

Workflow and Business Process Management

Dominic Duggan

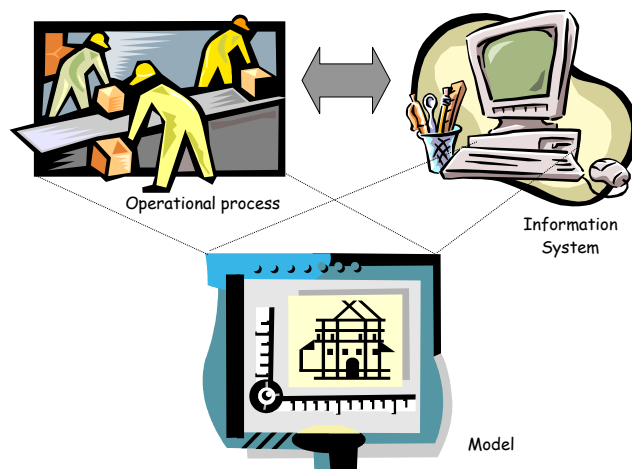
Stevens Institute of Technology

Based on material by Wil M.P. van der Aalst

1

1

Focus on Models



2

2

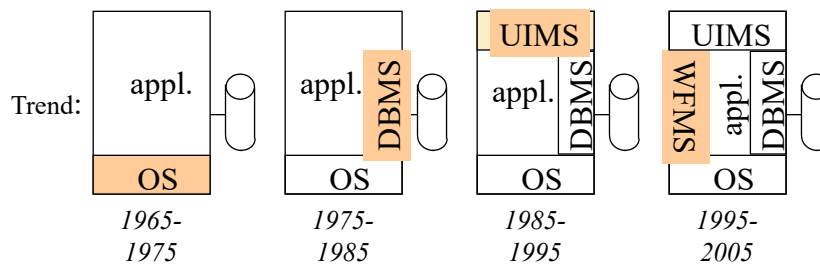
Workflow Management

- Goal
 - To manage the flow of work such that the work is done at the right time by the proper person.
- Definitions
 - A **workflow management system (WFMS)** is a software package that can be used to support the definition, management and execution of workflow processes.
 - A **workflow system (WFS)** is a system based on a WFMS that supports a specific set of business processes through the execution of computerized process definitions

3

3

Relevance of Workflow Mangt



Processes:

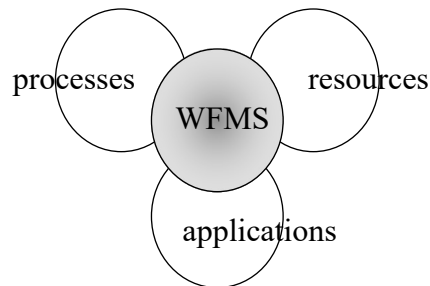
- are becoming more important
- are subject to frequent changes
- are becoming more complex
- are increasing in number

⇒ **Workflow Management System**

4

4

Basic Idea

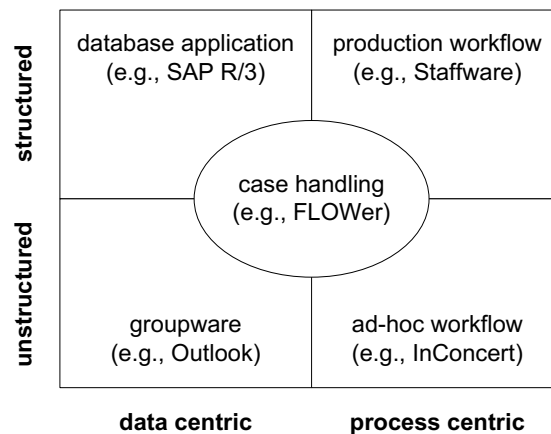


- separation of processes, resources and applications
- focus on the logistics of work processes, not on the contents of individual tasks

