Ch2 Modeling Workflows

1. Workflow Concepts
2. Petri Nets
3. Mapping Workflow Concepts onto Petri Nets

2.1 Workflow Concepts-The Case

- The primary objective of a workflow system is to deal with cases.
- case type: Abstarct of a group of similar cases
- Each case has a unique identity.
- A case has a limited lifetime defined by process.
- state: consists of three elements:
  - case attributes: A range of variables can be associated with each case which are used to manage it. the value of a case attribute may change as the case progresses.
  - conditions: Show how far a case has progressed. A condition is regarded as a requirement that must be met before a particular task may be carried out.
  - the content of the case: not managed by the workflow management system.

2.1 Workflow Concepts-The Task

- A task is a logical unit of work which is indivisible and thus is always carried out in full.
  - rollback
  - Task Types: manual, automatic and semi-automatic tasks.
- work item: One is the combination of a case and a task which is just about to be carried out.
- activity: refers to the actual performance of a work item.

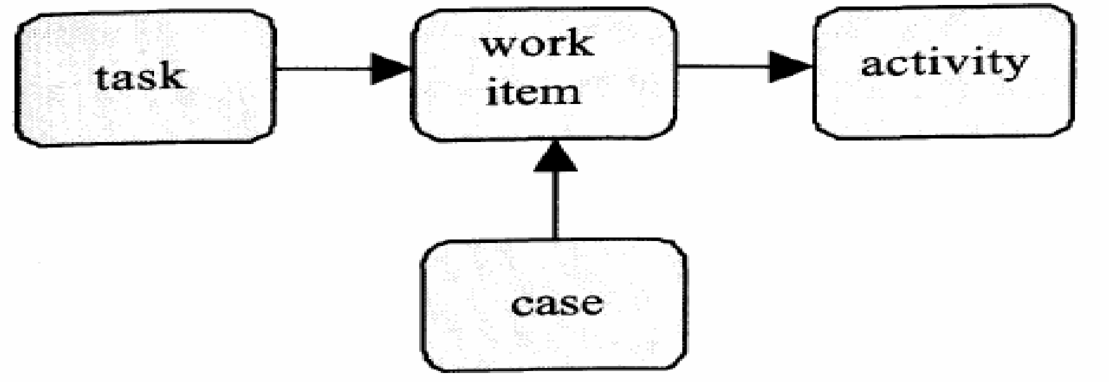


Figure 2.1
The relationship between the terms task, case, work item, and activity

2.1 Workflow Concepts-The Process

- Process:The way in which a particular category of cases should be carried out is described by the relevant process, which indicates which tasks need to be carried out and the order in which these tasks should be done.
- We can also regard a process as a procedure for a particular case type. many different cases are handled using a single process.
- a process is constructed from tasks and conditions.
- Conditions are used to decide which order is followed.
- subprocesses, reuse, hierarchical structure
- The lifecycle of a case is defined by a process.

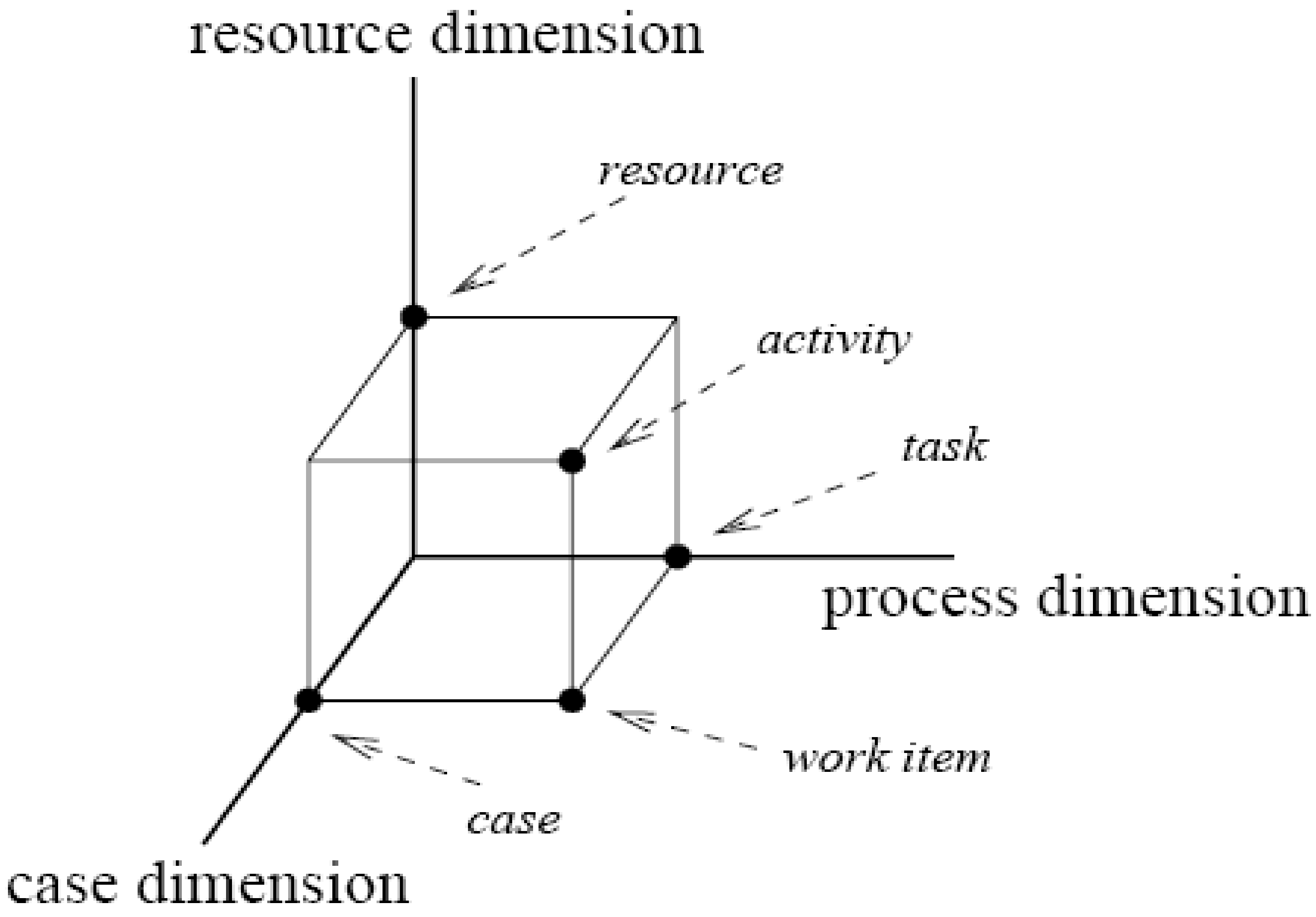
2.1 Workflow Concepts-Routing

- the routing of the case: Routing along particular branches determines which tasks need to be performed (and in which order).
- four basic routing constructions:
  - sequential routing
  - parallel routing: AND-split, AND-join
  - selective routing: OR-split, OR-join
  - iterative routing

2.1 Workflow Concepts-Enactment

- triggering: the work item is triggered by a resource
- three types of triggers:
  - a resource initiative
  - an external event
  - a time signal
- Work items which must always be carried out immediately—without the intervention of external stimuli— do not require a trigger.

2.1 Workflow Concept Space



2.2 Petri Nets

- Petri nets: based upon which establish formalism for the modeling and analysis of processes
- Petri nets were devised in 1962 by Carl Adam Petri as a tool for modeling and analyzing processes.
  - Graphical
  - a strong mathematical basis, entirely formalized

2.2 Petri Nets - Classical Petri nets

- A Petri net consists of places, transitions and directed arc.
- two types of arcs: those that run from a place to a transition and those that run from a transition to a place.

