Introduction to Process Migration of Business Processes

Rushikesh K. Joshi
Computer Science & Engineering
IIT Bombay
rkj@cse.iitb.ac.in

Long Processes

Online businesses have long process, which run over days

For example, order-delivery chain can take a few days from initiation till completion

Examples: application-approval systems, scientific funding, e-governance processes

Process Evolution

Service provider is running a process

The service provider makes a change to the process and updates it for new features, improvements, removal of features, optimizations

This means a process evolves over time

... and we keep in mind that we have long processes

The problem of process migration

What happens when a process evolves, but there are instances still running the old process model?

Should they restart?

Or is there a “Marking” in the new process to which they can migrate?

The Prediction .... What do you think will happen?

Possibilities -- address difficulty in each

1 Let old instances keep running on old process

2 Wait for all to complete and only

3 Backward compatibility

4 Cancel current instances and ask them to restart

Dynamic Process Migration

We need strategies to let old instances continue into the new process from some ‘sensible’ point

Which means we need to find an “equivalent” marking in the new process

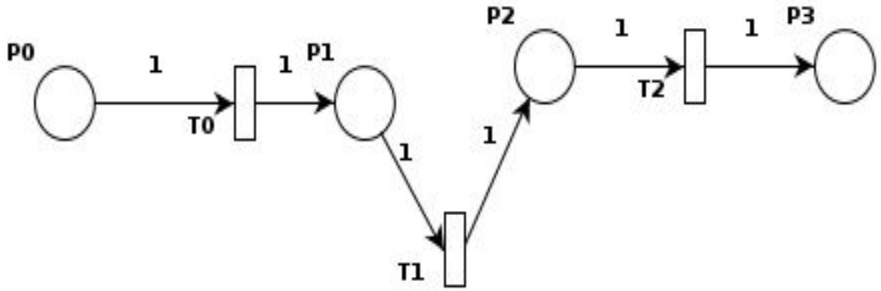
Equivalence Criteria

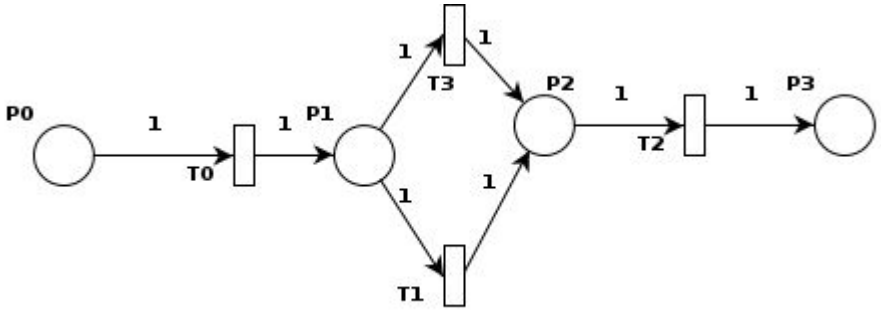
Each process application may use different methods of equivalence

- Just migrate the current marking as-it-is
- Look into transition history of the past
- Look into possible transitions in future

