

YAWL

YAWL

BPM BP PAIS BPMS WFMS SOA BPMN
WS- BPEL WSDL SOAP XML HTTP
UDDI JEE WCP CPN ...

Yet Another Workflow Language & Co.

Overview



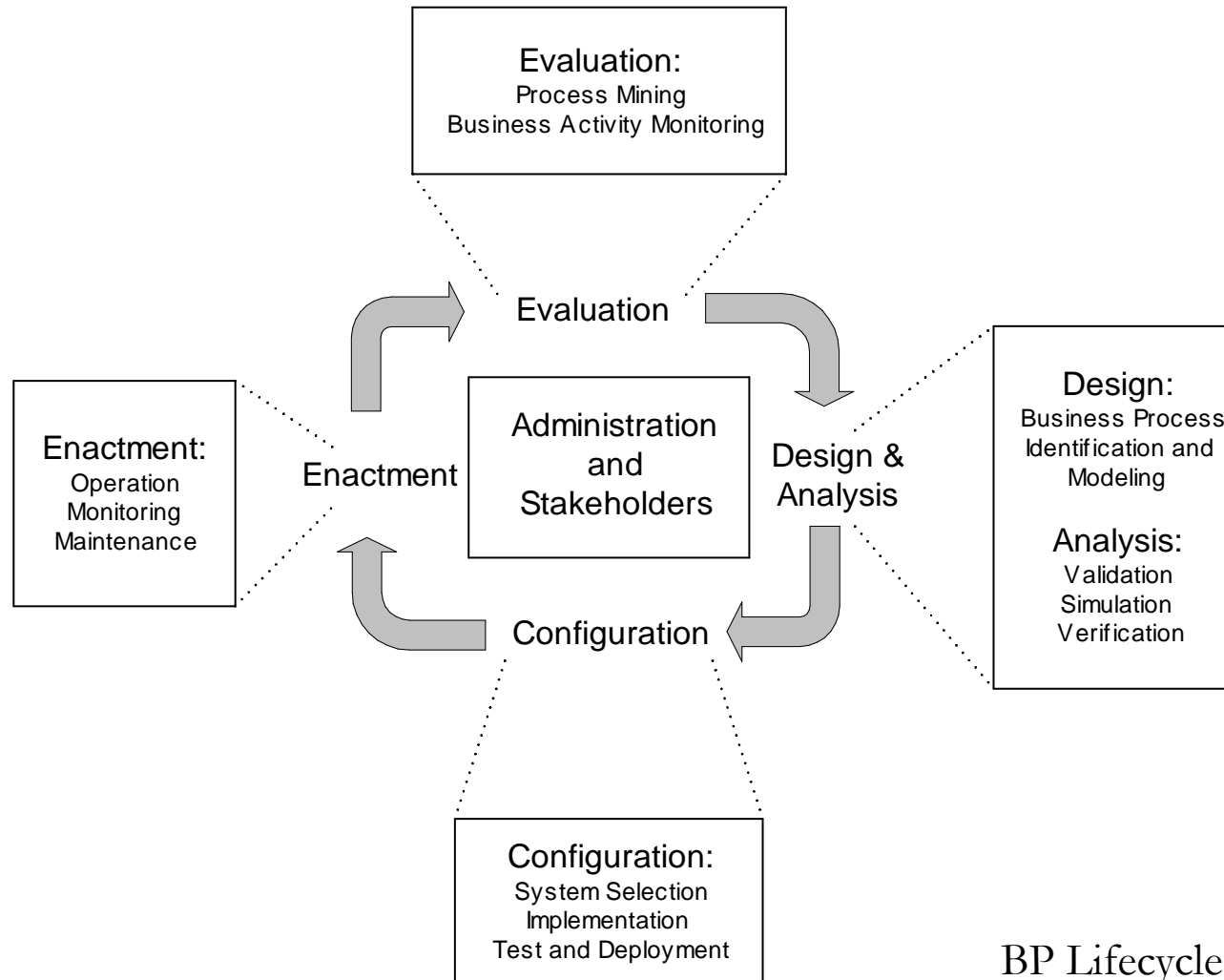
- Business Process Management (BPM)
- Process-Aware Information System (PAIS)
- Service Oriented Architecture (SOA)
- YAWL System Architecture
 - ▣ Workflow Management System (WFMS)
- Yet Another Workflow Language (YAWL)
- YAWL System demo
 - ▣ YAWL Editor
 - ▣ Il caso Genko Oil

Business Process Management (BPM)



- Le *organizzazioni* hanno bisogno di coordinare diversi *agenti* per raggiungere i propri obiettivi.
- Un insieme di attività fra loro correlate possono costituire un *business process* (BP).
 - output ben definito
 - strutturato, compiti precisi, ripetitivo
 - \Rightarrow bassa incertezza (es. processo \neq progetto)
- BPM \Leftrightarrow Gestione processi aziendali
 - Miglioramento continuo (costi, qualità, ...)
 - Strumenti informatici di supporto

Business Process Lifecycle



BP Lifecycle (Weske2007)

Process-Aware Information Systems (PAIS)



- Sistema informatico guidato da rappresentazioni esplicite di BP con l'obiettivo di coordinare gli agenti nello svolgimento delle loro attività
 - ▣ Business Process Management System (BPMS)
 - ▣ Workflow Management System (WFMS)
- Visual business process modeling languages (BPMLs) with textual executable specifications.
 - ▣ token-based semantics, Petri Nets
 - ▣ OMG BPMN + OASIS WS-BPEL, YAWL
- Engine (interprete + varie ed eventuali)

Service Oriented Architecture (SOA)



- Architettura software per applicazioni distribuite di supporto ai processi aziendali.
- Obiettivi principali:
 - ▣ Modularità
 - ▣ Integrazione sistemi esistenti
 - ▣ Adattabilità alle esigenze organizzative
 - ▣ Componenti riutilizzabili \Rightarrow *service*
 - ▣ Separazione regole aziendali dal codice macchina
 - flessibilità
 - separazione ruoli: programmatori, consulenti, ecc...