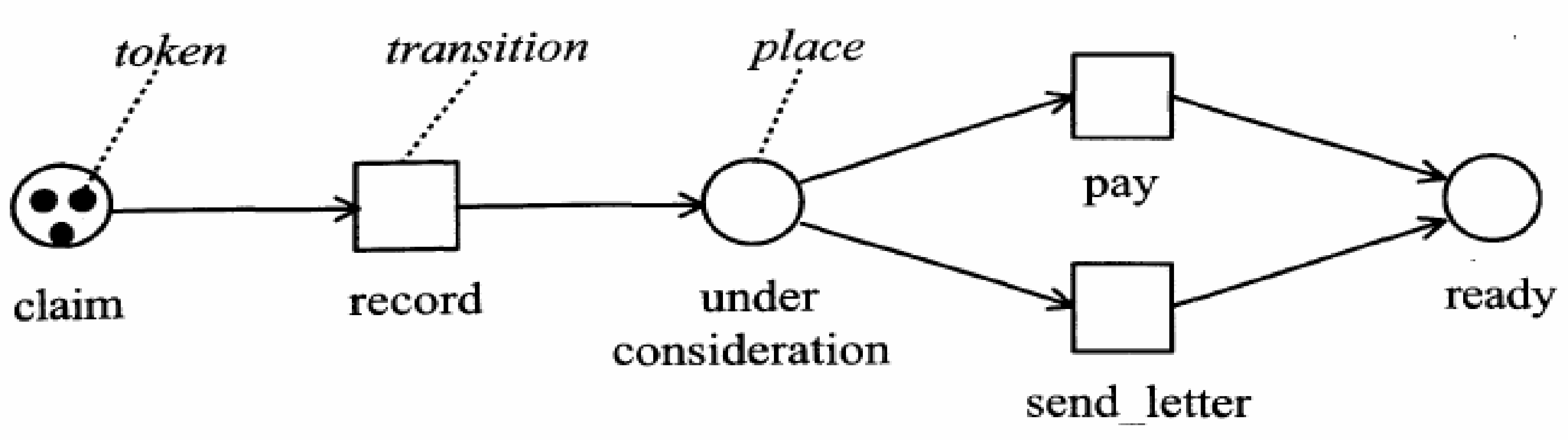


Figure 2.2
A classic Petri net

2.2 Petri Nets - Classical Petri nets

- input place & output place
- token: The structure of a Petri net is fixed; however, the distribution of its tokens among the places can change.
- The state of a Petri net is indicated by the distribution of tokens amongst its places.
- A transition may only fire if it is enabled. This occurs when there is at least one token at each of its input places.
- firing: one token is removed from each input place and one token added to each output place.

2.2 Petri Nets - Classical Petri nets

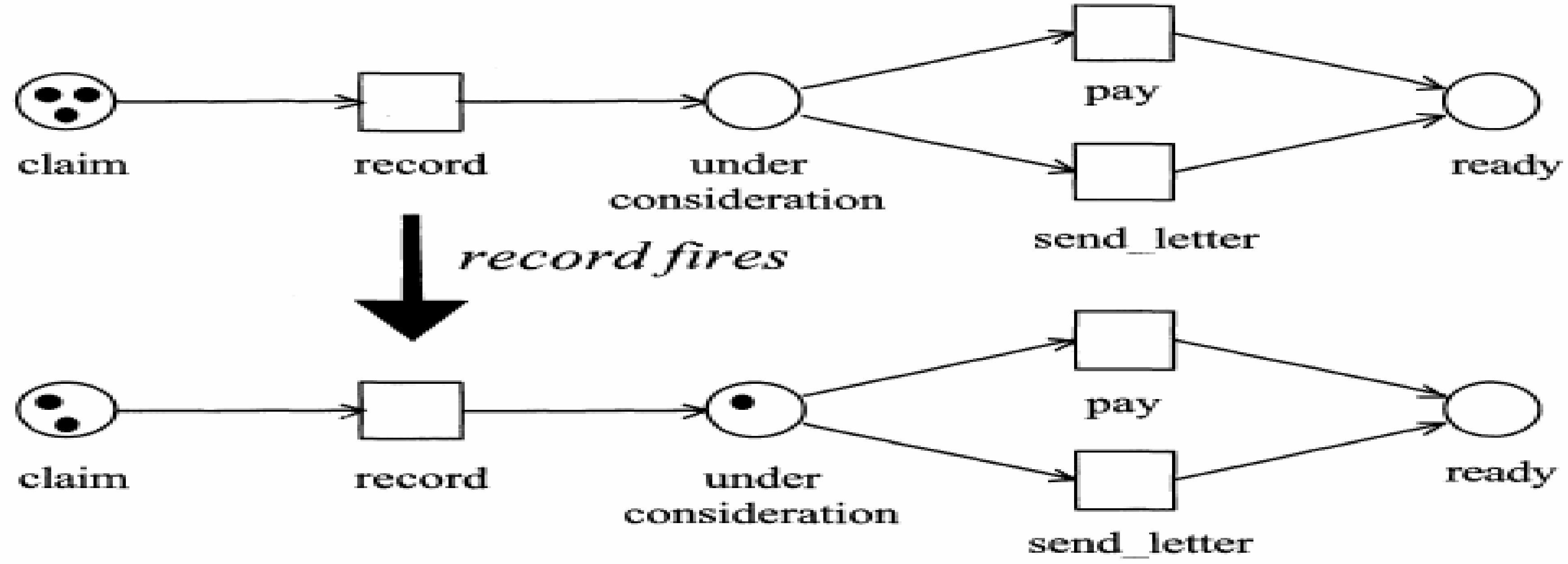


Figure 2.3
State before and after the transition "record" fires

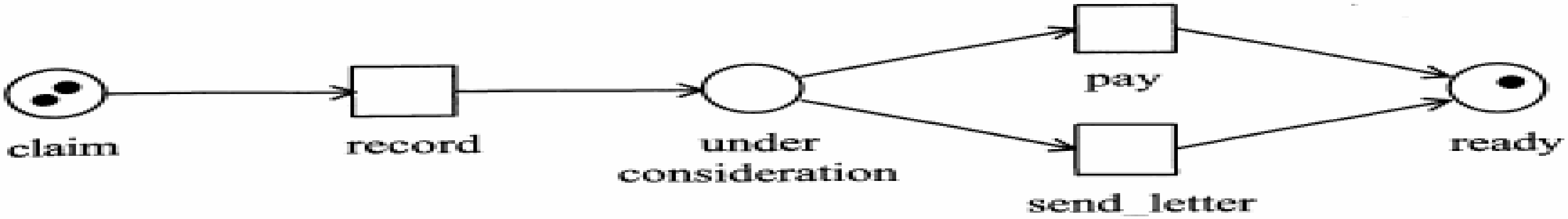


Figure 2.4
State after "pay" fires

2.2 Petri Nets - Classical Petri nets

- Transitions are the active components in a Petri net. By firing a transition, the process being modeled shifts from one state to another. A transition therefore often represents an event, an operation, a transformation, or a transportation.
- The places in a Petri net are passive, in the sense that they cannot change the network's state. A place usually represents a medium, buffer, geographical location, (sub)state, phase, or condition.
- Tokens often indicate objects. These can be physical ones, but also objects representing information.

2.2 Petri Nets - Classical Petri nets

- limit the number of cases which can be under consideration at the same time to a maximum of one