5

5

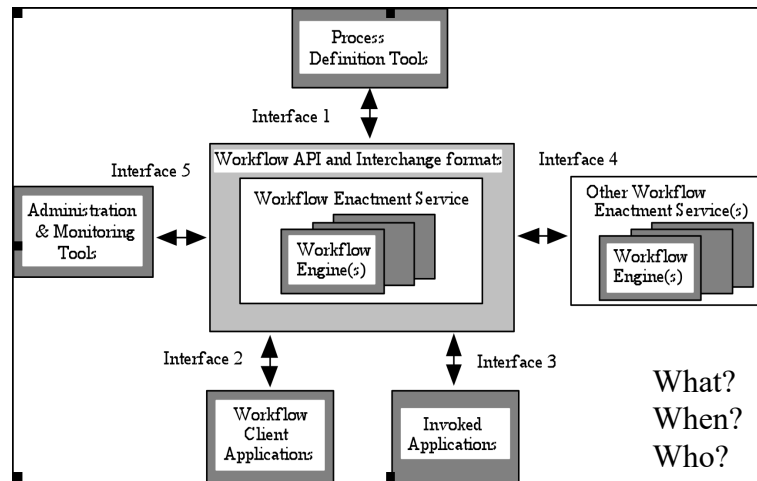
Computer Supported Cooperative Work (CSCW) spectrum



6

6

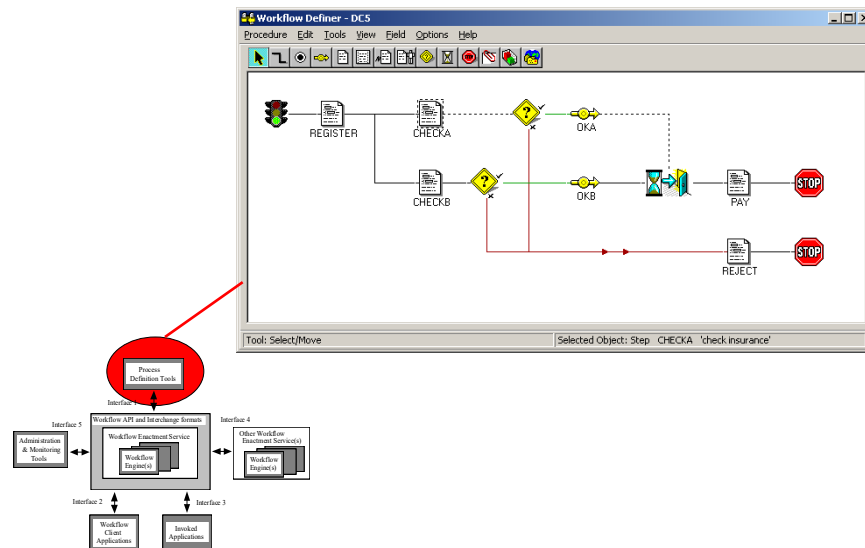
Reference model of the Workflow Management Coalition



7

7

Developer designs the process model



8

8

Worker selects a work item from a queue

The screenshot shows the 'Staffware 2000 System Administrator : DC_register - Work Items' window. The 'Work Queues' pane on the left lists various queues under 'User Queues', 'Group Queues', and 'Test Queues'. The 'DC_register' queue is selected. The 'Form: register' dialog is open, showing the 'Register claim' form with the text 'This is the first step of the Double Check Process' and a text field for 'Enter name' containing 'Case Jansen'. A 'Close Form' dialog is also visible, asking to 'Close the Form Window and' with 'Release' and 'Keep' options.

The diagram below illustrates the workflow engine architecture. It shows a central 'Workflow Engine' box with several interfaces: 'Interface 1' (Process Definition Tools), 'Interface 2' (Workflow Client Applications), 'Interface 3' (Invoked Applications), 'Interface 4' (Other Workflow Engines), and 'Interface 5' (Administration & Monitoring Tools). The 'Workflow Engine' box also contains 'Workflow Engine(s)' and 'Workflow Engine(s)'.

9

Administrator defines user accounts

The screenshot shows the 'Staffware Administration Managers : staffw_admin' window. The 'User Manager' tab is active, displaying a list of users. The 'Staffware User Manager' window is also open, showing the 'Users' tab with a list of users. The 'Attributes for: wvdalst' window is open, showing the 'Current Value' for the user 'wvdalst' as 'Wij van der Aalst'.

