Slides contain material from www.yawlfoundation.org
Developed by Wil van der Aalst, Michael Adams, Arthur ter Hofstede, Nick Russel

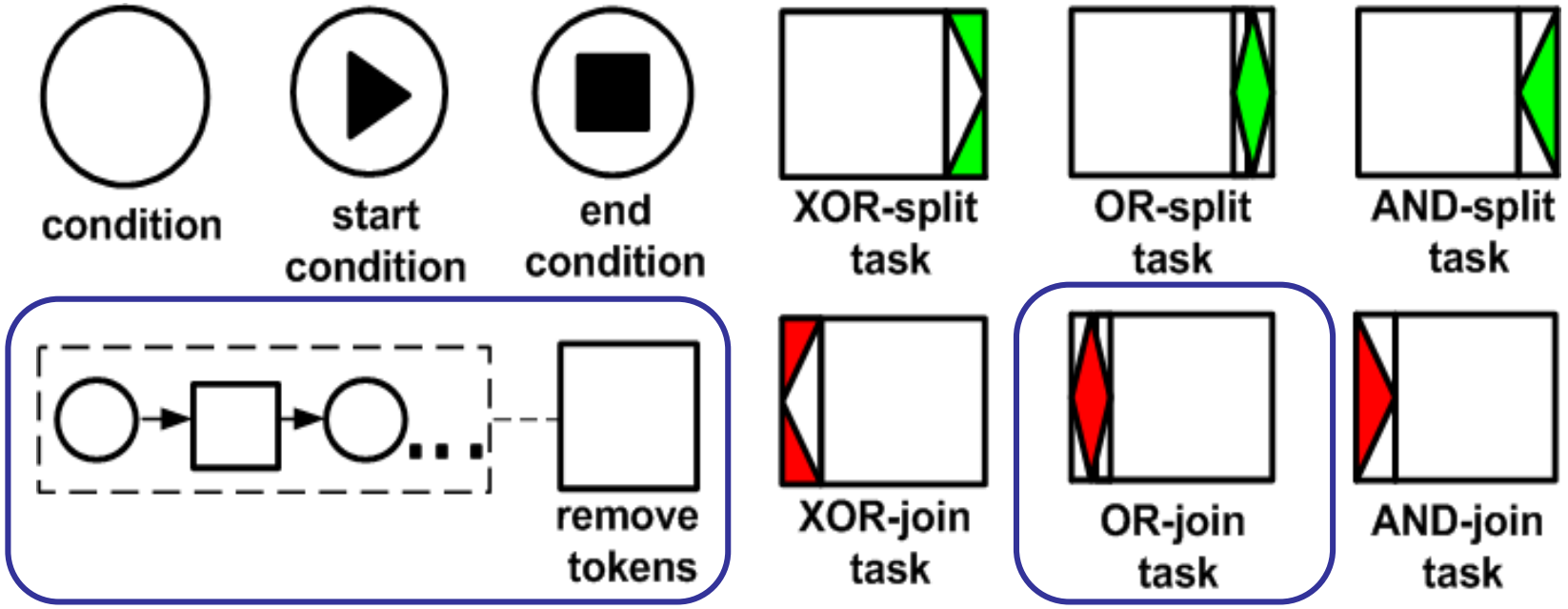
THE YAWL CONTROL FLOW PERSPECTIVE

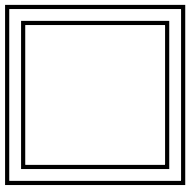
Business 2710 – Class 14

Learning Objectives

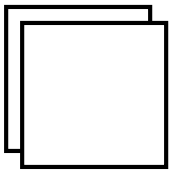
- Be able to identify common workflow patterns and model them using YAWL
- Be able to use the YAWL language to model simple business processes