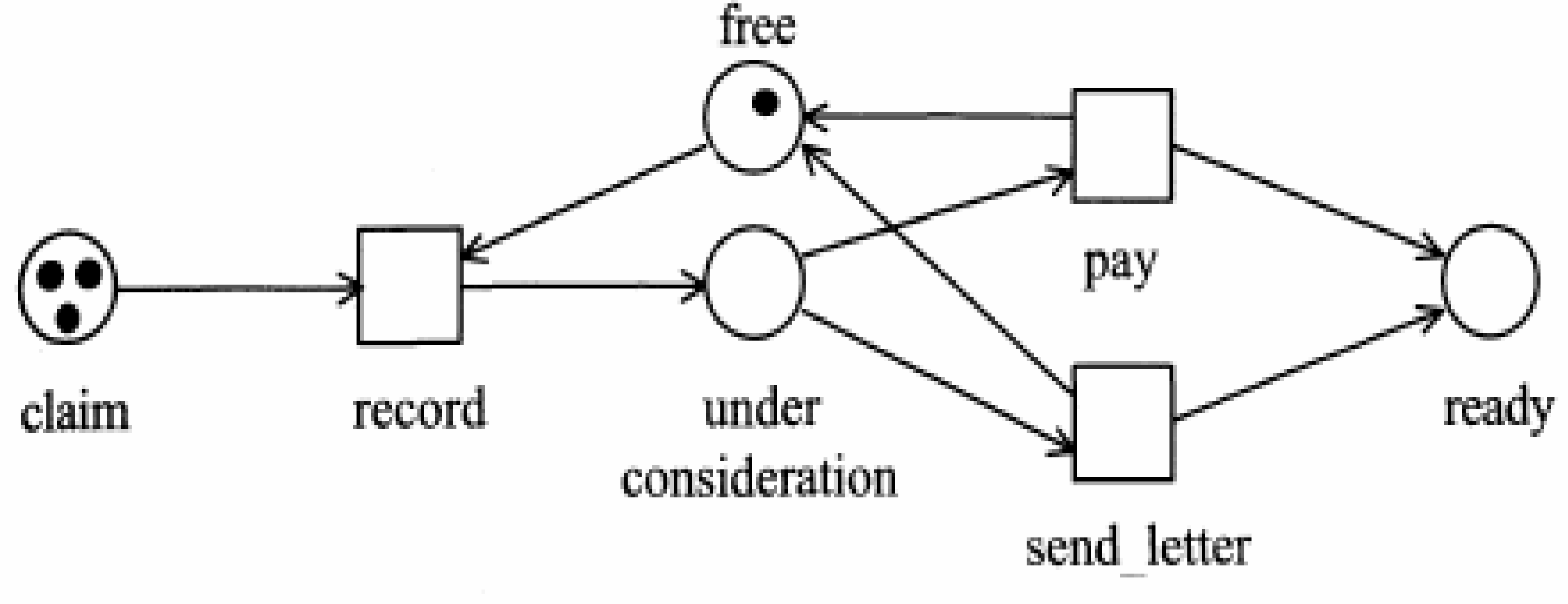


Figure 2.5
The modified Petri net

2.2 Petri Nets - Classical Petri nets

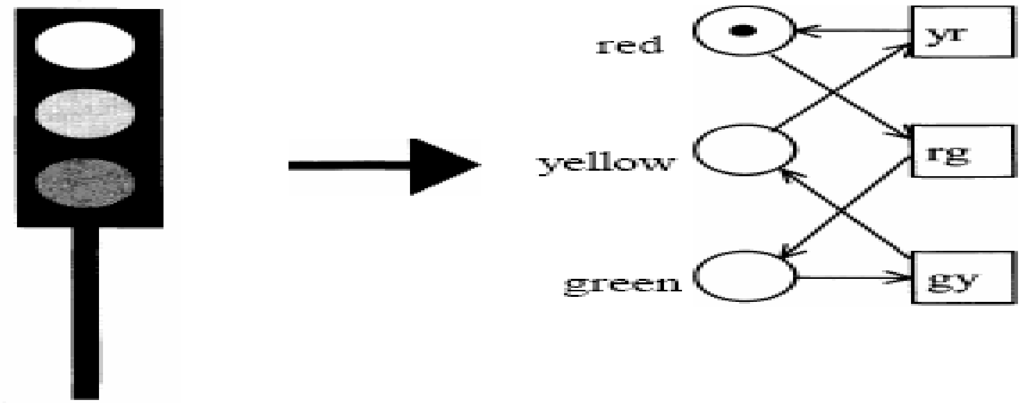


Figure 2.6
A set of traffic lights illustrated on a Petri net

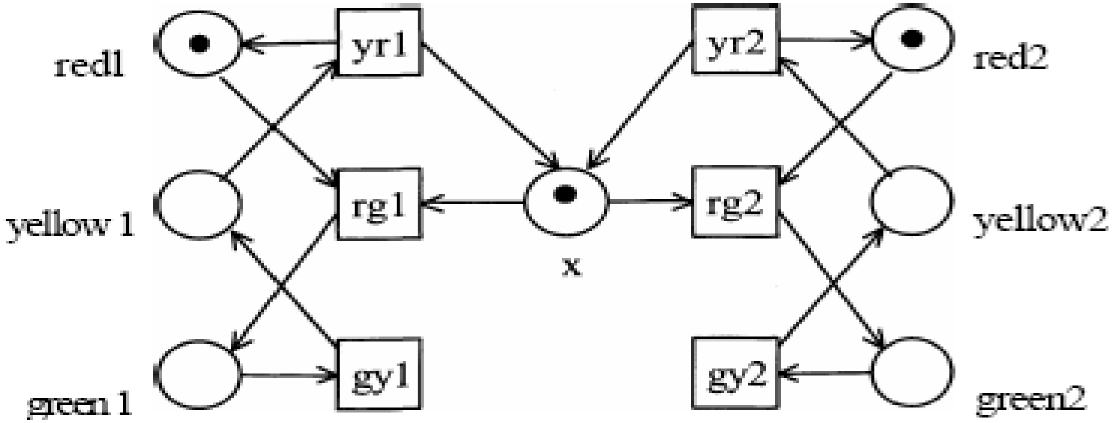


Figure 2.7
Two sets of traffic lights

- Using Petri nets to describe processes that are repetitive in nature

2.2 Petri Nets - High-level Petri nets

- Color extension
  - Tokens are used to model a whole range of things.
  - This extension ensures that each token is provided with a value or color.
  - A firing transition produces tokens that are based upon the values of those consumed during firing.The value of a produced token therefore may depend upon those of consumed ones.
  - the number of tokens produced is determined by the values of those consumed
  - a transition is only enabled once there is a token at each of the input places and the *preconditions* have been met.
  - A transition's precondition is a logical requirement connected with the values of the tokens to be consumed.

2.2 Petri Nets - High-level Petri nets

- Color extension
  - In contrast with the classic Petri net, it is now possible for an output place not to receive a token.

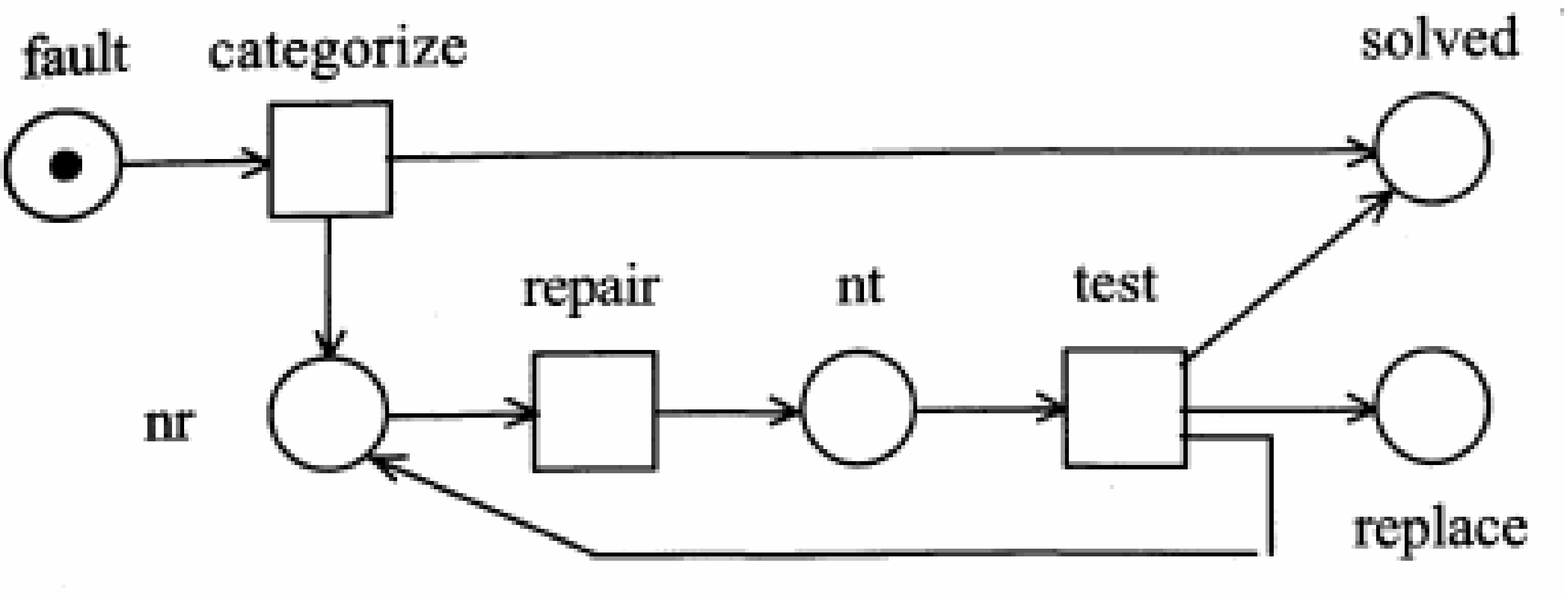


Figure 2.8
The process for dealing with faults

2.2 Petri Nets - High-level Petri nets

- Color extension
  - use a precondition to "synchronize" tokens.
  - The number of incoming arrows thus shows how many tokens there must be at the input point from which they come.

