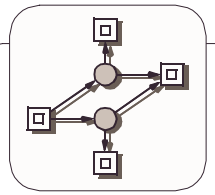


BrandenburgTechnical
University at Cottbus,
Computer Science Institute

CONCURRENCY PATTERNS
-
A PETRI NET PERSPECTIVE

-- WORK IN PROGRESS --

MONIKA HEINER

mh@informatik.tu-cottbus.de
http://www.informatik.tu-cottbus.de

HIGHLY COMPETITIVE COMPETITION

- my new car !

- my new software toolkit ?

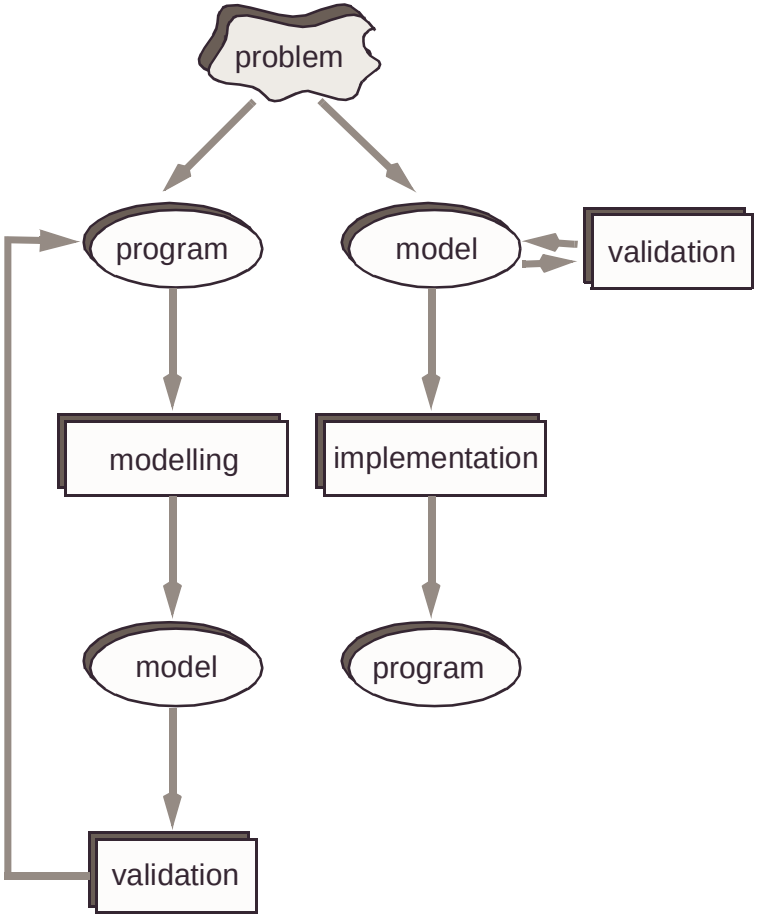
-> IEC 61508
Functional safety of
electrical/electronic/programmable electronic
safety-related systems

Fault Prevention

- Gamma, E. et al.:
  Design Patterns: Elements of Reusable Object-Oriented Software; Addison Wesley 1994.
- Fowler, M.:
  Analysis Patterns, Reusable Object Models; Addison-Wesley 1997.
- Grand, M.:
  Patterns in Java, Vol. 1, A Catalog of Reusable Design Patterns Illustrated with UML; Wiley 1998.
- . . .

- Rising, L. (ed):
  Design Patterns in Communication Software; Cambridge Univ. Press 2001.
- Pont, M. J.:
  Patterns for Time-Triggered Embedded Systems; Addison-Wesley 2001.