YAWL Notation





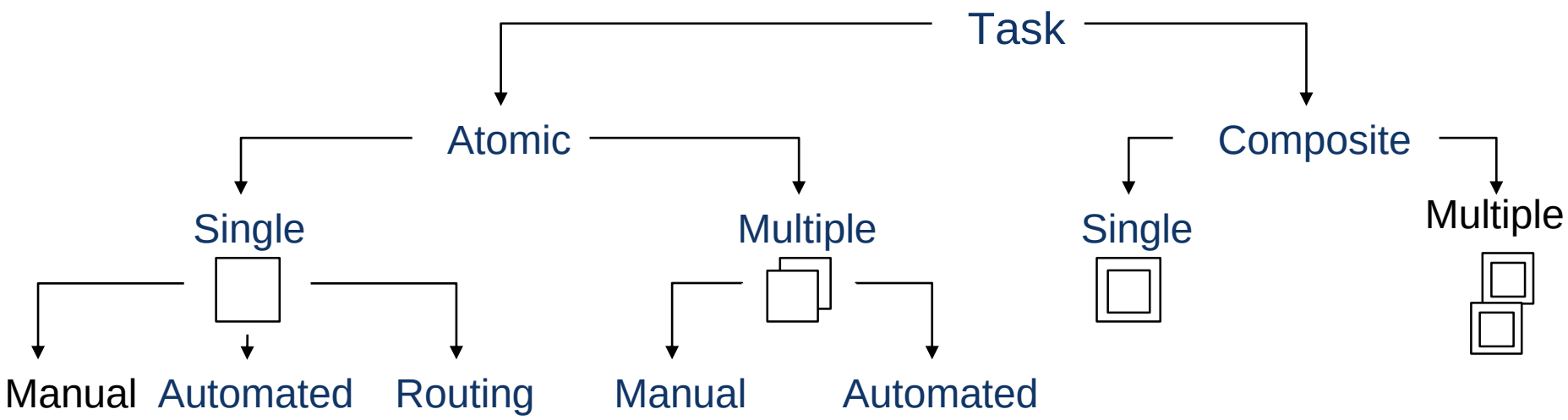
Composite task



Multiple Instance task

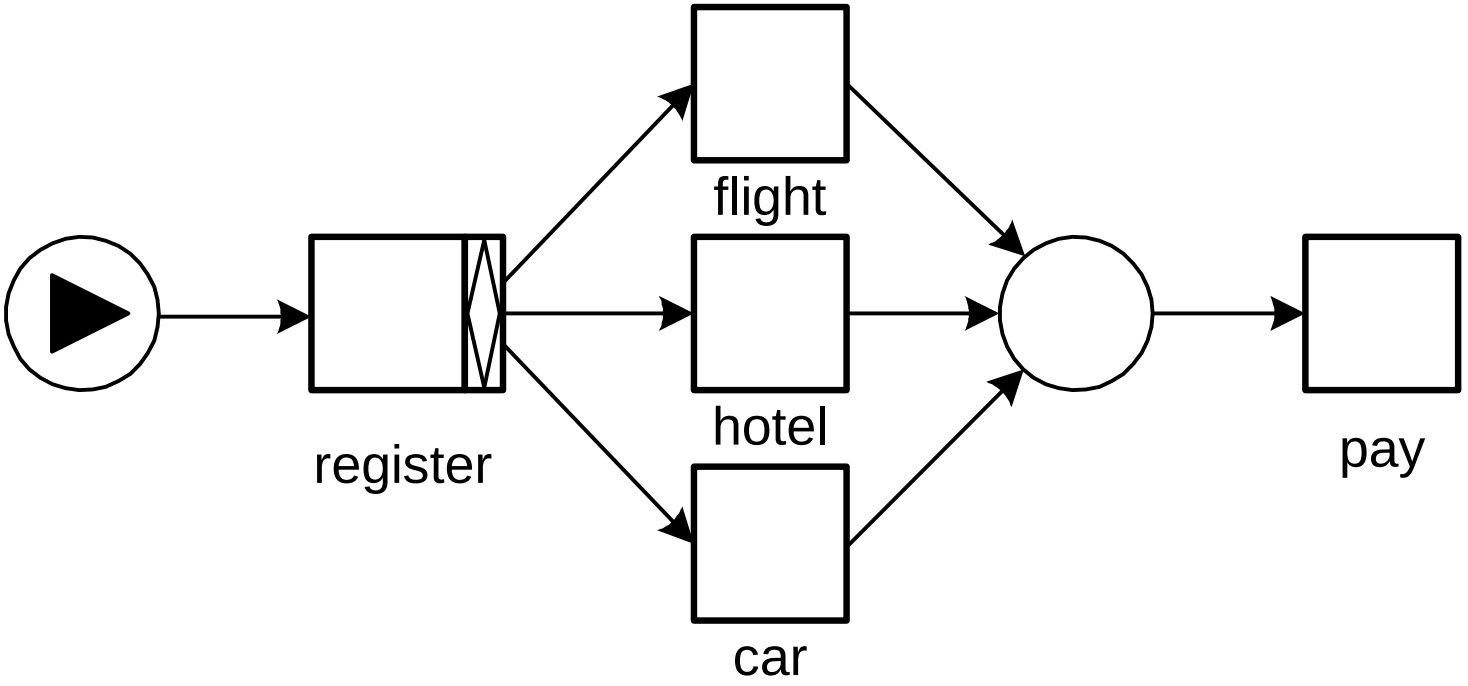
Setting up the Process Control Logic

- Task hierarchy



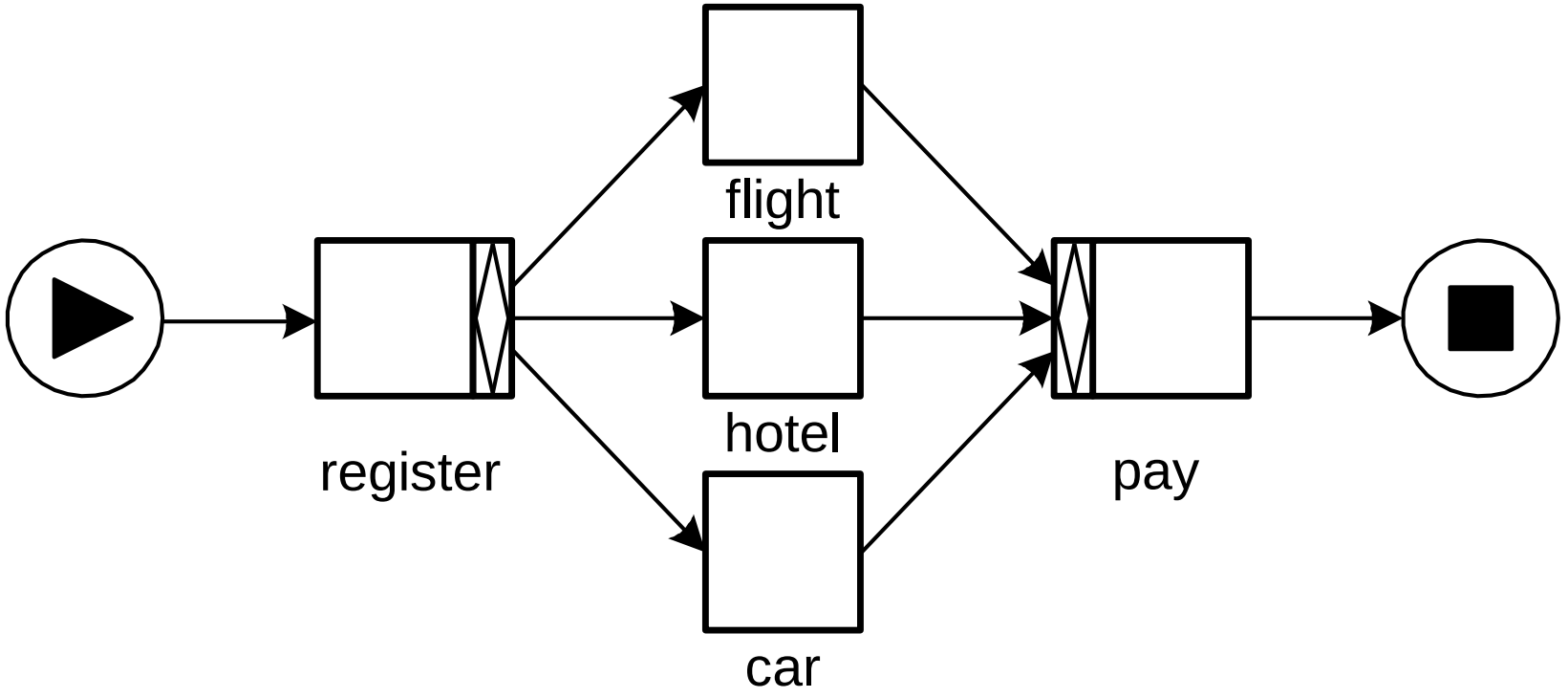
General YAWL Example