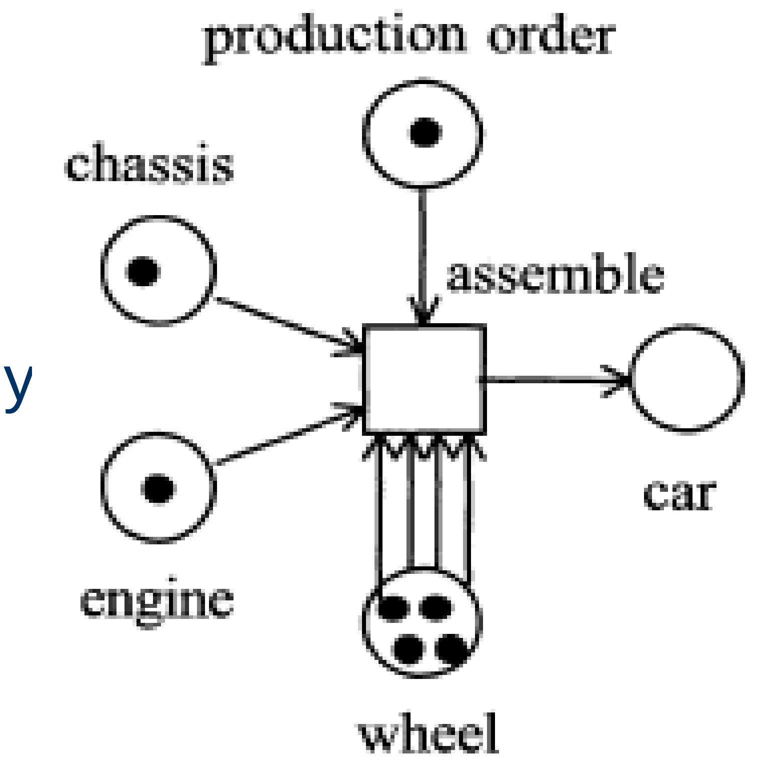


Figure 2.9
The transition "assemble

2.2 Petri Nets - High-level Petri nets

- Color extension: the graphic representation no longer contains all the information. For each transition, the following factors must be specified:
  - Whether there is a precondition. If there is a precondition, then this must be defined precisely.
  - The number of tokens produced per output place during each firing. This number may depend upon the values of the tokens consumed.
  - The values of the tokens produced. This, too, may depend upon the values of the tokens consumed.

2.2 Petri Nets - High-level Petri nets

- Time extension
  - tokens receive a *timestamp* as well as a value
  - the *enabling time* of a transition is the earliest moment at which its input places contain sufficient *available* tokens.
  - Tokens are consumed on a FIFO (first-in, first-out) basis. The token with the earliest timestamp thus is the first to be consumed. Furthermore, it is the transition with the earliest *enabling time* that fires first.
  - If there is more than one transition with the same enabling time, a non-deterministic choice in made. Moreover, the firing of one transition may affect the enabling time of another.
  - The timestamp of a produced token is equal to the time of firing plus this delay. The length of the delay may depend upon the value of the tokens consumed. However, it is also possible that the delay has a fixed value (for example, 0) or that the delay is decided at random. Firing itself is instantaneous and takes no time.

2.2 Petri Nets - High-level Petri nets

- Time extension

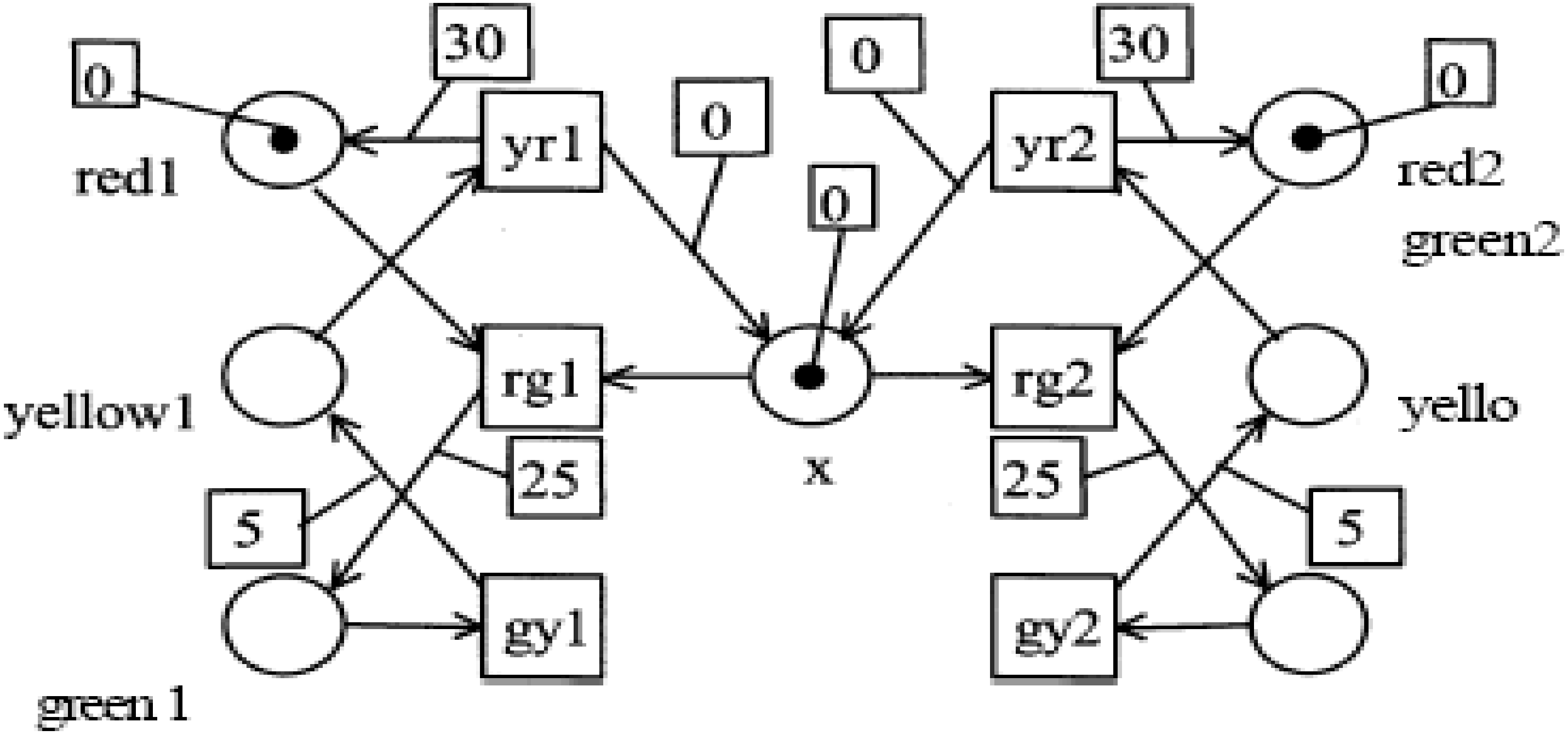


Figure 2.10
The two sets of traffic lights with time

2.2 Petri Nets - High-level Petri nets

- Hierarchical extension
  - The hierarchical extension therefore ensures that it becomes possible to add structure to the Petri net model.
  - Subprocess
  - a process can take two forms: (1) as a subprocess within a hierarchically superior process (2) as the definition of the process
  - When modeling complex processes, a hierarchical method of description is often an absolute necessity.