Principi SOA

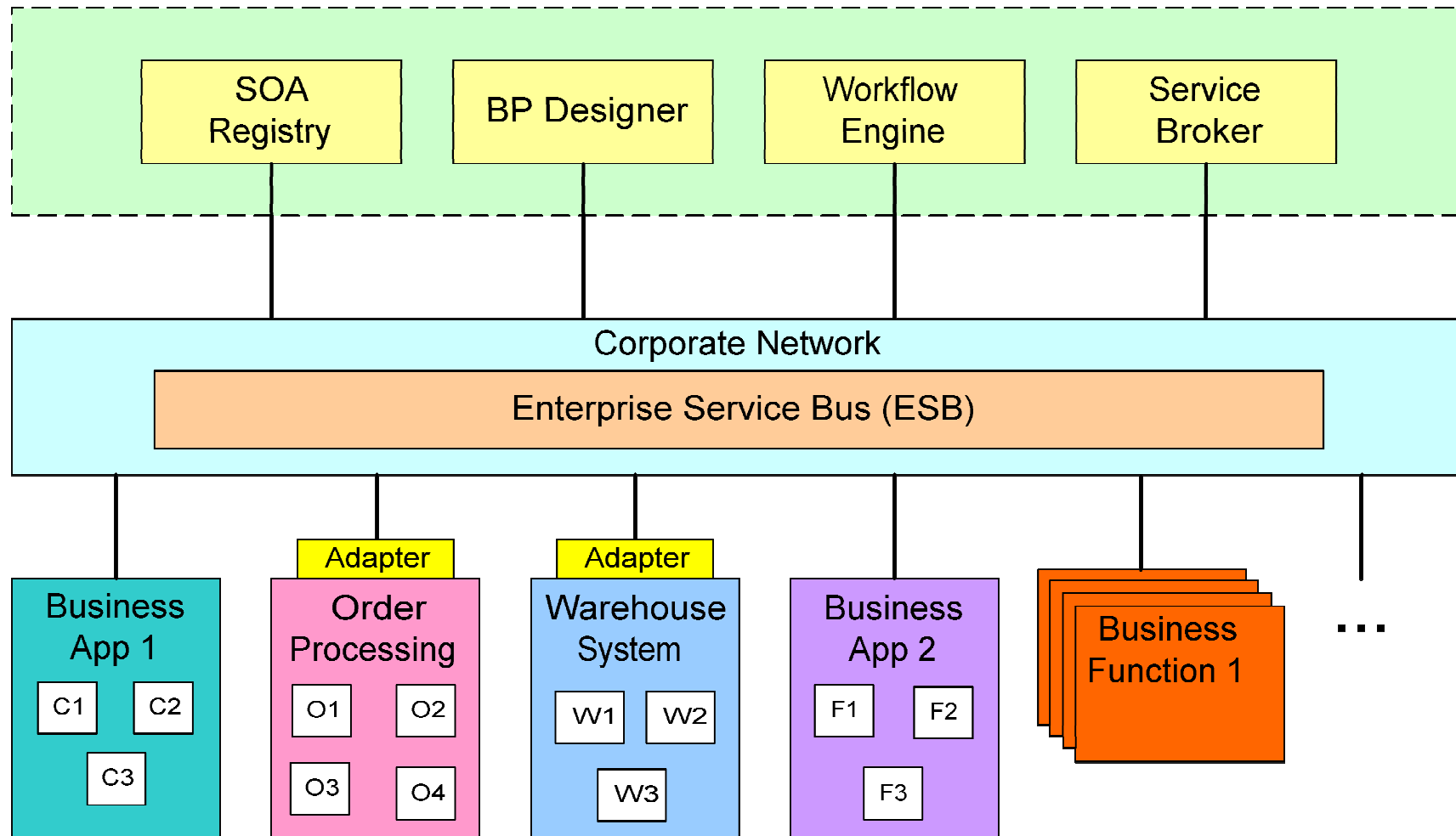


- Incapsulamento sistemi esistenti come servizi
- Servizi debolmente accoppiati con interfacce stabili
- Servizi distribuiti tramite la rete aziendale

Servizio → modulo software con una funzione ben definita che non dipende dallo stato di altri servizi.

Applicazione → utilizzo dei servizi seguendo una sequenza predefinita di attività dettata dalle esigenze aziendali: workflow!

Overview SOA



Web Services

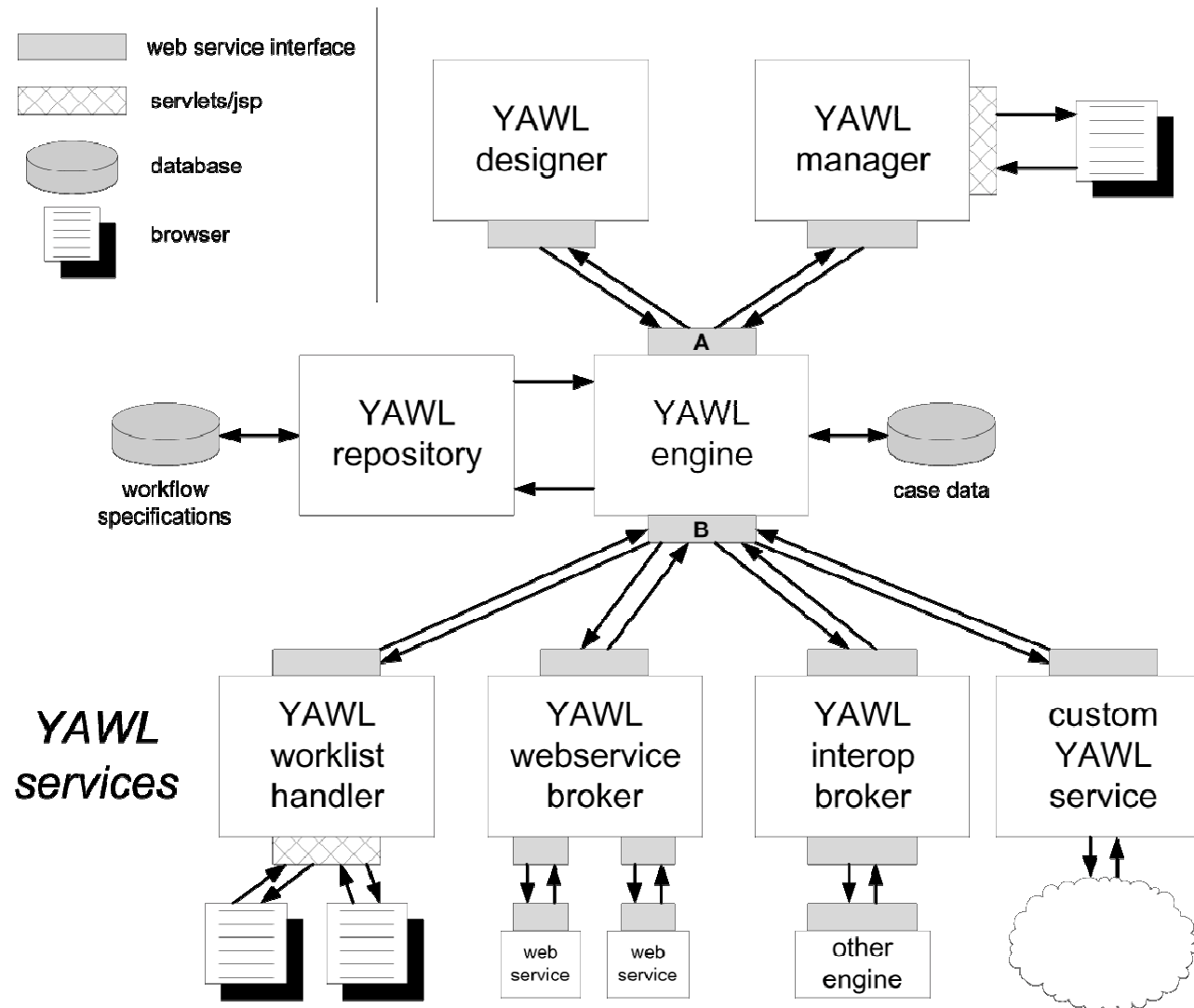
- Un componente è riutilizzabile solo se conforme agli standard accettati dall'intera industria.
- eXtensible Markup Language (XML)
- Web Services Description Language (WSDL)
 - ▣ interfaccia pubblica del componente software
- Simple Object Access Protocol (SOAP)
 - ▣ XML-RPC, HTTP, XML Schema
- Universal Description Discovery & Integration (UDDI)
 - ▣ registro per i servizi
- Business Process Execution Language (WS-BPEL)
- Business Process Modeling Notation (BPMN)

