Dirk **Fahland**

about.me/dirk.fahland

Artifacts
Processes with Multiple Instances

A Distributed Process
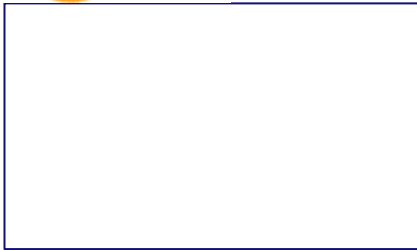

- activities to reach a specific goal, e.g. deliver books
- involves several participants
- any kind of process (not tied to business)
  goal = termination in a specific state

A Distributed Process

- activities to reach a specific goal, e.g. deliver books
- involves several participants
- any kind of process (not tied to business)
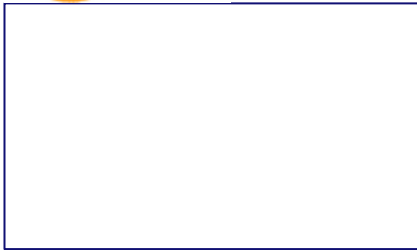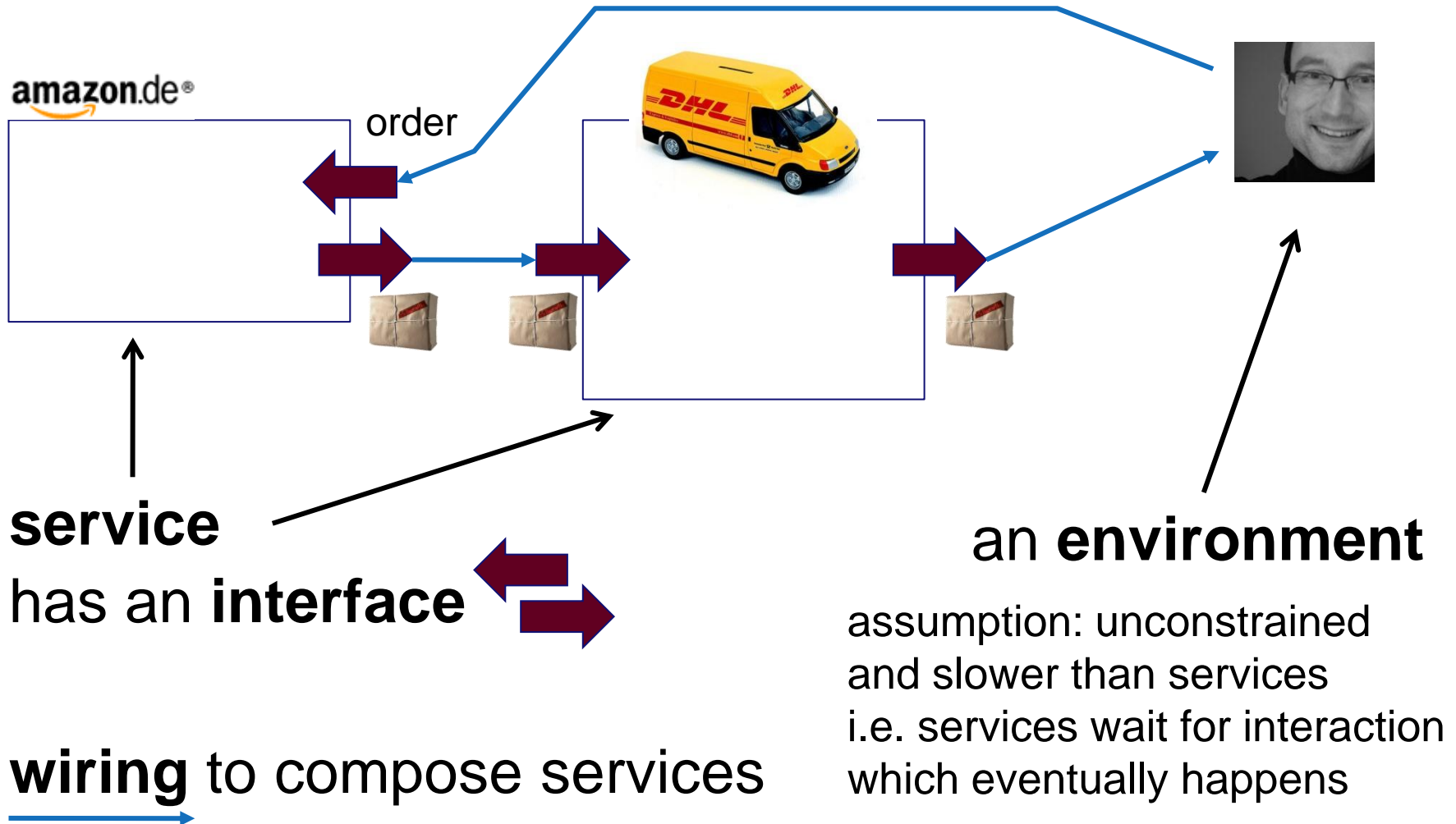 goal = termination in a specific state

A Distributed Process

- activities to reach a specific goal, e.g. deliver books
- involves several participants
- any kind of process (not tied to business)
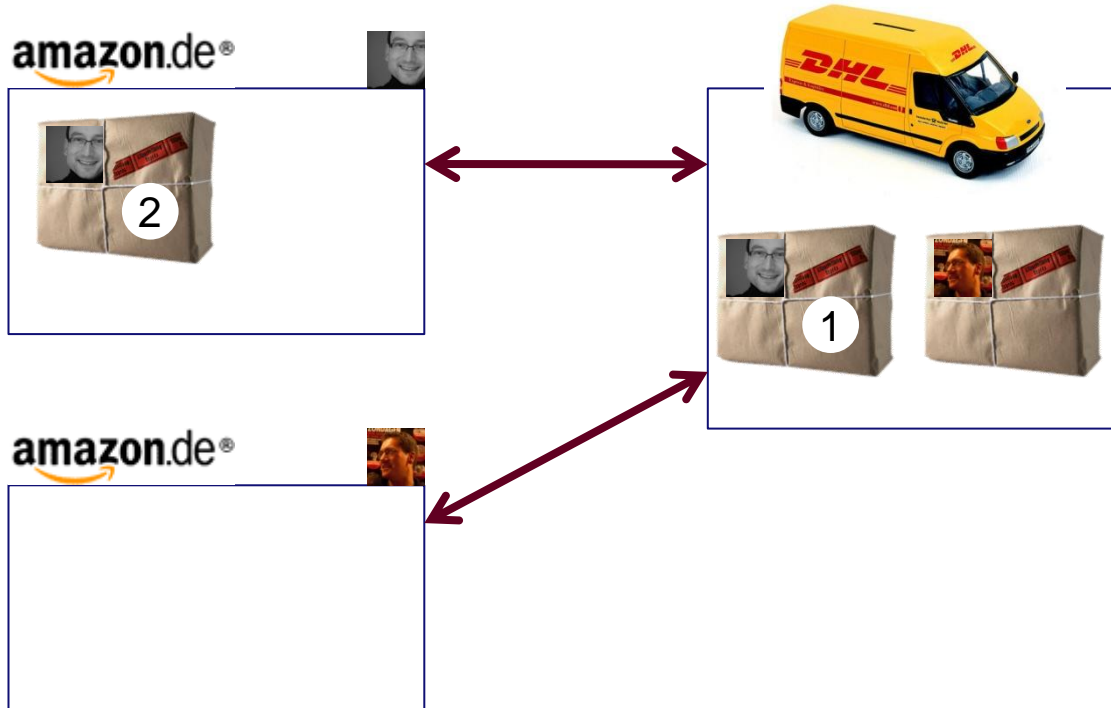 goal = termination in a specific state

Services: Process Modularity

order

service has an **interface**

wiring to compose services

an **environment**

assumption: unconstrained and slower than services i.e. services wait for interaction which eventually happens

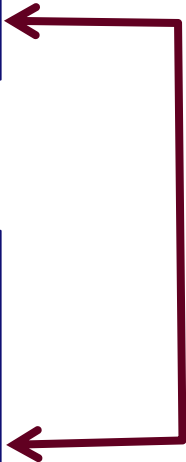# A process that needs multiple instances

A process that needs multiple instances
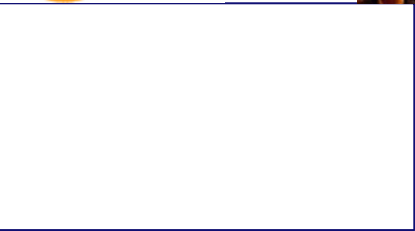

complex relations
between service
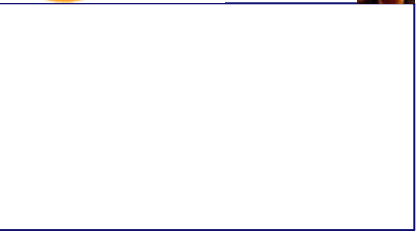**instances**

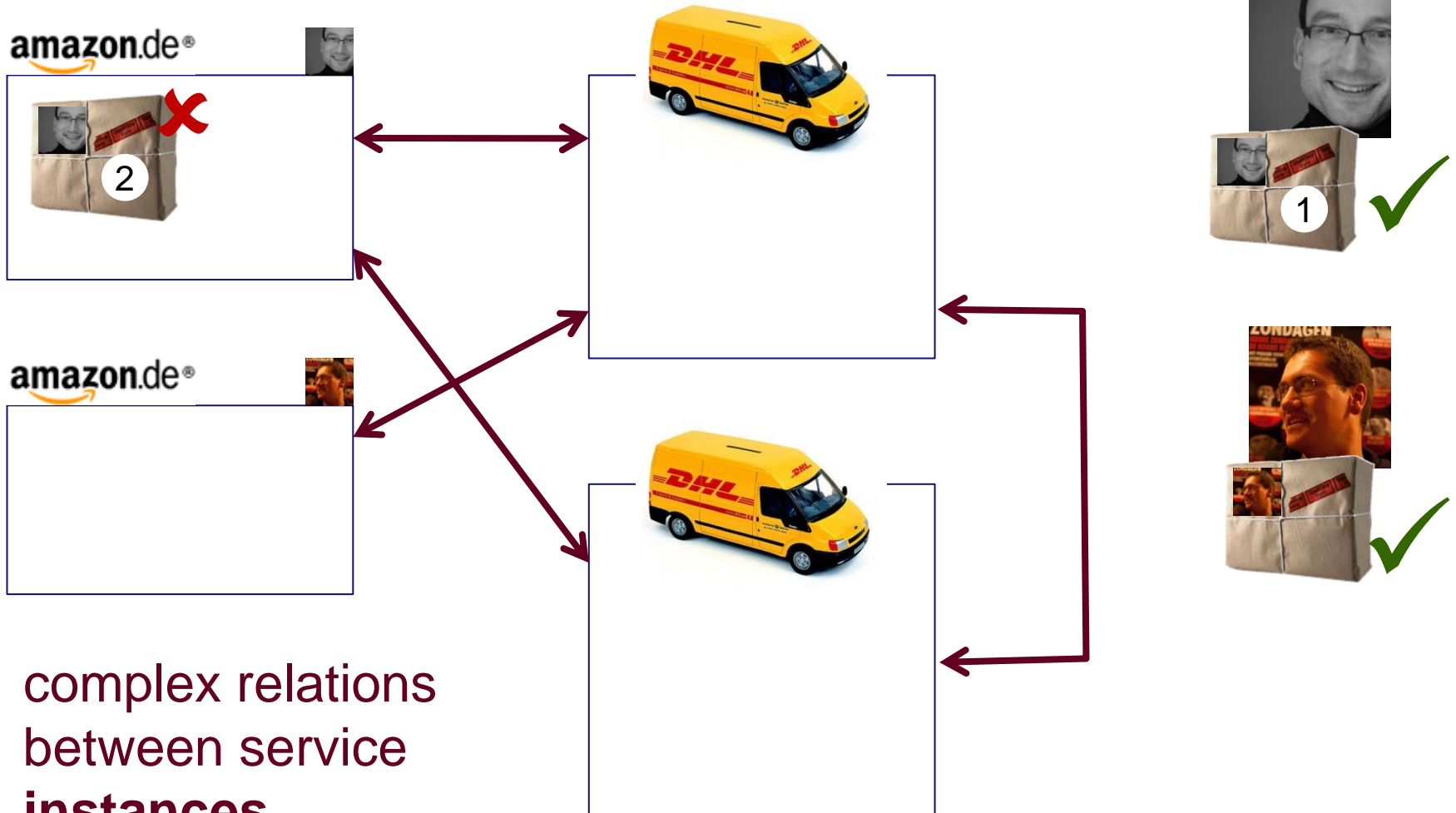# A process that needs multiple instances