The diagram below illustrates the workflow engine architecture. It shows a central 'Workflow Engine' box with several interfaces: 'Interface 1' (Process Definition Tools), 'Interface 2' (Workflow Client Applications), 'Interface 3' (Invoked Applications), 'Interface 4' (Other Workflow Engines), and 'Interface 5' (Administration & Monitoring Tools). The 'Workflow Engine' box also contains 'Workflow Engine(s)' and 'Workflow Engine(s)'.

10

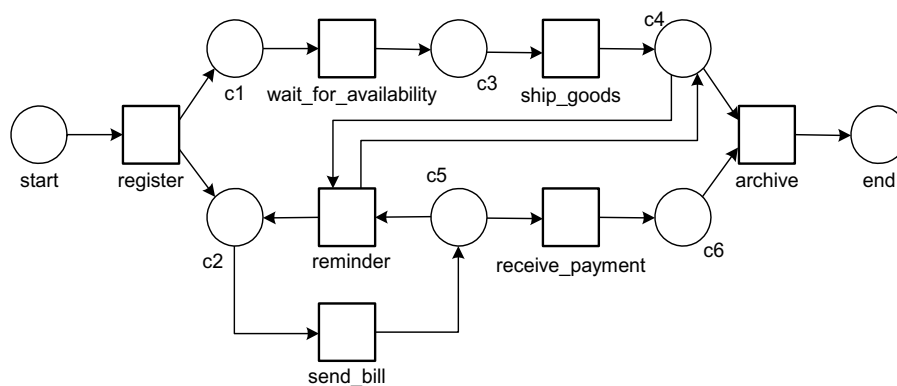
Workflow perspectives

- Process perspective (tasks and the routing of cases)
- Resource perspective (workers, roles, 4-eyes principle, etc.)
- Case/data perspective (process instances and their attributes)
- Operation/application perspective (forms, application integration, etc.)
- ...