Software Engineering & Models, Two Approaches



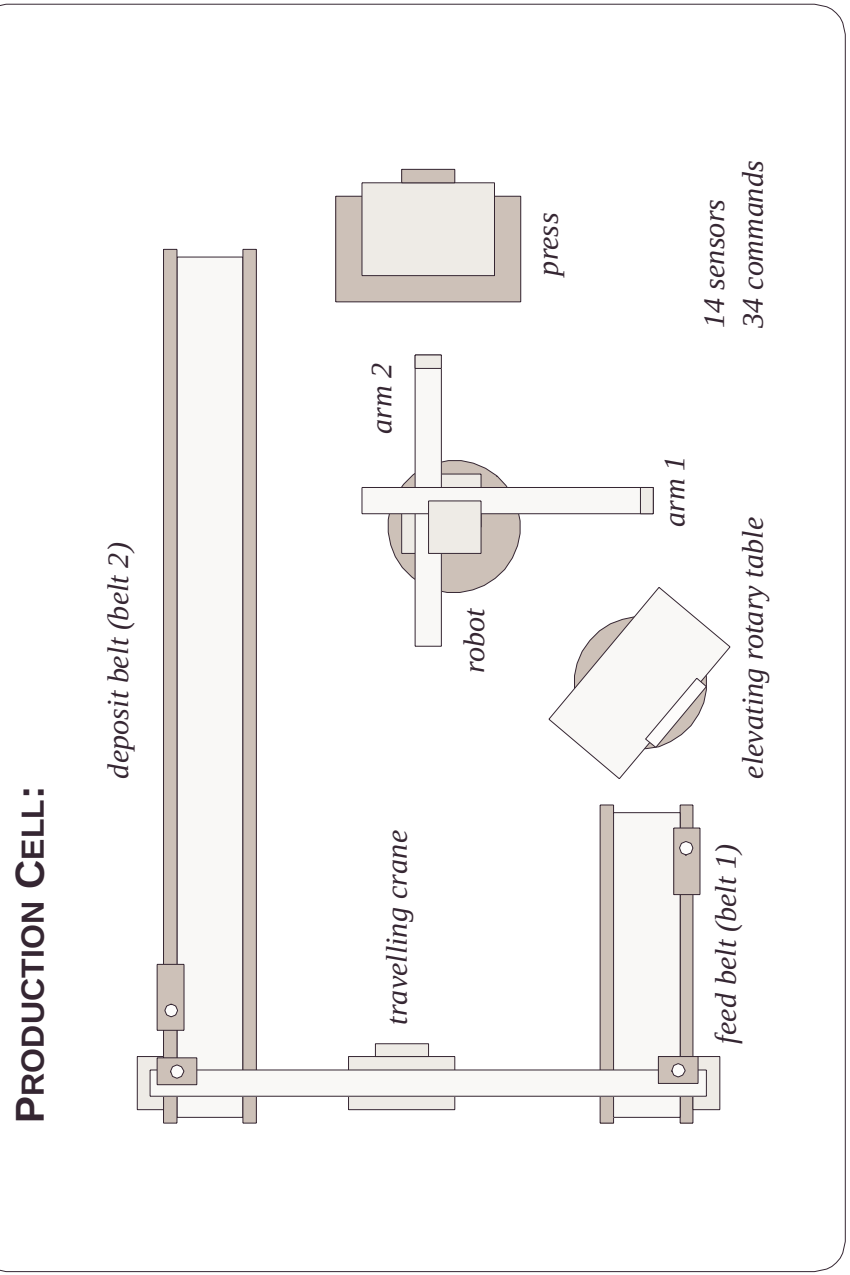
CASE STUDIES:

ACADEMIC:

- low-level mutex algorithms
- Dijkstra's philosophers
- Milner's schedulers
- solitaire
- . . .

MORE REALISTIC

- production cell, Karlsruhe
- production cell, Cottbus
- concurrent pushers
- 2-hand switch
- plc press controller
- . . .



INFORMAL SAFETY REQUIREMENTS (Σ 21):

- The press must not be moved downward,
  if sensor 1 is true, and
  t must not be moved upward, if sensor 3 is true.
  -> *Restrictions of machine mobility.*

- The press may only be closed,
  when no robot arm is positioned inside it.
  -> *Avoidance of machine collisions.*

- The feed belt may only convey a blank through its light barrier, if the table is in loading position.
  -> *Blanks are not dropped outside safe areas.*

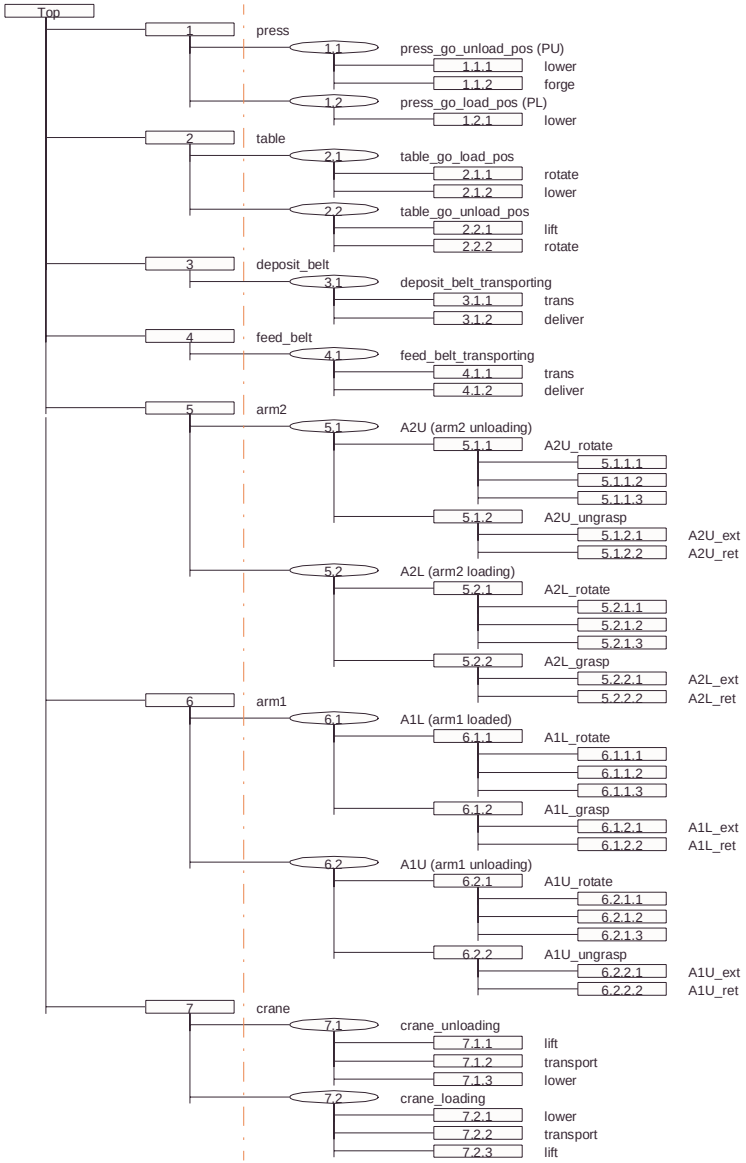
- Blanks may not be put into the press, if it is already loaded.
  -> *Insurance of a sufficient distance between consecutively processed blanks.*

additional requirements related to design consistency:

- The robot swivel is either stopped or moves in exactly one direction.