Not sound!



General YAWL Example

Sound

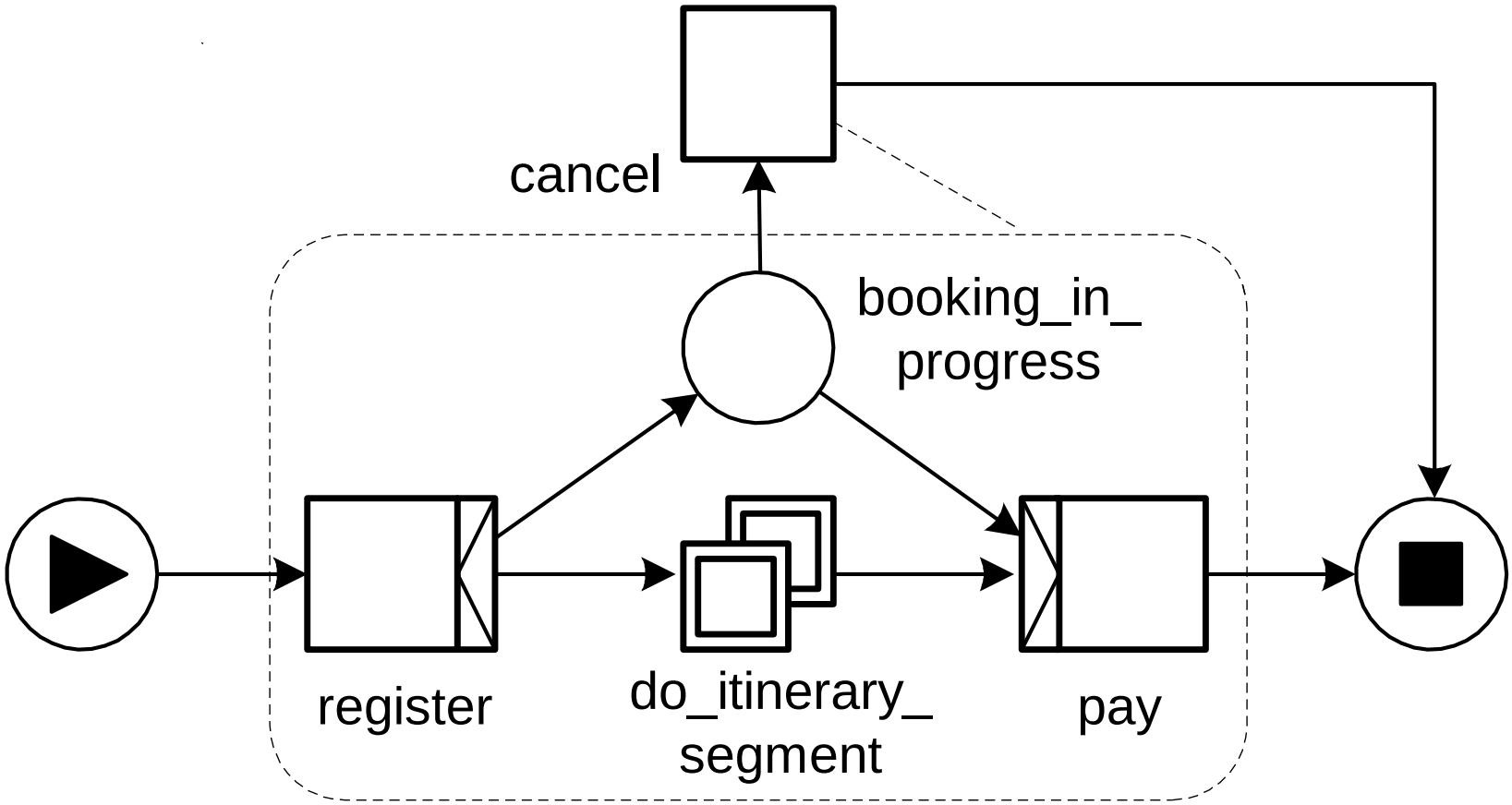


Cancellation in YAWL

- Syntactically, a cancellation region consists of a number of tasks and places that are attached to a so-called **cancellation task.**