11

11

Petri Net Process Model



12

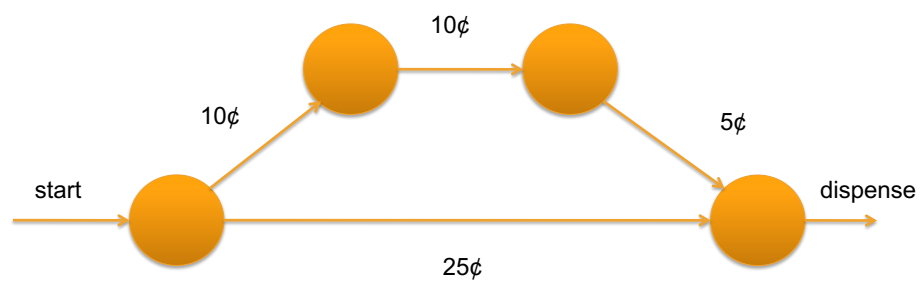
12

PETRI NETS

13

13

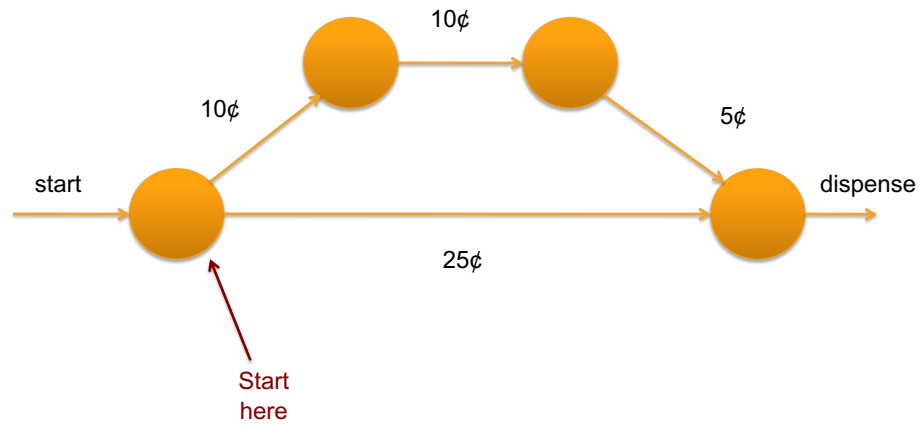
State Machines



14

14

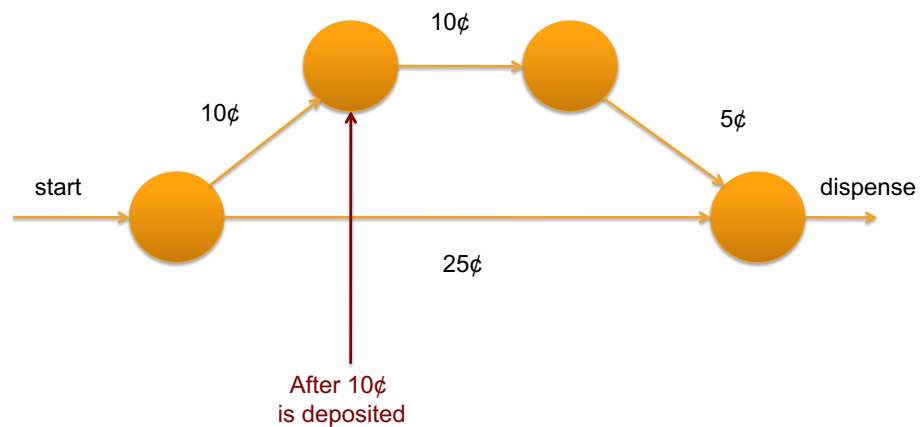
State Machines



15

15

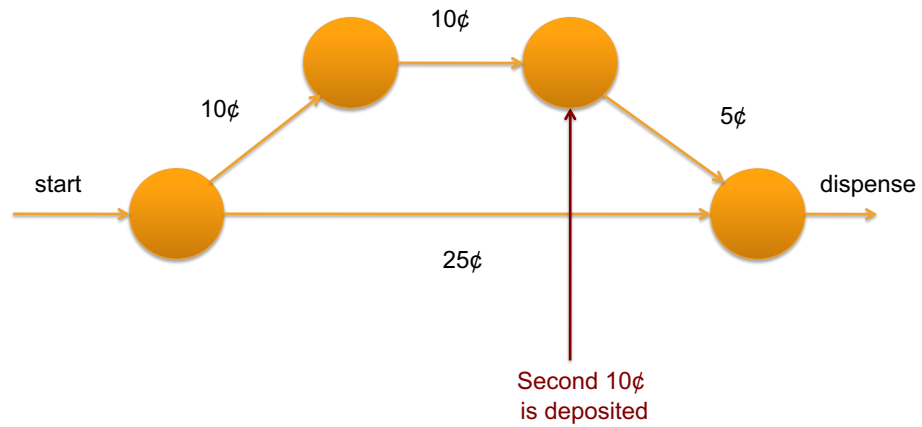
State Machines



16

16

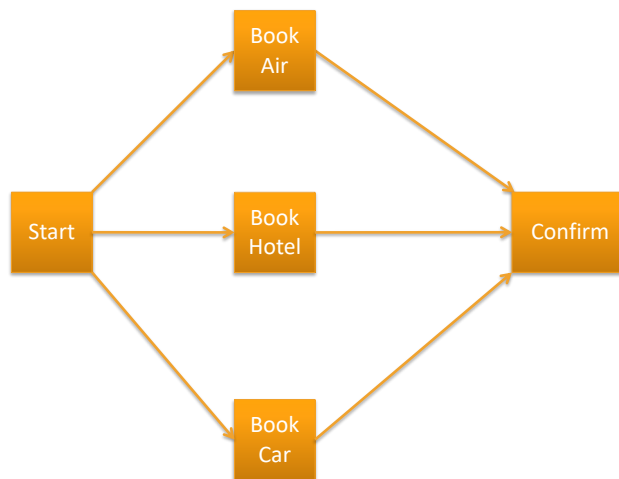
State Machines



17

17