Migrate the Marking as-it-is

Consider the given old and new processes and make a list of old and new

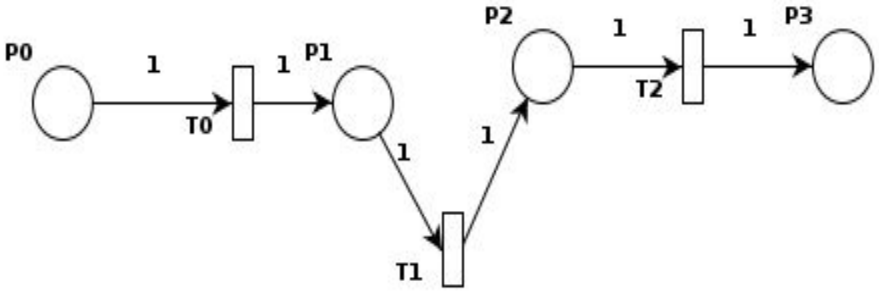


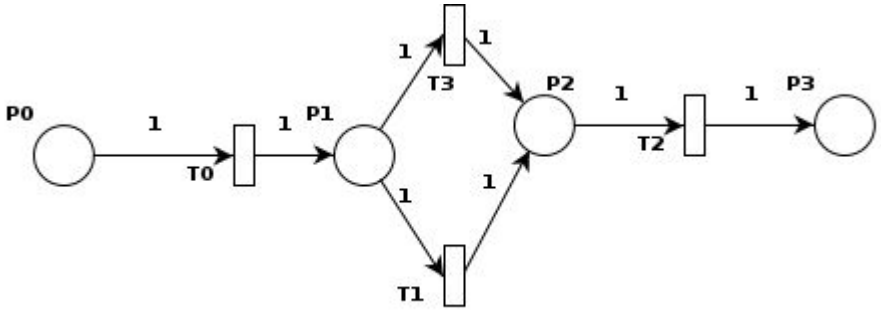


Tasks
T0: Select Goods
T1: Pay by Cash
T2: Take delivery
T3: Pay by Credit-card

Migrate the Marking as-it-is

Consider the given old and new processes and make a list of old and new





Tasks
T0: Select Goods
T1: Pay by Cash
T2: Take delivery
T3: Pay by Credit-card

Traces in old design:
T0-T1-T2

Traces in new design:
T0-T1-T2
Or
T0-T3-T2

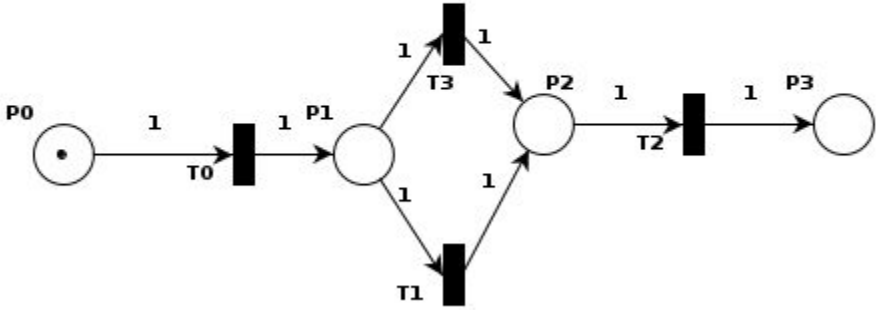
The Migration Table

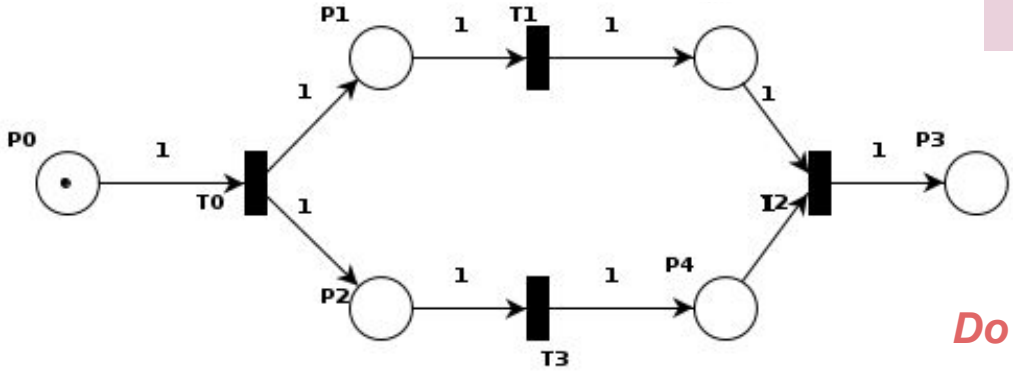
Old Marking	New Marking
{P0}	{P0}
{P1}	{P1}
{P2}	{P2}
{P3}	{P3}

All markings are migratable
Which means the marking continues as it is for all old markings, and there is no issue

Migrate the Marking as-it-is-- Another Example

Consider the given old and new processes and make a list of old and new mappings



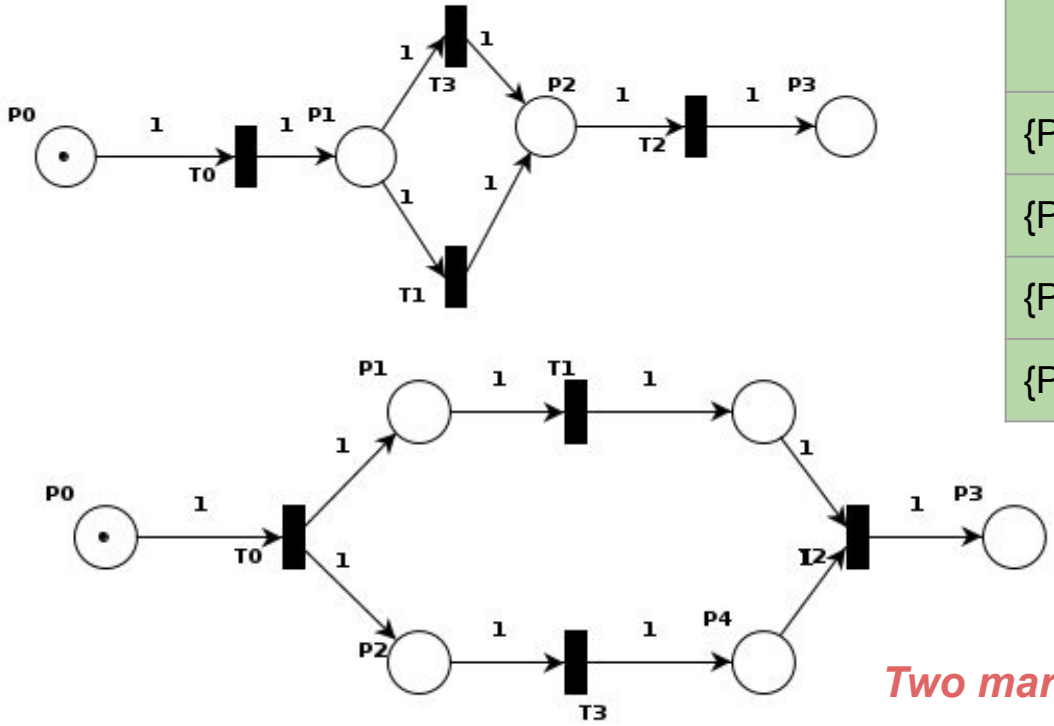


Tasks
T0: Make leave application
T1: approval from TA instructor
T2: approval from facad
T3: approval from guide

Do you see any problem?