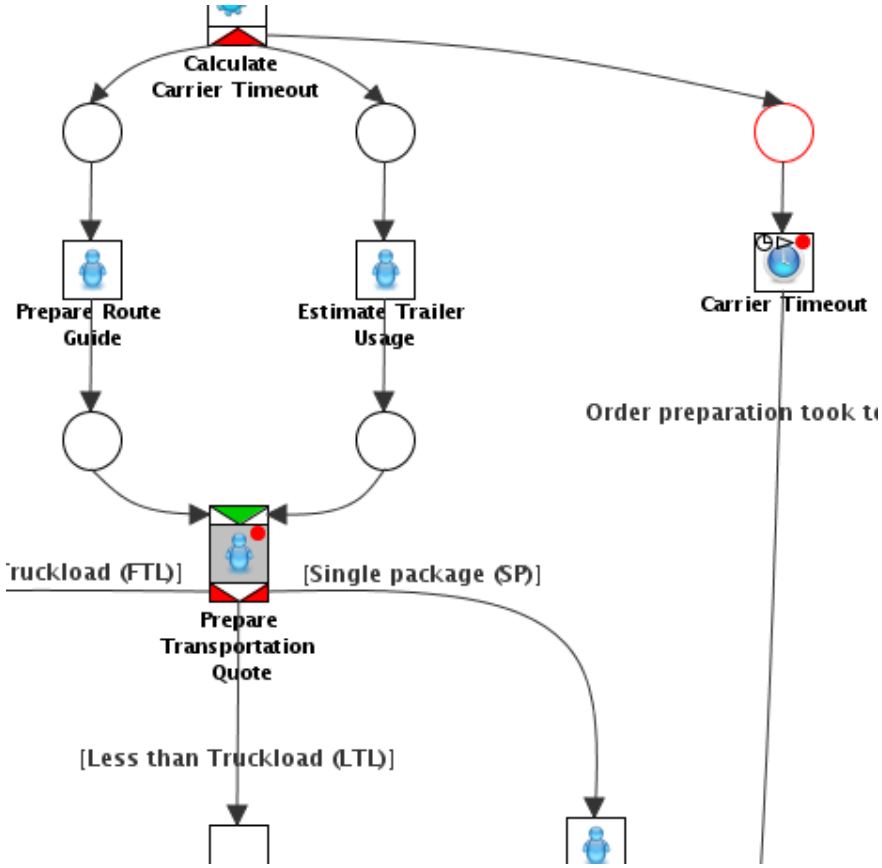
- Upon **completion** of a cancellation task, all tokens in a cancellation region (or in decompositions of tasks in that region, etc.) are removed