A process that needs multiple instances

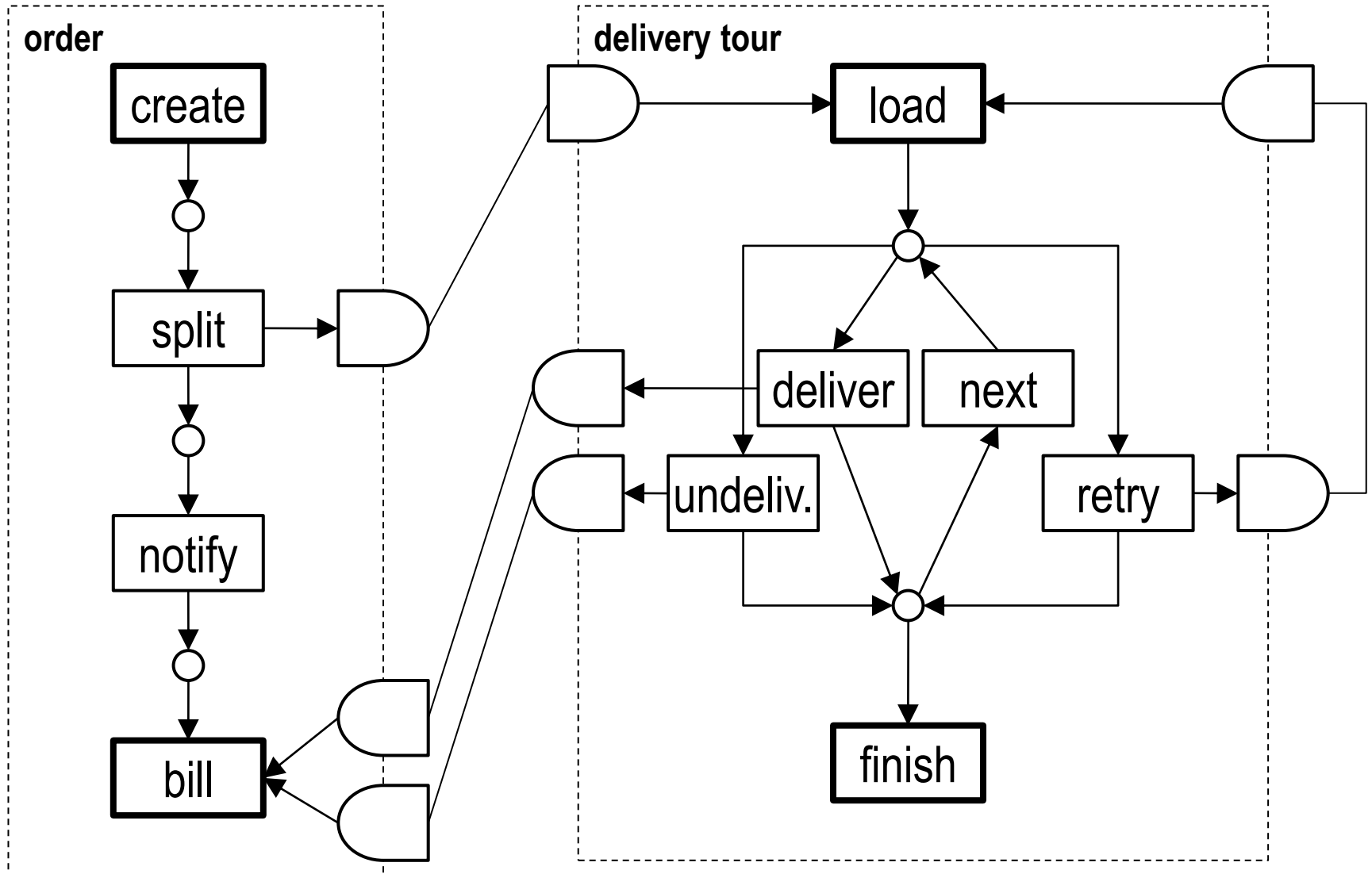# A process that needs multiple instances

complex relations
between service
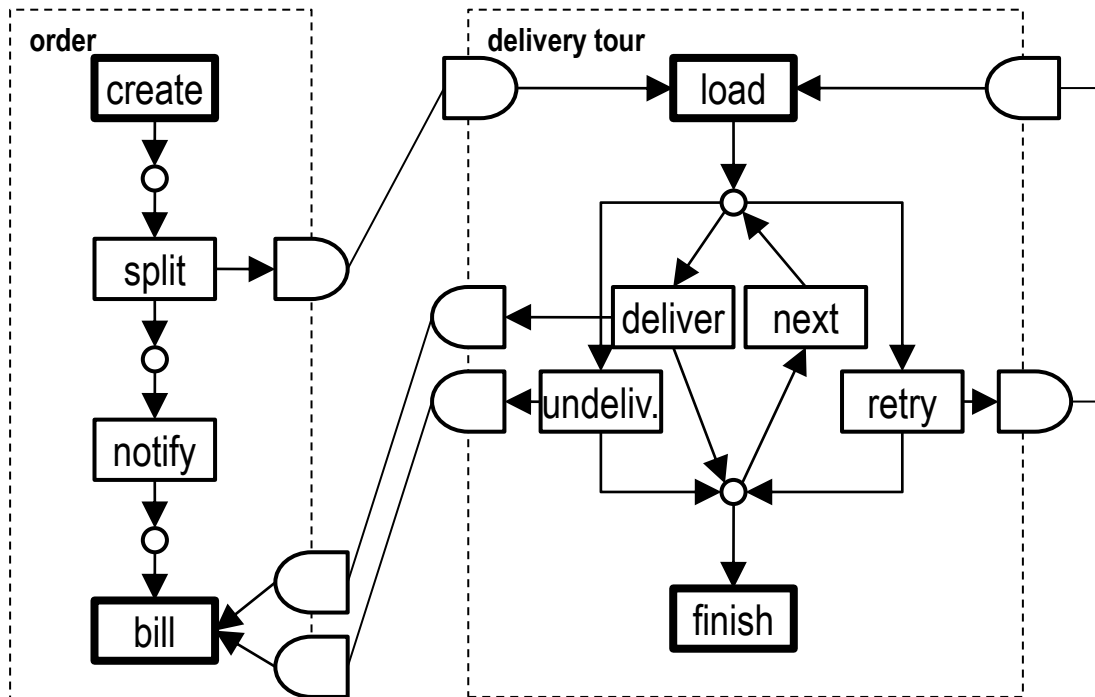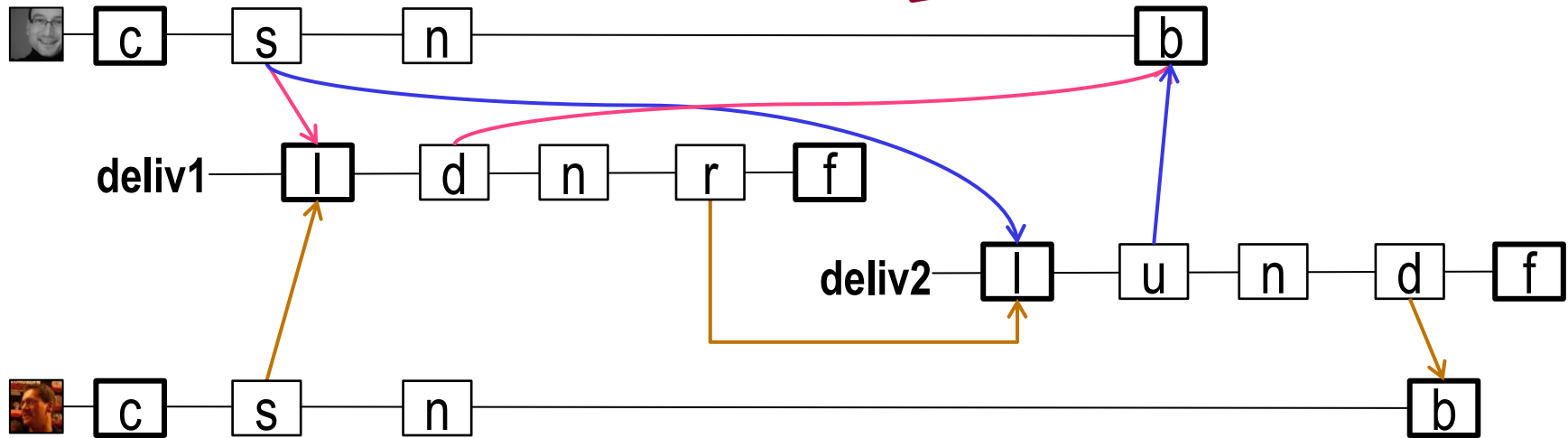**instances**

Describe Multi-Instance Processes

- **Artifacts**
- Creating a complete artifact model
- A new paradigm
- Research problems

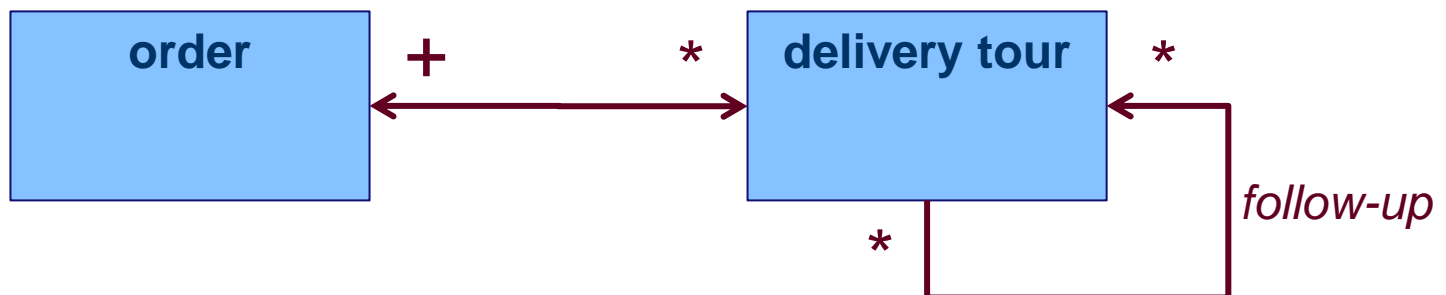# Describe Services

An example execution

service model: incomplete
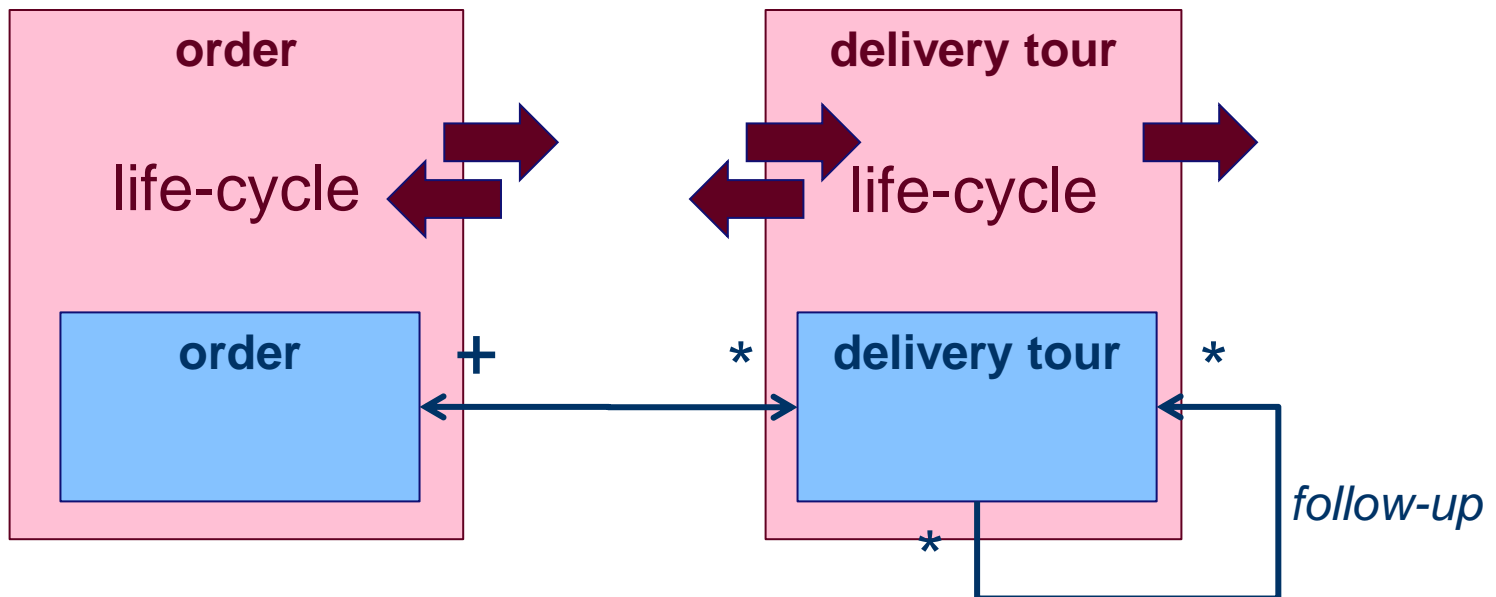
The reason of complex relations