2.2 Petri Nets - High-level Petri nets

- Hierarchical extension

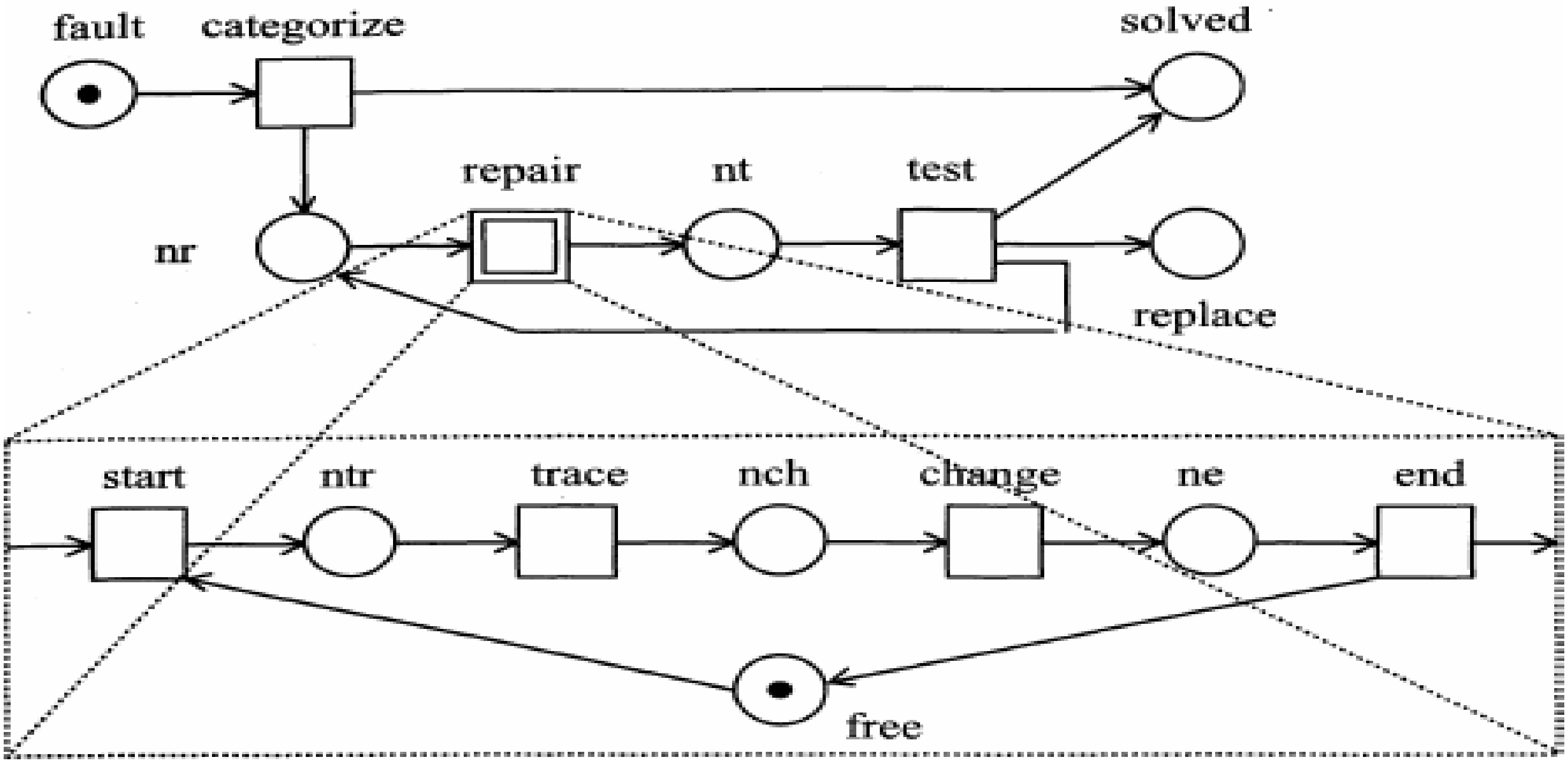


Figure 2.11
The process "solve fault" contains one subprocess: "repair

2.3 Mapping Workflow Concepts onto Petri Nets

- Workflow Concepts: case, task, condition, process, trigger
- Process
    - conditions -- places, tasks -- transitions
    - "entrance" -- source place, "exit" -- sink place
    - Case – token
    - The conditions play two important roles:
        - the order of tasks proceeding
        - the state of the case

2.3 Mapping Workflow Concepts onto Petri Nets

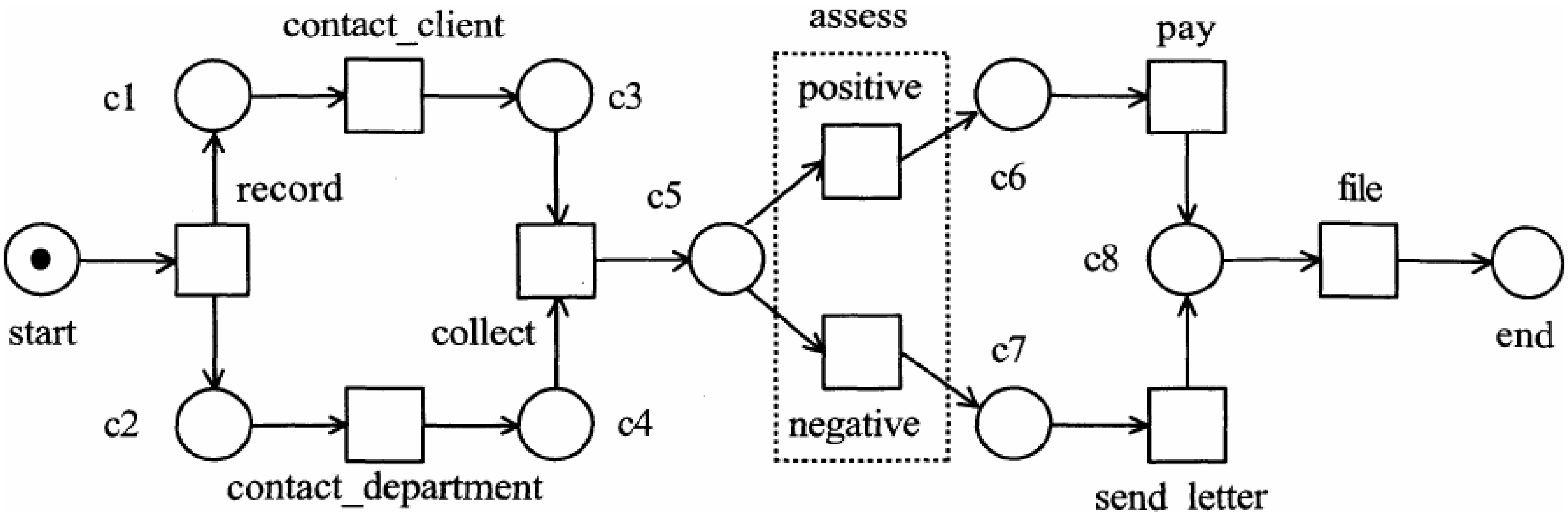


Figure 2.12
The process "handle complaint" modeled as a Petri net

- each process should fulfil two requirements: (1) it should at any time be possible to reach—by performing a number of tasks—a state in which there is a token in end; and (2) when there is a token in end, all the others should have disappeared.

2.3 Mapping Workflow Concepts onto Petri Nets

- The state of a case is not determined solely by the conditions that have been met. the case may have one or more attributes. For these, it seems obvious to use the color extension. The value of a token contains information about the attributes of the case in question.