Migrate the Marking as-it-is -- the difficulty

Consider the given old and new processes and make a list of old and new mappings

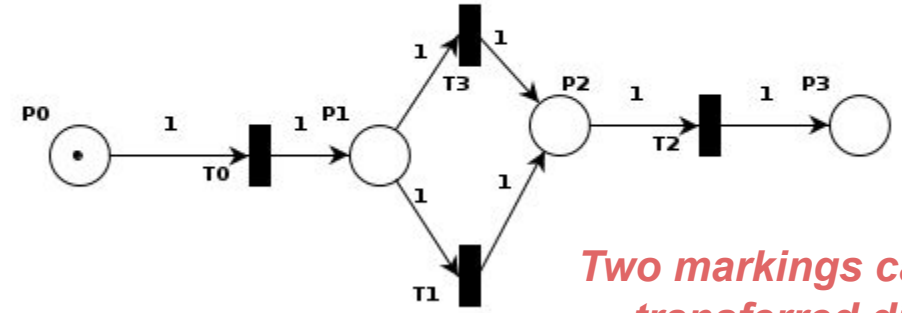


Old Marking	Is the marking valid in new process
{P0}	yes
{P1}	no
{P2}	no
{P3}	yes

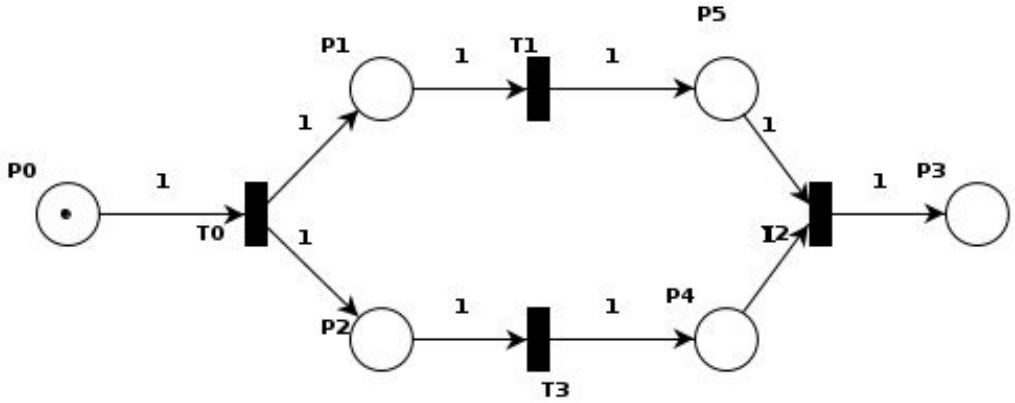
Two markings cannot be transferred directly

Migrate the Marking as-it-is -- the difficulty

Consider the given old and new processes and make a list of old and new mappings



Two markings cannot be transferred directly



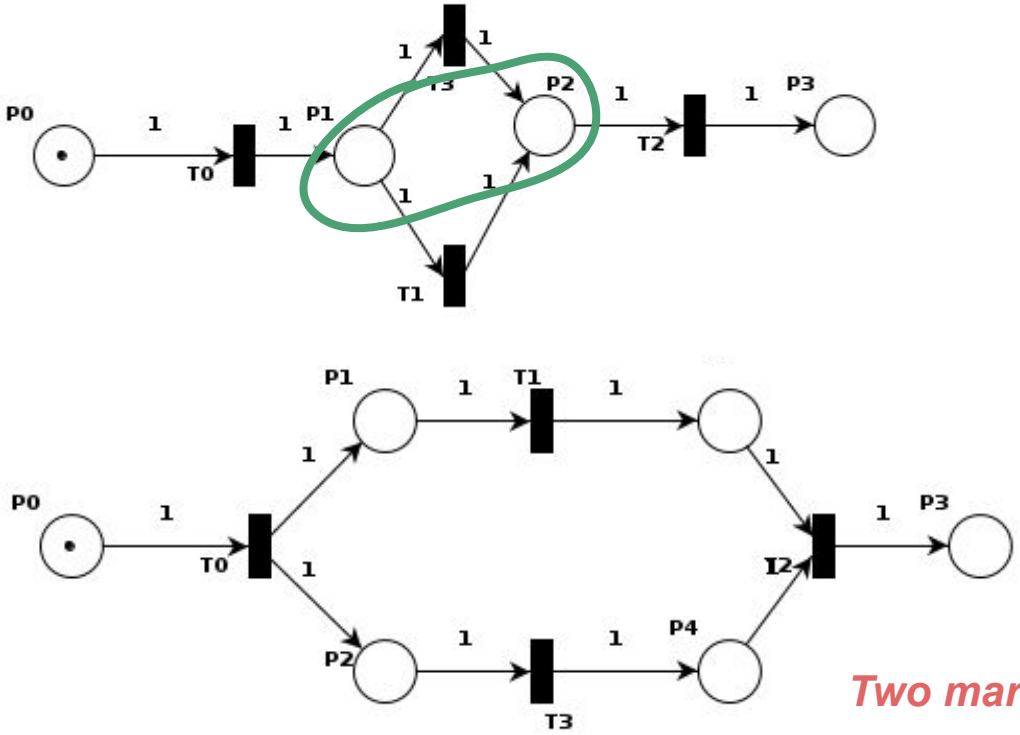
Old Marking	Is the marking valid in new process
{P0}	yes
{P1}	**no**
{P2}	**no**
{P3}	yes

Traces in old design:
T0-T1-T2
T0-T3-T2

Traces in new design:
T0-T1-T2
Or
T0-T2-T1

Change Region

It is the region in the old process, which is not migratable due to absence of consistent migration markings in new process



Old Marking	Is the marking valid in new process
{P0}	yes
{P1}	no
{P2}	no
{P3}	yes

Two markings cannot be transferred directly

This is the most important takeaway that everyone has to remember:

Find a consistent mapping into the new process for every running instance. The mapping depends on consistency model used.

Not every marking may be migratable!