- **Data**

- each process is founded in its (data) **objects**
  - an order
  - a delivery tour
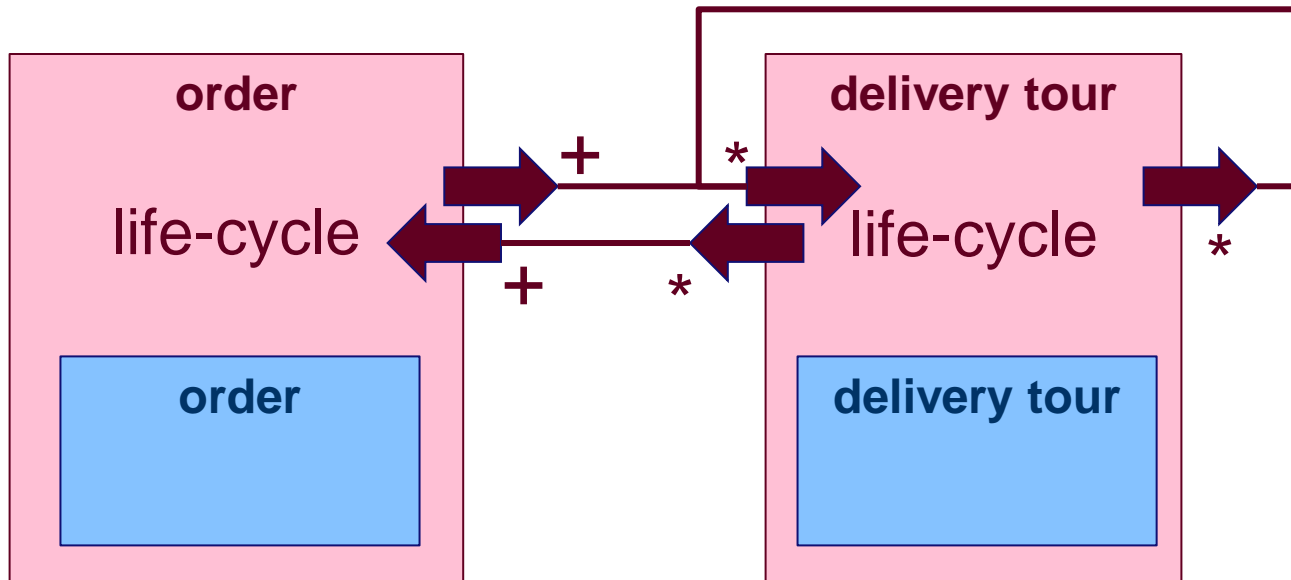- objects can be in complex **relationships**

Artifacts

- encapsulate each class of objects in a service
- service behavior = object life-cycle
- service interface = object interaction

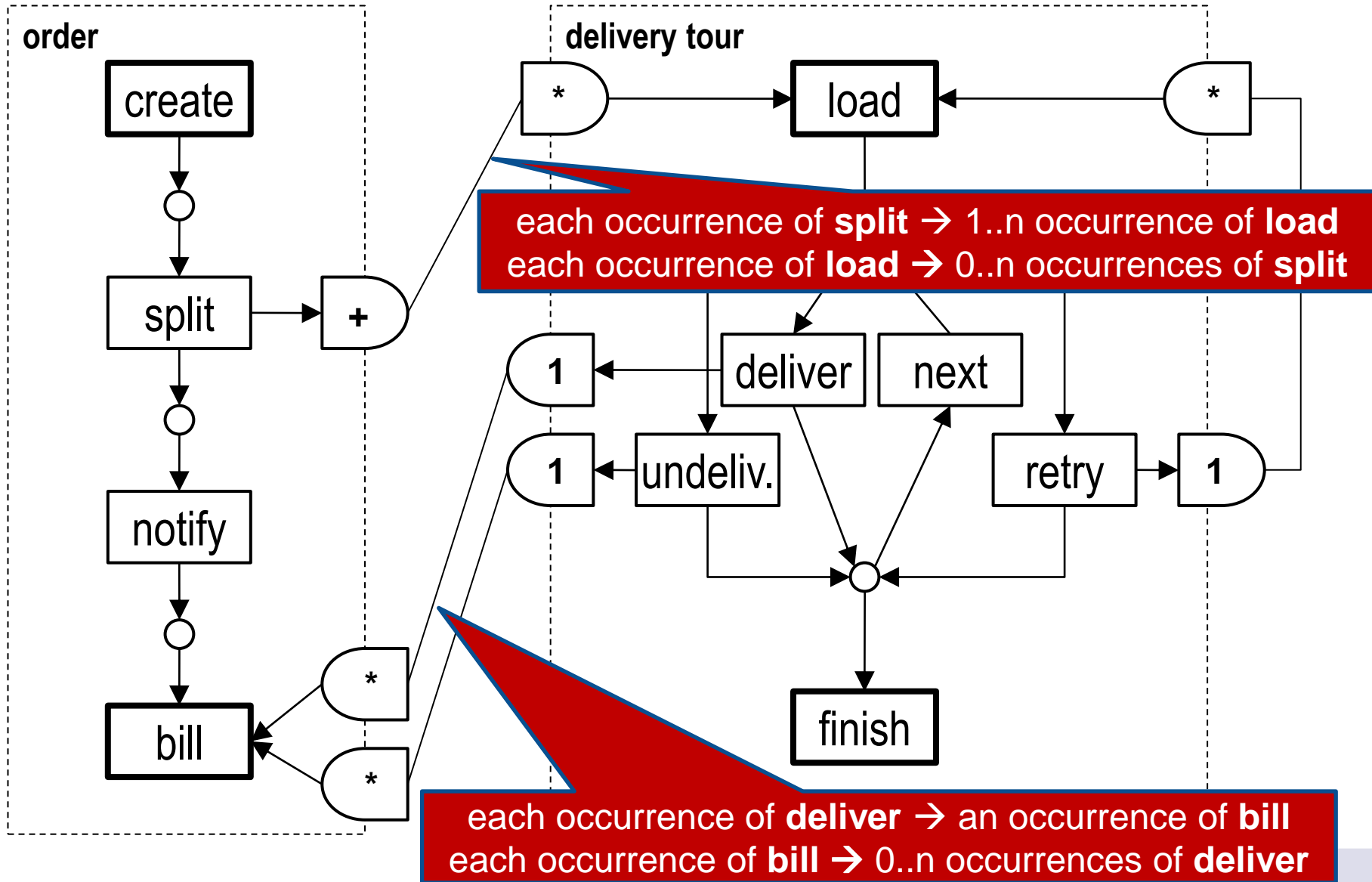[Nigam, Caswell 2003], [Cohn, Hull 2009]
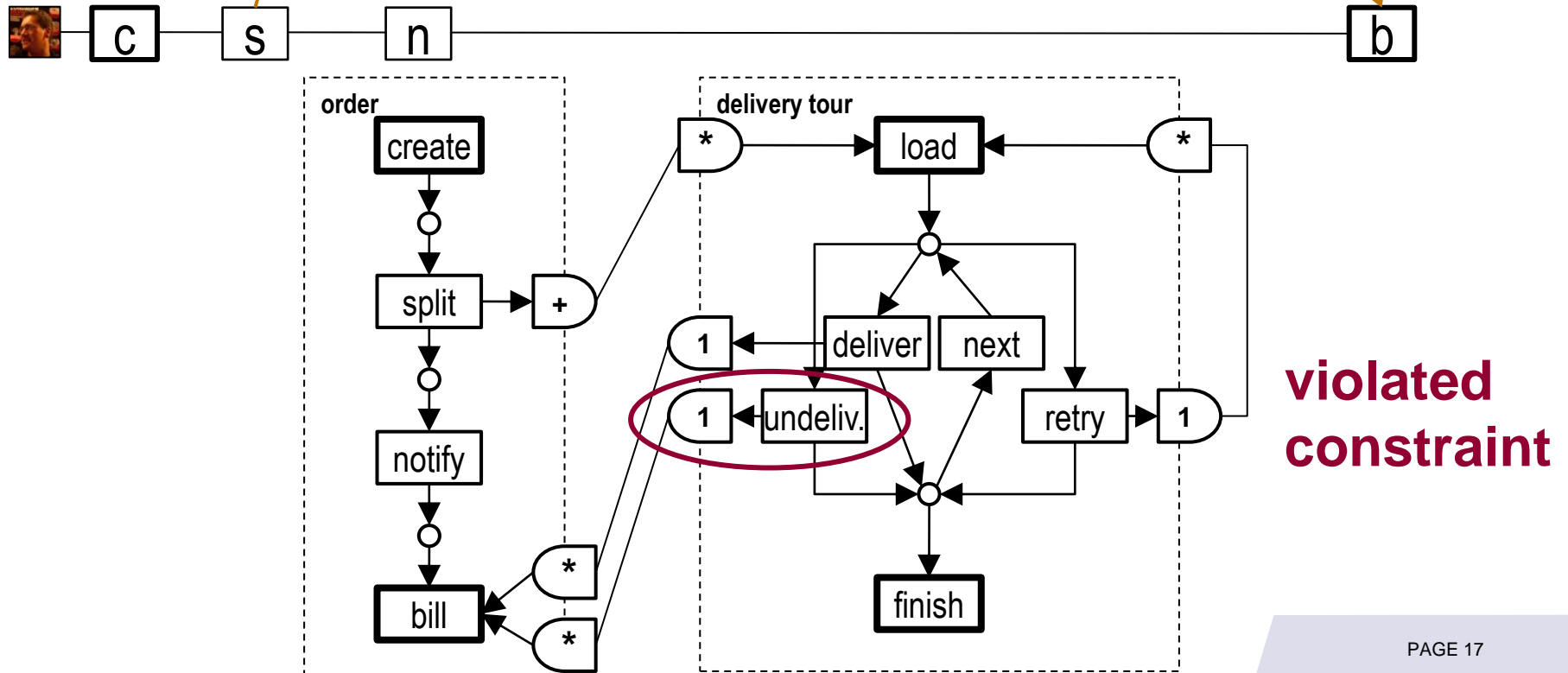
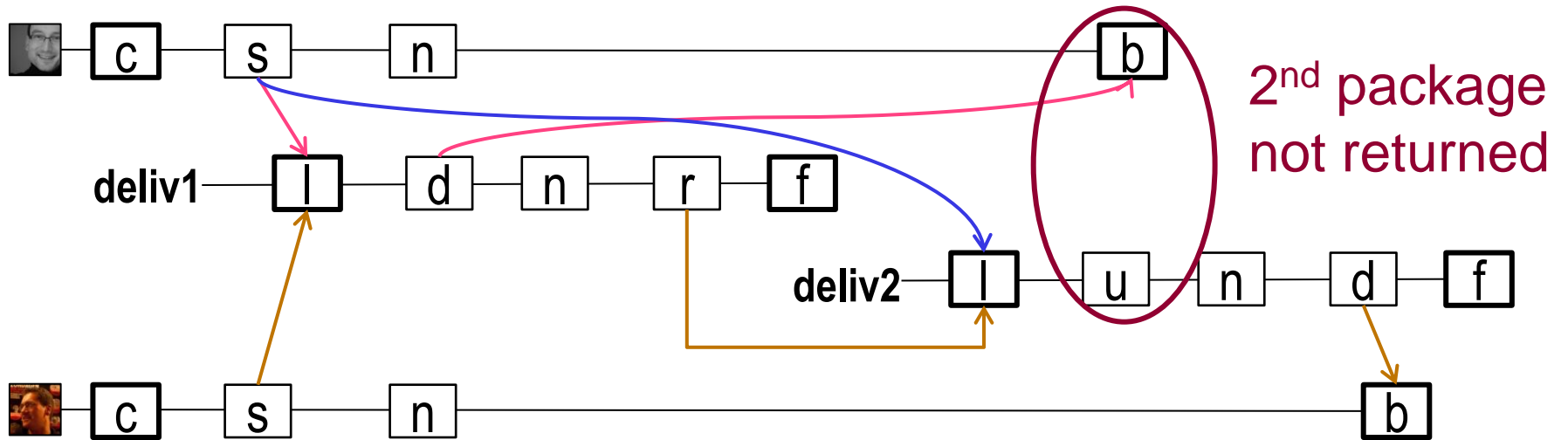# Service wiring inherits relationships

