z^{(n+n_C) \times m}.$$

Methods of Synthesis

Theorem: **(T1)**

Synthesis of Controllers based on Place Invariants

Given the set of linear state constraints that the supervised system must follow, written as

$$L\mu_P \le b, \quad \mu_P \in N_0^{\,n}, \quad L \in Z^{n_C \times n} \quad and \quad b \in Z^{n_C}.$$

If $b - L\mu_{P_0} \ge 0$, then the controller with incidence matrix and the initial marking, respectively

$$D_C = -LD_P, \quad \text{and} \quad \mu_{C_0} = b - L\mu_{P_0},$$

enforce the constraints to be verified for all markings obtained from the initial marking.

Methods of Synthesis

Theorem - proof outline :

The constraint $L\mu_P \le b$ can be written as $L\mu_P + \mu_C = b$, using the slack variables μ_C. They represent the marking of the n_C places of the controller.

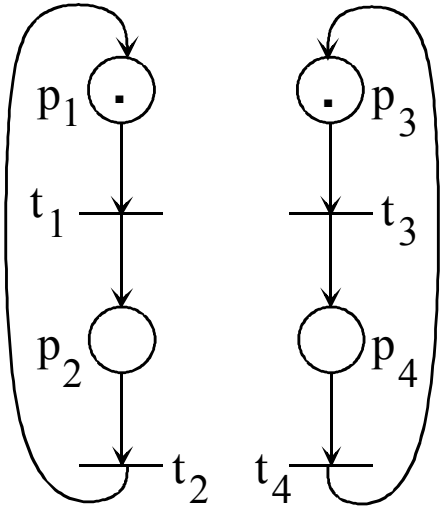
To have a place invariant, the relation $x^T D = 0$ must be verified and in particular, given the previous constraint:

$$x^T D = \begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_P \\ D_C \end{bmatrix} = 0, \text{ resulting } \boxed{D_C = -LD_P.}$$

From $L\mu_{P_0} + \mu_{C_0} = b$, follows that $\boxed{\mu_{C_0} = b - L\mu_{P_0}.}$

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion



Linear constraint: $\mu_2 + \mu_4 \leq 1$

That can be written as:

$$L\mu_P \leq b \qquad \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \leq 1.$$

Incidence Matrix $D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$ and initial marking $\mu_{P_0} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion

1) Test

$$b - L\mu_{P_0} = 1 - \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = 1 \geq 0.$$

OK.

2) Compute

$$D_C = -LD_P = -\begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix},$$

and

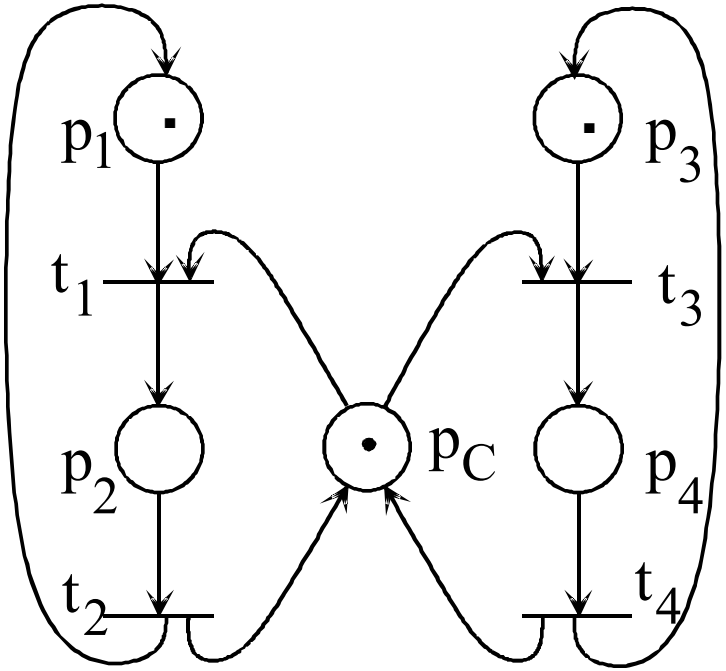
$$\mu_{C_0} = b - L\mu_{P_0} = 1.$$

OK.