Cancellation Example



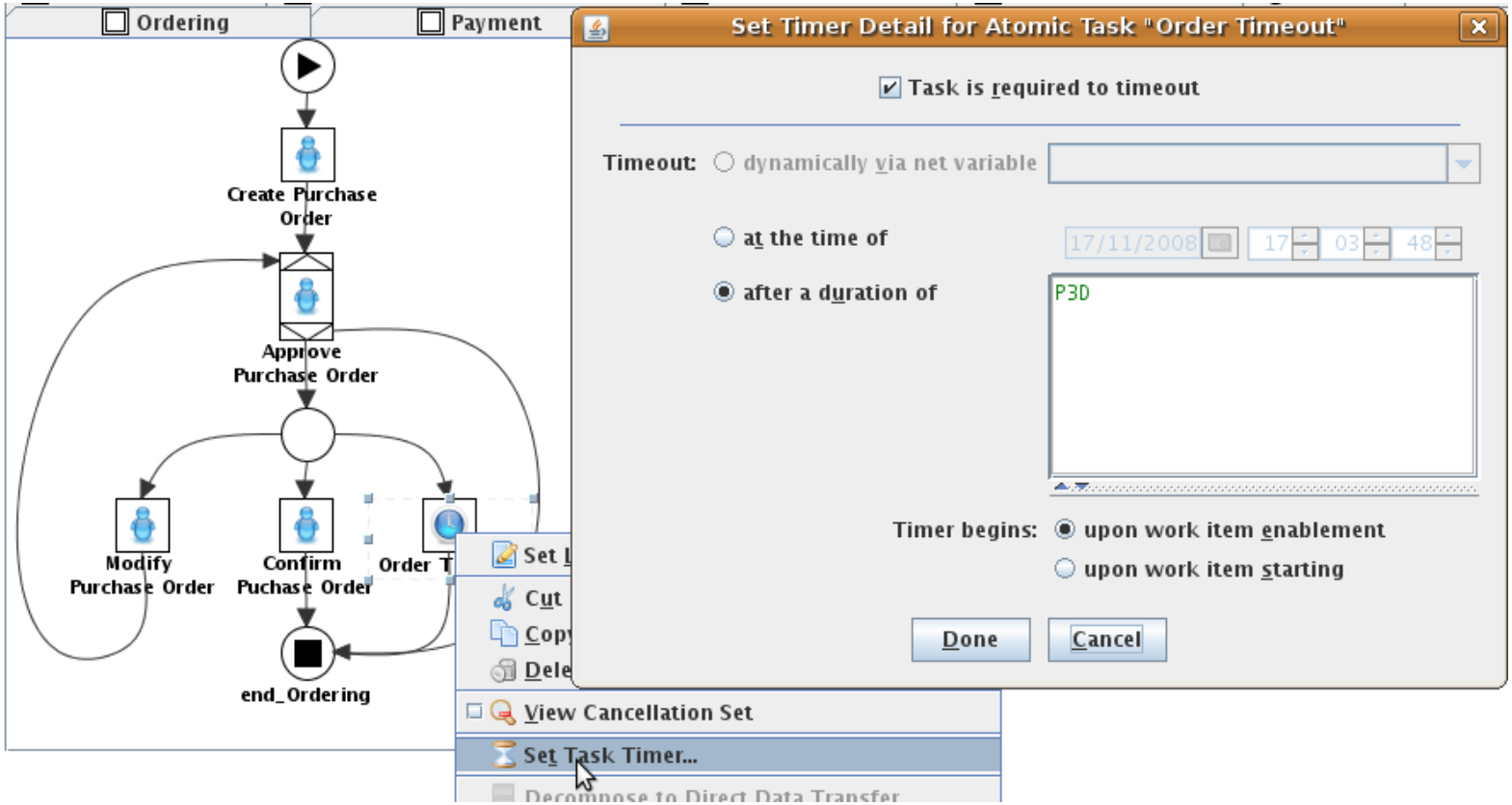
Cancellation Set

- Activation of a cancellation set
  - Tasks that are associated with a cancellation set are marked by a red dot
  - Open context menu of a task
  - Select View, Cancellation set



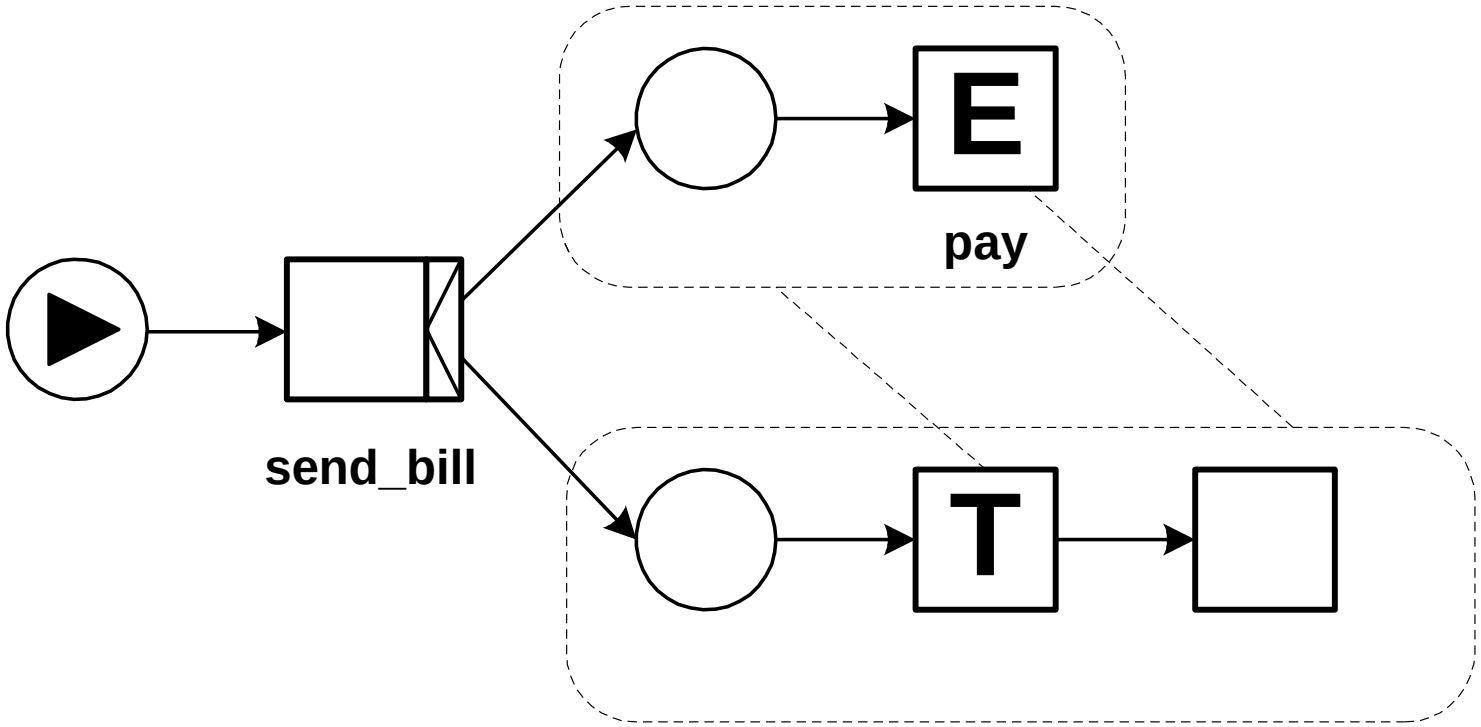
Timer Task

- Automatic single instance task executed within a given timeframe



YAWL Example with Time

T represents a time-out task



Control Flow Patterns

A Selection Only

Classes of control-flow patterns

- **Branching Patterns:** capture branching scenarios in processes.
- **Synchronisation Patterns:** describe synchronization scenarios arising in processes.
- **Repetition Patterns:** describe various ways in which repetition may be specified.
- **Multiple Instances (MI) Patterns:** delineate situations with multiple threads of execution in a workflow which relate to the same activity.