- objects in complex **relationships**
- relationships lift to **service wiring:**
  how many instances interact with each other
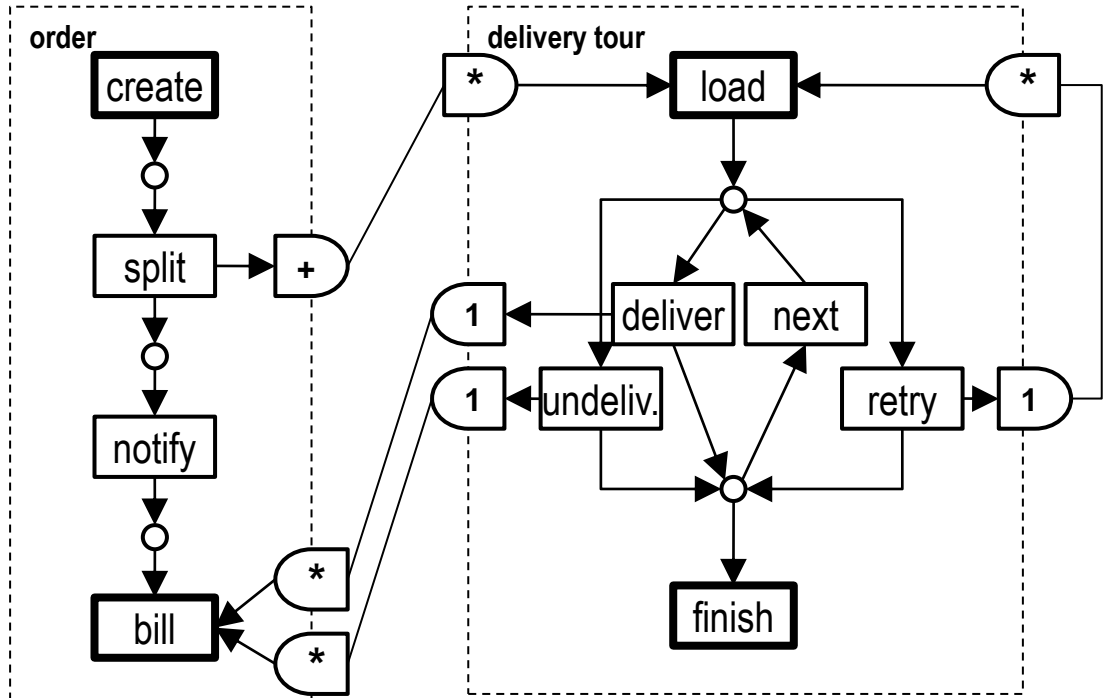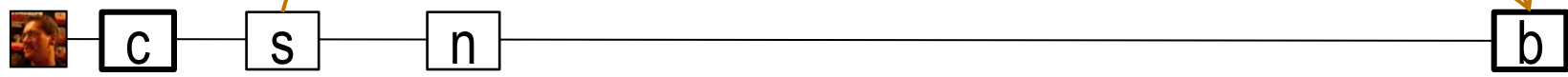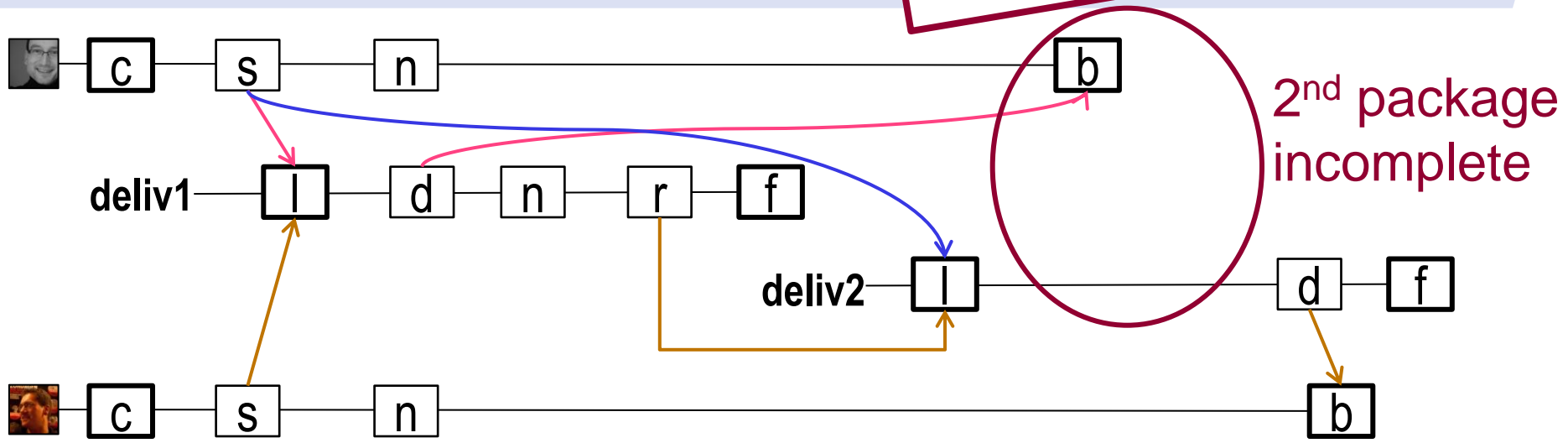
Cardinalities between events

order

create → split → notify → bill

delivery tour

each occurrence of **split** → 1..n occurrence of **load**
each occurrence of **load** → 0..n occurrences of **split**

each occurrence of **deliver** → an occurrence of **bill**
each occurrence of **bill** → 0..n occurrences of **deliver**

load, deliver, next, undeliv., retry, finish

Cardinalities constrain allowed executions

2nd package not returned

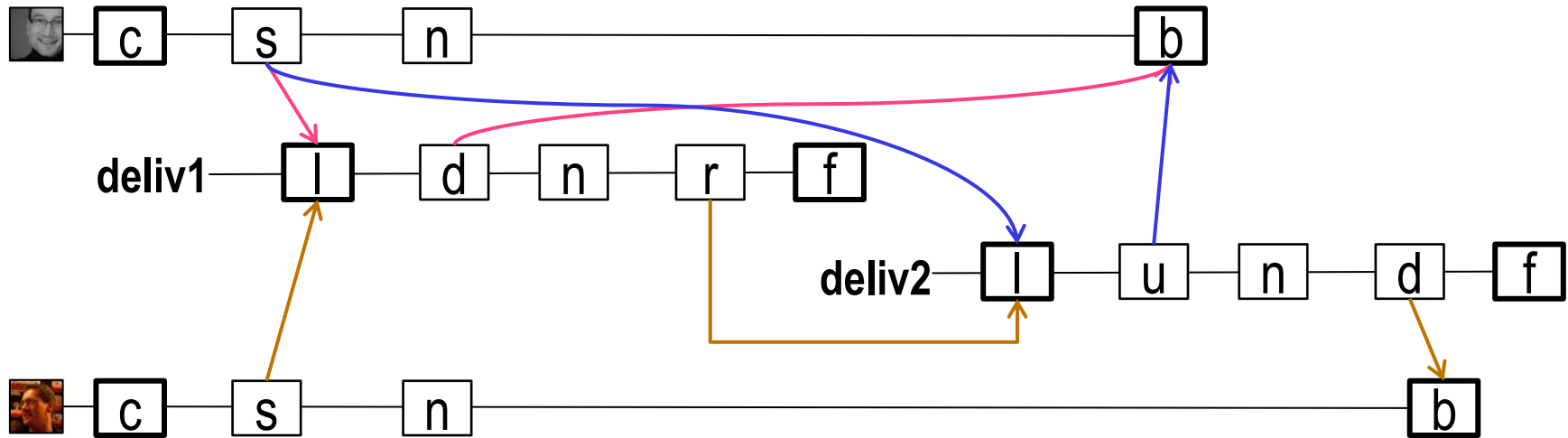violated constraint

Cardinalities are not enough

2nd package incomplete

Outline

- Artifacts
- **Creating a complete artifact model**
- Active and passive artifacts
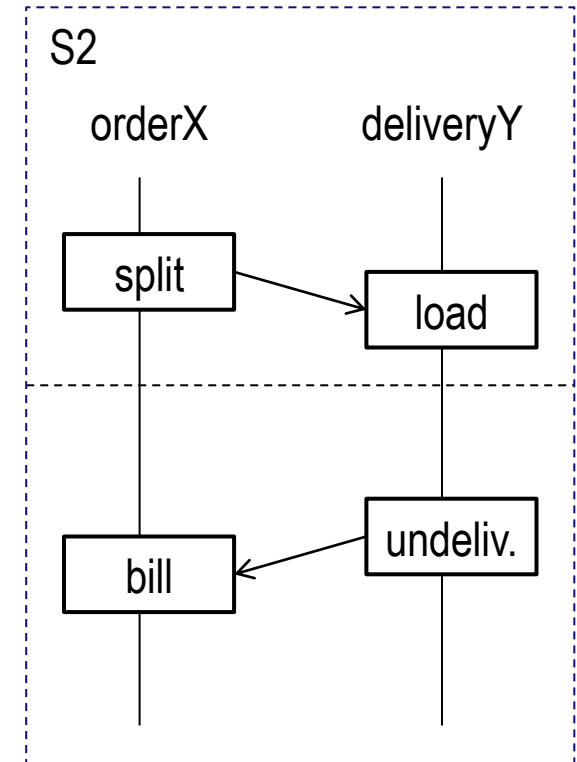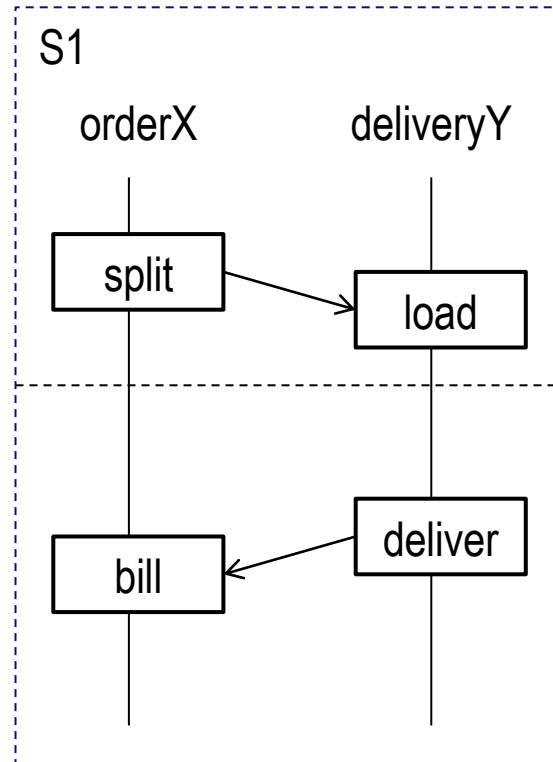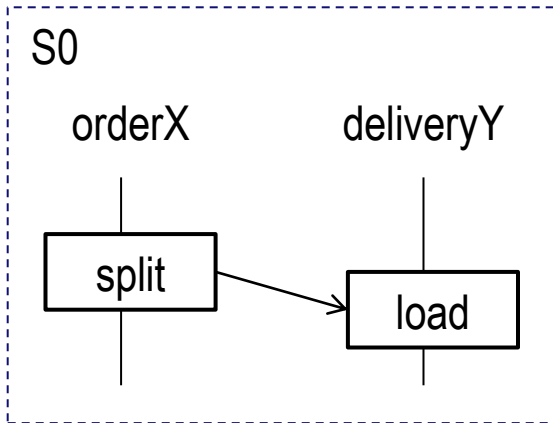- Research problems