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

OK.
UAU!!!.

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion

```
% The Petri net D=Dp-Dm, and m0
%  (Dplus-Dminus= Post-Pre)

Dm= [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1];

Dp= [0 1 0 0;
     1 0 0 0;
     0 0 0 1;
     0 0 1 0];

m0= [1 0 1 0]';

% Supervisor constraint
%
L= [0 1 0 1];
b= 1;

% Computing the supervisor
%
[Dfm, Dfp, ms0] = linenf(Dm, Dp, L, b, m0);
Df= Dfp-Dfm
ms0
```

Result using the function LINENF.m of the toolbox SPNBOX:

```
Df =

    -1     1     0     0
     1    -1     0     0
     0     0    -1     1
     0     0     1    -1
    -1     1    -1     1


ms0 =

     1
     0
     1
     0
     1
```

Methods of Synthesis

Definition:

Maximal permissivity occurs when (i) all the linear constraints are verified and (ii) all legal markings can be reached.

Lemmas:

L1) **The controllers** obtained with T1 **have maximal permissivity.**

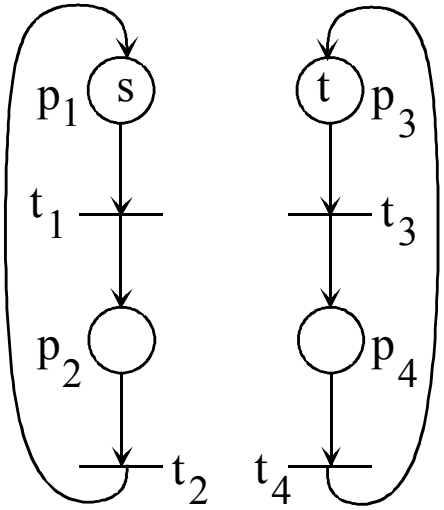
L2) Given the linear constraints used, **the place invariants** obtained with the controller synthesized with T1 **are the same** as the invariants associated with the initial system.

Methods of Synthesis

Example of controller synthesis

$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$

Readers / Writers



Linear constraints for *n* books: $\mu_2 + n\mu_4 \leq n$

That can be written as:

$$L\mu_P \leq b \qquad \begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \leq n.$$

Incidence Matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

and initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Methods of Synthesis

Example of controller synthesis

Readers / Writers

1) Test

$$b - L\mu_{P_0} = n - \begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix} = n \geq 0.$$

OK.

2) Compute

$$D_C = -LD_P = -\begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -n & n \end{bmatrix},$$

and

$$\mu_{C_0} = b - L\mu_{P_0} = n.$$

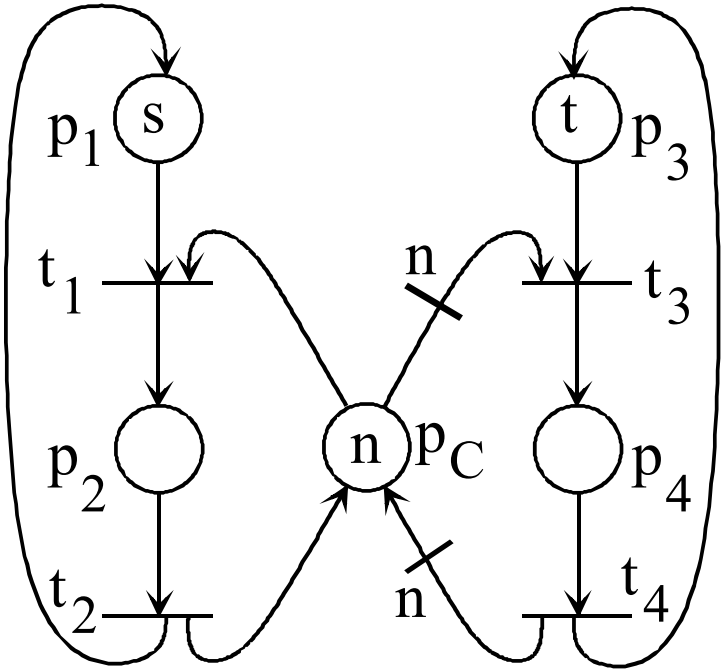
OK.

