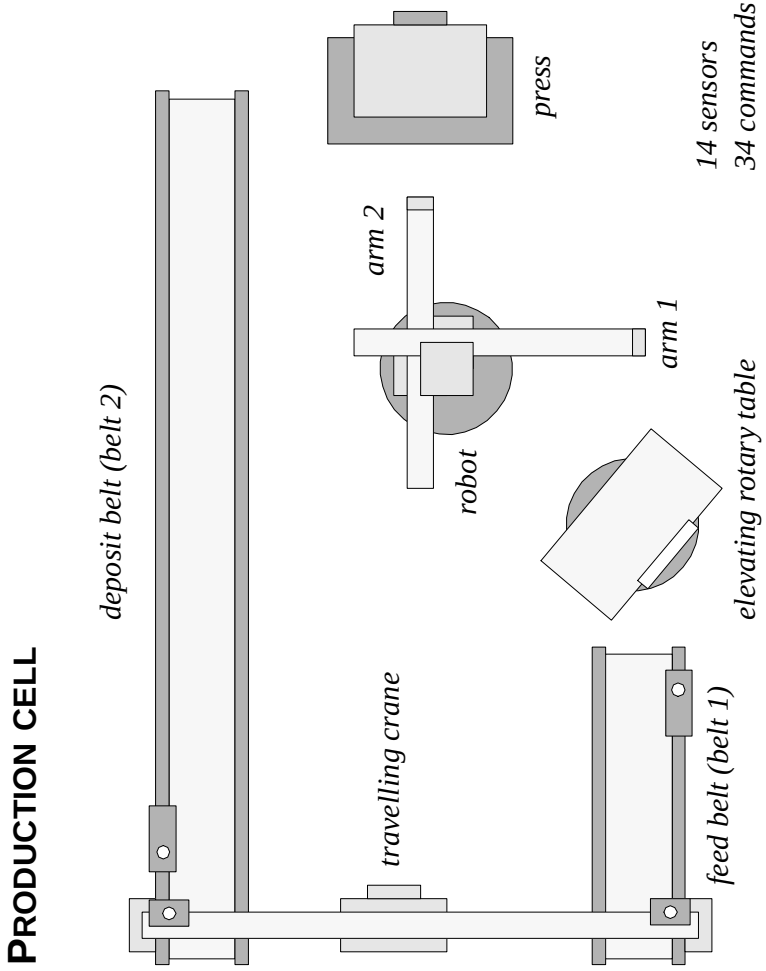
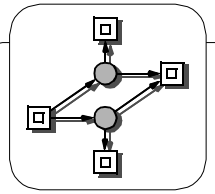
BrandenburgTechnical
University at Cottbus,
Computer Science Institute

CASE STUDY - PRODUCTION CELL

MONIKA HEINER
PETER DEUSSEN
JOCHEN SPRANGER

{mh, pd, jsp}@informatik.tu-cottbus.de
http://www.informatik.tu-cottbus.de





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.*

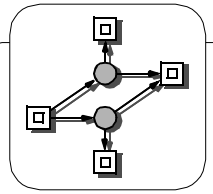
- The feed belt may only convey a blank through its light barrier, if the table is in loading position.
  -> *Blanks are not dropped outside safe areas.*

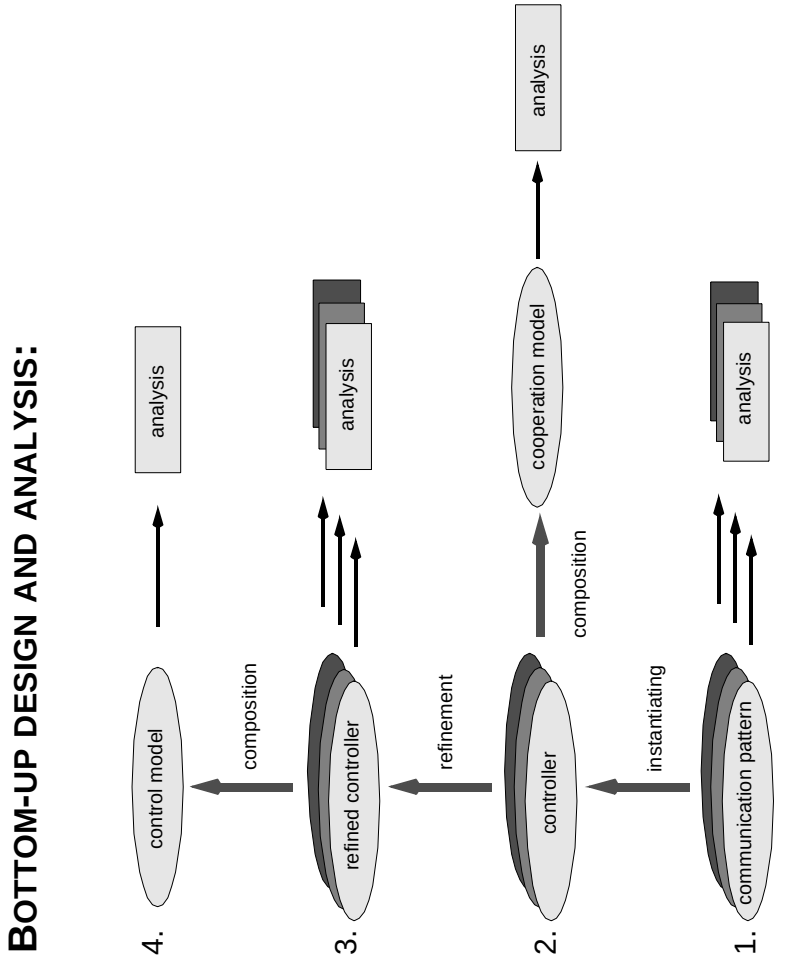
- Blanks may not be put into the press, if it is already loaded.
  -> *Insurance of a sufficient distance between consecutively processed blanks.*

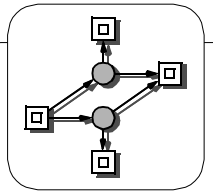
additional requirements related to design consistency:

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

- ...