Summary

Processes evolve

Instances can be migrated provided that

- There is a consistency criteria which can find a new marking

Change region: set of markings in the old processes which are non-migratable

Marking based, past based, future based criteria

Or do any one of the following (1 and 2 individual, unless the problem is sizable)

Possibilities of research/practice experience

1) Existing paper: read and present
2) New reExisting paper: read and presentsearch/possibility of paper communication (discuss with me)
3) Existing sizable process:Understand and implement in cms or plain php/html
4) New process: dessign, draw in tool and analyze by converting to petri nets--implementation a in 3 is not needed
5) Whole batch one problem as team work: for example, bpmn editor and petri net translator (needs more time and coordination)

Next Class:

Consistency criteria for migration
- History based
- Future based
- Trace based

Now to richer models of consistency

Use of Time

Past, Present, future

Comparison Operator

Set equality, subset, superset

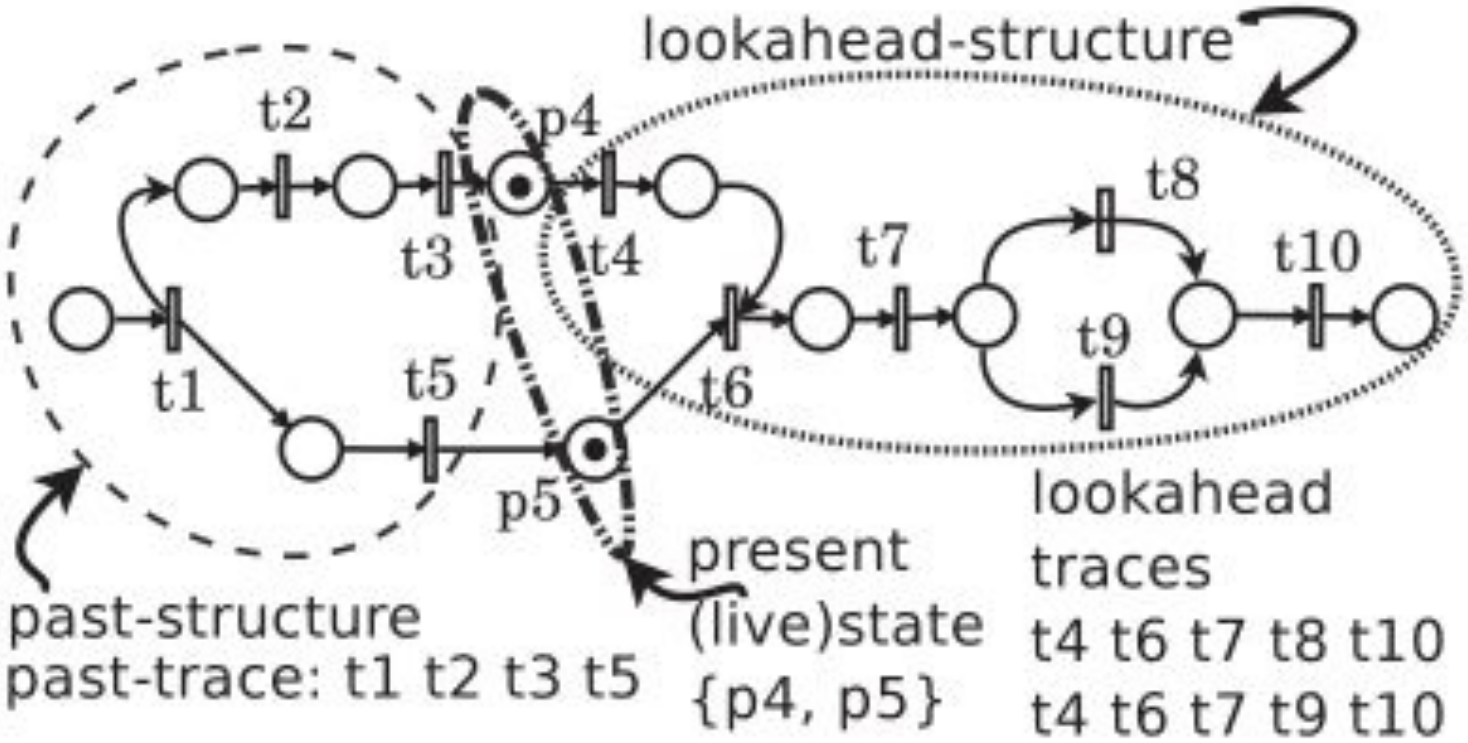
Trace Representations

Sequences, Sets, Purged Sequences, Purged Sets

Role of Net

Net (options) that was available, Net that will be available

Three regions to consider..



Trace Based Consistency Models

Compare traces based on past or future

Trail based consistency: uses past traces

Lookahead based consistency: uses future traces

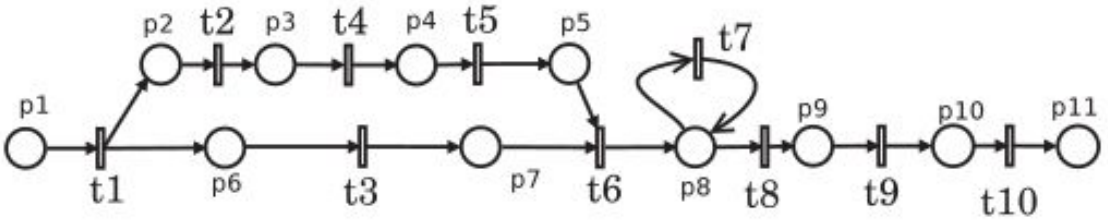
If removal of tasks (deleted tasks) is permitted, we have purged models