Methods of Synthesis

Example of controller synthesis

Readers / Writers

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ -1 & 1 & -n & n \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ n \end{bmatrix}$$

OK.
UAU!!!.

Supervision of DES

Advantages of the Method of the Place Invariants [ISIS docs]:

Other characteristics that can impact on the solutions?

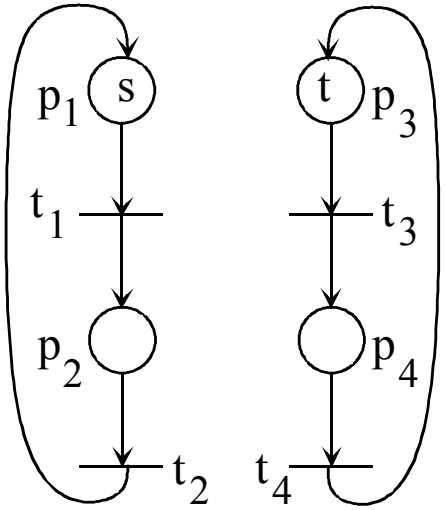
- Existence and uniqueness
- Optimality of the solutions (e.g. maximal permissivity)
- Existence of transition non-controllable and/or not observable (remind definitions for time-driven systems)

In general the solutions can be found solving:

Linear Programming Problems, with Linear Constraints

Methods of Synthesis

Example of controller synthesis: s Producers / t Consumers



Incidence matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Let p2= #machines working, t2= product produced
p3= #consumers, t3= request to consume (e.g. transport product)

Q: How to write *consume only when produced* ? What is the linear constraint?

Not possible to write it as a linear constraint on places $L\mu_p \leq b$.
Is it impossible to solve this problem with the proposed method ?

Methods of Synthesis

Generalized linear constraint

Let the generalized linear constraint be

$$L\mu_P + Fq_P + Cv_P \le b,$$
$$\mu_P \in N_0^{\,n}, v_P \in N_0^{\,m}, q_P \in N_0^{\,m},$$
$$L \in Z^{n_C \times n}, F \in Z^{n_C \times m}, C \in Z^{n_C \times m}, e \quad b \in Z^{n_C},$$

where

* μ_P is the marking vector for system P;
* q_P is the firing vector since t_0;
* v_P is the number of transitions (firing) that can occur, also designated as Parikh vector.

Methods of Synthesis

Function LINENF of SPNBOX

Theorem: Synthesis of Controllers based on Place Invariants, for Generalized Linear Contraints

Given the generalized linear constraint $L\mu_P + Fq_P + Cv_P \le b,$

if $b - L\mu_{P_0} \ge 0,$ then the controller with incidence matrix and initial marking, respectively

$$D_C^- = \max\left(0,\, LD_P + C,\, F\right)$$
$$D_C^+ = \max\left(0,\, F - \max\left(0, LD_P + C\right)\right) - \min\left(0,\, LD_P + C\right),$$

$$\mu_{C_0} = b - L\mu_{P_0} - Cv_{P0},$$

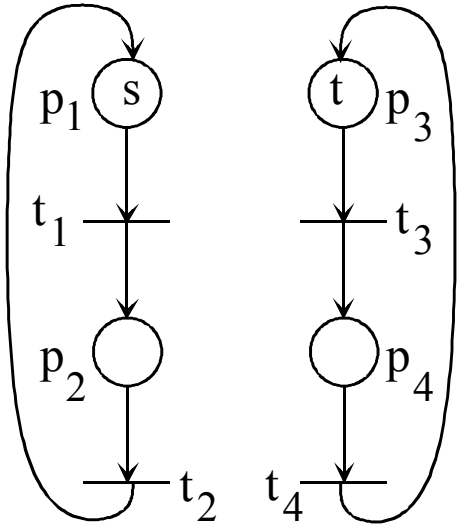
guarantees that constraints are verified for the states resulting from the initial marking.