Classes of Control-Flow Patterns

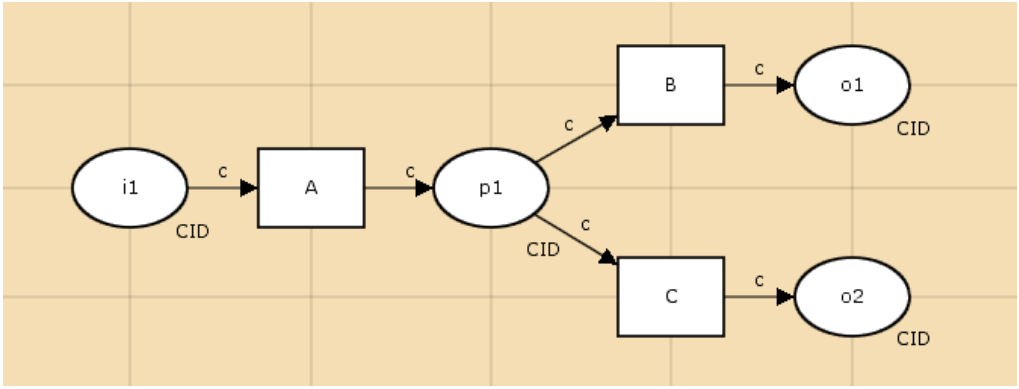
- **Concurrency Patterns**: reflect situations where restrictions are imposed on the extent of concurrent control-flow in a process instance
- **Trigger Patterns:** catalogue the different triggering mechanisms appearing in a process context.
- **Cancellation and Completion Patterns:** categorise the various cancellation scenarios that may be relevant for a workflow specification.
- **Termination Patterns:** address the issue of when the execution of a workflow is considered to be finished.

Branching Constructs

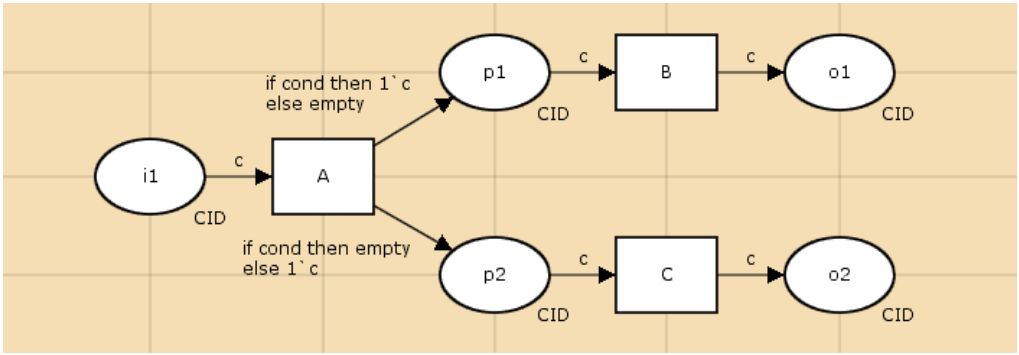
- AND Split
  - **Parallel split**, initiation of parallel threads
- OR Split
  - **Multi-choice**, thread of control is passed to one or more outgoing branches
- XOR Split
  - **Exclusive choice**, thread of control is passed to exactly one of the outgoing branches
  - **Deferred choice**, thread of control is passed to exactly one of the outgoing branches. Selection decision is deferred to the user and/or operating environment
- Thread Split
  - **Thread split**, thread of control is split into multiple concurrent threads in the same branch

Deferred choice vs Exclusive choice

- Deferred choice
  - **Context condition:** only one instance of the construct can operate at any time in a case



- Exclusive choice
  - **Context condition:** choice construct has access to all required resources when making routing decisions



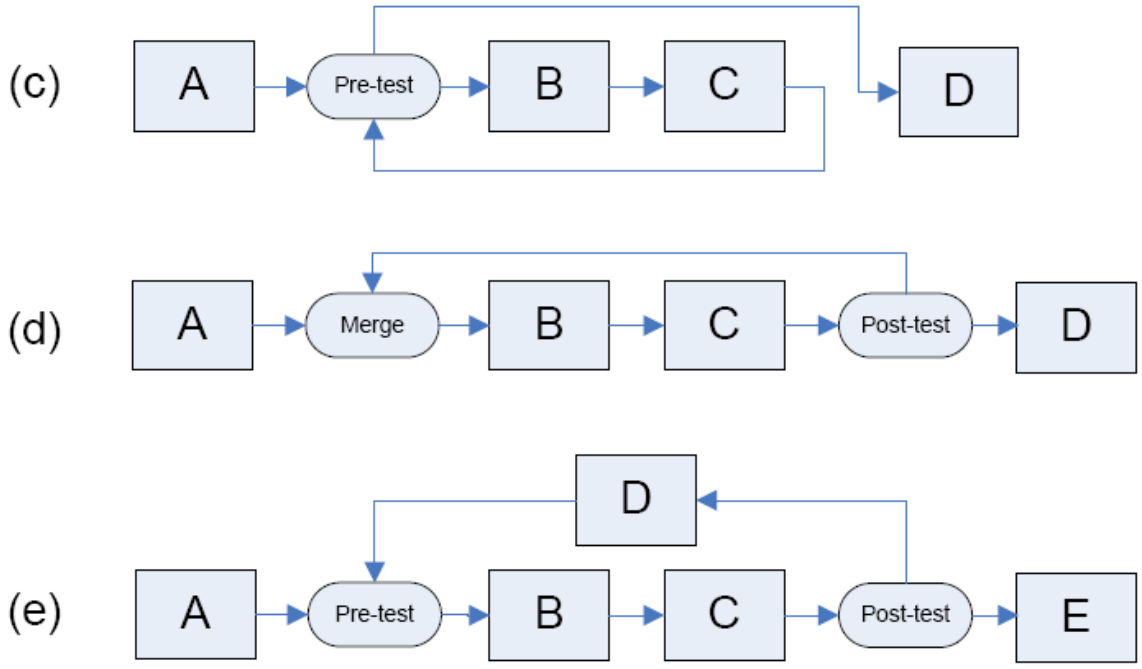
A (very) complex CF pattern: General synchronizing merge

- Basically: *Wait only if you have to*
- Problems:
  - How should you treat other OR-joins?
  - How do you deal with cycles?
  - How do you treat decomposed tasks?
  - Is an analysis of the future possible? How complex will it be?