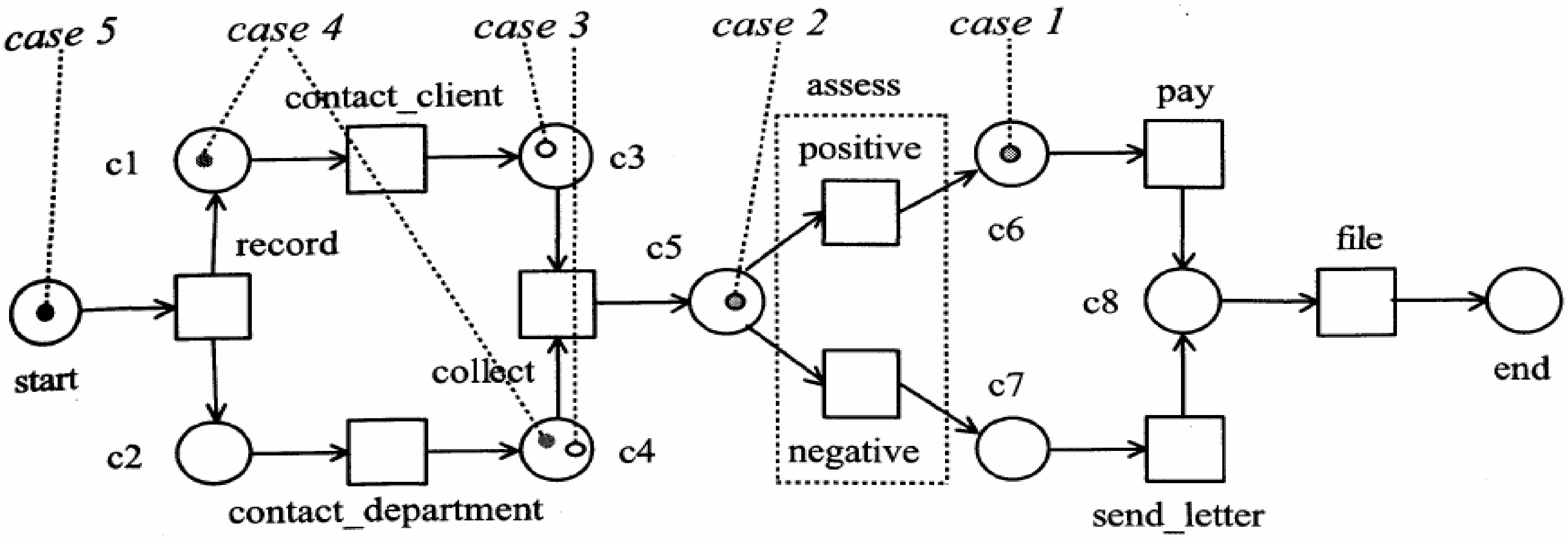


Figure 2.13
Each case is illustrated using one or more tokens

2.3 Mapping Workflow Concepts onto Petri Nets

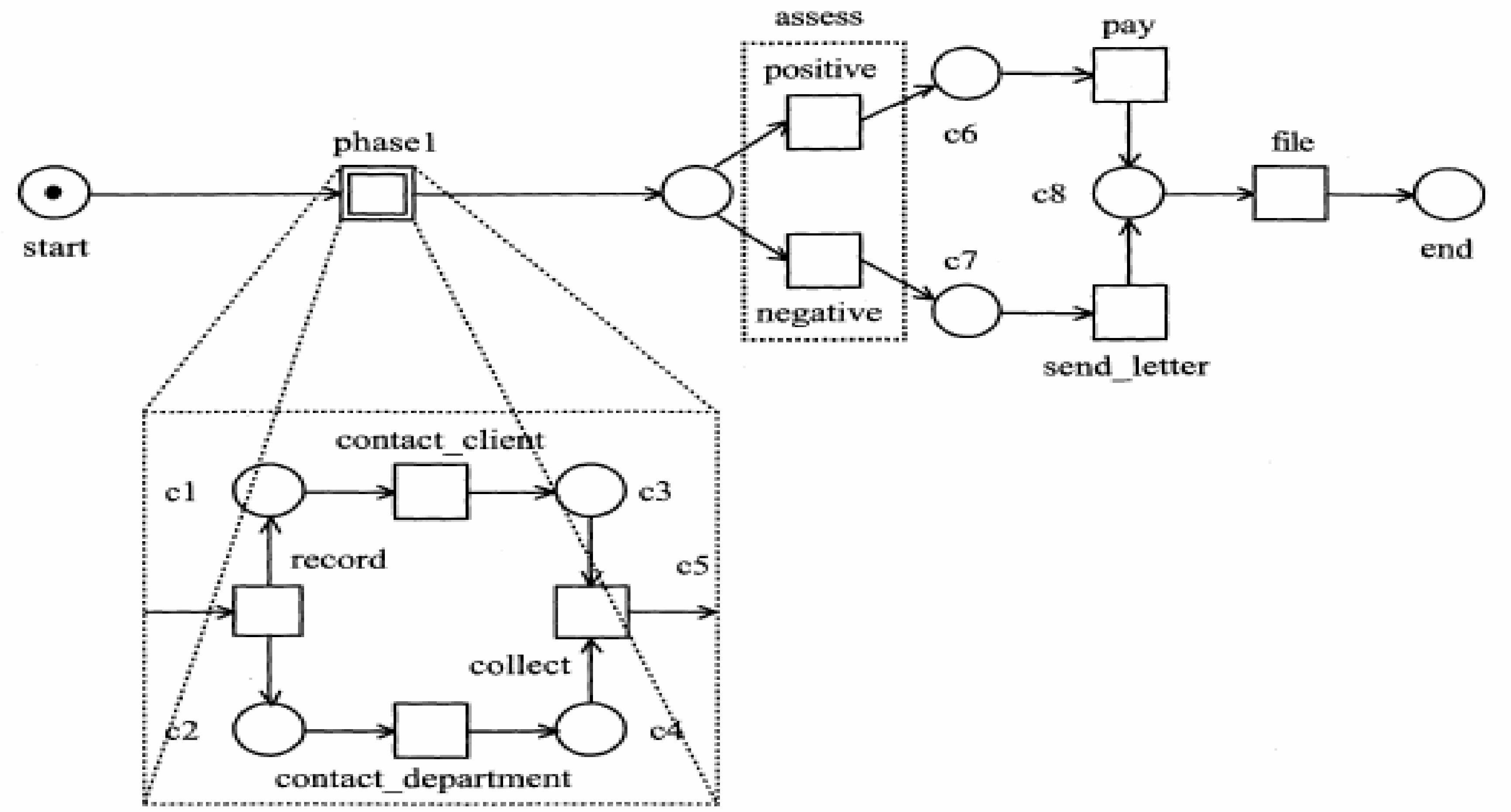


Figure 2.14
The process "handle complaint" now contains the subprocess "phase 1

2.3 Mapping Workflow Concepts onto Petri Nets

- Routing
  - Sequential routing: linking the two tasks using a place
  - Parallel routing: AND-split and the AND-join. "artificial" additions *management tasks,*

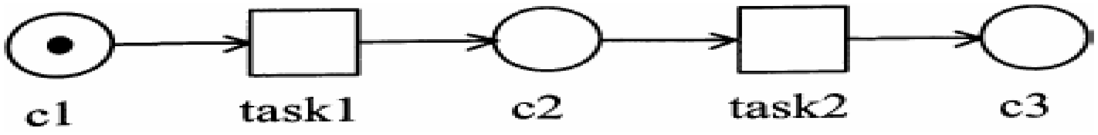


Figure 2.15
Sequential routing

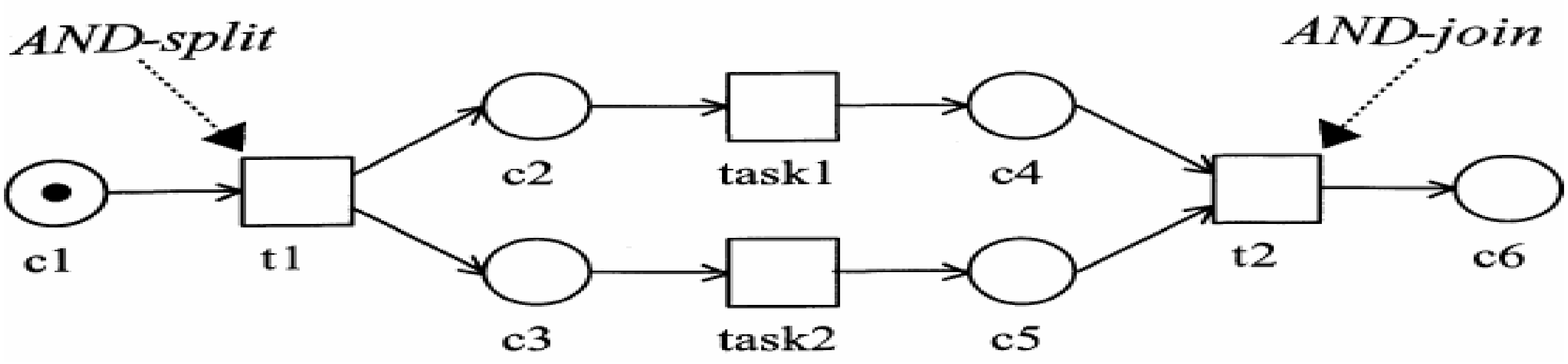


Figure 2.16
Parallel routing

2.3 Mapping Workflow Concepts onto Petri Nets

- Nondeterministic Select: the selection is left to the environment of the workflow system.
  - explicit model: Fig2.17, choice is made at the moment when there is a token in cl
  - implicit model: Fig2.18, choice is made at the moment when either task1 or task2 has to be carried out

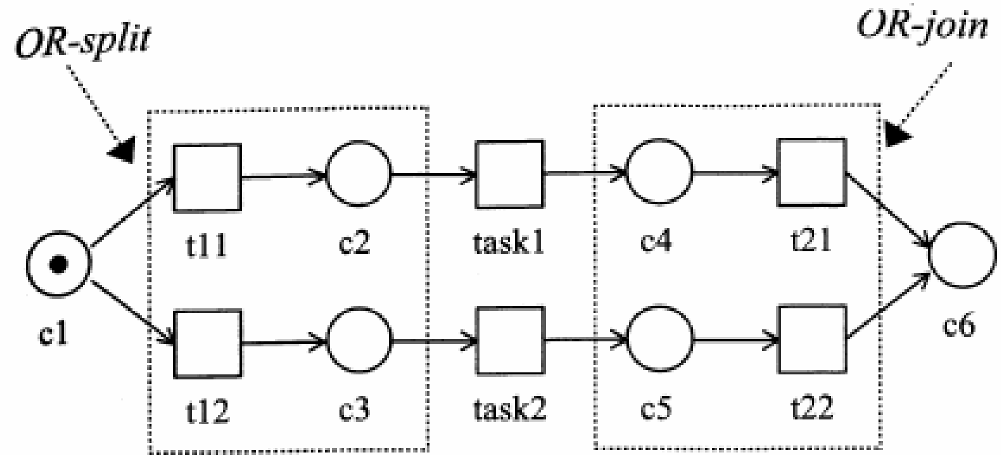


Figure 2.17
Selective routing (1)

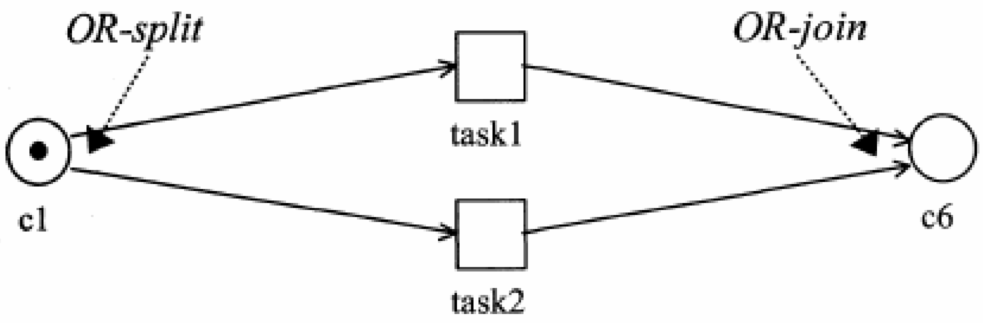


Figure 2.18 Selective routing (2)