YAWL System

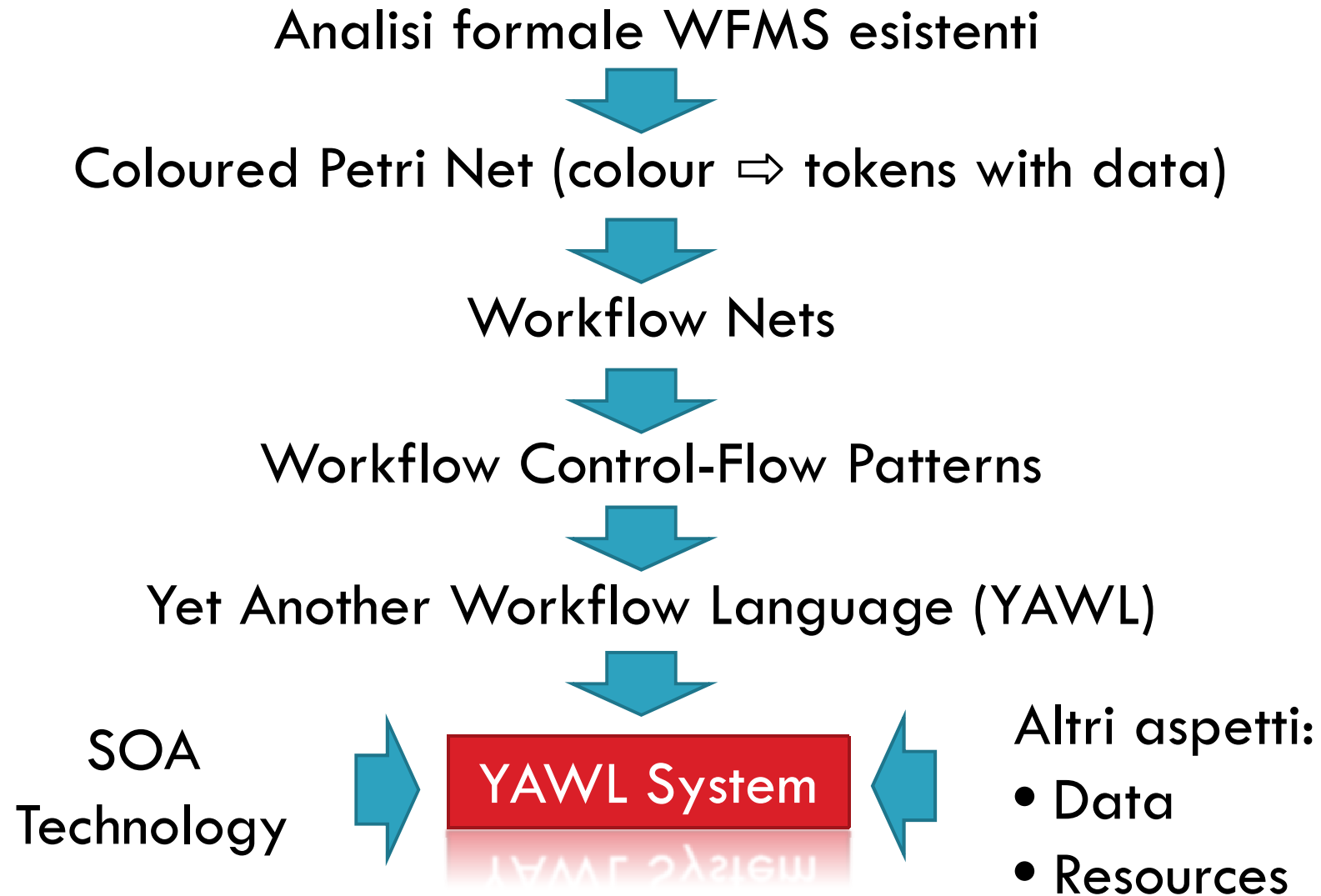


- YAWL System è un WFMS open source che supporta il linguaggio di modellazione YAWL.
- Implementazione basata su SOA, Web Services e tecnologie Java Enterprise Edition (JEE)
- Componenti principali:
 - ▣ YAWL Editor ⇒ Java Rich Client
 - ▣ Engine ⇒ Web Application JEE
 - Apache Tomcat
 - PostgreSQL

YAWL System – Architettura



YAWL System – Storia



Workflow Control-Flow Patterns (WCP) - 1

Basic Control Flow Patterns

1. Sequence
2. Parallel Split
3. Synchronization
4. Exclusive Choice
5. Simple Merge

State-based Patterns

16. Deferred Choice
17. Interleaved Parallel Routing
18. Milestone
39. Critical Section
40. Interleaved Routing

Iteration Patterns

10. Arbitrary Cycles
21. Structured Loop
22. Recursion

Advanced Branching and Synchronization Patterns

6. Multi-Choice
7. Structured Synchronizing Merge
8. Multi-Merge
9. Structured Discriminator
28. Blocking Discriminator
29. Cancelling Discriminator
30. Structured Partial Join
31. Blocking Partial Join
32. Cancelling Partial Join
33. Generalised AND-Join
37. Local Synchronizing Merge
38. General Synchronizing Merge
41. Thread Merge
42. Thread Split

Workflow Control-Flow Patterns (WCP) - 2

Cancellation and Force Completion Patterns

- 19. Cancel Task
- 20. Cancel Case
- 25. Cancel Region
- 26. Cancel Multiple Instance Activity
- 27. Complete Multiple Instance Activity