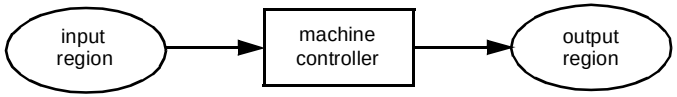




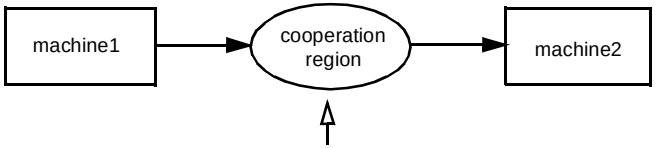


COOPERATION MODEL, BASIC DESIGN PRINCIPLES

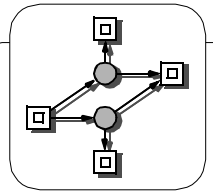
- production cell = pipeline of machines
- each machine
  - takes plates from some input places;
  - processes them;
  - puts plates on some output places;



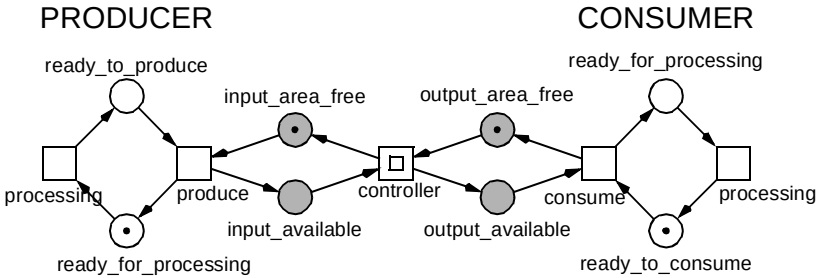
- cooperation region between two consecutive machines



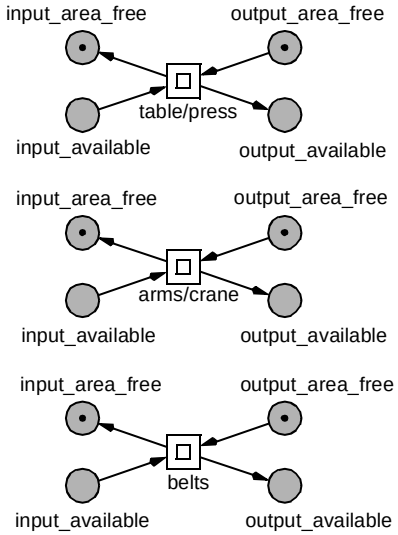
- mutual exclusive shared resources
  - robot swivel
    - (to rotate both arms)
  - physical regions
    - (intersection of trajectories of different machines)

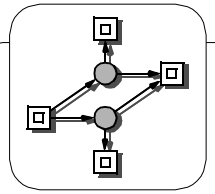


BOUNDED PROCON PATTERN



THREE TYPES OF COOPERATION PATTERN



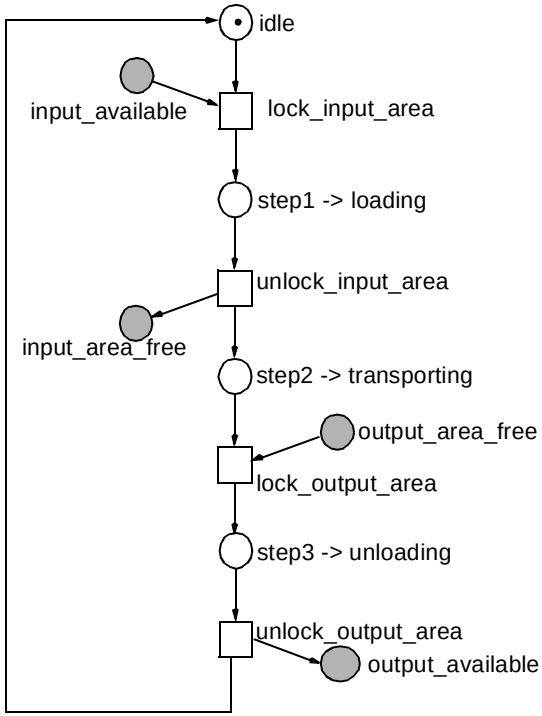


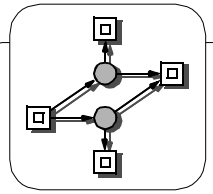
(A) INDEPENDENT INPUT/OUTPUT

- arms/crane:
  step-wise synchronization with only one of the adjacent controllers,

- pattern property, e.g.
  $\mathbf{G_A}(step1 \rightarrow \neg(input\_available \vee input\_area\_free))$

-



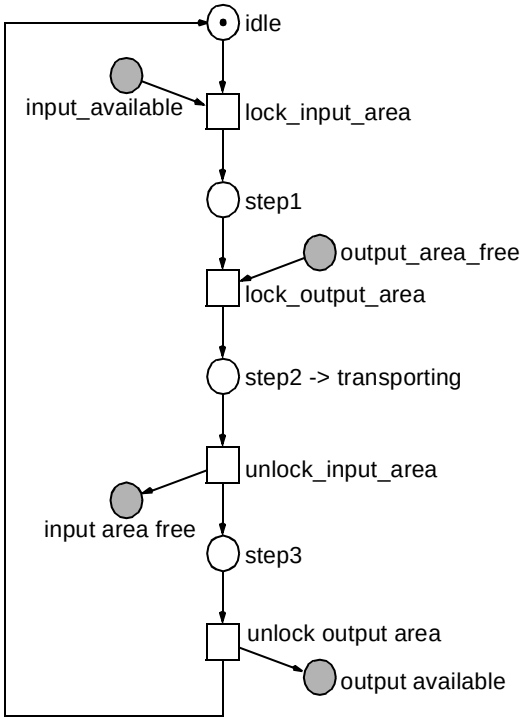


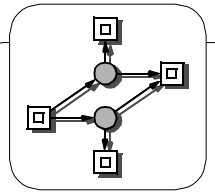
(B) DEPENDENT INPUT/OUTPUT

- belts:
  simultaneous control of input and output region

- pattern property
  $\mathbf{G_A}(step2 \rightarrow$
  $\neg(input\_available \vee input\_area\_free$
  $\vee output\_area\_free \vee output\_available))$