Conversation

- a set of corresponding communication events
- between a fixed set of instances
  - C1: order1, delivery1
  - C2: order2, delivery1, delivery2
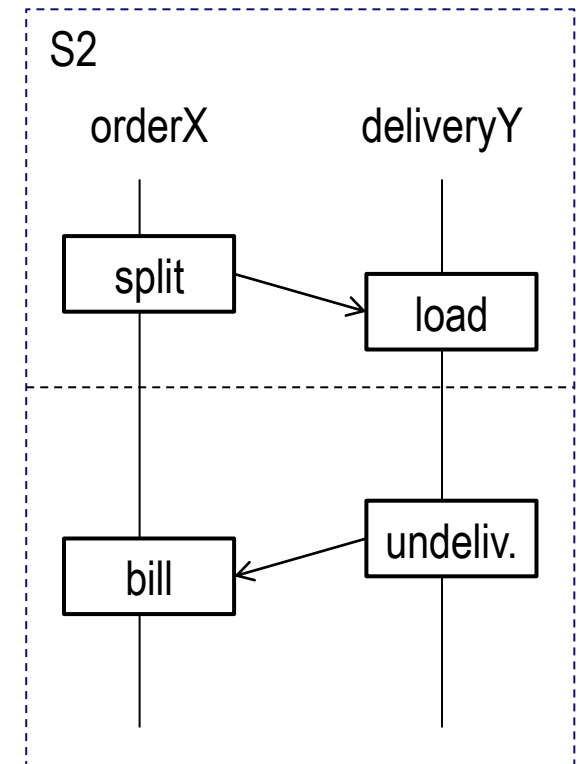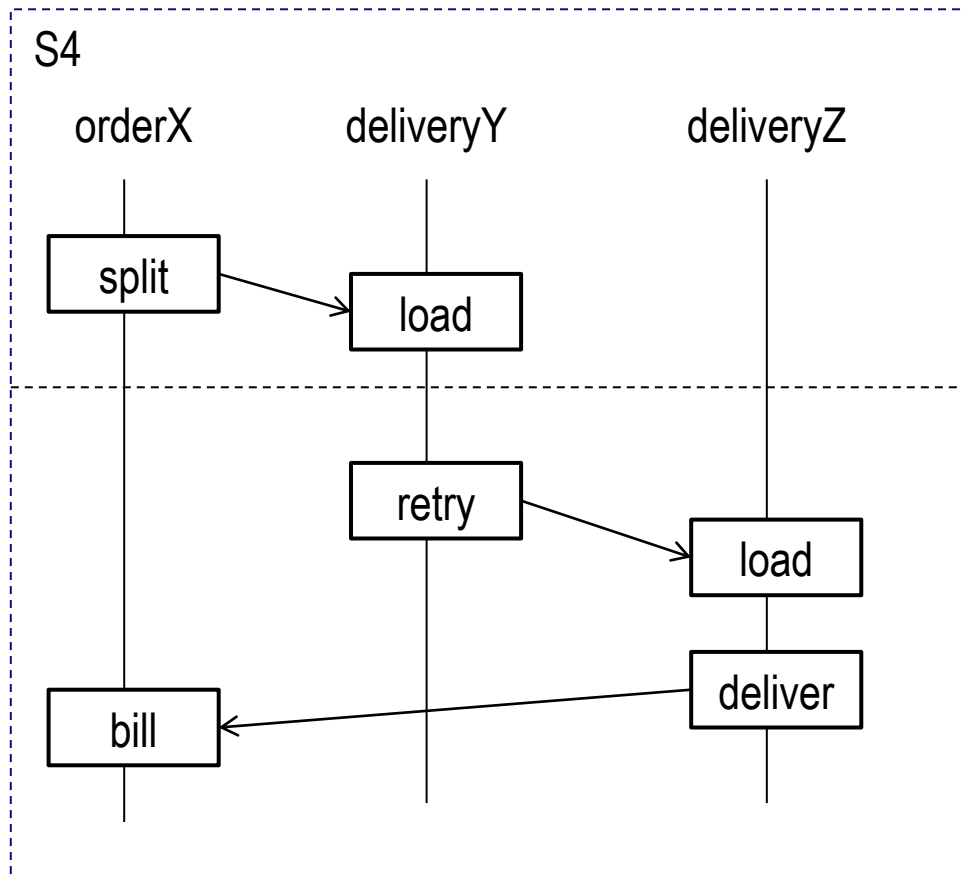 - C3: order1, delivery2
- conversations can overlap

Specify Conversations

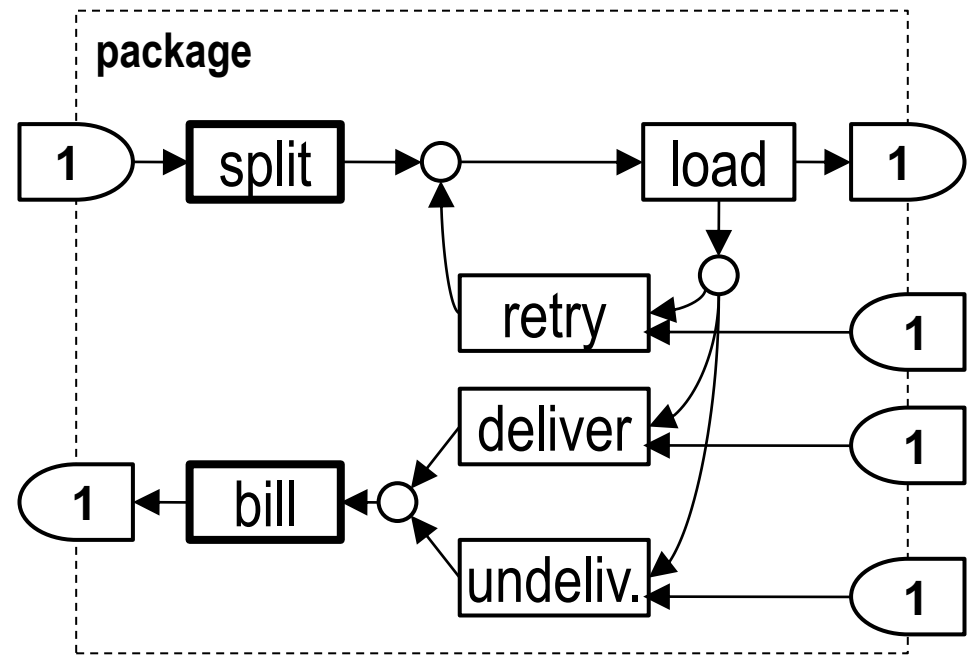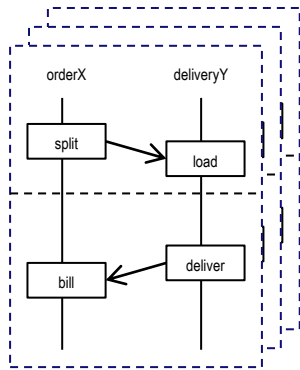- by interaction **scenarios** between **instances**

Specify Conversations

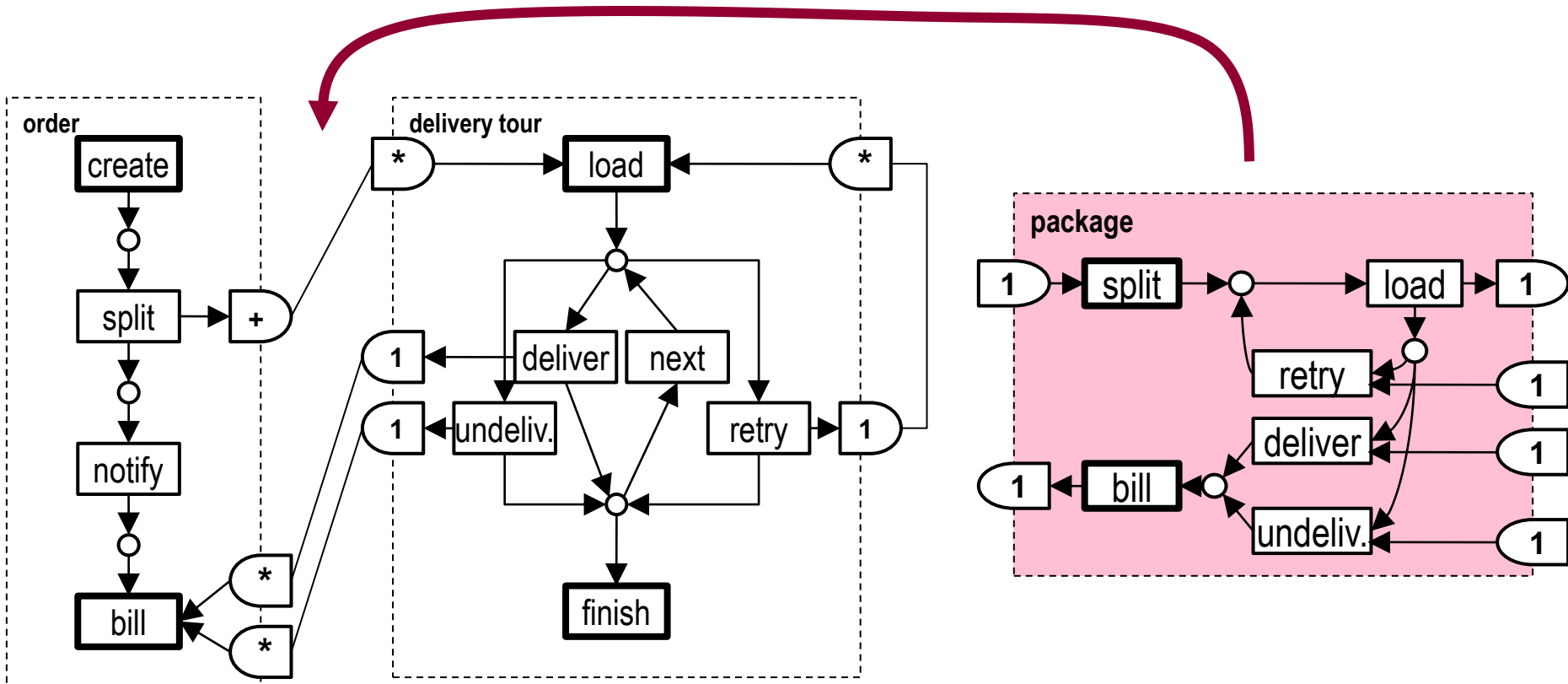- by interaction **scenarios** between **instances**