Termination Patterns

- 11. Implicit Termination
- 43. Explicit Termination

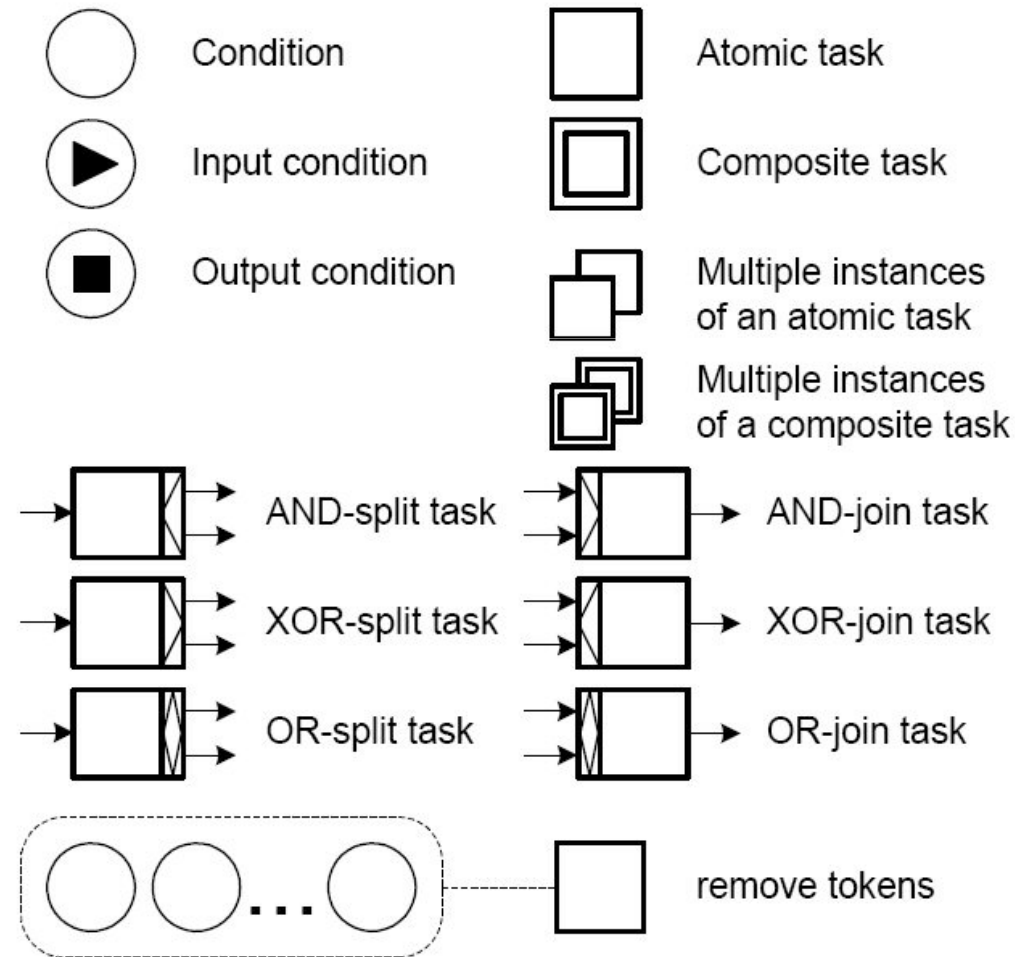
Trigger Patterns

- 23. Transient Trigger
- 24. Persistent Trigger

Multiple Instance Patterns

- 12. Multiple Instances without Synchronization
- 13. Multiple Instances with a Priori Design-Time Knowledge
- 14. Multiple Instances with a Priori Run-Time Knowledge
- 15. Multiple Instances without a Priori Run-Time Knowledge
- 34. Static Partial Join for Multiple Instances
- 35. Cancelling Partial Join for Multiple Instances
- 36. Dynamic Partial Join for Multiple Instances

YAWL – Elementi principali



YAWL – Task atomici e composti

□ Elementi principali del linguaggio

▣ Task – unità di lavoro elementari manuali o automatiche



Atomic task



Composite task

▣ Condizioni o place – attività interne eseguite dall'interprete e usate per modellare diversi cammini di esecuzione



Condition

▣ Ogni definizione di processo ha un'unica condizione di input e un'unica condizione di output



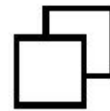
Input condition



Output condition

YAWL – Task con istanze multiple

- Un task (composto o atomico) può avere *istanze multiple*.



Multiple instances
of an atomic task

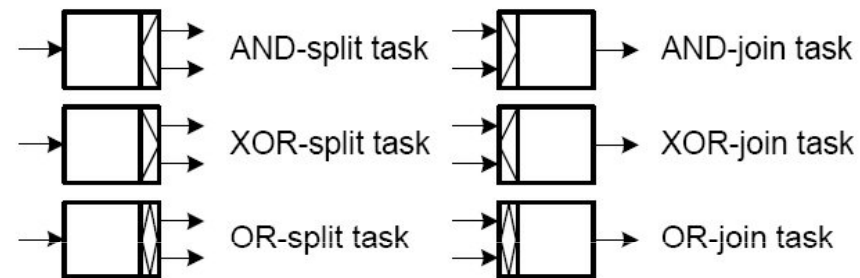


Multiple instances
of a composite task

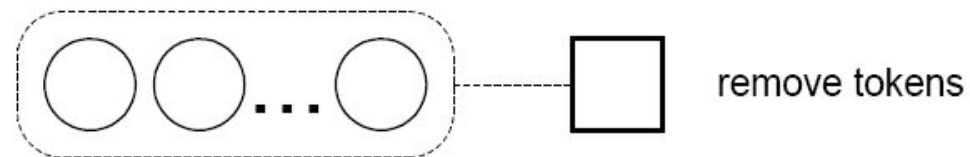
- Si può specificare:
 - ▣ limite inferiore e superiore di istanze create
 - ▣ soglia di istanze completate oltre la quale il task termina
 - ▣ decidere che il numero di istanze sia statico oppure dinamico

Il linguaggio YAWL

- Costrutti per descrivere il flusso di controllo



- Cancellazione di task in esecuzione (cancellation region)