Methods of Synthesis

Example of controller synthesis

$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$

Producer / Consumer



Linear constraint: $v_3 \leq v_2$

That can be written as:

$$Cv_P \leq b$$

$$L = 0, F = 0$$

$$\begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \leq 0.$$

Incidence matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Methods of Synthesis

Example of controller synthesis

Producer / Consumer

1) Test $b - L\mu_{P_0} = 0 - 0 \geq 0.$ **OK.**

2) Compute

$$D_C^- = \max(0, LD_P + C, F)$$
$$D_C^+ = \max(0, F - \max(0, LD_P + C)) - \min(0, LD_P + C),$$

$$D_C^- = \max(0, [0\ -1\ 1\ 0], 0) = [0\ 0\ 1\ 0]$$
$$D_C^+ = \max(0, -[0\ 0\ 1\ 0]) - \min(0, [0\ -1\ 1\ 0])$$
$$= [0\ 0\ 0\ 0] - [0\ -1\ 0\ 0] = [0\ 1\ 0\ 0]$$

and $\mu_{C_0} = b - L\mu_{P_0} - Cv_{P0},$

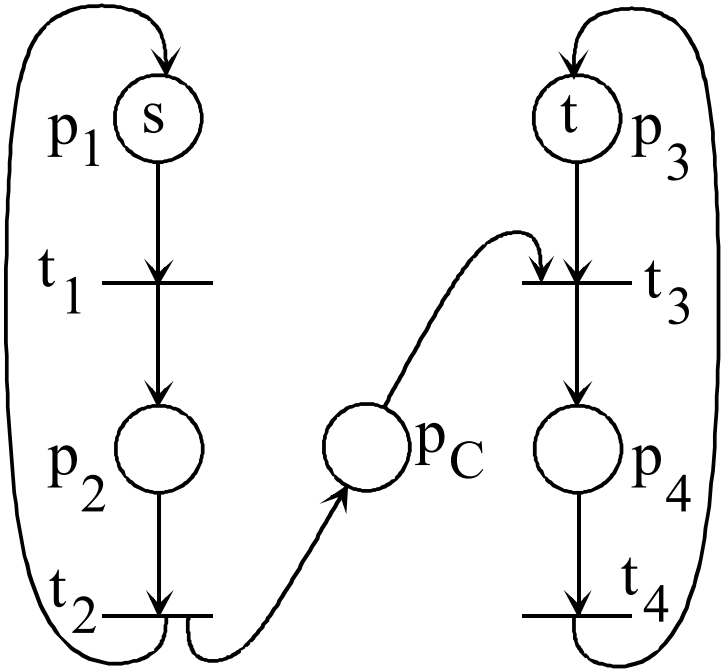
$\mu_{C_0} = b - L\mu_{P_0} = 0 - 0 = 0.$ **OK.**

Methods of Synthesis

Example of controller synthesis

Producer / Consumer

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ 0 \end{bmatrix}$$

OK.
UAU!!!.

Methods of Synthesis

```
% The Petri net D=Dp-Dm, and m0
%  (Dplus-Dminus= Post-Pre)

Dm= [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1];

 Dp= [0 1 0 0;
      1 0 0 0;
      0 0 0 1;
      0 0 1 0];

m0= [1 0 1 0]';

% Supervisor constraint
%
L= []; F= [];  C= [0 -1 1 0];
b= 0;

% Computing the supervisor
%
[Dfm, Dfp, ms0] = linenf(Dm, Dp, L, b, m0, F, C)
Df= Dfp-Dfm
ms0
```

Example of controller synthesis: Producer Consumer

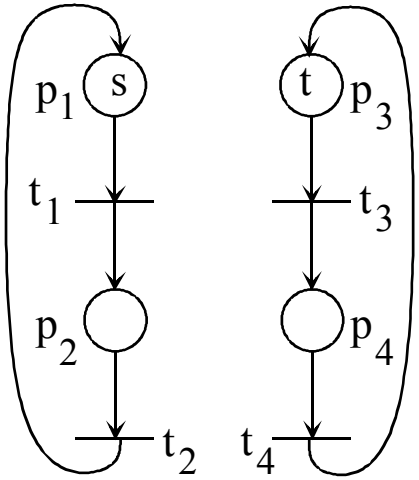
Result using the function LINENF.m of the toolbox SPNBOX:

```
Df =

    -1     1     0     0
     1    -1     0     0
     0     0    -1     1
     0     0     1    -1
     0     1    -1     0


ms0 =

     1
     0
     1
     0
     0
```