-





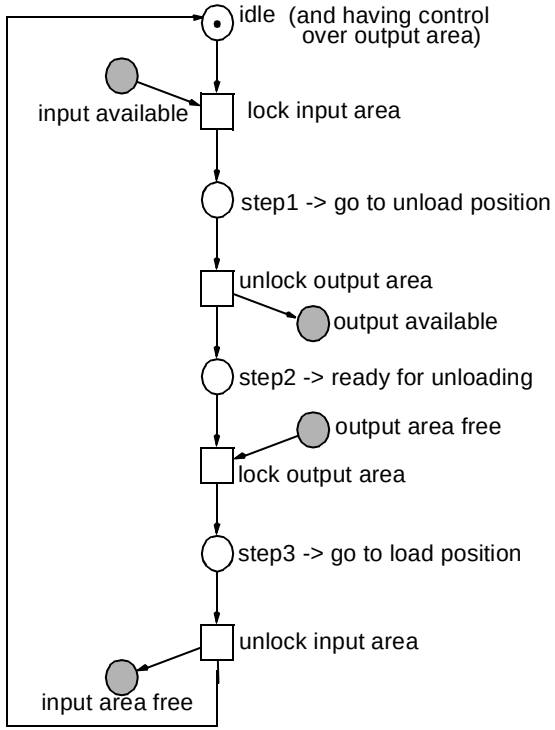
(C) Mutually Exclusive Input/Output

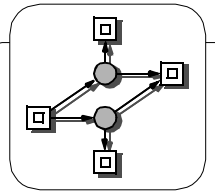
- table/press:
  the controller must always hold a lock on one of its cooperation regions;

- pattern property

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

-





Three types of cooperation pattern, Summary

(A) Independent input/output
arms/crane:

step-wise synchronization with only one of its adjacent controllers,
e.g. crane:

$\mathbf{G_A}(\neg(ch\_DC\_free \wedge ch\_DC\_full) \vee \neg(ch\_CF\_free \wedge ch\_CF\_full))$

(B) Dependent input/output
belts:

simultaneous control of input and output region
e.g. feed belt:

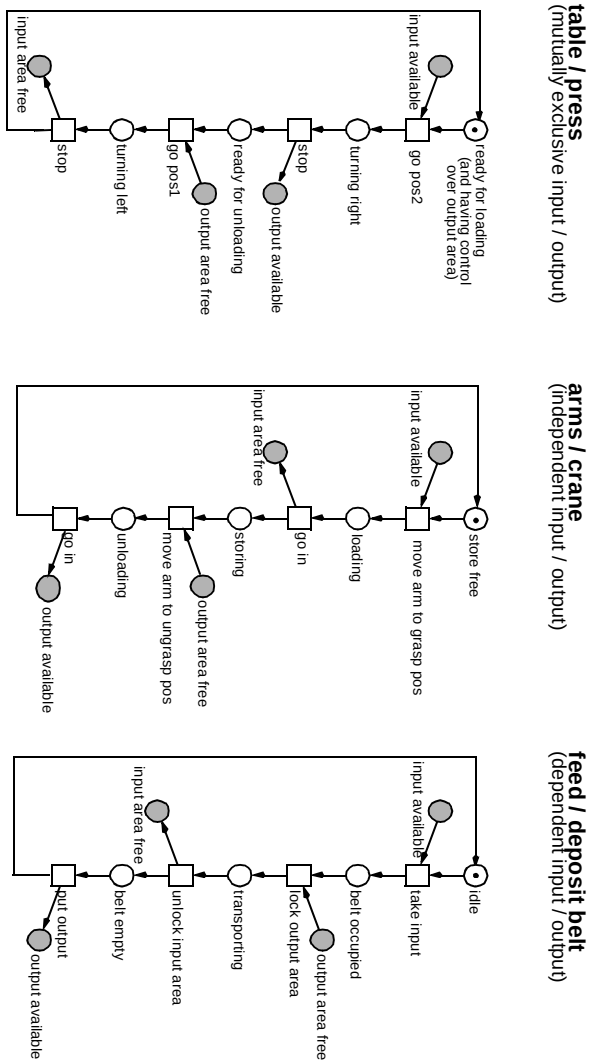
$\mathbf{G_A}(feed\_belt\_transporting \rightarrow$
$\neg(ch\_CF\_free \vee ch\_CF\_full \vee ch\_FT\_free \vee ch\_FT\_full))$

(C) Mutually exclusive input/output
table/press:

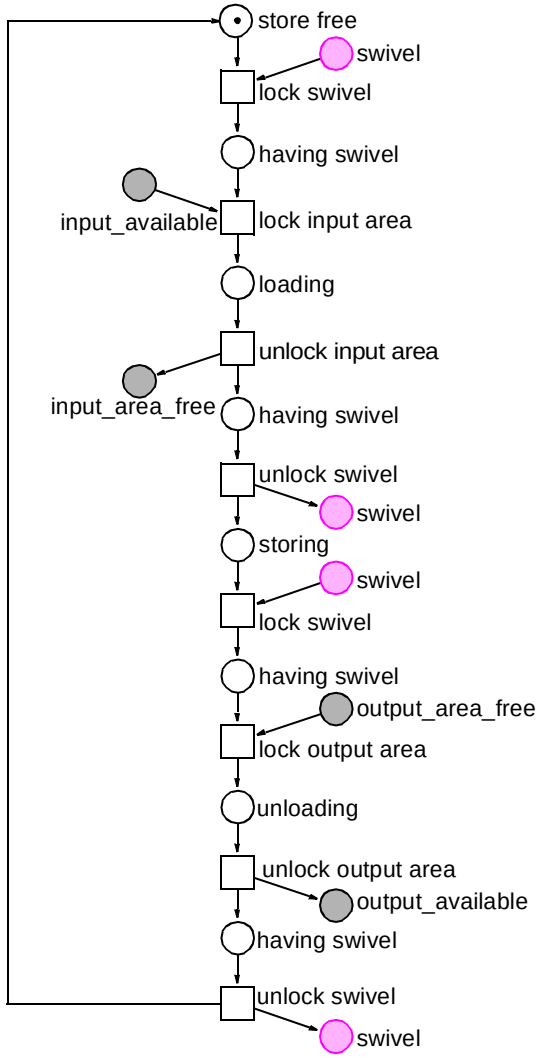
the controller must always hold a lock on one of its cooperation regions,
e.g. table:

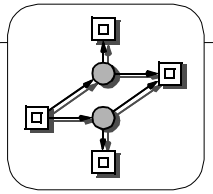
$\mathbf{G_A}(\neg(ch\_FT\_full \vee ch\_FT\_free) \vee \neg(ch\_TA1\_full \vee ch\_TA1\_free))$

THREE TYPES OF COOPERATION PATTERN, SUMMARY

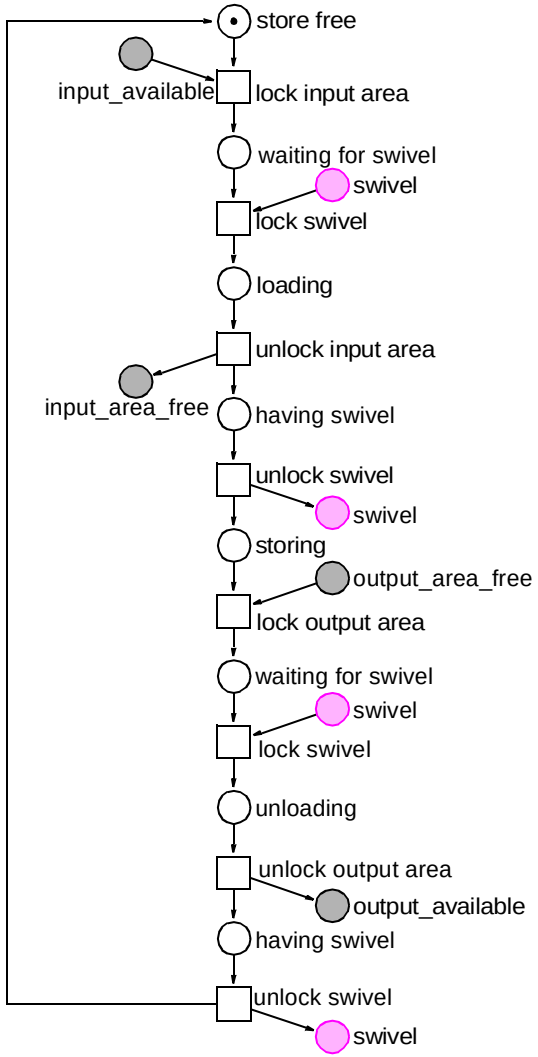


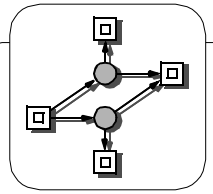
ARM VERSION2



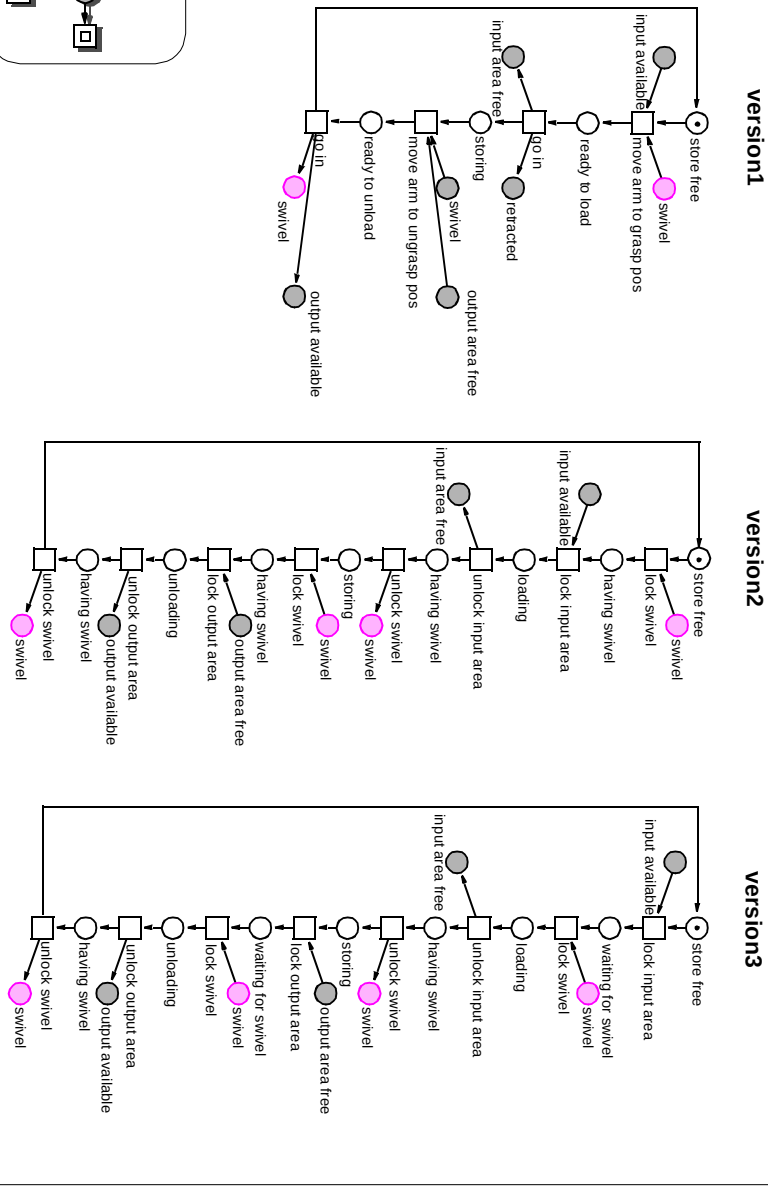


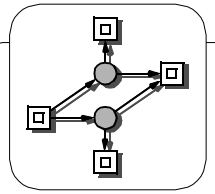
ARM VERSION3





THREE ARM VERSIONS, SUMMARY





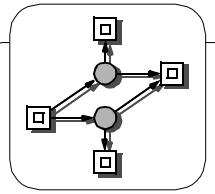
SOURCE TEXT EXAMPLES [CASAIS 94A,B]