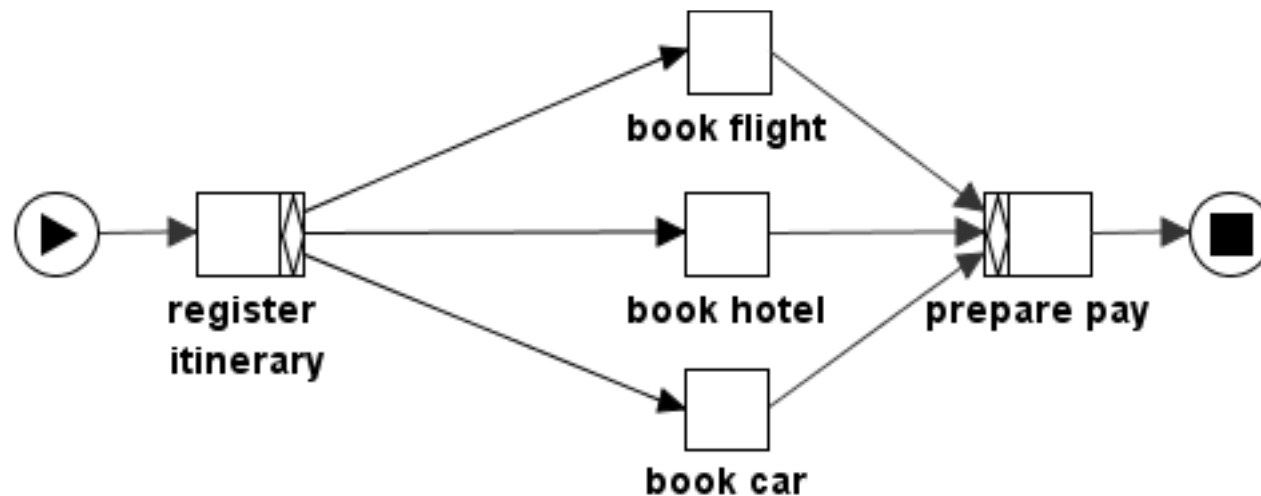


YAWL – Decomposizione Gerarchica

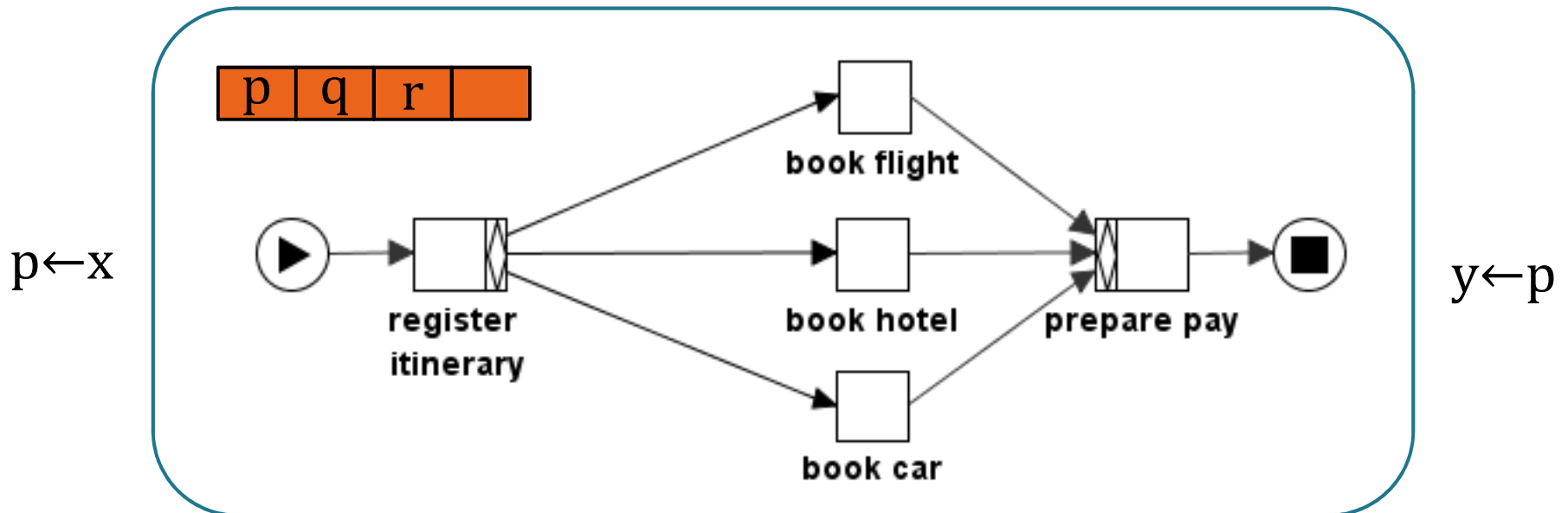
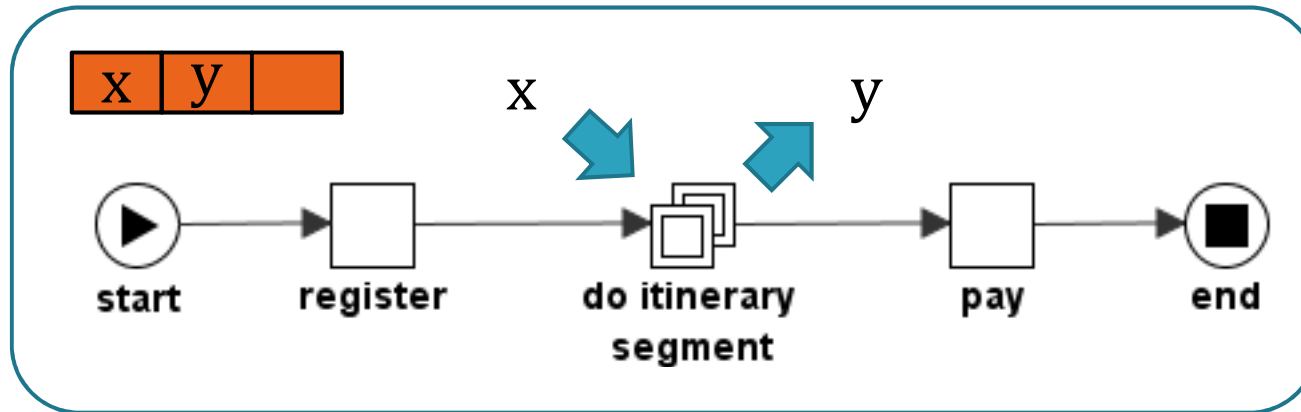


- ❑ Un workflow (process specification) in YAWL è una gerarchia formata da una o più workflow net
- ❑ Struttura ad albero
- ❑ Esiste una workflow net radice
- ❑ Esiste una workflow net per ogni task composto
- ❑ I task atomici sono le foglie dell'albero
- ❑ La decomposizione è di tipo funzionale