Use either sets or sequences to compare

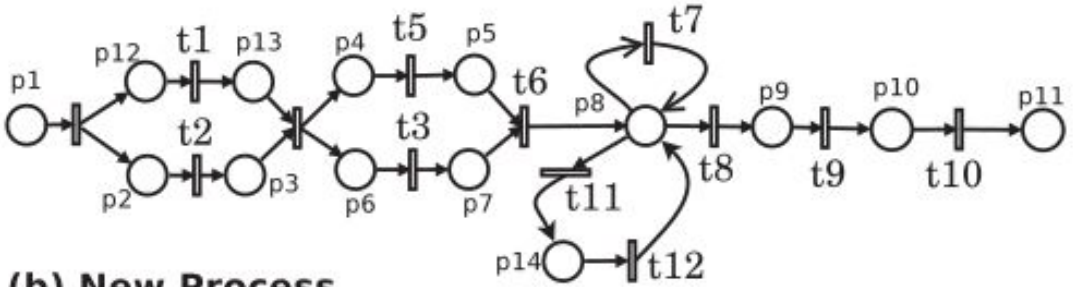
Trail based models

- Trace equivalence: The trace through which old marking was arrived is available in new process, from initial to some marking in new process
- History equivalence: The set of transitions through which the old marking was arrived is obtainable in new one through some trace, from initial to some marking in new process
- Purged trace equivalence: keep only the transitions that are common to both nets (purge the rest) and apply trace equivalence
- Purged history equivalence: keep only the transitions that are common to both nets (purge the rest) and apply history equivalence

Trail based models example



(a) Old Process

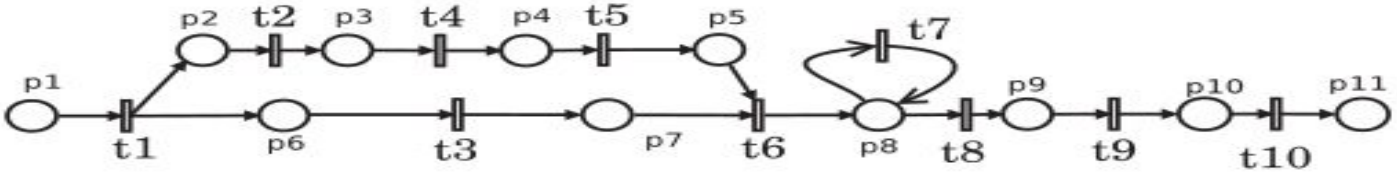


(b) New Process

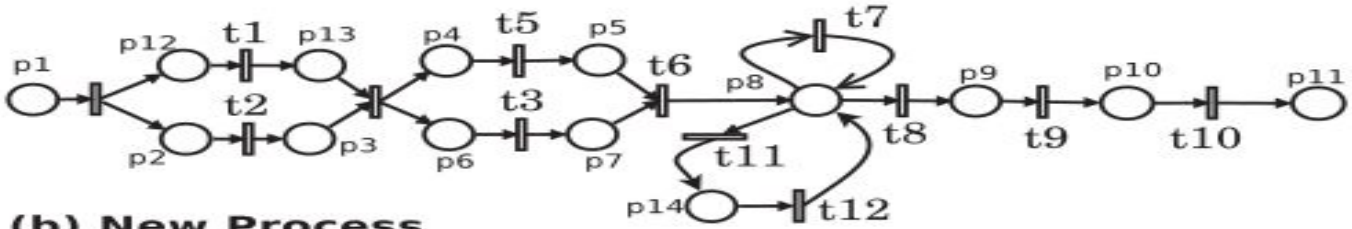
t1:select room t2: check-in register t3:prepare room
t4: advance payment t5: generate key-card
t6: furnish room t7: room service t8: inspect room
t9: generate invoice t10:check-out payment
t11: purchase t12: add to bill

Fig. 13. Trail-based models: Example of an accommodation process.

Trace Equiv.



(a) Old Process

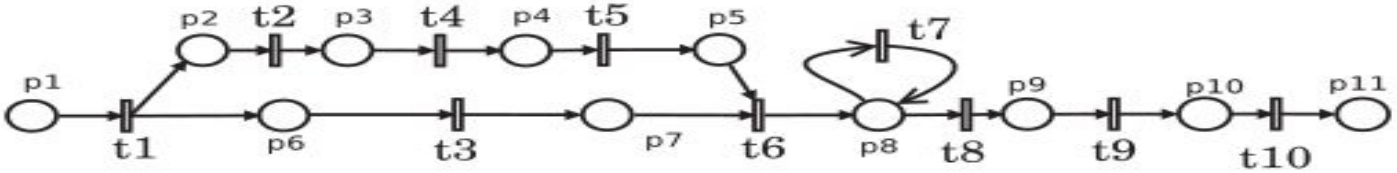


(b) New Process

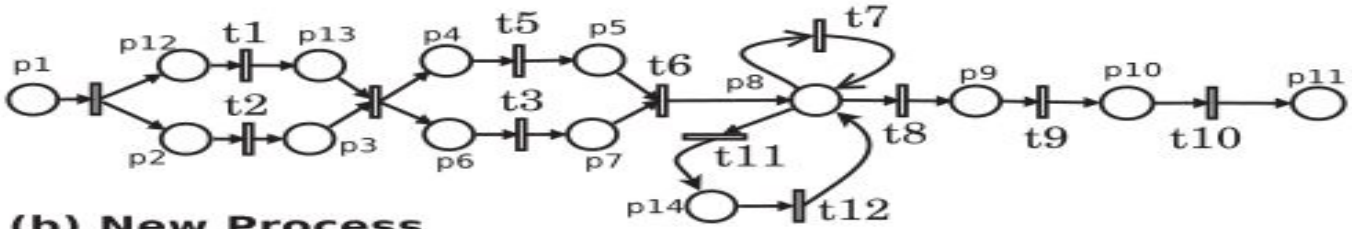
t1:select room t2: check-in register t3:prepare room
t4: advance payment t5: generate key-card
t6: furnish room t7: room service t8: inspect room
t9: generate invoice t10:check-out payment
t11: purchase t12: add to bill