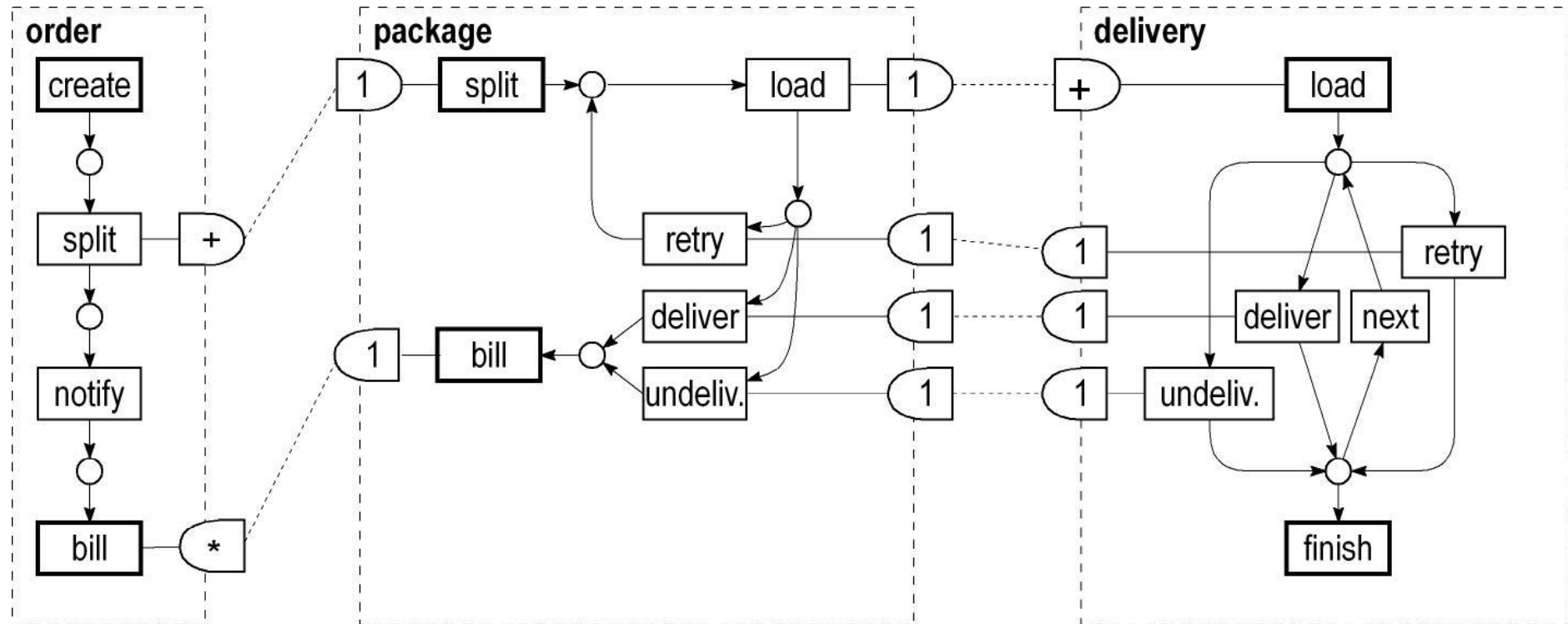# Synthesize Conversation Artifact

scenarios of the conversation

artifact model of the conversation
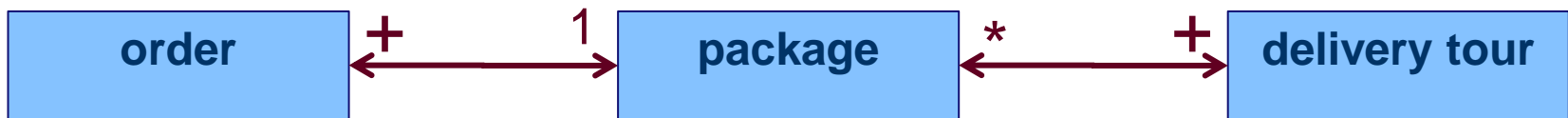
Refine Process Model

Refine Process Model

- interaction: synchronous
  with dynamic number of events
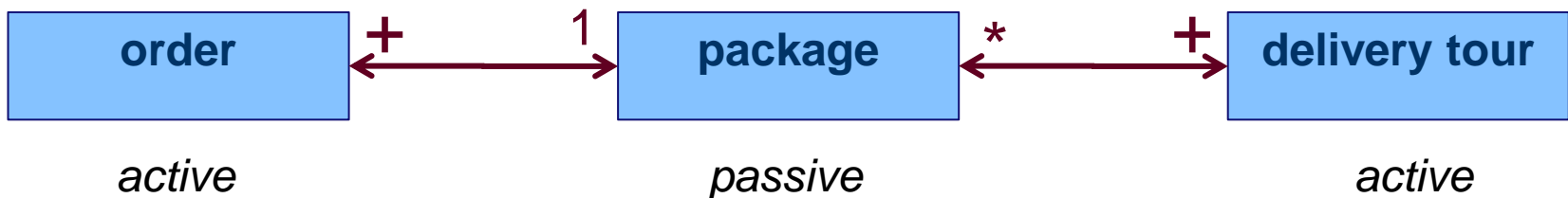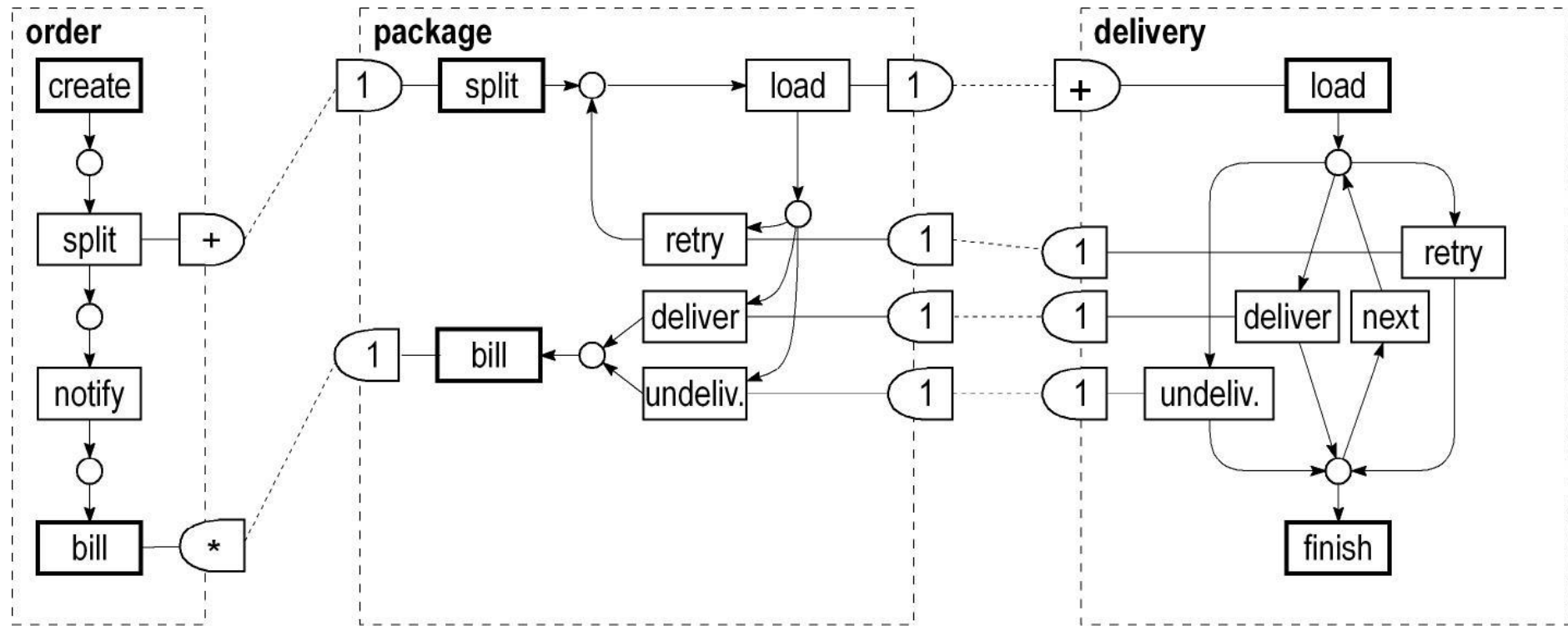
Outline

- Artifacts
- Creating a complete artifact model
- **Active and passive artifacts**
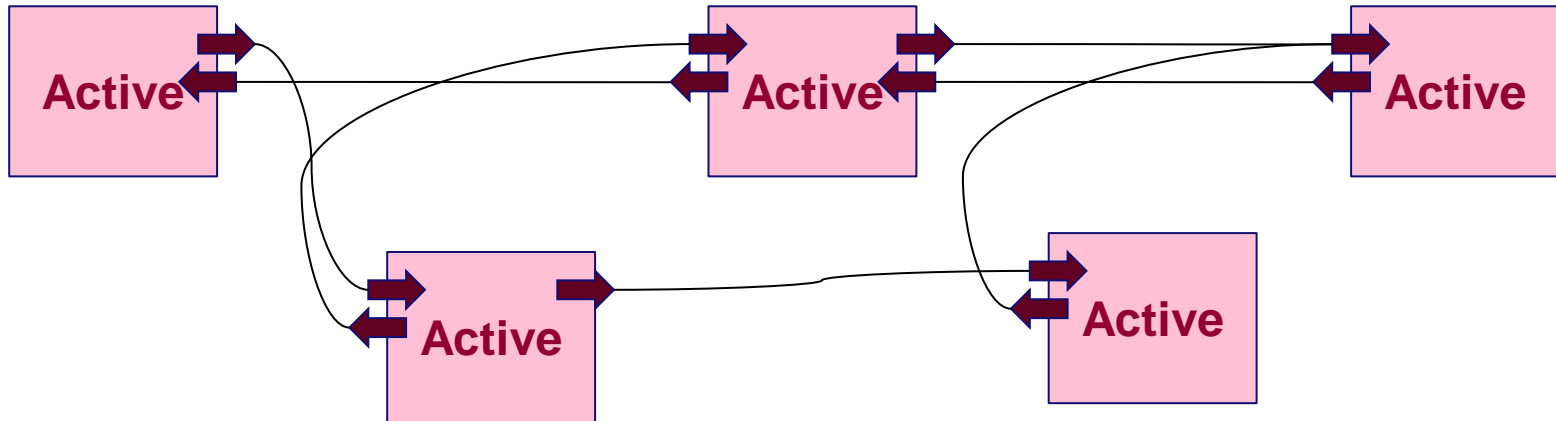- Research problems

1st Observation: Refined Class Model
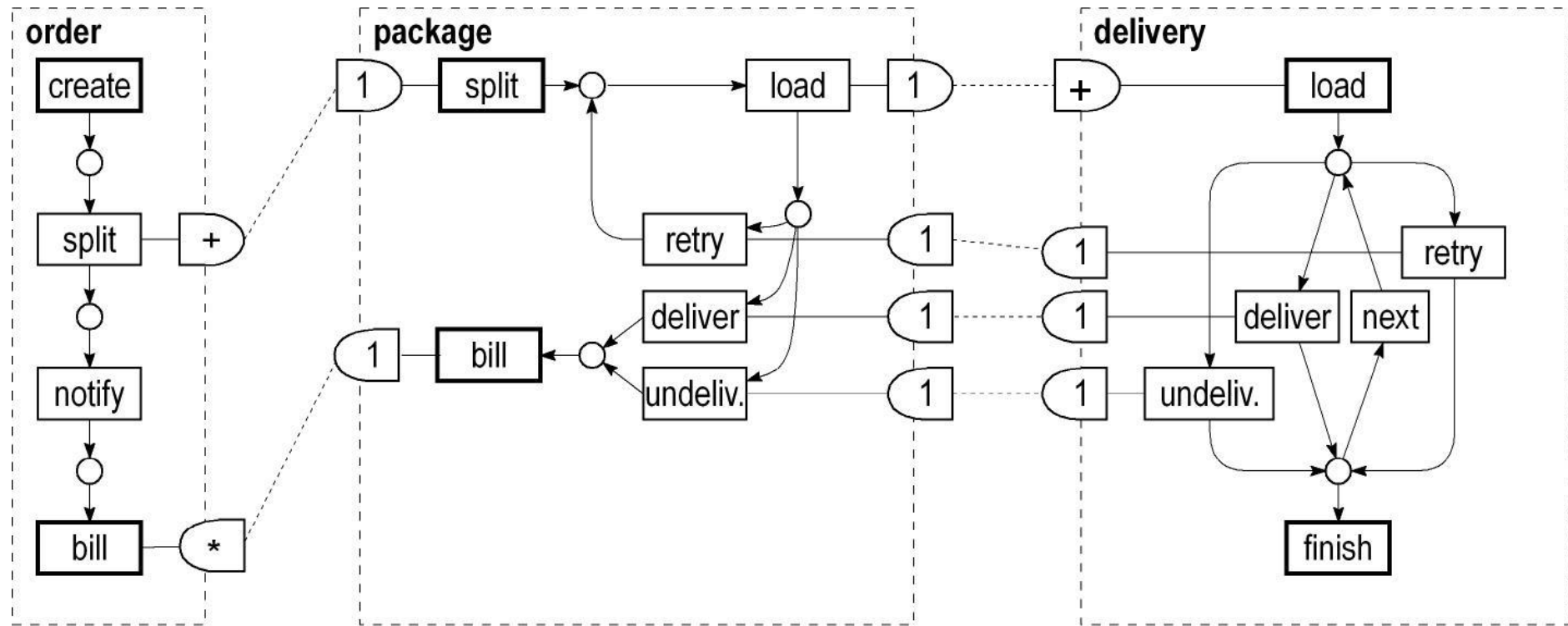
2nd Observations: Active and Passive