arm: procedure to take a plate

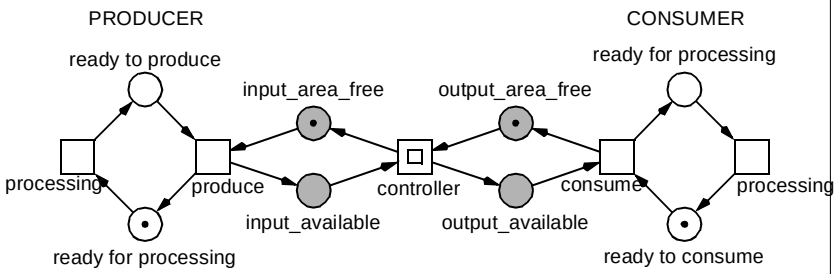
```
Take /* version2 */
    acquire locks on shared resources (swivel)
    acquire lock on input area
        move_arm_to_grasppos
        do_grasp
        go_in
    release lock on input area
    release locks on shared resources (swivel)

Take /* version3 */
    acquire lock on input area
    acquire locks on shared resources (swivel)
        move_arm_to_grasppos
        do_grasp
        go_in
    release lock on input area
    release locks on shared resources (swivel)
```

belt: procedure to transport a plate

```
Transport
    acquire lock on input area
    acquire lock on output area
        transport
    release lock on input area
    release lock on output area
```

CONTROLLER ANALYSIS

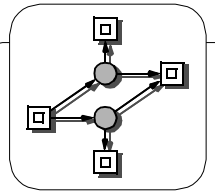


ARMS

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

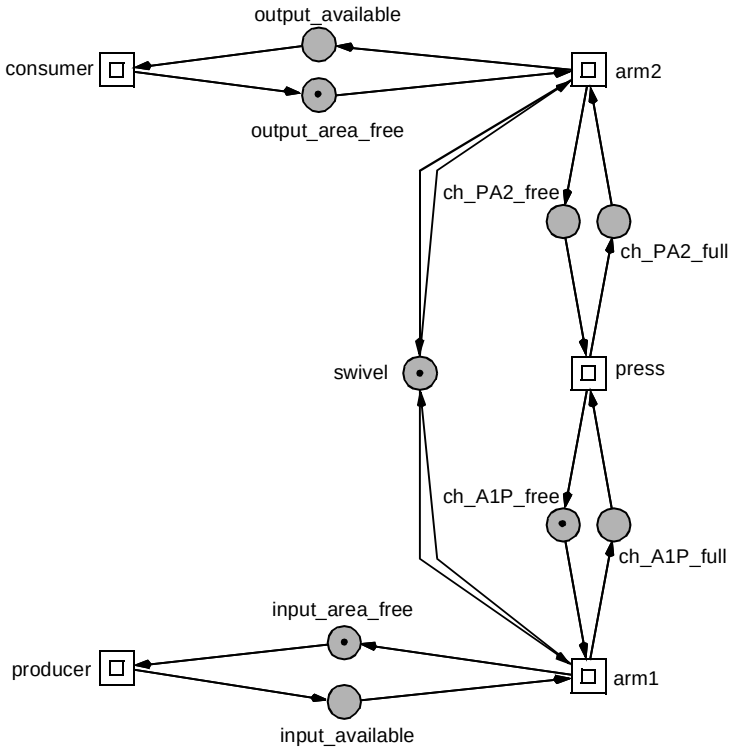
ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	Y	Y	Y	N	N	N	N	Y	N	Y	Y	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	



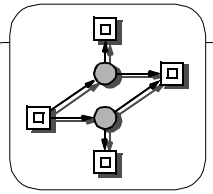
STEP-WISE COMPOSITION

E.G. SUBSYSTEM: ARM1 - PRESS - ARM2

(ARMS: VERSION2)



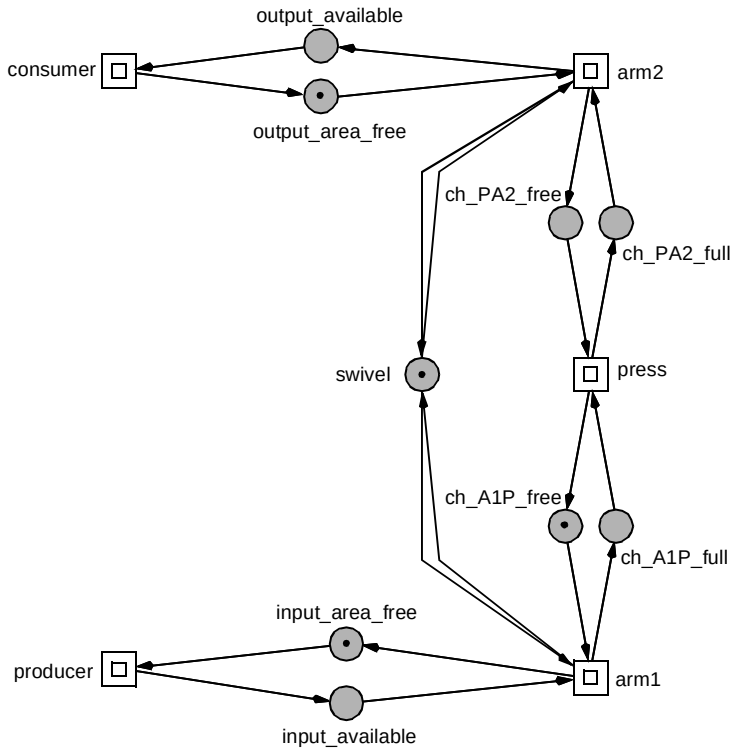
ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
N	Y	Y	N	Y	Y	Y	Y	N	Y	N	N	?	N	N	N	