Esempio – Workflow 2 Livelli

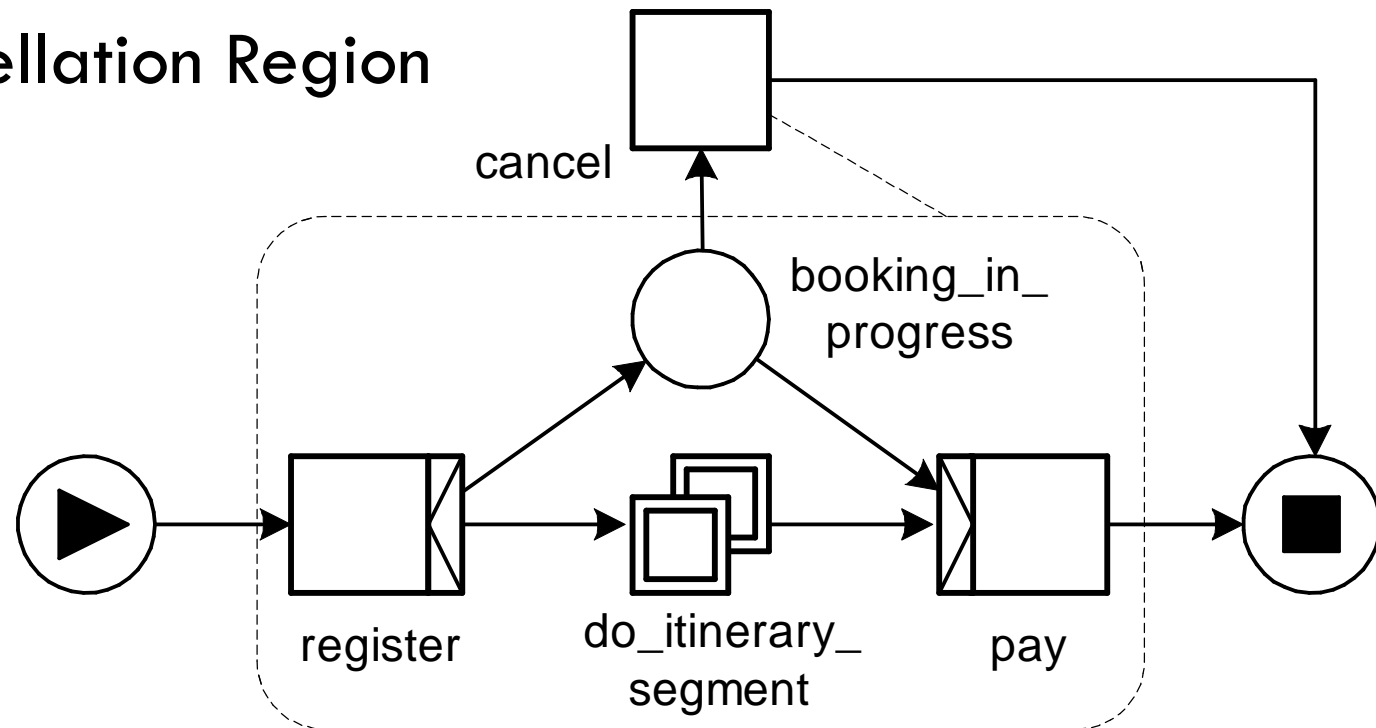


Esempio – Net Variables



Esempio – Pattern Avanzati

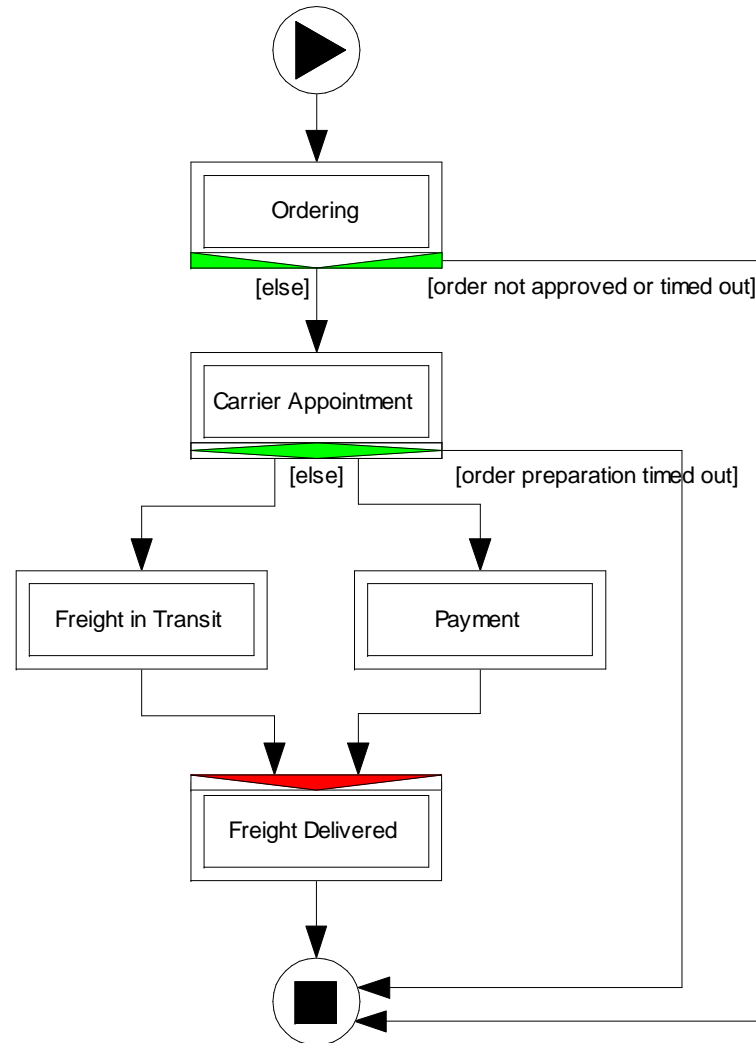
- Use Condition (Place)
- Deferred Choice
- Cancellation Region