2.3 Mapping Workflow Concepts onto Petri Nets

- Deterministic Select: the decision is made according to the specific properties of the case
    - color extension
    - model1: Fig2.19
    - model2: Fig2.20
    - the OR-splits in figures 2.19 and 2.20 are equivalent ? p57

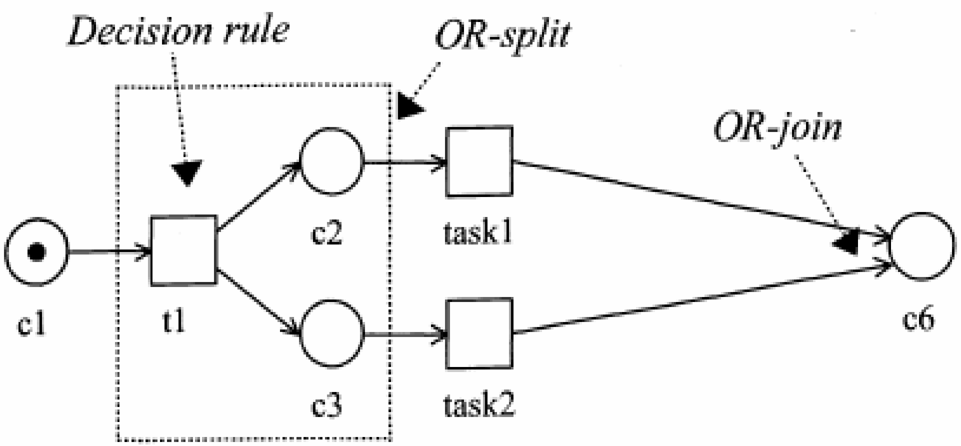


Figure 2.19
Selective routing (3)

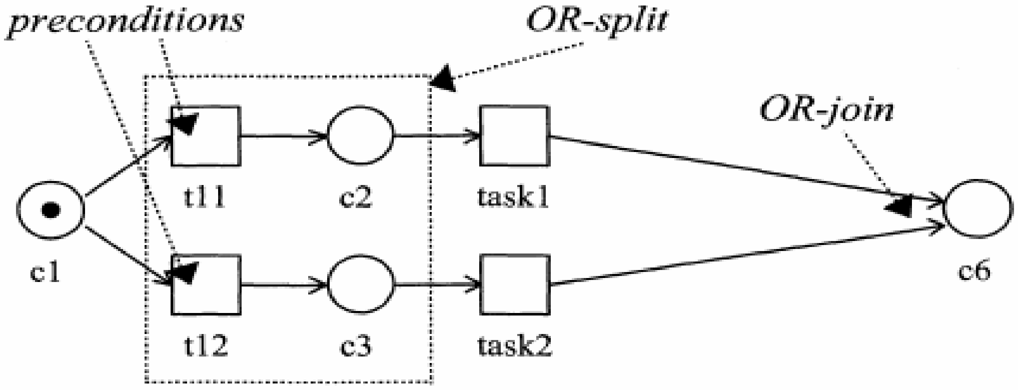


Figure 2.20
Selective routing (4)

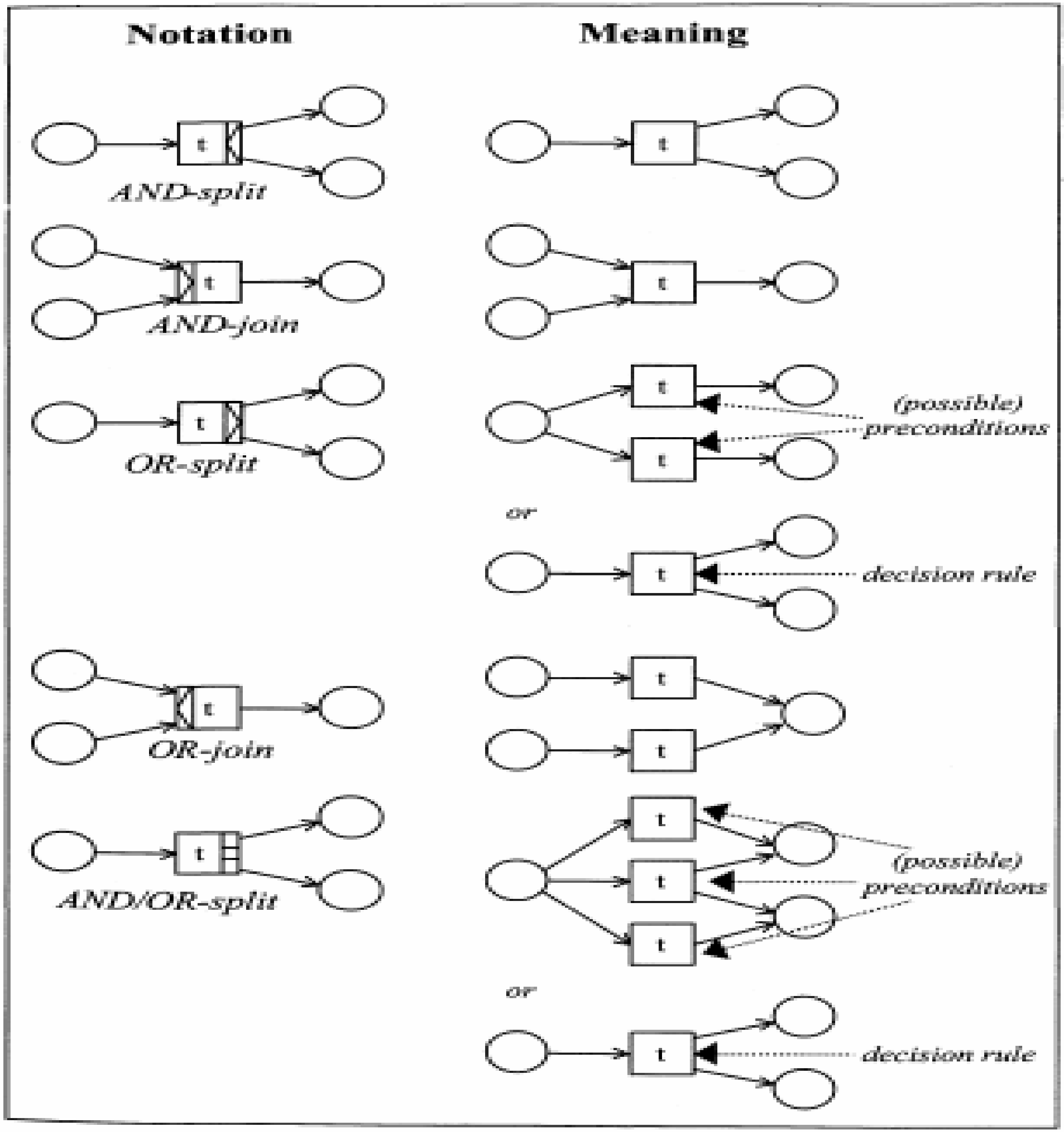
2.3 Mapping Workflow Concepts onto Petri Nets

- In the case of parallel routing, there may be more than one token assigned to the same case.
- we must ensure that a change to a case attribute caused by the performance of a task updates the value of every token pertaining to that case.
- Model 1 disobey the transition rule of Classical Petri Net.

2.3 Mapping

Figure 2.21
Notation method for common constructions

Modeling Workflows
59



2.3 Mapping Workflow Concepts onto Petri Nets

- Iterative routing
  - Fig 2.22: repeat ... until ...
  - Fig 2.23: while ... do ...