Methods of Synthesis

Example of controller synthesis

Bounded Producer / Consumer



Incidence matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

TWO linear constraints:

$$\begin{cases} v_3 \le v_2 \\ v_2 \le v_3 + n \end{cases} \Leftrightarrow \begin{cases} v_3 - v_2 \le 0 \\ v_2 - v_3 \le n \end{cases}$$

$$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$$

The two linear constraints can be written as:

$$Cv_P \le b \quad i.e.\ L = 0, F = 0 \Leftrightarrow \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \le \begin{bmatrix} 0 \\ n \end{bmatrix}$$

Methods of Synthesis

Example of controller synthesis

Bounded Producer / Consumer

1) Test

$$b - L\mu_{P_0} = b = \begin{bmatrix} 0 \\ n \end{bmatrix} \geq 0.$$

OK.

2) Compute

$$D_C^- = \max\left(0, \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}, 0\right) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$D_C^+ = \max\left(0, 0 - \max\left(0, \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}\right)\right) - \min\left(0, \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

and

$$\mu_{C_0} = b - L\mu_{P_0} = \begin{bmatrix} 0 \\ n \end{bmatrix}.$$

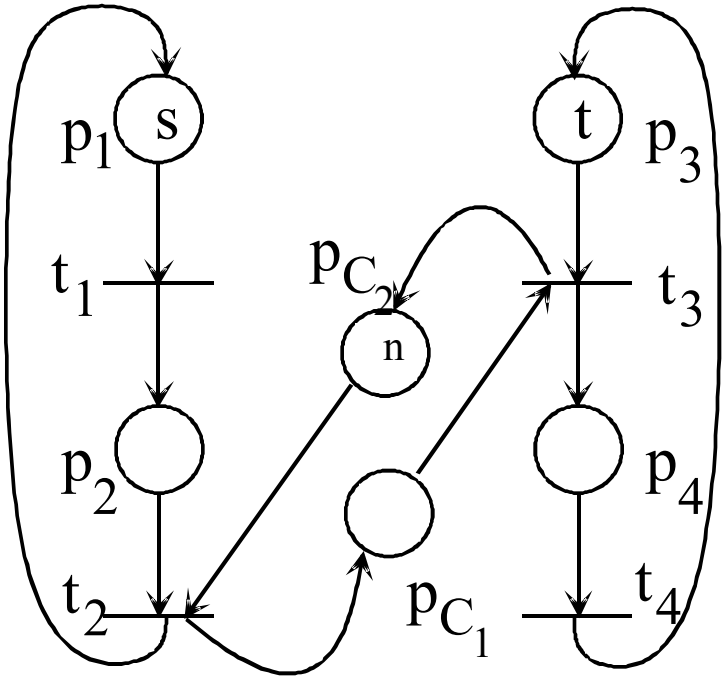
OK.

Methods of Synthesis

Example of controller synthesis

Bounded Producer / Consumer

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ 0 \\ n \end{bmatrix}$$

OK.
UAU!!!.

Methods of Synthesis:
adding Uncontrollable and Unobservable transitions

Definition of **Uncontrollable Transition:**

A transition is uncontrollable if its firing **cannot be inhibited** by an external action (e.g. a supervisory controller).

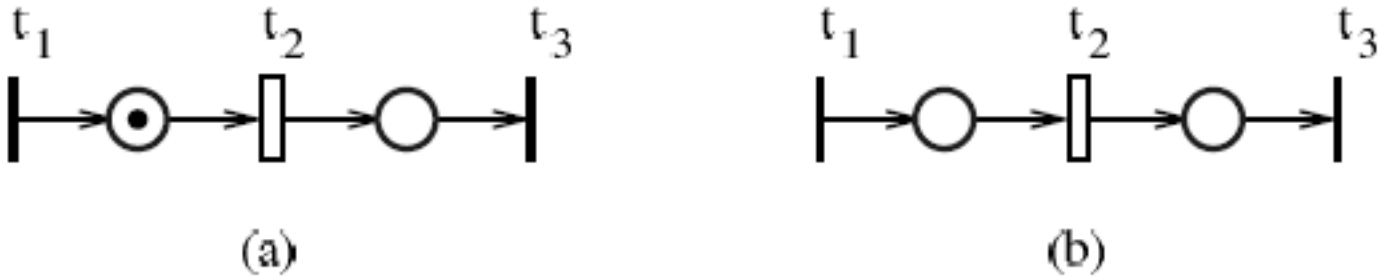
Definition of **Unobservable Transition:**

A transition is unobservable if its firing **cannot be detected or measured** (therefore the study of any supervisory controller can not depend from that firing).