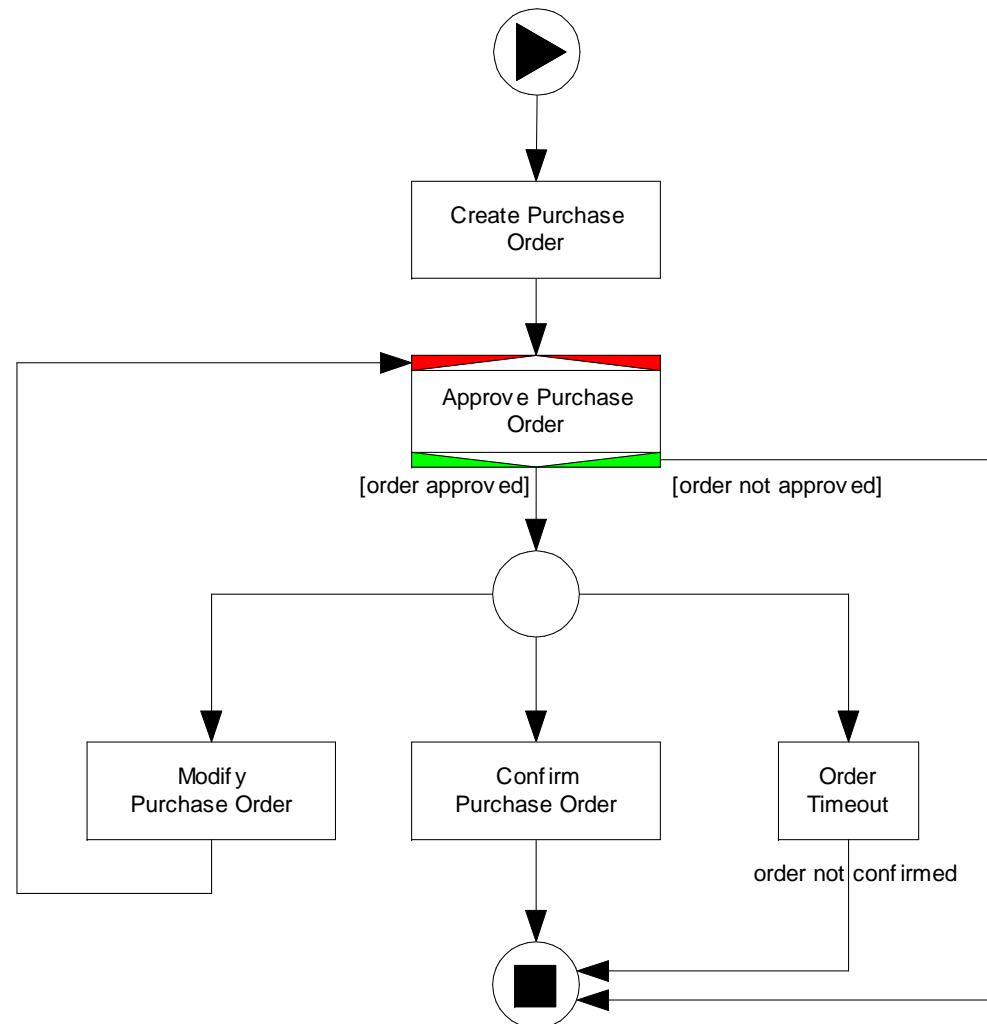
Example - Genko Oil

| Department | Code | Users |
|------------------------------|------|--|
| Chief Executive Officer | CEO | Don Vito Corleone |
| Order Management Department | OD | Carmine Marino, Fredo Corleone, Sal Tessio, Tom Hagen, Vincent "Vinnie" Mancini Corleone, ... |
| Supply Department, Warehouse | SD | Billy Van Arsdale, Carlo Rizzi, Jaggy Jovino, Joe Lucadello, Jack Woltz, Tom Hagen, Connie Corleone Rizzi, ... |
| Carrier Department | CD | Carmine Cuneo, Johnny Fontaine, Kay Adams, Mama Corleone, Momo Barone, Marcello La Rosa, Peter Clemenza, Stefano Clemenza, ... |
| Finance Department | FD | Arturo de Ofstede, Captain McCluskey, Emilio Barzini, Virgil "The Turk" Sollozzo, ... |

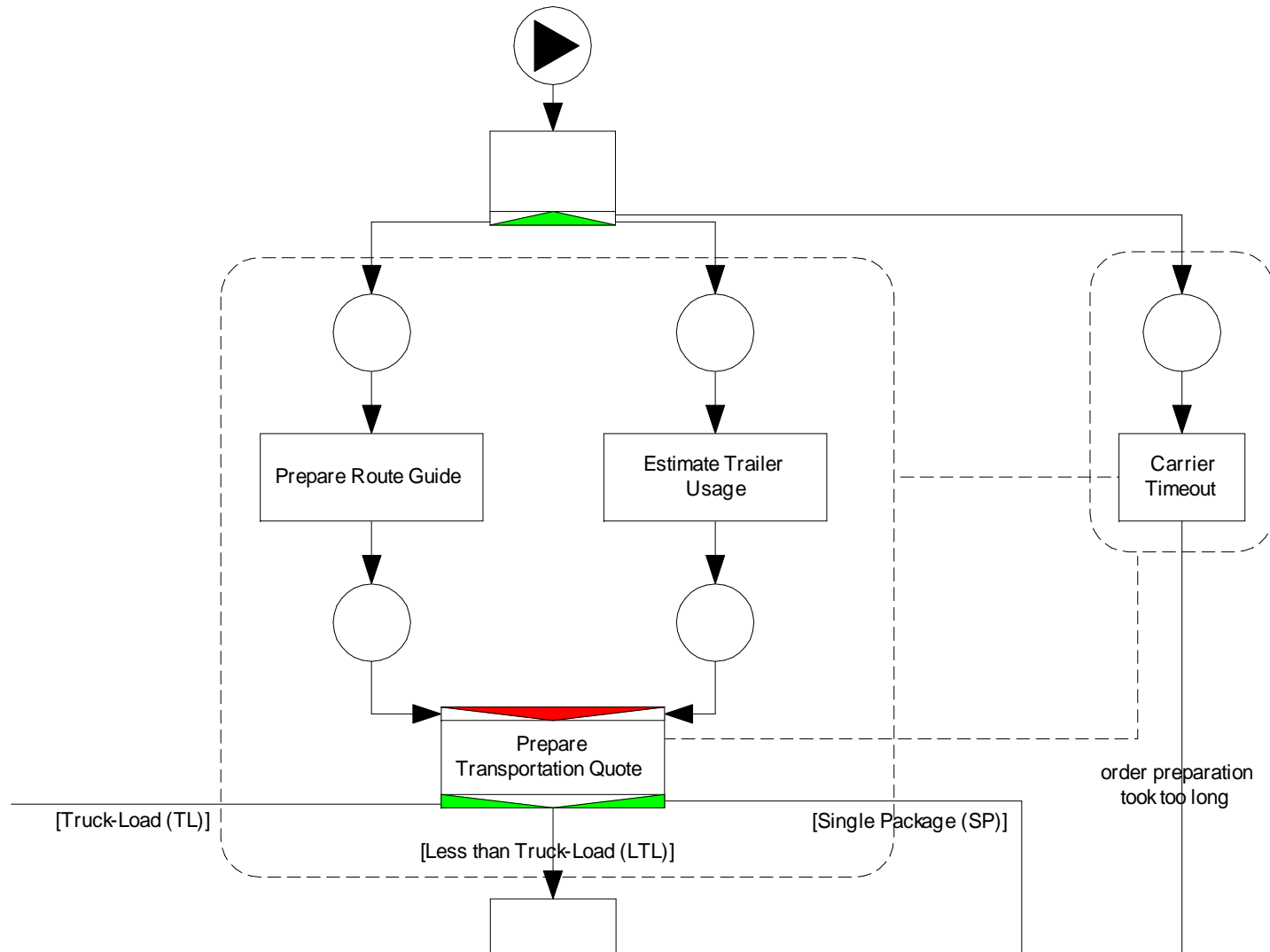
Example – Overall (Main Process)