Model	Old Trace (σ)	Old Marking (M)	New Consistent Marking (M')
Trace	ϵ	$\{p_1\}$	$\{p_1\}, \{p_2, p_{12}\}$
	t_1	$\{p_2, p_6\}$	$\{p_2, p_{13}\}$
	$t_1 t_2$	$\{p_3, p_6\}$	$\{p_3, p_{13}\}, \{p_4, p_6\}$
	$t_1 t_2 t_3$	$\{p_3, p_7\}$	$\{p_4, p_7\}$

History Equiv.



(a) Old Process



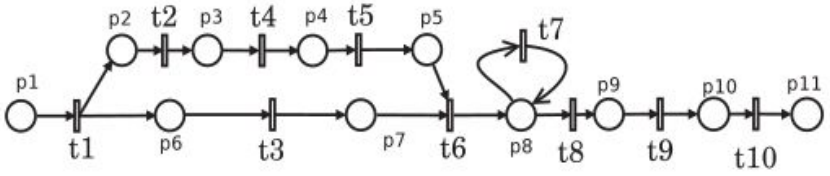
(b) New Process

t1:select room t2: check-in register t3:prepare room
t4: advance payment t5: generate key-card
t6: furnish room t7: room service t8: inspect room
t9: generate invoice t10:check-out payment
t11: purchase t12: add to bill

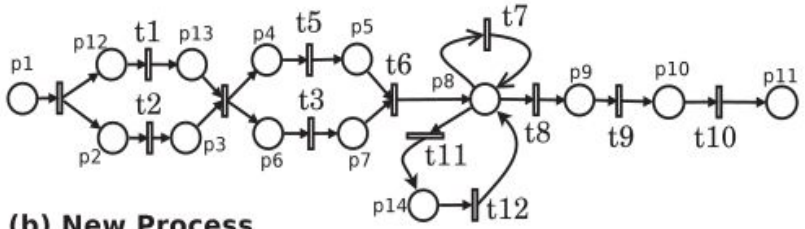
History			
	ϵ	$\{p_1\}$	$\{p_1\}, \{p_2, p_{12}\}$
	t_1	$\{p_2, p_6\}$	$\{p_2, p_{13}\}$
	$t_1 t_2$	$\{p_3, p_6\}$	$\{p_3, p_{13}\}, \{p_4, p_6\}$
	$t_1 t_2 t_3, t_1 t_3 t_2$	$\{p_3, p_7\}$	$\{p_4, p_7\}$

Purged Trace

Purged-trace			
	ϵ	$\{p_1\}$	$\{p_1\}, \{p_2, p_{12}\}$
	t_1	$\{p_2, p_6\}$	$\{p_2, p_{13}\}$
	t_1t_2	$\{p_3, p_6\}$	$\{p_3, p_{13}\}, \{p_4, p_6\}$
	$t_1t_2t_3$	$\{p_3, p_7\}$	$\{p_4, p_7\}$
	$t_1t_2t_3t_4$	$\{p_4, p_7\}$	$\{p_4, p_7\}$
	$t_1t_2t_3t_4t_5$	$\{p_5, p_7\}$	$\{p_5, p_7\}$
	$t_1t_2t_3t_4t_5t_6$,	$\{p_8\}$	$\{p_8\}, \{p_{14}\}$
	$t_1t_2t_3t_4t_5t_6t_7$,		
	$t_1t_2t_3t_4t_5t_6t_7t_7$,		
	... [expression:		
	$t_1t_2t_3t_4t_5t_6(t_7)^*$]		
	$t_1t_2t_3t_4t_5t_6(t_7)^*t_8$	$\{p_9\}$	$\{p_9\}$
	$t_1t_2t_3t_4t_5t_6(t_7)^*t_8t_9$	$\{p_{10}\}$	$\{p_{10}\}$
	$t_1t_2t_3t_4t_5t_6(t_7)^*t_8t_9t_{10}$	$\{p_{11}\}$	$\{p_{11}\}$



(a) Old Process

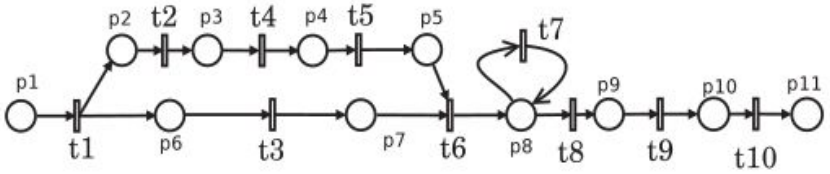


(b) New Process

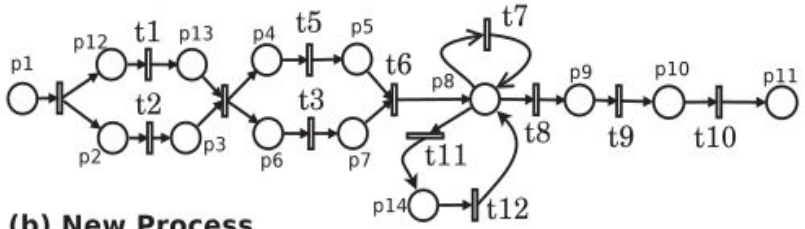
t1:select room t2: check-in register t3:prepare room
t4: advance payment t5: generate key-card
t6: furnish room t7: room service t8: inspect room
t9: generate invoice t10:check-out payment
t11: purchase t12: add to bill