Proposition:

In a Petri net based controller, both input and output arcs to/from plant transitions are used to trigger state changes in the controller. *Since a controller can not have arcs connecting to unobservable transitions, then all* ***unobservable transitions are also implicitly uncontrollable****.*

Methods of Synthesis:
adding Uncontrollable and Unobservable transitions



Assuming that **t1 is controllable** and **t2 is uncontrollable**:
- case (a) t2 cannot be directly inhibited; it will eventually fire
- case (b) **t2 can be indirectly prevented** from firing by inhibiting t1.

Assuming that **t2 is unobservable** and **t3 is observable**, i.e. we cannot detect when t2 fires. The state of a supervisor is not changed by firing t2. However we can **indirectly detect that t2 has fired**, by detecting the firing of t3.

Methods of Synthesis

Definition: A **marking μ_P** is **admissible** if

i) $L\mu_P \leq b$ and **ii)** $\forall \mu' \in R(C, \mu_P)$ verifies $L\mu' \leq b$

Definition: A **Linear Constraint (L, b)** is **admissible** if

i) $L\mu_{Po} \leq b$ and

ii) $\forall \mu' \in R(C, \mu_{Po})$ such that $L\mu' \leq b$

μ' is an admissible marking.

Note: ii) indicates that the firing of uncontrollable transitions can never lead from a state that satisfies the constraint to a new state that does not satisfy the constraint.

Methods of Synthesis

Proposition: Admissibility of a constraint

A linear constraint is admissible ***iff***

- The initial markings satisfy the constraint.
- There **exists a controller** with maximal permissivity that forces the constraint and **does not inhibit any uncontrollable transition**.

Two sufficient (not necessary) conditions:

Corollary: given a system with uncontrollable transitions,

$\boxed{l^T D_{uc} \leq 0}$ implies admissibility.

Corollary: given a system with unobservable transitions,

$\boxed{l^T D_{uo} = 0}$ implies admissibility.

Methods of Synthesis

Function MRO_ADM of SPNBOX

Lemma: Structure of Constraint transformation

Let $R_1 \in Z^{n_C \times n}$ *such that* $R_1 \mu_P \geq 0,$

$R_2 \in Z^{n_C \times n_C}$ be a matrix with positive elements in the diagonal,

If there exists $L' = R_1 + R_2 L$

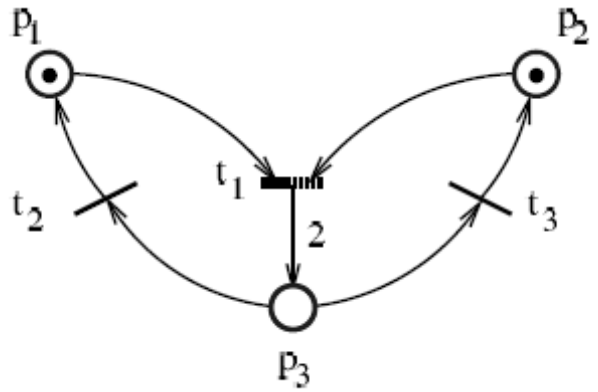
$b' = R_2 (b+1) - 1,$

such that $L' \mu_P \leq b'$

then it is also verified that $L \mu_P \leq b.$

Methods of Synthesis

Example: design controller with t1 unobservable (1/3)



$$D = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 2 & -1 & -1 \end{bmatrix}, \quad D_{uo} = \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}$$