Active and Passive Artifacts

- Active Artifact
  - "service": activities, performed by agents driving the process, e.g. split an order, write an invoice, …
  - can communicate (asynchronously)

Active and Passive Artifacts

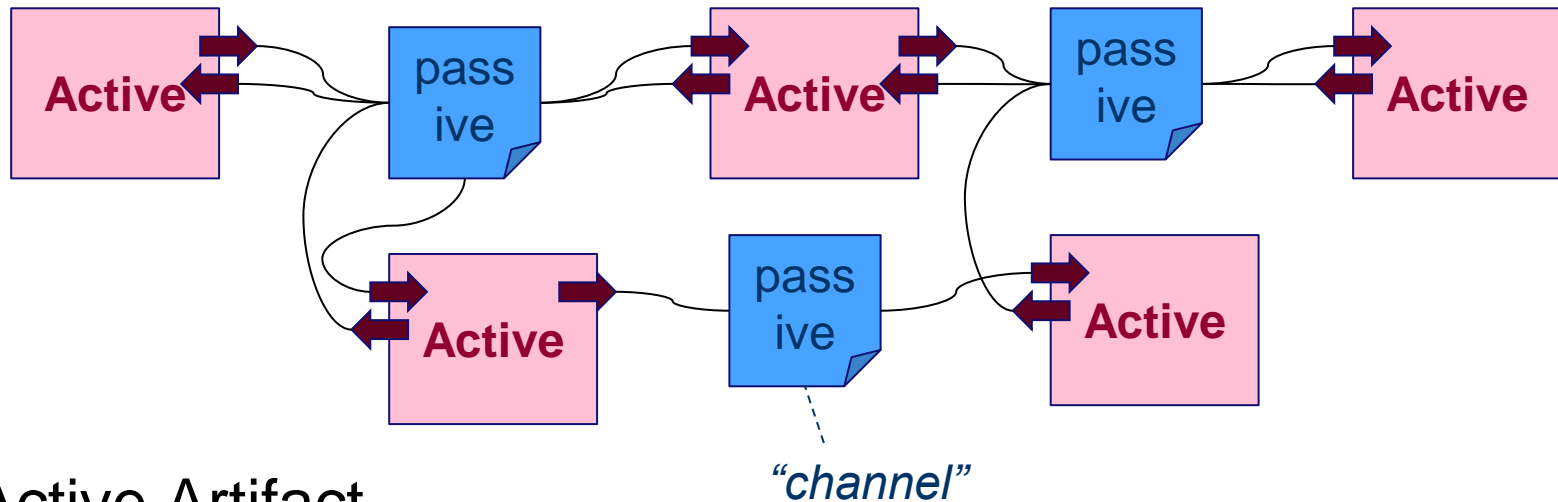*"channel"*
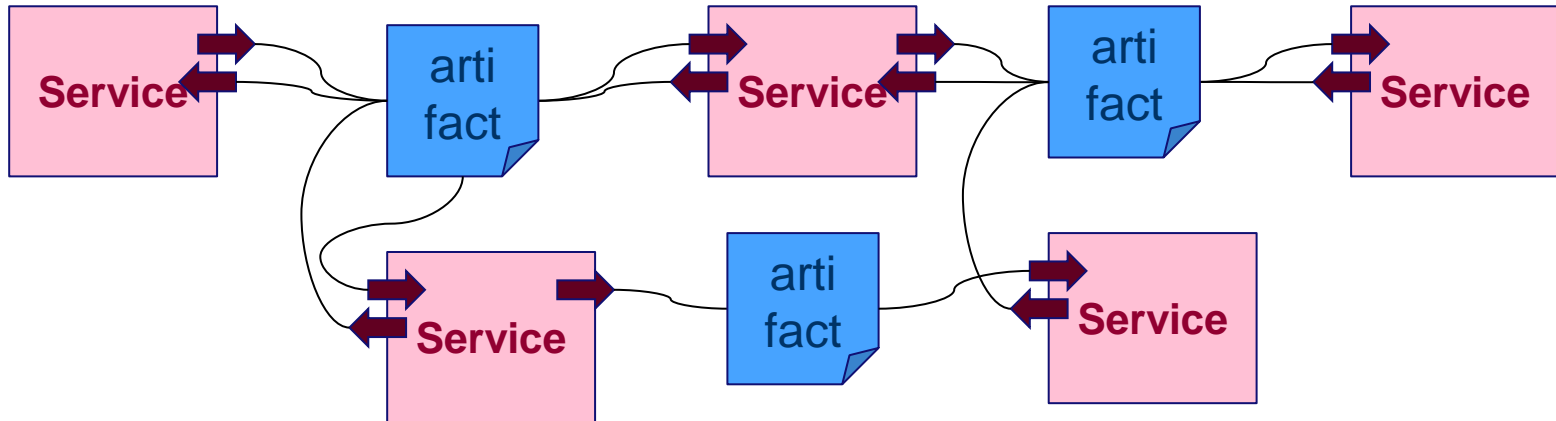
- Active Artifact
  - "service": activities, performed by agents driving the process, e.g. split an order, write an invoice, …
  - can communicate (asynchronously)
- Passive Artifact
  - "data object": attributes, can be updated, e.g. a package
  - order of updates is constrained, triggered from outside
  - **restricts interaction between several active artifact instances**

Services and Artifacts

"channel"

- Service
  - activities, performed by agents driving the process, e.g. split an order, write an invoice, …
 - can communicate (asynchronously)
- Artifact = "data object"
  - attributes, can be updated, e.g. a package
  - order of updates is constrained, triggered from outside
  - **restricts interaction between several service instances**
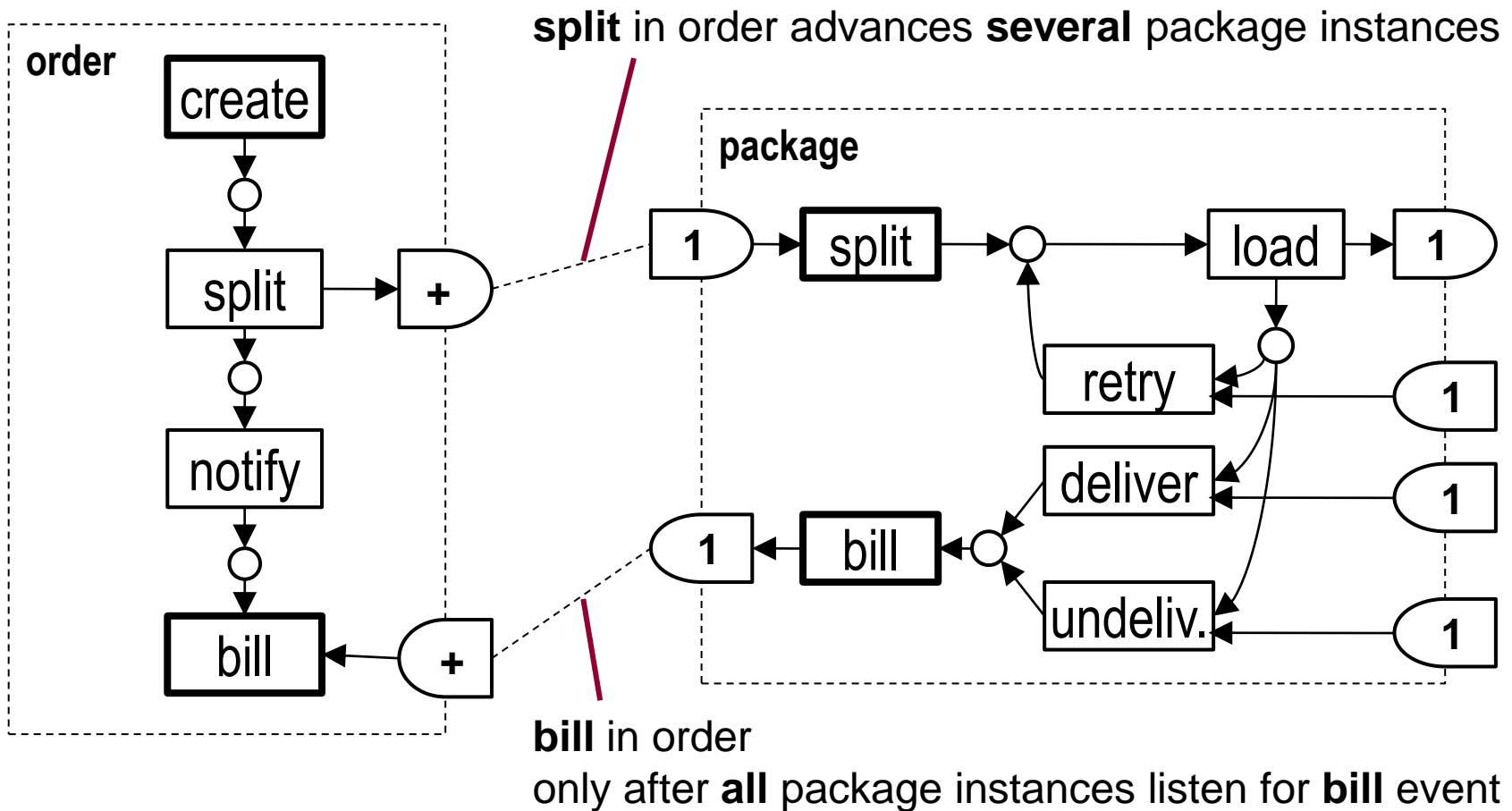
Relation to Behavioral Programming