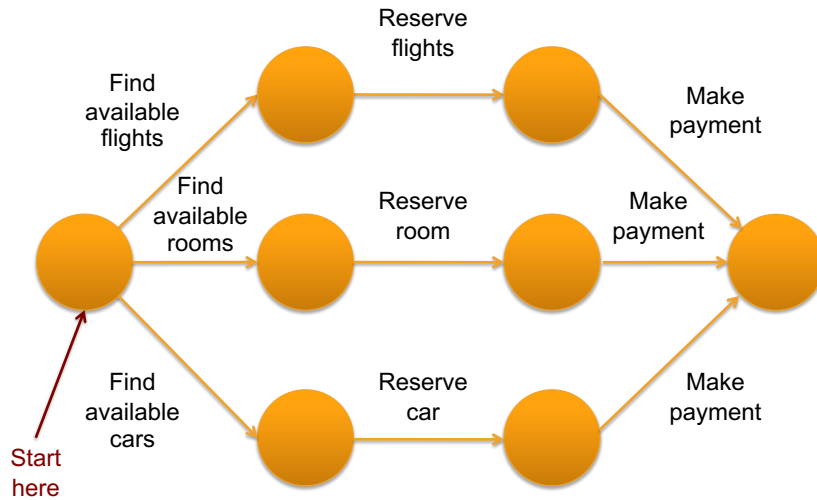
PERT Charts



18

18

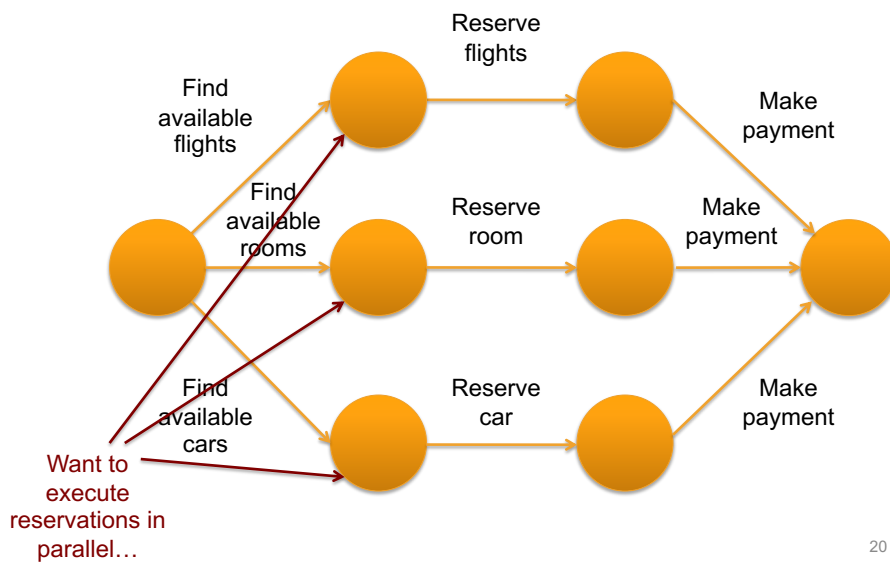
State Machines



19

19

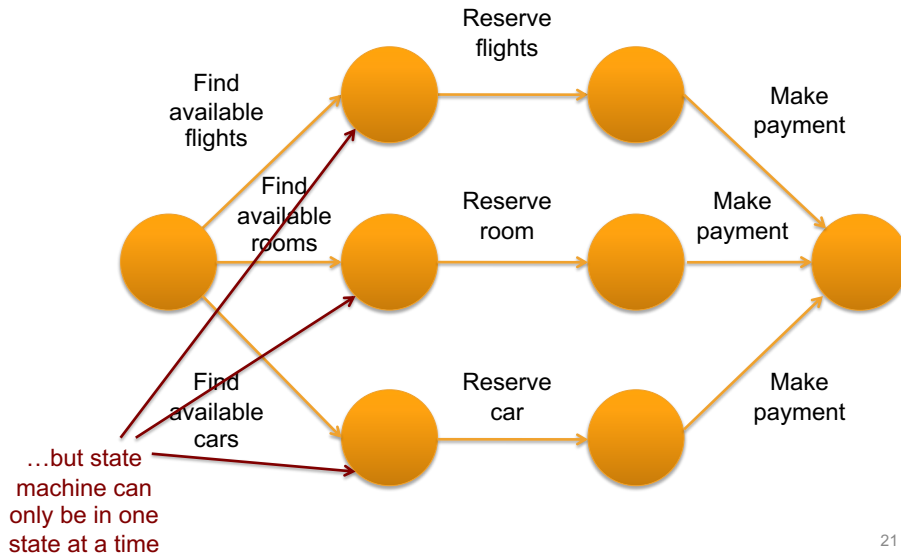
State Machines



20

20

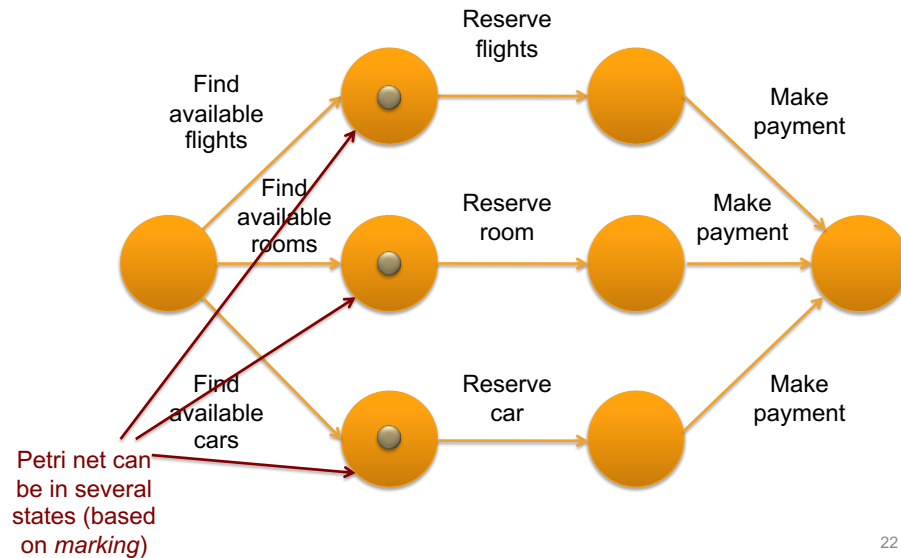
State Machines



21

21

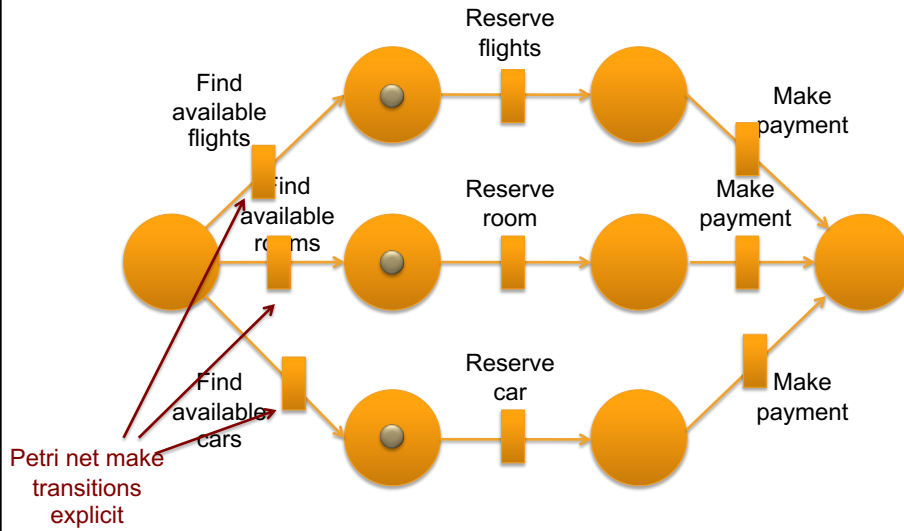
State Machines



22

22

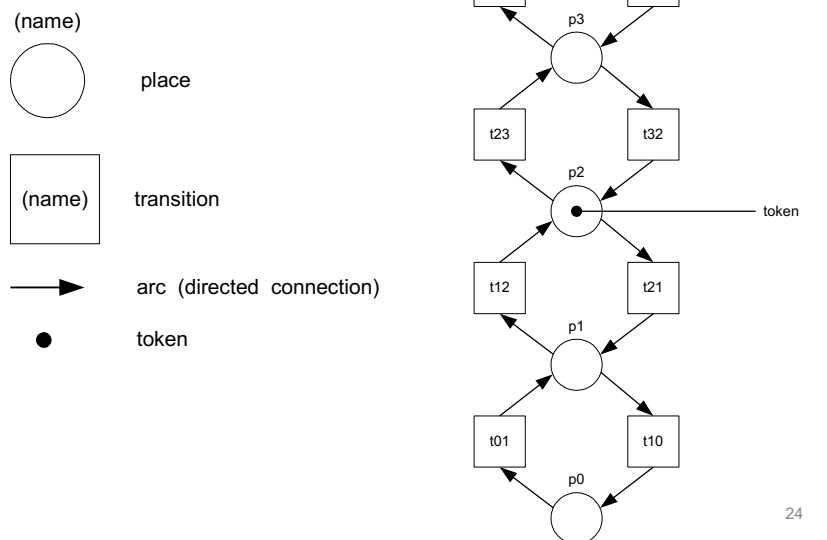
State Machines



23

23