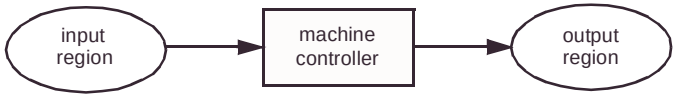
- ...

NET HIERARCHY:

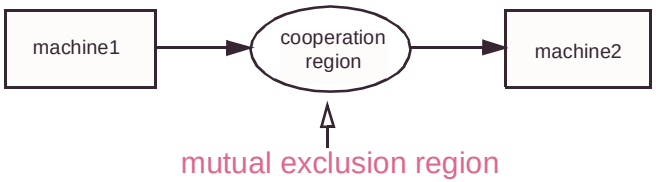


Cooperation Model, Basic Design Principles:

- production cell = pipeline of machines
- each machine
  - takes plates from some input places;
  - processes them;
  - puts plates on some output places;

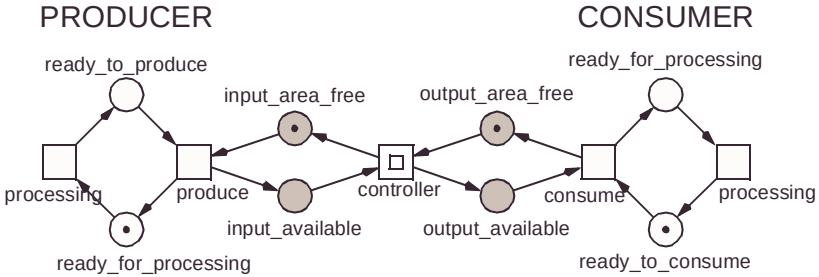


- cooperation region between two consecutive machines

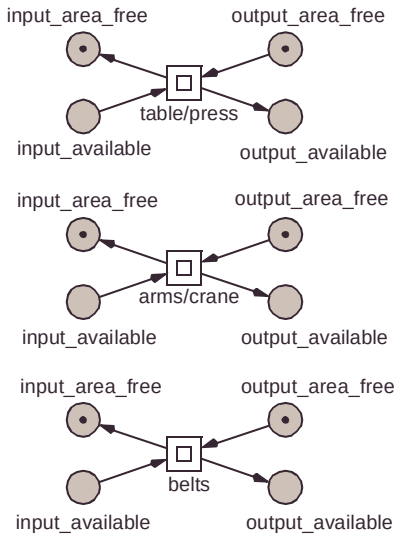


- mutual exclusive shared resources
  - robot swivel
    - (to rotate both arms)
  - physical regions
    - (intersection of trajectories of different machines)

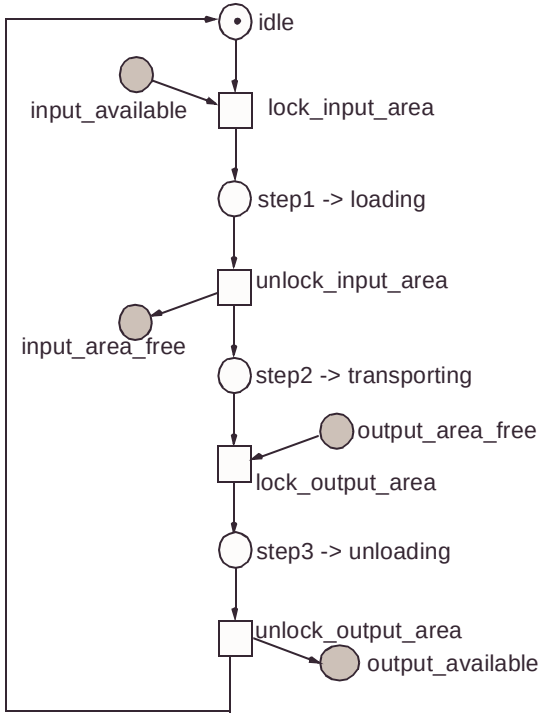
Bounded Producer Consumer Pattern:



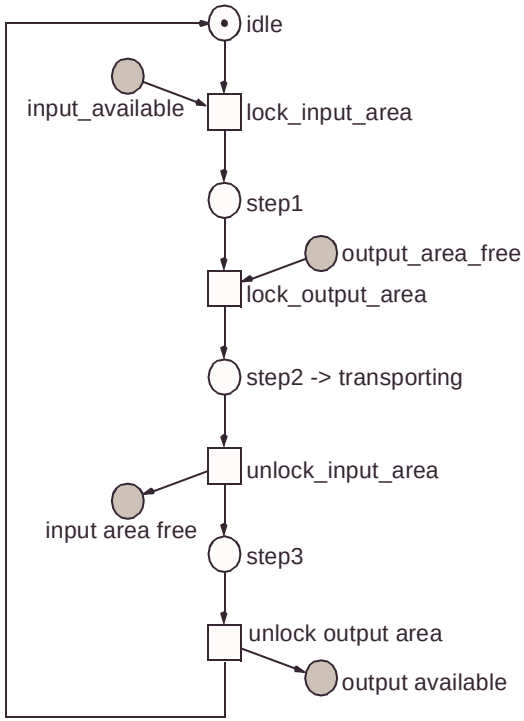
THREE TYPES OF COOPERATION PATTERN



(A) Independent Input/Output

- arms/crane:
  step-wise synchronization with only one of the adjacent controllers,
- pattern property, e.g.
  $\mathbf{G_A}(step1 \rightarrow \neg(input\_available \vee input\_area\_free))$
- 

(B) Dependent Input/Output

- belts:
  simultaneous control of input and output region necessary,
- pattern property
  $\mathbf{G_A}(step2 \rightarrow \neg(input\_available \vee input\_area\_free \vee output\_area\_free \vee output\_available))$
- 

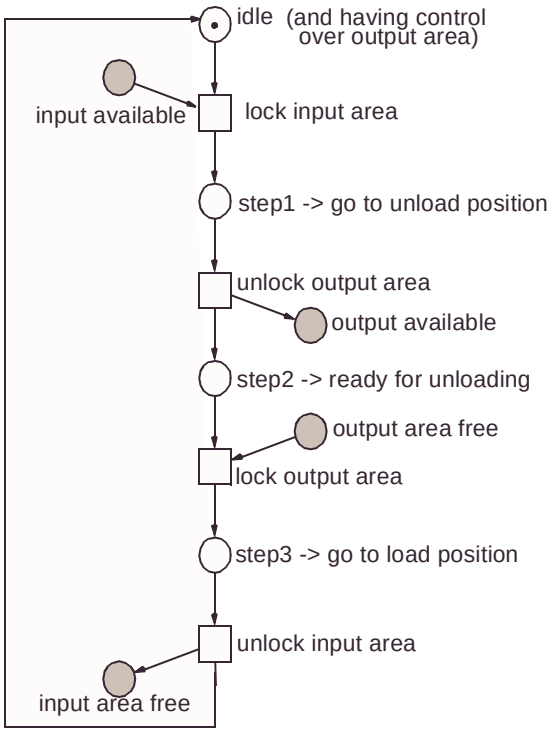
(C) Mutually Exclusive Input/Output

- table/press:
  the controller must always hold a lock on one of its cooperation regions;

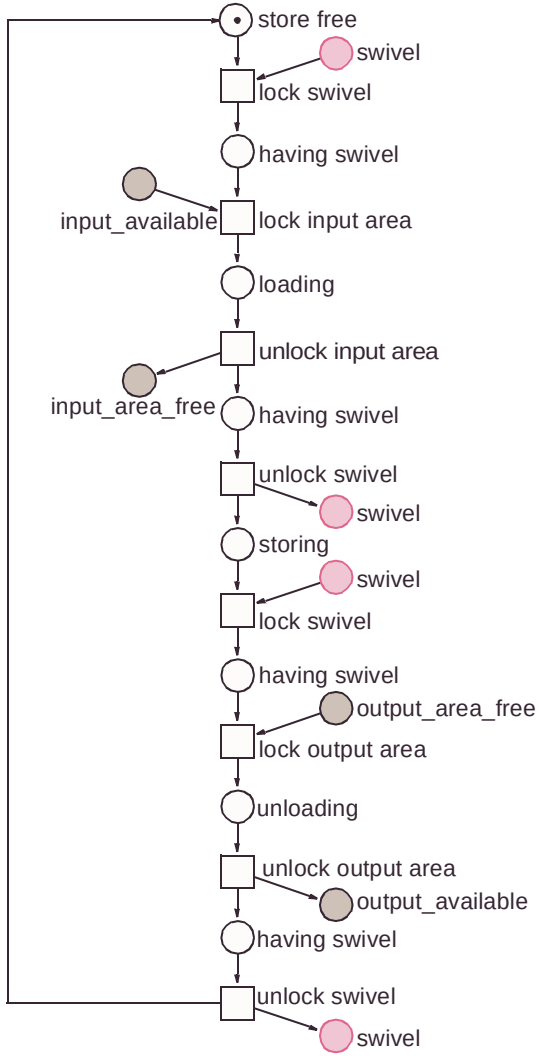
- pattern property

  $\mathbf{G_A}(\neg(input\_available \vee input\_area\_free) \vee$
  $\neg(output\_available \vee output\_area\_free))$