(for a detailed discussion see Chapter 3)

Repetition patterns -1

Structured loop, the ability to execute a task or sub-process repeatedly on the basis of a pre or post test where there is a single entry/exit point

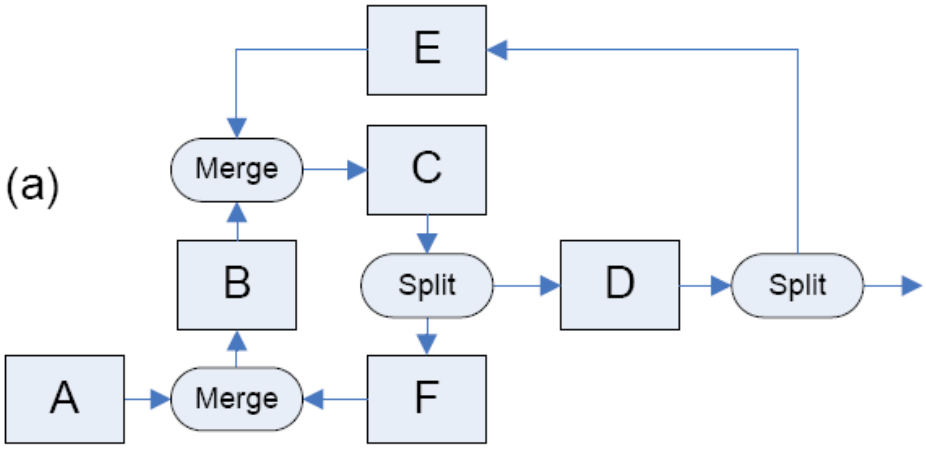


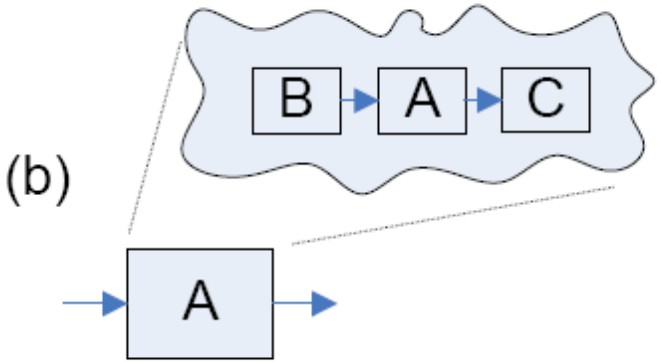
Repetition Patterns -2

Describe various ways in which repetition of tasks or sub-processes may be specified