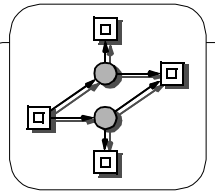
STEP-WISE COMPOSITION

E.G. SUBSYSTEM: ARM1 - PRESS - ARM2

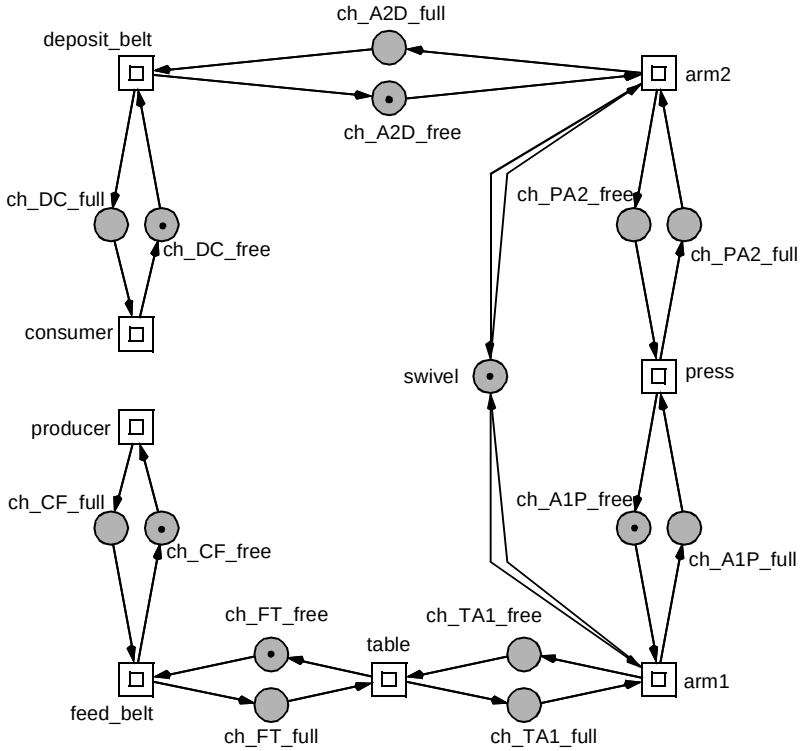
(ARMS: VERSION3):



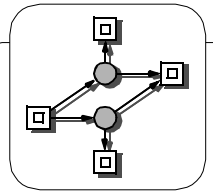
ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	



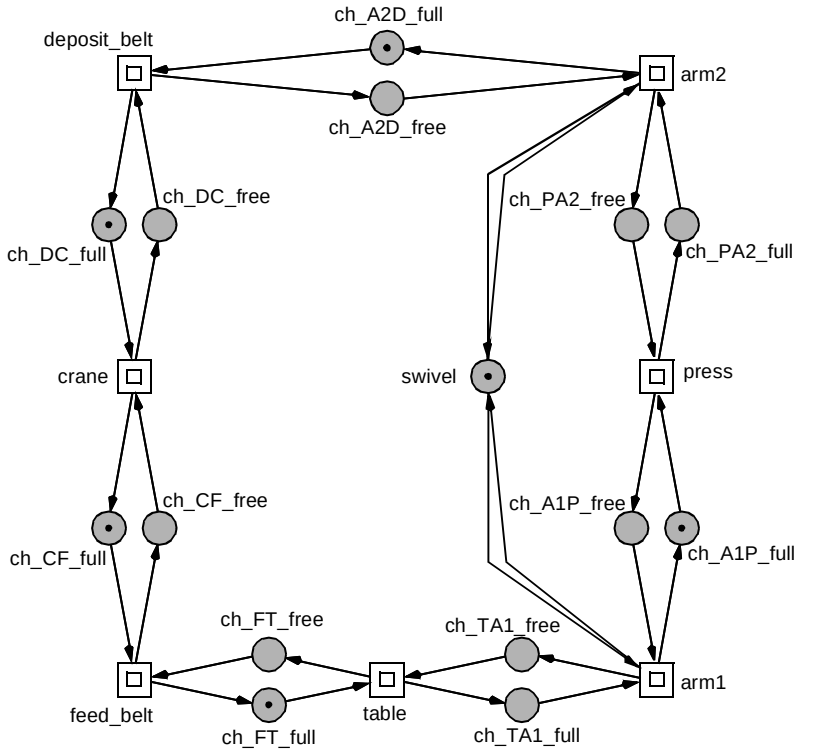
OPEN SYSTEM, COARSE STRUCTURE



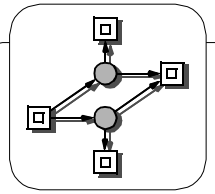
ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	



CLOSED SYSTEM, COARSE STRUCTURE

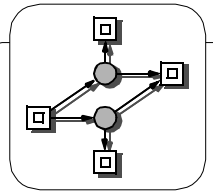


ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	



ANALYSIS EFFORTS, COOPERATION MODEL

	places/ transitions	DTP	R_{stub}	R
table / press with init part without init part	 13 / 9 12 / 8	 (N) 28	 12 8	 28 24
crane	12 / 8	31	11	48
arms version 1 version 2 version 3	 13 / 8 17 / 12 17 / 12	 38 109 88	 11 15 15	 48 112 96
belts	12 / 8	26	8	36
subsystem with arm version 1 arm version 2 arm version 3	 25 / 16 33 / 24 33 / 24	 175 3.851 (N) 725	 47 75 140	 640 1.984 1.800
open system	51 / 36	1.145	299	77.760
closed system with 1 plate with 2 plates with 3 plates with 4 plates with 5 plates	51 / 36	1.140	 36 7 2 94 98 121	 864 4.776 12.102 16.362 12.144