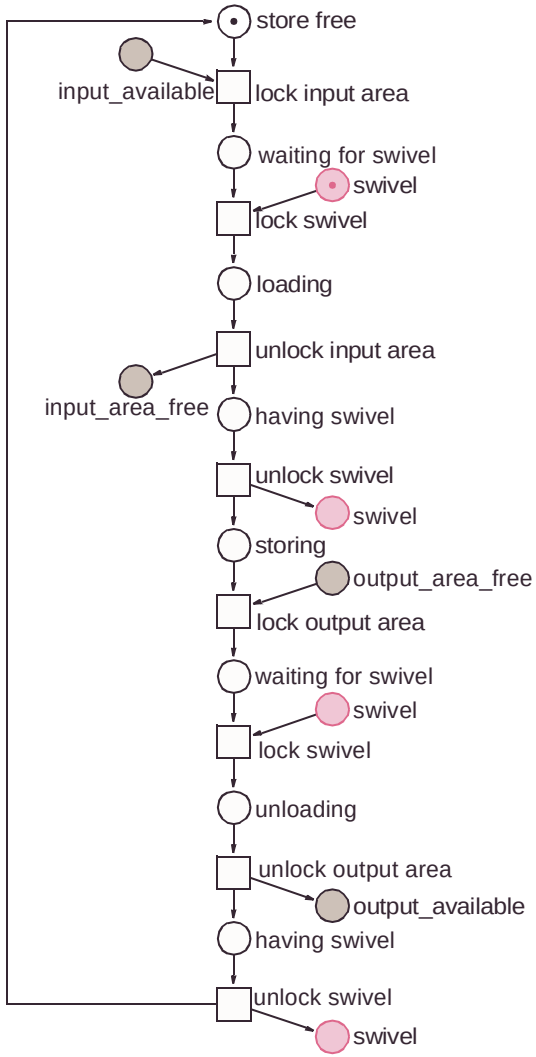
-



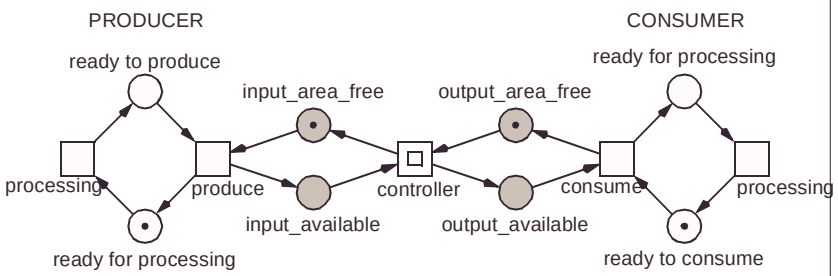
Arm Version2:



ARM VERSION3



CONTROLLER ANALYSIS:



ARMS

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	

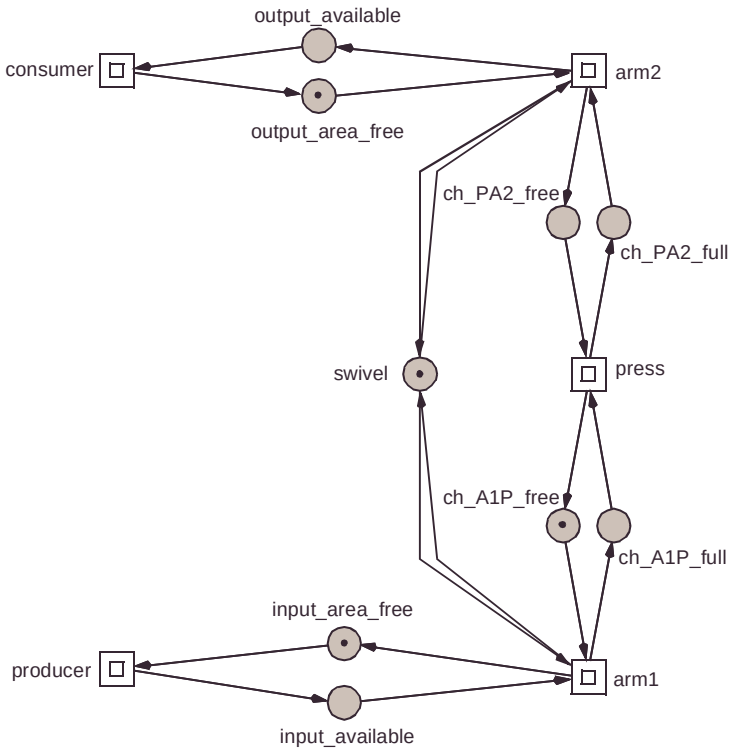
ELSE

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	Y	Y	Y	N	N	N	N	Y	N	Y	Y	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	

STEP-WISE COMPOSITION:

E.G. SUBSYSTEM: ARM1 - PRESS - ARM2

(ARMS: VERSION2)



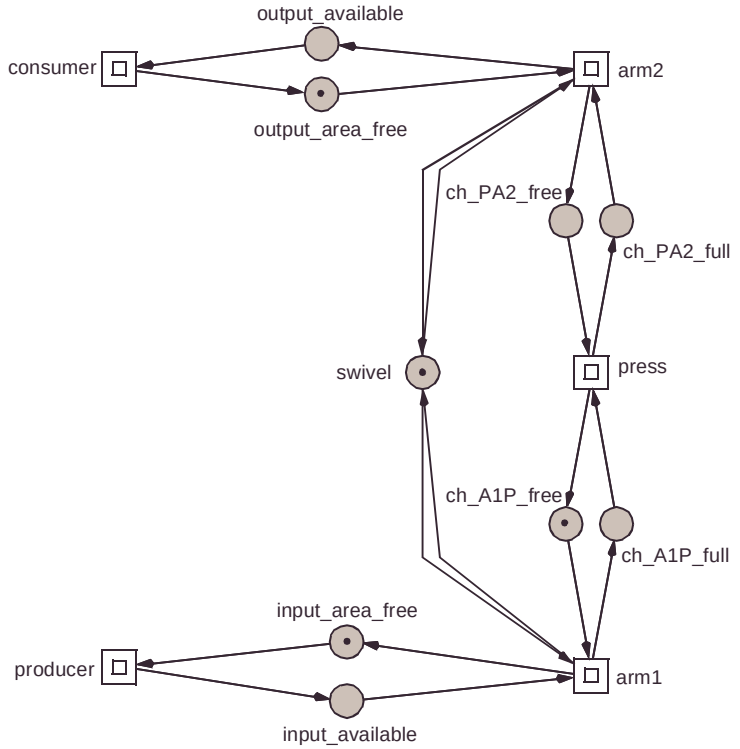
ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y

DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
Y	Y	Y	N	Y	Y	Y	Y	N	Y	N	N	?	N	N	N

STEP-WISE COMPOSITION:

E.G. SUBSYSTEM: ARM1 - PRESS - ARM2

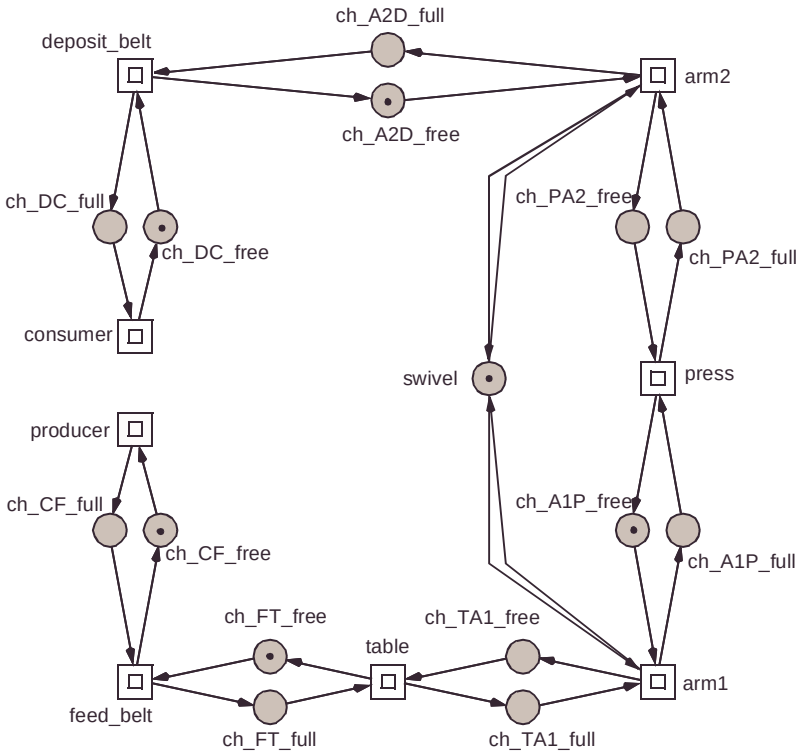
(ARMS: VERSION3):



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y

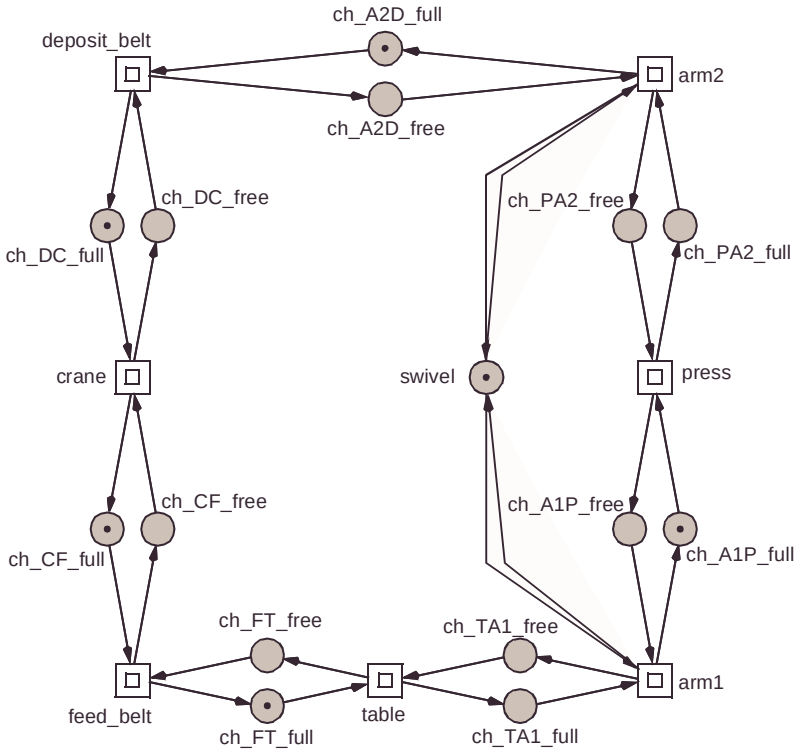
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y

OPEN SYSTEM COARSE STRUCTURE:



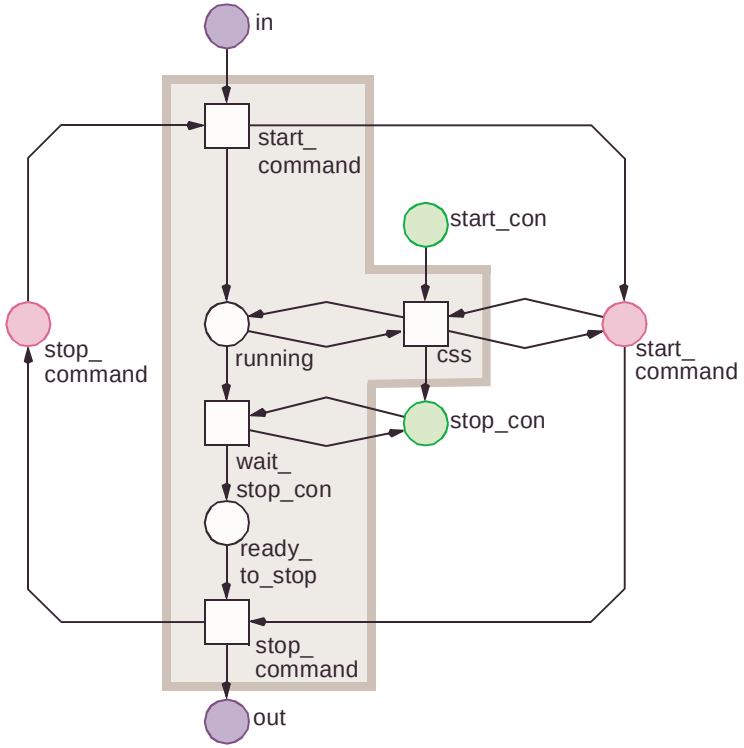
ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	

CLOSED SYSTEM COARSE STRUCTURE:



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	

Basic Motion Step + Environment:

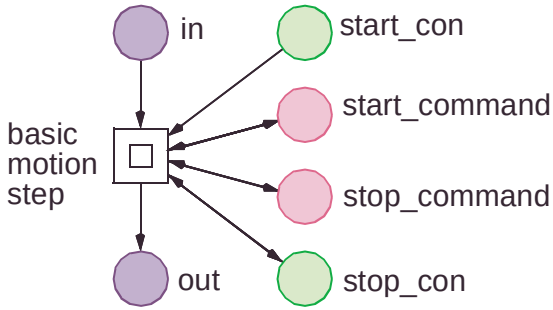


fusion nodes:

- interface
- actuator states
- sensor states

css - change sensor state

Macro Net of Basic Motion Step:



formal parameters

actual parameters, e.g.:

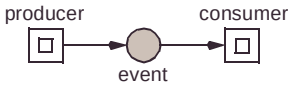
press_forge	press_lift
press_at_middle_pos	press_at_lower_pos
press_upward	press_up
press_stop	press_stop
press_at_upper_pos	press_at_middle_pos

Main Lessons Learnt

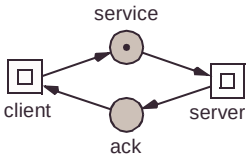
- management of medium-sized Petri nets
  -> *hierarchical structure + fusion nodes;*
- the whole model is composed of a few patterns
  -> *bounded producer/consumer pattern*
  -> *communication patterns for producer/consumer pipeline*
    - *independent input/output*
    - *dependent input/output*
    - *mutually exclusive input/output*
  -> *mutex pattern*
  -> *basic motion step pattern*
    - *sequence*
    - *alternative*
- new editor feature: parameter substitution
  -> *library of reusable Petri net components;*
- interleaving rule of communication and mutex synchronisation
  -> *lock a mutex resource always as late as possible*
- pattern properties
  -> *model consistency criteria*
  -> *to be generated for each instance*

Other Concurrency Patterns

- n mutex resource pattern
  -> *n=2: dining philosophers*
  -> *pattern property (deadlock freedom): acyclic access structure, hierarchical resource organisation [Brinch Hansen]*
- unbounded producer/consumer pattern