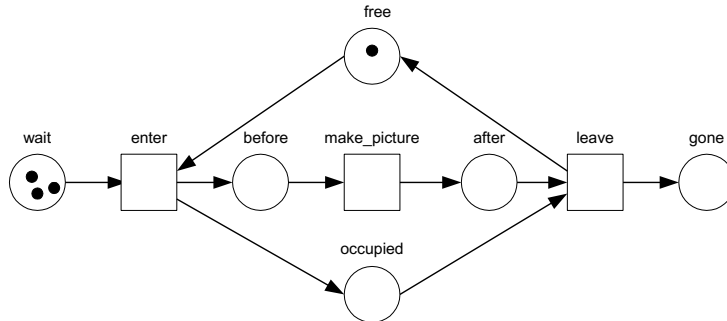
Elements of Petri Nets



24

24

Rules



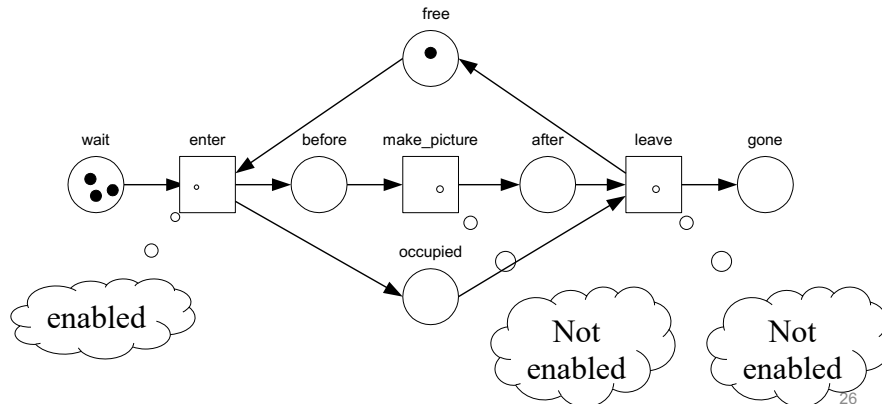
- Connections (**arcs**) are directed
- Only connect **place**→**transition** or **transition**→**place**
- **Places** may hold zero or more **tokens**

25

25

Enabled

- A transition is **enabled** if each of its input places contains at least one token.