Arbitrary cycles, cycles in a process with more than one entry or exit point

Recursion, repetition of a task through self-invocation





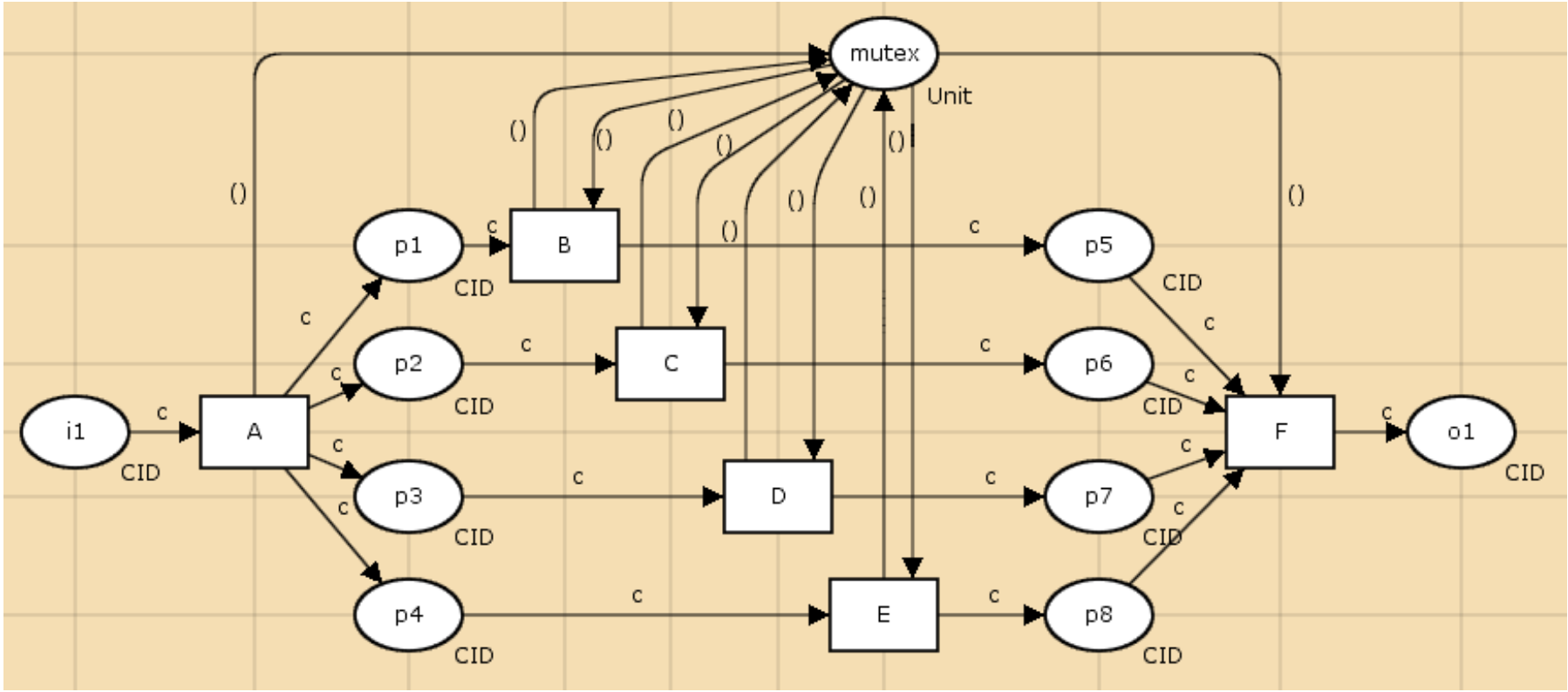
Concurrency Patterns

Reflect situations where restrictions are imposed on the extent of concurrent control-flow in a process instance

- **Sequence**, sequential task execution
- **Interleaved routing**, execution of a set of tasks can occur in any order but not concurrently
- **Interleaved parallel routing**, a set of tasks, which have a partial ordering amongst them, and execute in order order that satisfies this sequence but not concurrently
- **Critical section**, two or more regions in a process model that cannot be active simultaneously
- **Milestone**, the execution of a nominated task can only proceed when the process instance is in a specified state

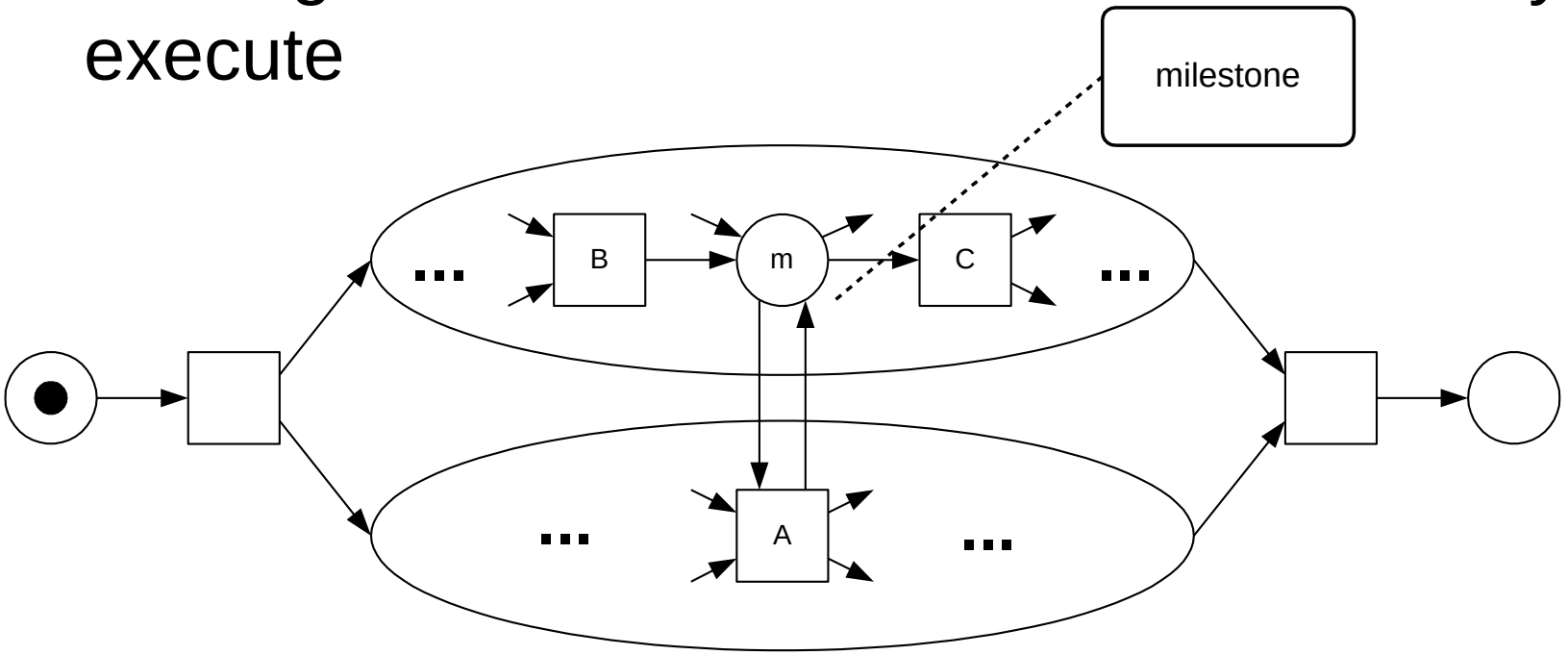
Interleaved Routing: Definition

Context condition: tasks initiated and completed sequentially and cannot be



Milestone

- In some cases a thread needs to check whether some other parallel thread has reached a certain point (but has not moved beyond it)
- As long as this is the case a certain task may execute



Cancellation and Completion Patterns

Categorize the various cancellation scenarios that may be relevant for a workflow specification

- **Cancel task**, withdraw a specified task instance
- **Cancel case**, withdraw all task instances in a case
- **Cancel region**, withdraw task instances in a specified region of a process
- **Cancel MI task**, withdraw all instances of a specified MI task
- **Complete MI task**, withdraw all remaining instances of a specified MI task, completed instances are unaffected and the MI task is deemed to have sucessfully completed