Purged History

Purged-history			
	ϵ	$\{p_1\}$	$\{p_1\}, \{p_2, p_{12}\}$
	t_1	$\{p_2, p_6\}$	$\{p_2, p_{13}\}$
	t_1t_2	$\{p_3, p_6\}$	$\{p_3, p_{13}\}, \{p_4, p_6\}$
	$t_1t_2t_3, t_1t_3t_2$	$\{p_3, p_7\}$	$\{p_4, p_7\}$
	$t_1t_2t_3t_4, t_1t_2t_4t_3,$ $t_1t_3t_2t_4$ [expression: $t_1(t_2t_4 \| t_3)$]	$\{p_4, p_7\}$	$\{p_4, p_7\}$
	$t_1(t_2t_4t_5 \| t_3)$	$\{p_5, p_7\}$	$\{p_5, p_7\}$
	$t_1(t_2t_4t_5 \| t_3)(t_7)^*$	$\{p_8\}$	$\{p_8\}, \{p_{14}\}$
	$t_1(t_2t_4t_5 \| t_3)(t_7)^*t_8$	$\{p_9\}$	$\{p_9\}$
	$t_1(t_2t_4t_5 \| t_3)(t_7)^*t_8t_9$	$\{p_{10}\}$	$\{p_{10}\}$
	$t_1(t_2t_4t_5 \| t_3)(t_7)^*t_8t_9t_{10}$	$\{p_{11}\}$	$\{p_{11}\}$



(a) Old Process



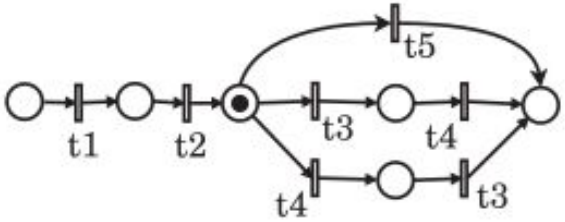
(b) New Process

t1:select room t2: check-in register t3:prepare room
t4: advance payment t5: generate key-card
t6: furnish room t7: room service t8: inspect room
t9: generate invoice t10:check-out payment
t11: purchase t12: add to bill

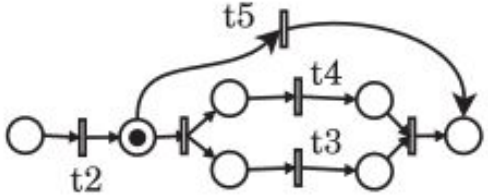
Lookahead based models

- Stongly consistent look ahead consistency
  - Set of all firing sequences in futures are equal
- Accomodative lookahed consistency
  - The new set of future firing sequences is a superset of the old set
- Weakly consistent lookahead
  - The new set is a subset of the old set, and is non-empty
- Accomodative purged lookahead
  - Superset, but in purged form- a few sections can be omitted in the lookahed traces
- Weakly consistent lookahead
  - Subset, purged form is acceptable as above

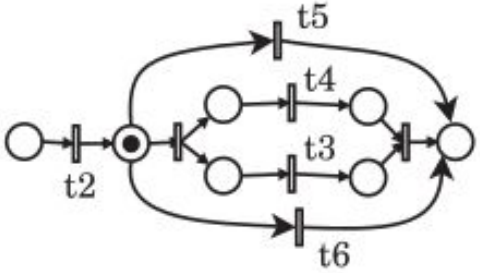
Examples



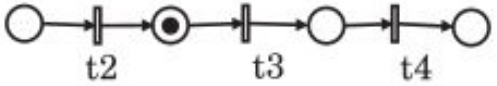
(a) *Old Workflow Marking*



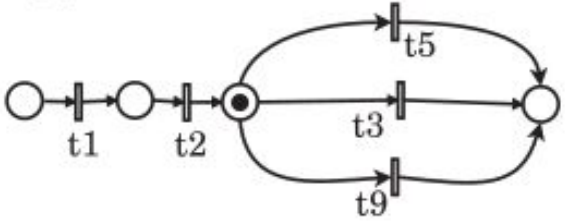
(b) *Strong Lookahead*



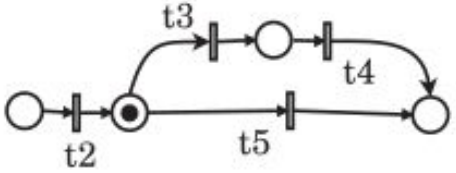
(c) *Accommodative Lookahead*



(d) *Weak Lookahead*



(e) *Accommodative-Purged Lookahead*



(f) *Strong-Set Lookahead Weak Lookahead*

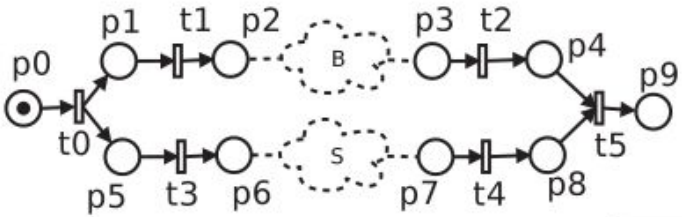
Weak lookahead example

Marking {p0}-->Marking {q0}. Set of new remaining traces is a subset of old

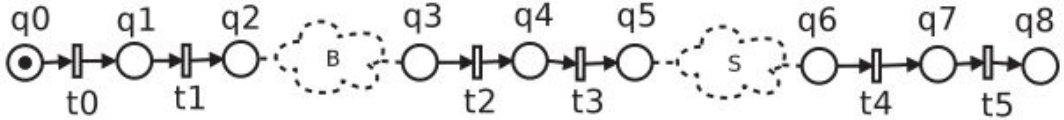
old traces: {t0-t1-t2-t3-t4-t5, t0t1-t3-t4-t2-t5,... etc}

New traces{t0-t1-t2-t3-t4-t5}

Find non-migratable markings by this criterion?