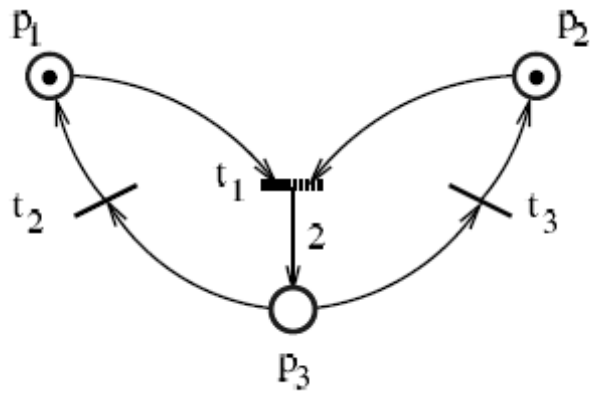
Objectives: $\mu_1 + \mu_3 \geq 1$ and $\mu_2 + \mu_3 \geq 1$ which can be written in matrix form as

$$L = -\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad b = -\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Example extracted from "Supervised Control of Concurrent Systems: A Petri Net Structural Approach", M. Iordache and P. Antsaklis, Birkhauser 2006.

Methods of Synthesis

Example: design controller with t1 unobservable (2/3)




```
D= [-1  1  0;
    -1  0  1;
    +2 -1 -1];
Tuo= 1;
Tuc= [];

L= -[1 0 1; 0 1 1];
b= -[1; 1];

[La, ba, R1, R2, how, dhow] = mro_adm(L, b, D, Tuc, Tuo)
```

Solution obtained with the function MRO_ADM.m of the SPNBOX toolbox:

```
La =
    -2     0    -1
     0    -2    -1

ba =
    -1
    -1
```

$$L_a = -\begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}, \quad b_a = -\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Note: verify that $L_a \mu \leq b_a$ implies $L\mu \leq b$

Methods of Synthesis

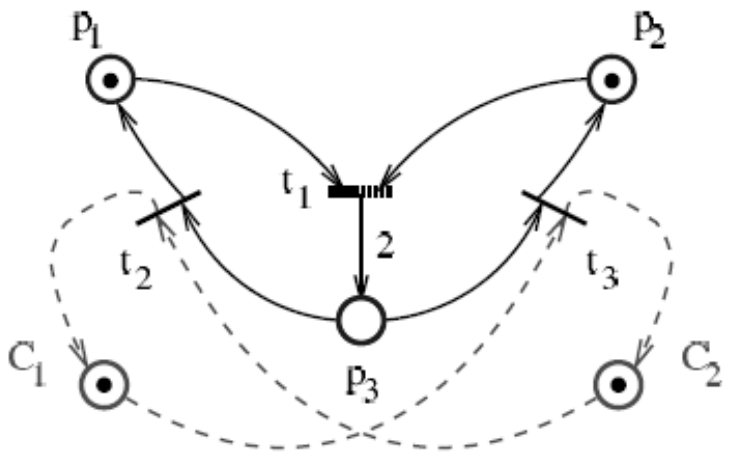
Example: design controller with t1 unobservable (3/3)

Finally the supervised controller is simply obtained from L_a and b_a:

$$D_c = -L_a D_p$$

$$= \begin{bmatrix} -2 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 2 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$



$$\mu_{c0} = b_a - L_a \mu_{p0}$$

$$= \begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} -2 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

This course is ending. What is next?

This course is ending .