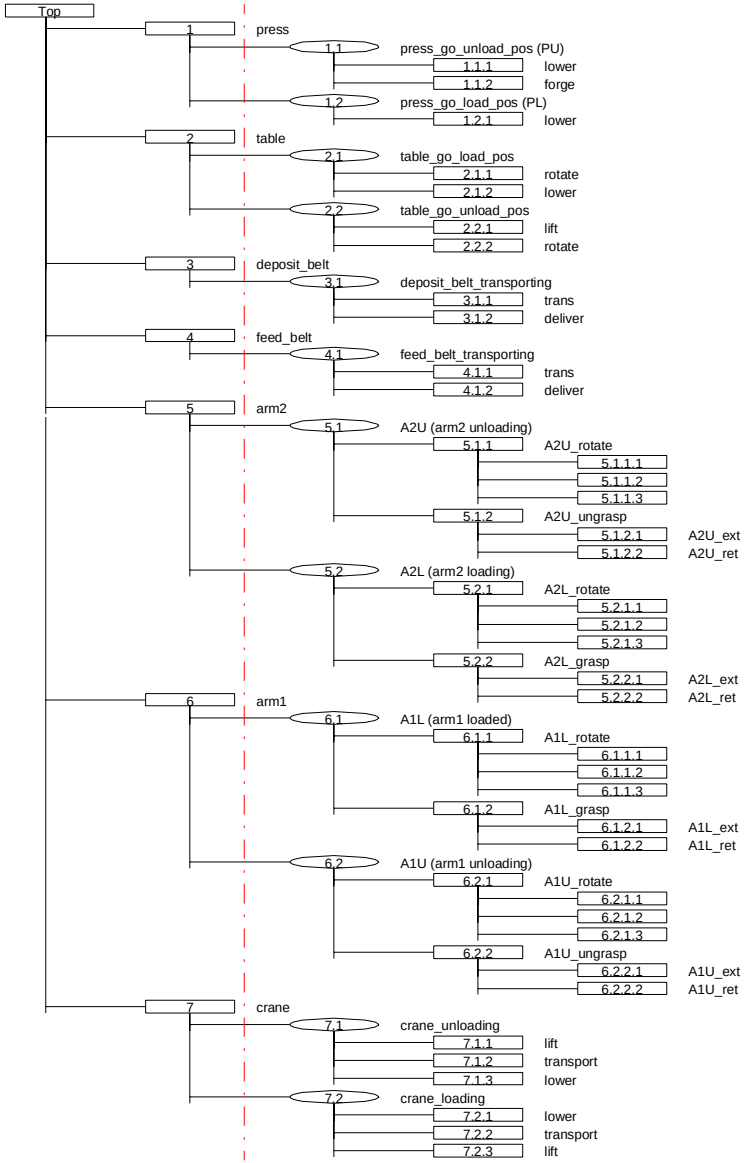


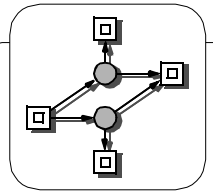
ANALYSIS EFFORTS BY SMV (BDD), COOPERATION MODEL

subsystem	without reordering		computation of reordering		with reordering	
	time	BDD nodes	time	BDD nodes	time	BDD nodes
controllers						
belt	0.10”	3962	0.12”	2762	0.04”	3723
table/press	0.09”	2902	0.14”	2149	0.12”	2656
crane	0.12”	4075	0.16”	2643	0.11”	3555
arm						
version 1 version 2 version 3	0.13” 0.22” 0.23”	4270 10017 9735	0.18” 0.29” 0.35”	2837 5073 5015	0.12” 0.18” 0.21”	3720 9806 8816
composed systems						
robot (arm version 3)	21.93”	41685	1.00”	7671	6.76”	11829
robot/press with						
arm version 1 arm version 2 arm version 3	1.88” 11.02” 13.38”	10292 11231 15618	3.14” 9.59” 10.14”	5799 6378 7012	1.08” 8.93” 6.10”	10093 10680 10365
open system	343.19”	103319	205.03”	31506	99.80”	44732
closed system						
with 1 plate with 2 plates with 3 plates with 4 plates with 5 plates	36.29” 77.14” 144.89” 182.46” 275.53”	48984 59467 94818 108414 180507	22.39” 57.40” 69.00” 75.38” 49.90”	8357 12041 16847 20292 14906	13.58” 23.48” 37.10” 54.06” 30.41”	11163 17662 27101 40188 12144

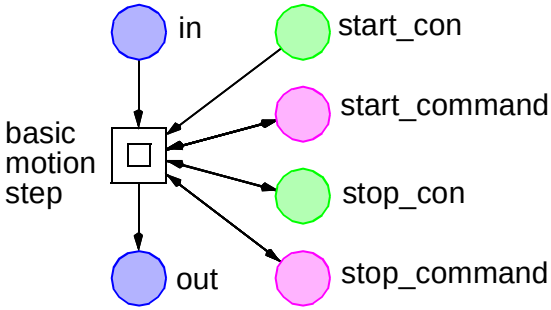
Machine: Hypersparc, 32 MB (britten)
Times: user time + system time
Model checking of the formula $\&_t$ AG EF en(t) (lifeness condition)
Comp. of variable reordering perfomed without model checking, smv options: -f -r inc

NET HIERARCHY





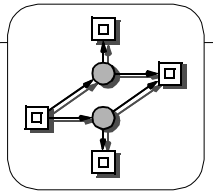
BASIC MOTION STEP, MACRO NET



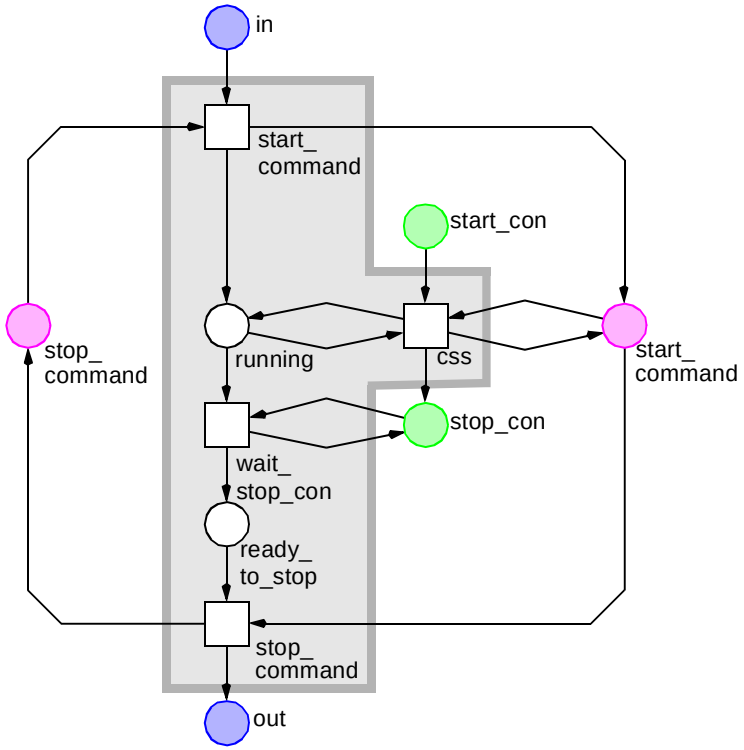
formal parameters

actual parameters, e.g.:

press_forge	press_lift
press_at_middle_pos	press_at_lower_pos
press_upward	press_up
press_at_upper_pos	press_at_middle_pos
press_stop	press_stop