Old Configuration of Billing and Shipping Workflow



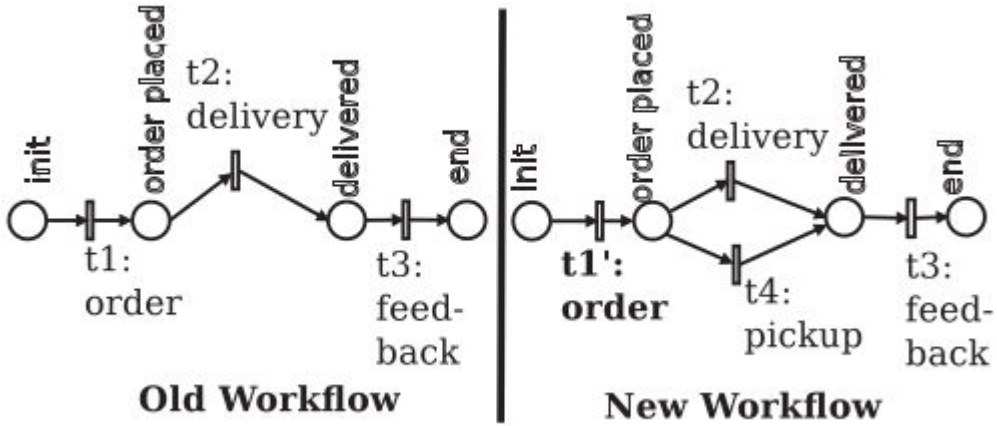
Billing and Shipping Workflow after Reconfiguration

t0: recv. order
t1:call bill
t2: recv bill details
t3: call ship
t4: recv ship details

Activities

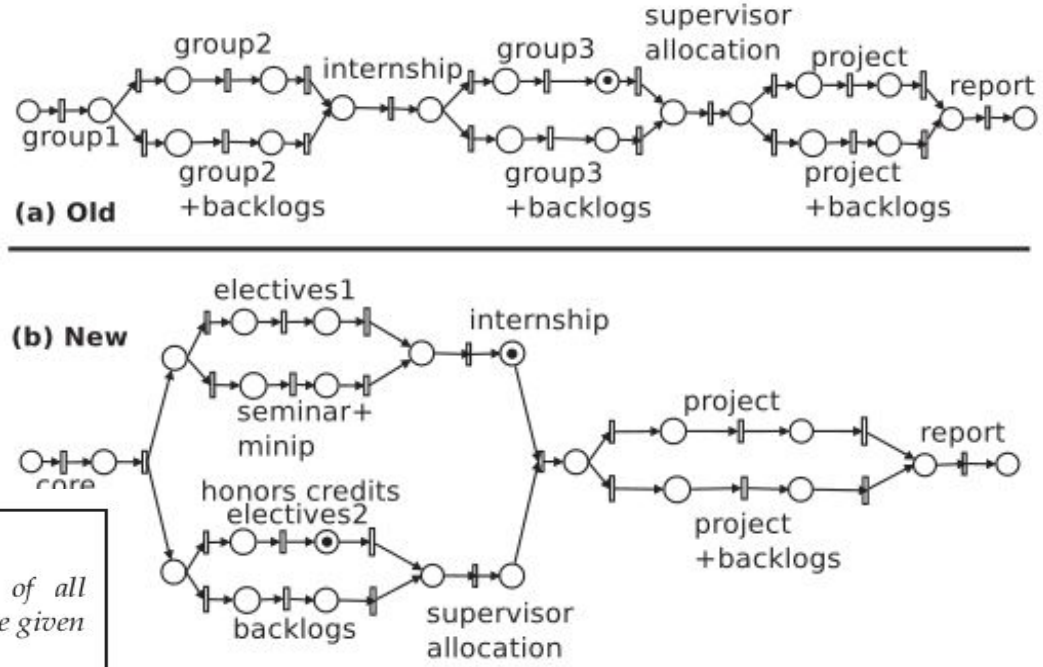
Live consistency

- “Present” based model
  - Current marking is used
  - The marking has to be migratable as-it-is
  - Change region-- set of all non-migratable markings (concept applicable to all models of consistency)



Business Consistency

- As per business rules
- Make a table of mappings of markings from old to new



"Migration Circular"

Credit substitution rules for immediate migration of all Master's students into the new academic curriculum are given as follows:

1) *Group 1 credits are treated as Core credits.*
2) *Group 2 credits are treated as Elective-1 credits.*
3) *Backlog students registered for Group 2 are inducted into Seminar+Mini-project path.*
4) *Group 3 courses are made optional and converted into honors credits.*
5) *Backlog students who do not complete Group 3 credits can not register for the honors credits, instead they will need to complete the backlog credits.*

Reference

A Taxonomy of Consistency Models in Dynamic Migration of Business Processes

Ahana Pradhan and Rushikesh Krishnakant Joshi

IEEE Transactions on Services Computing

May/June 2018

Pages 562-579