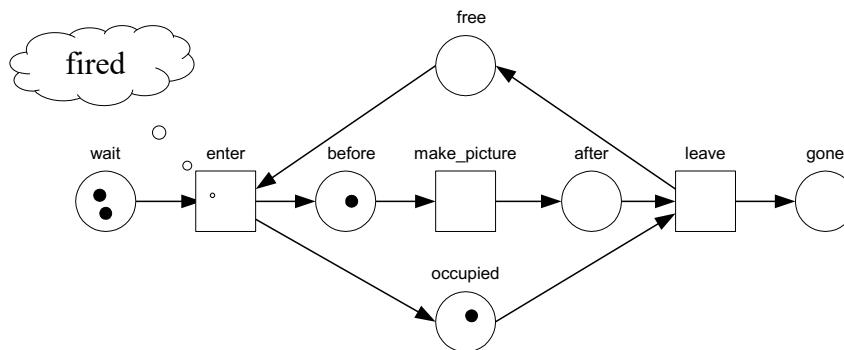


26

26

Firing

- **Firing** of an enabled transition:
 - consume a token from each input place
 - produce a token for each output place



27

27

Remarks

- Firing is atomic
- Multiple transitions may be enabled
 - Non-determinism: which transition fires?
- The number of tokens may vary
 - If #input places \neq #output places
- **Marking**

28

28

Non-determinism