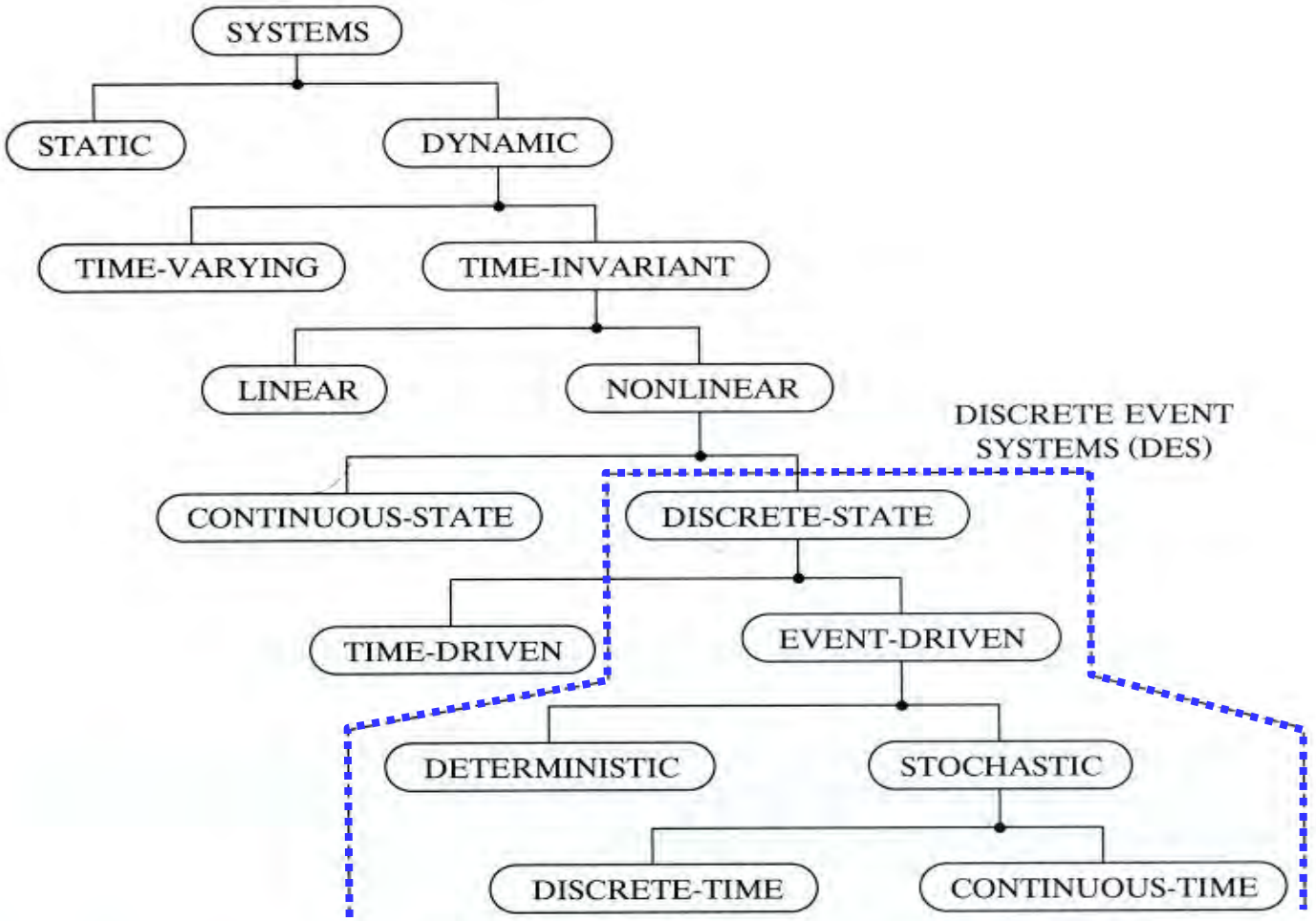


Figure 1.29. Major system classifications.

What is next ?

Top 10 Challenges in Logic Control for Manufacturing Systems

by Dawn Tilbury from University of Michigan

10. Distributed Control	(General management of distributed control applications, Open/distributed control -- ethernet-based control)
9. Theory	(No well-developed and accepted theory of discrete event control, in contrast to continuous control)
8. Languages	(None of the programming languages do what we need but nobody wants a new programming language)
7. Control logic synthesis	(automatically)
6. Standards	(Machine-control standards -- every machine is different, Validated standards, Standardizing different types of control logic programming language)
5. Verification	(Standards for validation, Simulation and verification of controllers)
4. Software	(Software re-usability -- cut and paste, Sophisticated software for logic control, User-unfriendly software)
3. Theory/Practice Gap	(Bridging the gap between industry and academia, Gap between commercial software and academic research)
2. Education	(Educating students for various PLCs, Education and keeping current with evolution of new control technologies, Education of engineers in logic control, Lack of curriculum in discrete-event systems)
	And the number one challenge in logic control for manufacturing systems is...
1. Diagnostics	(Integrating diagnostic tools in logic control, Standardized methodologies for design, development, and implementation of diagnostics)

The End .