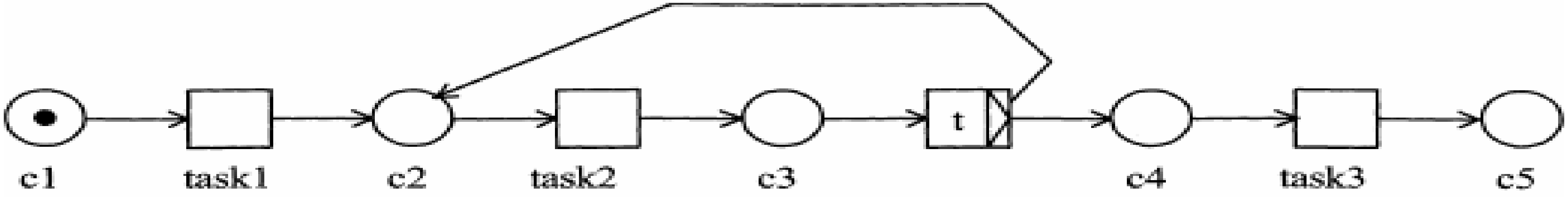


Figure 2.22
Iterative routing (1)

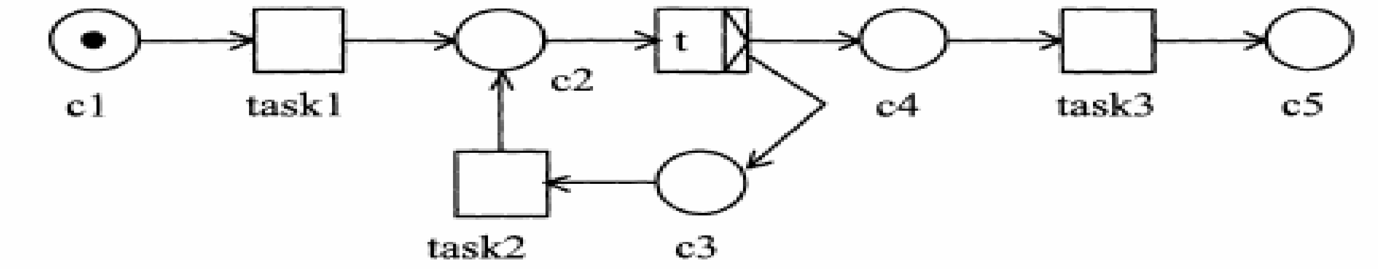


Figure 2.23
Iterative routing (2)

2.3 Mapping Workflow Concepts onto Petri Nets

- Enactment
  - A task corresponds with one or more transitions, a work item with a transition being enabled, and an activity with the firing of a transition.
  - three types of triggers: (1) a resource initiative; (2) an external event; (3) a time signal
  - To each transition belonging to a task requiring a trigger an extra input place is added
  - The triggering mechanism also shows that the timing of an OR-split choice is crucial.
  - atomic, rollback

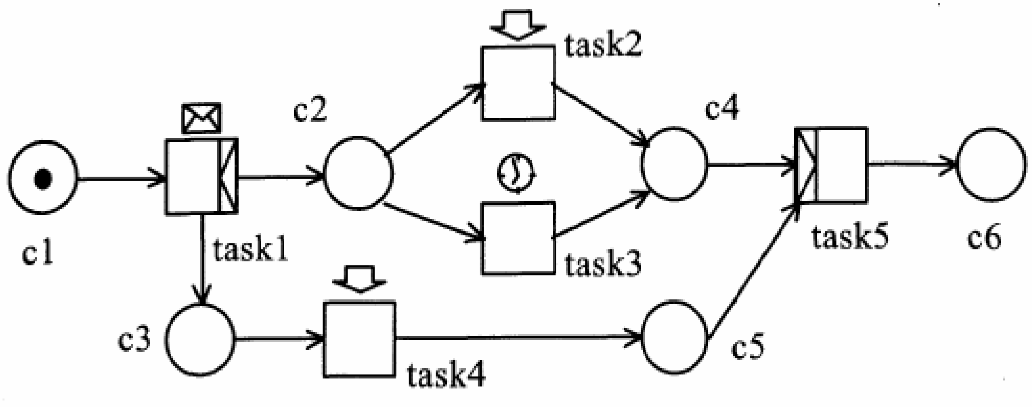


Figure 2.25
An example with various forms of triggering

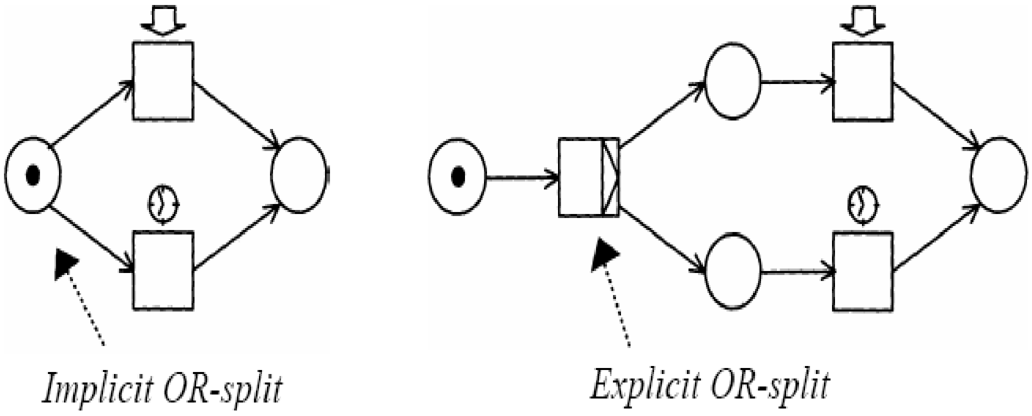


Figure 2.26
There is an essential difference between the implicit and explicit OR-split

2.3 Mapping Workflow Concepts onto Petri Nets

- Example: Travel agency

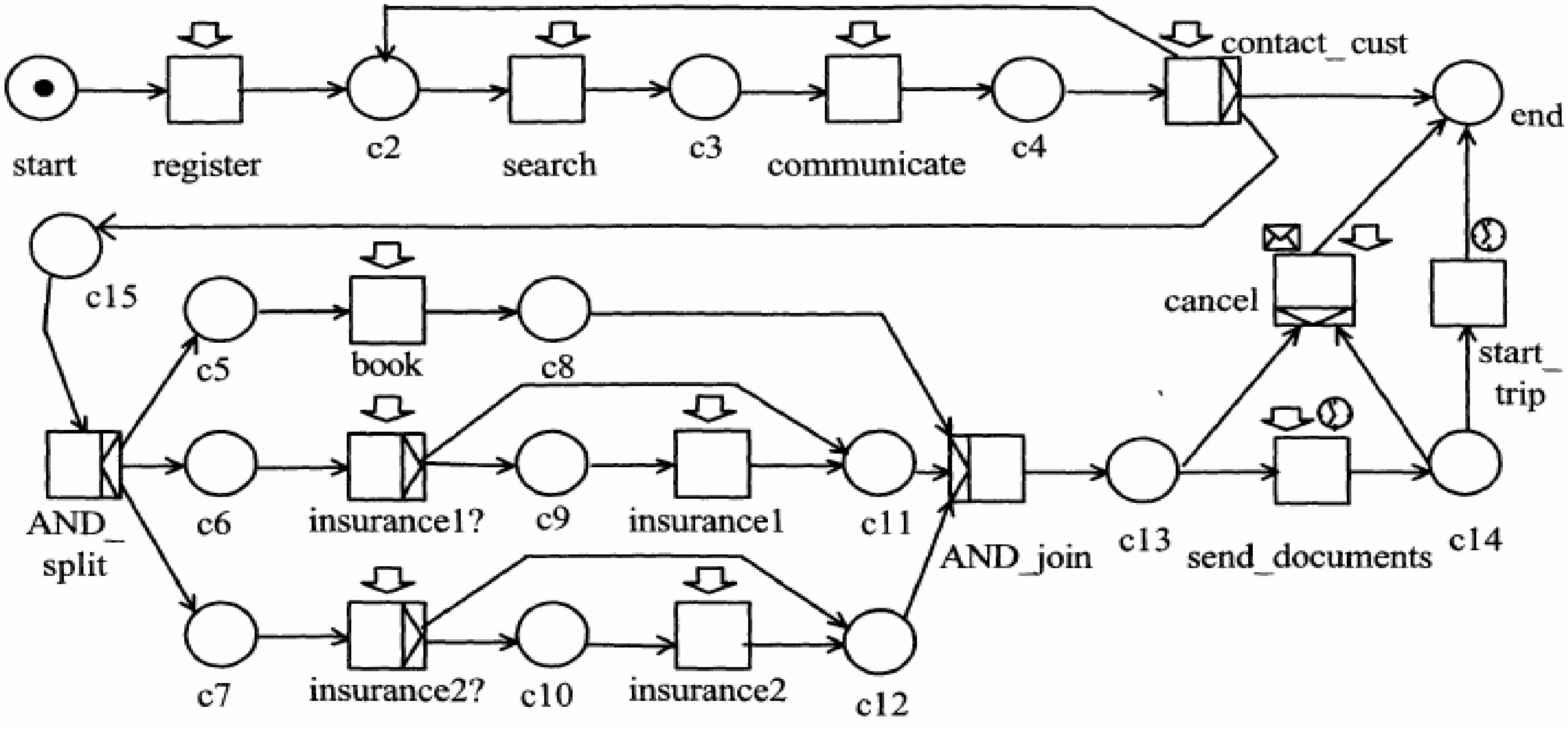


Figure 2.27
The travel agency

Homework

- P69 Exercise 2.2 Project X