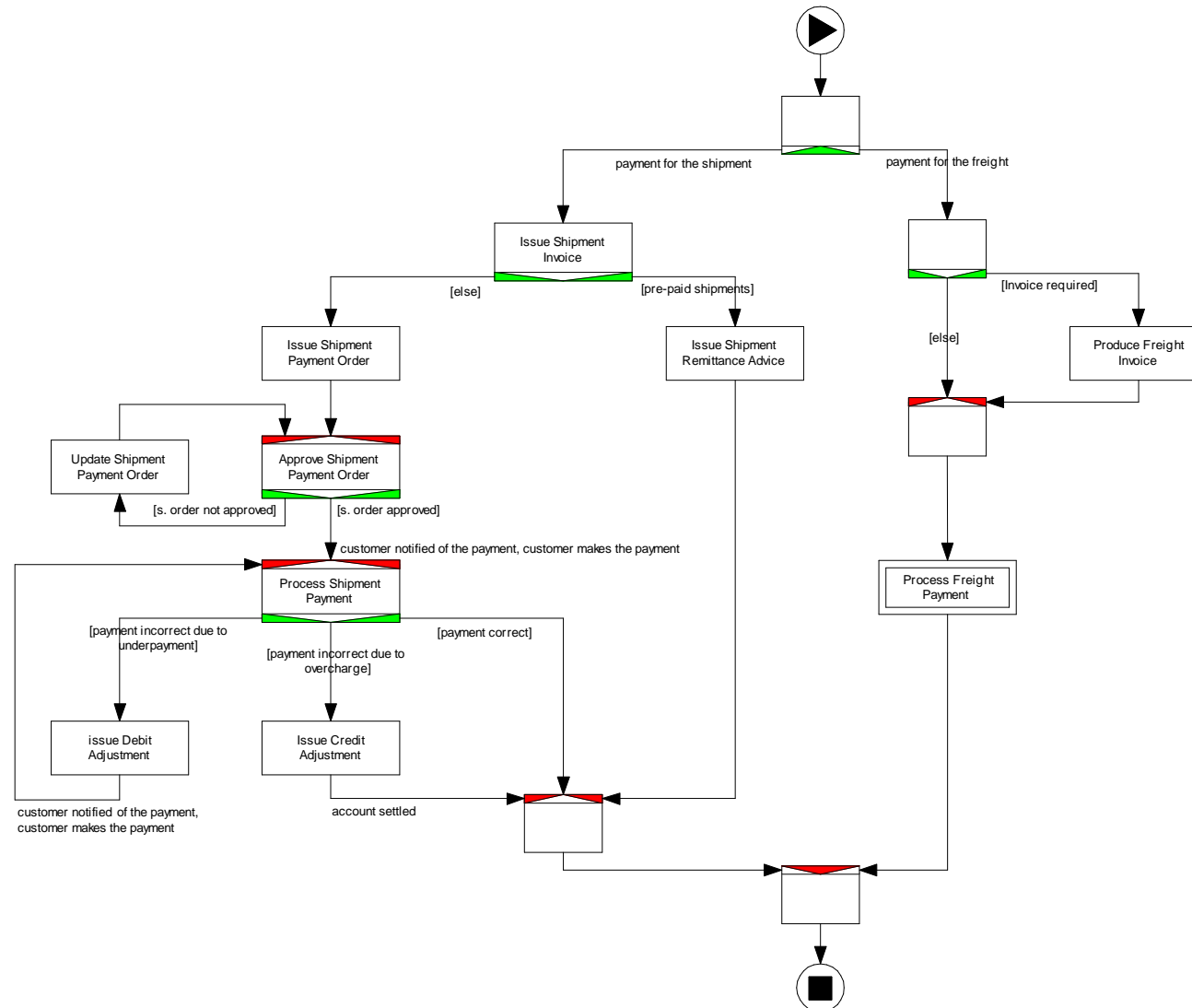
Example – Ordering



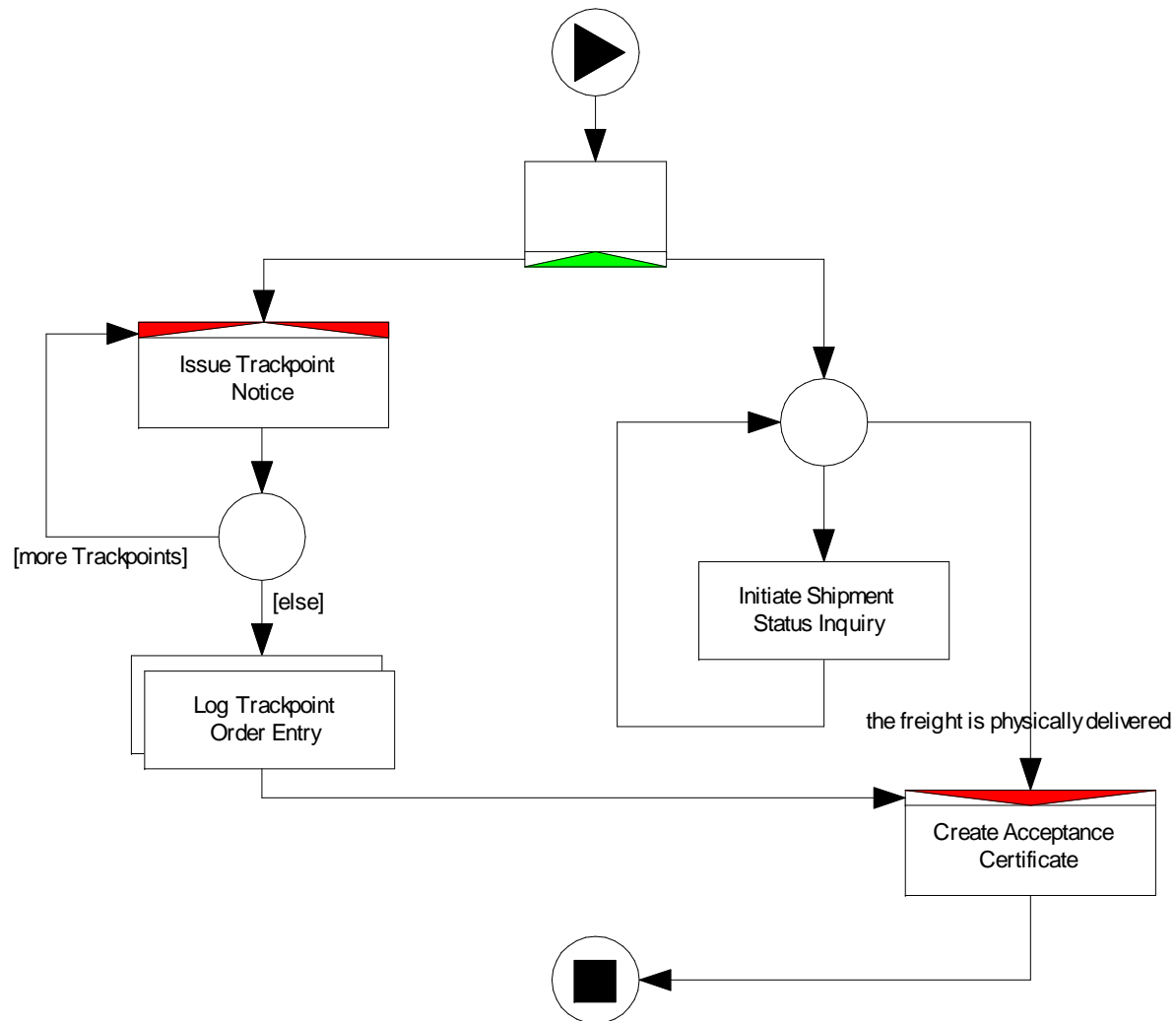
Example – Carrier_Appointment



Example – Payment



Example – Freight_in_Transit



Example – Freight_Delivered