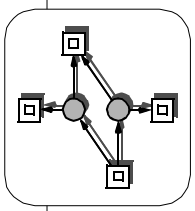


BASIC MOTION STEP + ENVIRONMENT



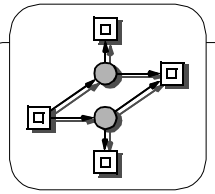
fusion nodes:
- interface
- actuator states
- sensor states

css - change sensor state



MAIN ANALYSIS RESULTS

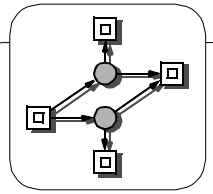
	cooperation model	control model
size # pages	51 P, 36 T 8 pages	231 P, 202 T 65 pages
general analysis	covered by P-Invariants → BOUNDED	
	DTP & ES → LIVE size $(RG_{stub})_5$: 121 → Deadlock-free	not ES size $(RG_{stub})_5$: 585 → Deadlock-free
	size $(prefix)_5$: 252 B, 159 E size $(RG)_5$: 12.144	size $(prefix)_5$: 1619 B, 768 E → LIVE size $(RG)_5$: 1.657.242
special analysis	PROD/CTL: rich, but too slowly AG ($\neg\varphi$): acceptable AG ($p \rightarrow AF\chi$): slowly	PROD/LTL PEP/CTL_0 * lack of quantification on pathes * lack of AF, AU



ANALYSIS EFFORTS (CONTROL MODEL)

subsystem	P / E	PEP	PROD					
		C / E	R	time	R_{stub} a)	time	R_{stub} b)	time
controllers								
crane	45/34	154/71	256	0.78”	51	0.16”	38	0.08”
feed belt	22/16	69/34	69	0.20”	31	0.10”	16	0.07”
table	32/24	82/37	88	0.38”	36	0.15”	24	0.09”
arm (version 3)	66/60	138/65	365”	1.19”	62	0.23”	51	0.09”
press	28/20	166/81	140	0.42”	48	0.10”	20	0.09”
deposit belt	22/16	69/34	69	0.20”	31	0.11”	16	0.07”
composed systems								
robot	124/120	3514/1752	63,232	11.26’	992	5.99”	205	0.21”
robot/press	140/132	1280/624	18,344	3.10”	557	3.46”	305	0.35”
open system	198/176	2773/1348	2,776,936	?	798	5.90”	507	0.62”
closed system	231/202							
with 1 plate		690/316	30,952	7.54’	162	0.68”	163	0.32”
with 2 plates		1670/792	543,480	ca. 3.3 h	406	2.53”	456	0.72”
with 3 plates		2009/960	> 1,7 Mio	> 20 h	523	4.51”	635	0.95”
with 4 plates		2164/1035	> 3.1 Mio	> 42 h	471	4.02”	678	1.06”
with 5 plates		1619/768	1,657,242	ca. 14 h	585	5.05”	608	0.98”

a) deletion algorithm
b) incremental algorithm



TEMPORAL LOGICS, EXAMPLES OF ANALYZED PROPERTIES

General analysis

- liveness

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

Special analysis

- design demanded properties, e.g. (PROD/LTL)

  $\mathbf{G}\left(\text{robot\_stop} \dot{\vee} \text{robot\_left} \dot{\vee} \text{robot\_right}\right)$

- functional properties, e.g.

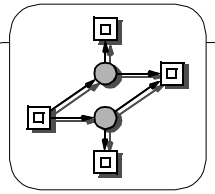
  $\mathbf{EF}(\text{arm1\_mag\_on} \wedge \text{arm2\_mag\_on})$ (PEP)

- safety properties, e.g.

  If a robot arm is loaded, its magnet is not deactivated until the robot is in its unloading position (PROD/LTL)

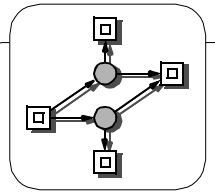
  $\mathbf{G}(\varphi \rightarrow \neg\chi\mathbf{U}\psi)$, where

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



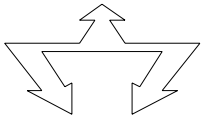
MAIN LESSONS LEARNT

- 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 procon 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&mutex synchronisation
  -> *lock a mutex resource always as late as possible*
- pattern properties
  -> *model consistency criteria*
  -> *to be generated for each instance*



CONCLUSIONS

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



required -> granted

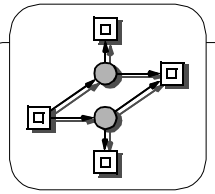
- properties taxonomy

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**

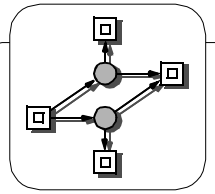
  e.g. liveness

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

- **INSIGHTS**

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

  $\mathbf{EF}$(arm1_mag_on $\wedge$ arm2_mag_on)



REFERENCES

[Casais 94a]
Casais, E.:
Eiffel; A Reusable Framework for Production Cells Developed with an Object-oriented Programming Language; in: Lewerentz, C.; Lindner, T. (eds.): Case Study "Production Cell" A Comparative Study in Formal Software Development, FZI-Publication 1/94, Forschungszentrum Informatik, Karlsruhe 1994, pp. 241-256.

[Casais 94b]
Casais, E.:
An Experiment in Framework Development; in: Lewerentz, C.; Lindner, T. (eds.): Case Study "Production Cell" A Comparative Study in Formal Software Development, FZI-Publication 1/94, Forschungszentrum Informatik, Karlsruhe 1994, pp. 95-124.

[Heiner 95]
Heiner, M.; Deussen, P.:
Petri Net Based Qualitative Analysis - A Case Study;
Techn. Report BTU Cottbus, I-08/1995, Dec. 1995.

[Heiner 98]
Heiner, M.; Deussen, P.; Spranger, J.:
A Case Study in Developing Control Software of Manufacturing Systems with Hierarchical Petri Nets; Journal paper

[Lewerentz 95]
Lewerentz, C.; Lindner, T.:
Formal Development of Reactive Systems - Case Study Production Cell;
LNCS 891, 1995.

[Michaelis 93]
Michaelis, M.:
Objektorientierte Modellierung einer Fertigungszelle mit Eiffel (in German);
Diplomarbeit Univ. Karlsruhe, Fakultät für Informatik, June 1993.