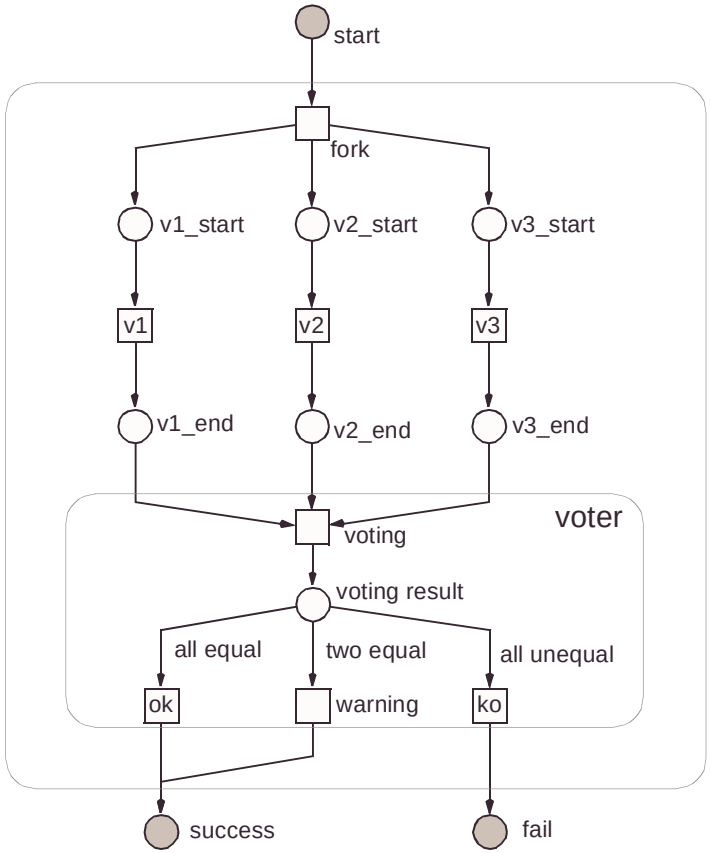
- client/server communication pattern



- n layered client/server pattern
  -> *pattern property (deadlock freedom): acyclic communication structure*
- fault-tolerant basic structures ?
  -> *n version programming*
  -> *recovery block scheme*

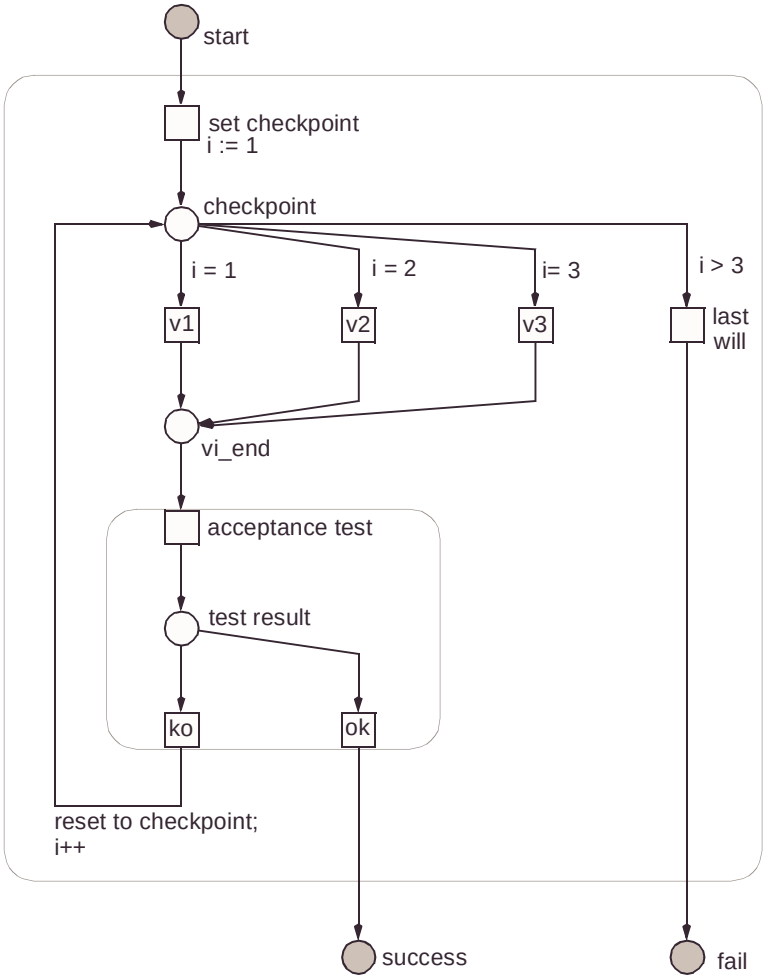
N VERSION PROGRAMMING

-> parallel execution of n program versions, followed by majority test



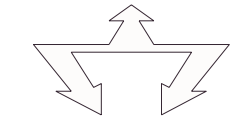
RECOVERY BLOCK SCHEME

-> alternative execution of n program versions, each followed by acceptance test



CONCLUSIONS

- catalogue of concurrency patterns
- step-wise system development
  +
  step-wise specification of system properties



required -> granted

- properties taxonomy

taxonomy I

general properties
boundedness
liveness
special properties
safety properties
progress properties
model consistency properties

taxonomy II

"must" properties -> fatal errors
"maybe" properties -> warnings
"fun" properties -> insights

PROPERTY TAXONOMY II

- **FATAL ERRORS**

e.g. safety properties

If a robot arm is loaded, its magnet is not deactivated until the robot is in its unloading position.

$\mathbf{G}(\varphi \rightarrow \neg\chi\mathbf{U}\psi)$, *where*

φ = arm1_mag_on
$\wedge$ arm1_pickup_angle
$\wedge$ arm1_pickup_ext
χ = arm1_mag_off
ψ = arm1_release_angle $\wedge$ arm1_release_ext

- **WARNINGS**

e.g. liveness

***AG(EF** en(t)) for each transition t*

- **INSIGHTS**

Is it possible, that both robot arms carry a plate at the same time?

$\mathbf{EF}(\text{arm1\_mag\_on} \wedge \text{arm2\_mag\_on})$

CHALLENGES

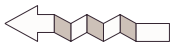
- structured (sequential) programming [Dijkstra]
  -> *to avoid uncontrolled use of goto's*

STRUCTURED CONCURRENT PROGRAMMING

-> *to avoid uncontrolled use of synchronisation / communication*

- problem frames

architecture styles



architecture patterns

design patterns

idioms