Artifact package restricts service behavior of order and delivery

load only **after** split, i.e. load is **blocked** in any other state
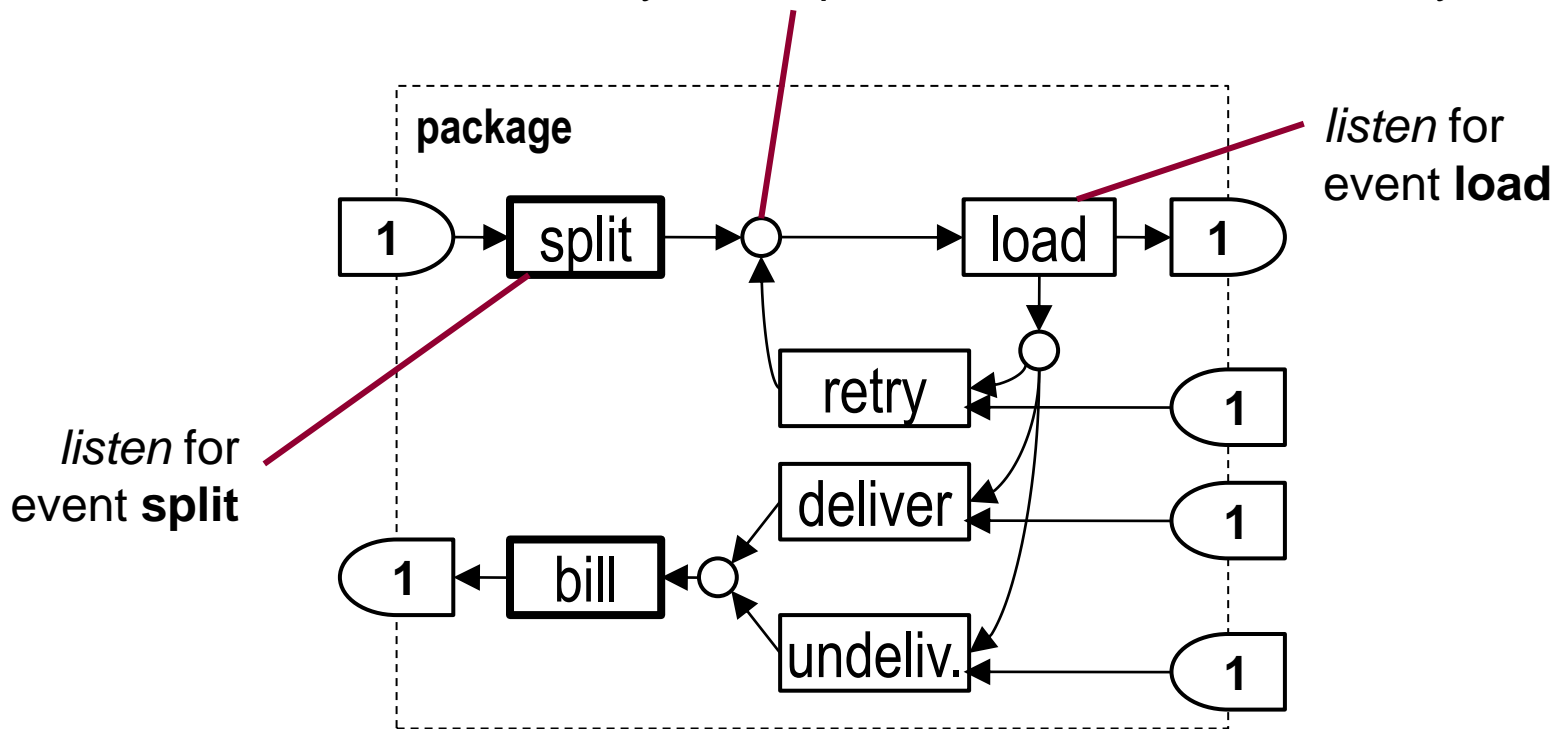
listen for event **load**

listen for event **split**

package

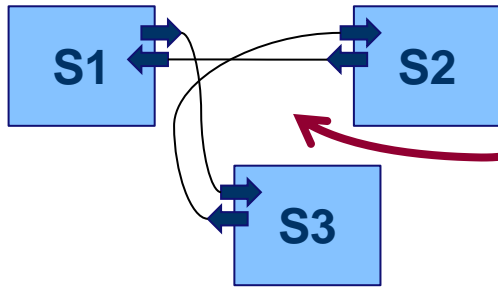# Relation to Behavioral Programming

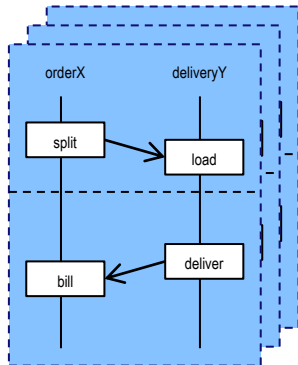Artifact package restricts service behavior of order and delivery

Outline

- Artifacts
- Creating a complete artifact model
- Active and passive artifacts
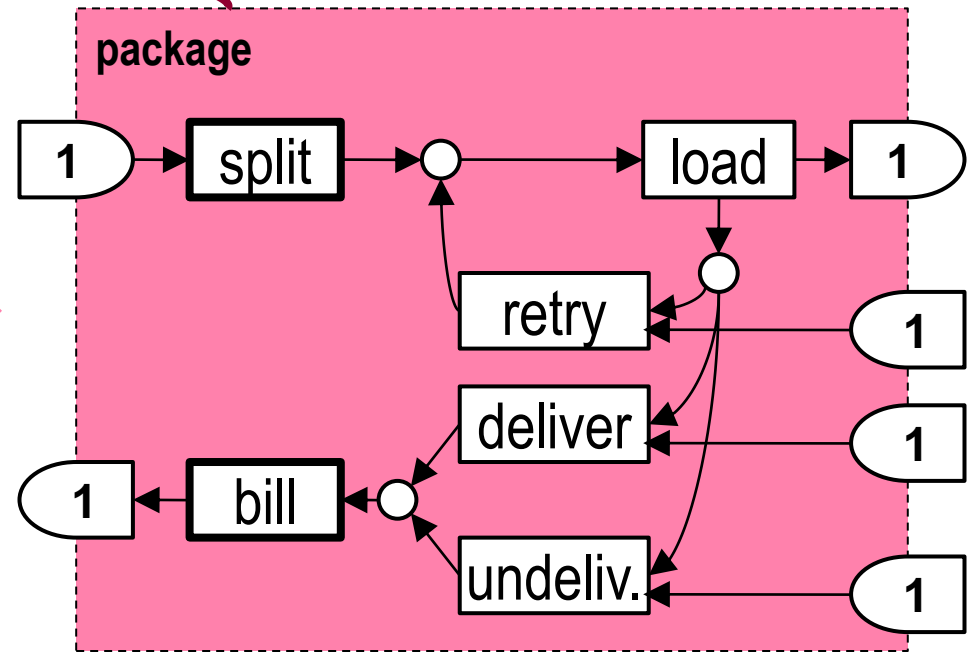- **Research problems**

1) Artifact Synthesis
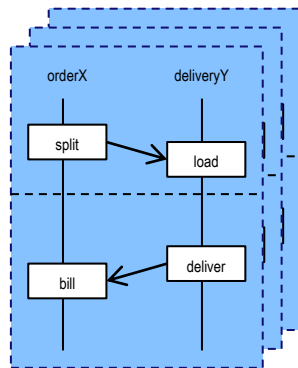

service choreography
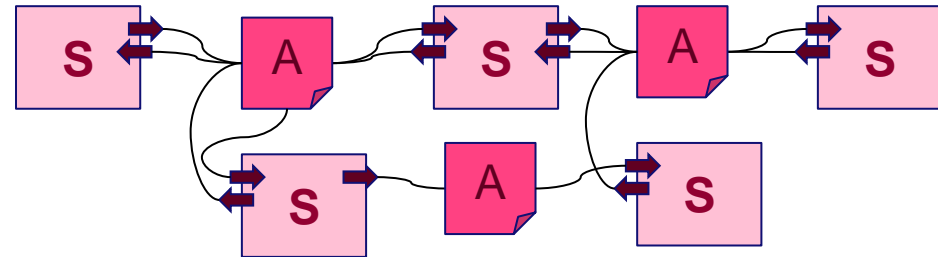
scenarios of the conversation

artifact model of the conversation

1b) Service + Artifact Synthesis

scenarios
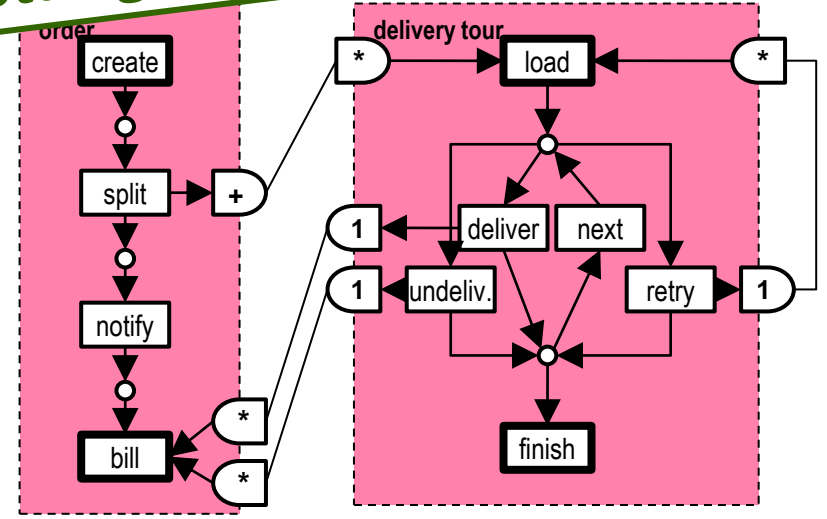between objects

**process model of
services + artifacts**

2) Service Synthesis
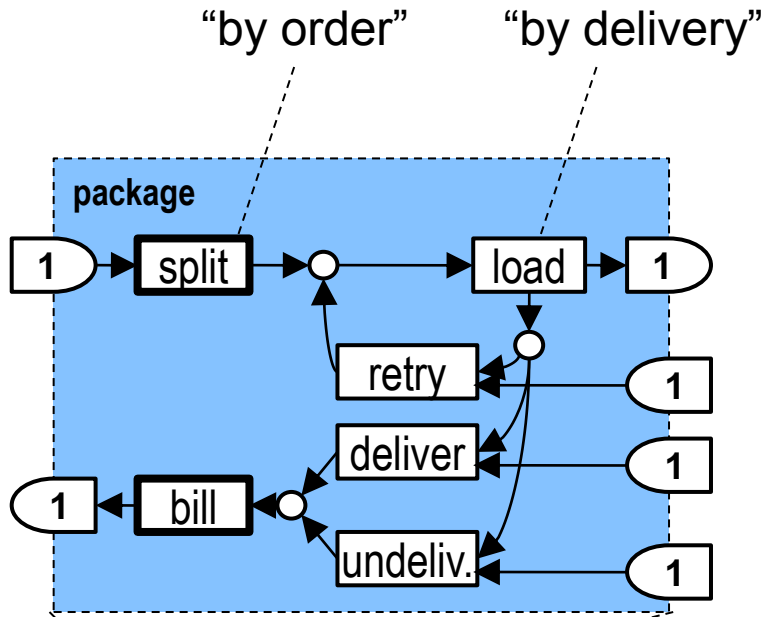
service models

s.t. artifacts + services always reach the goal states

annotated artifacts
+
goal states
(e.g. **bill** occurred)
+
behavioral constraints
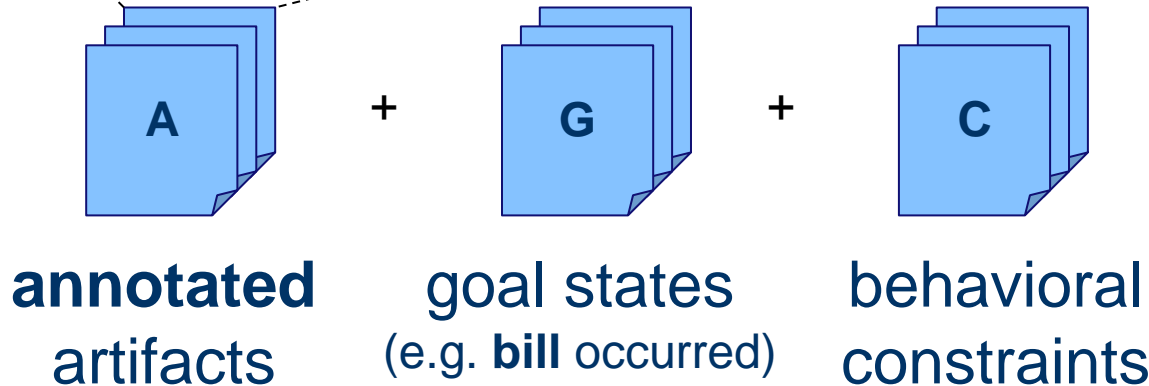
More Problems

3) Adapter Synthesis

- given services and artifacts
  where composition not always terminates

- synthesize an adapter (artifact or service)
  s.t. the extended composition terminates

4) Service/Artifact Discovery

- given an actual information system: writes log files

- synthesize artifacts + services
  s.t. the composition describes behavior in the log files

Wrap Up

- **Process = Services + Artifacts**

  - paradigm to capture multi-instance processes based on an object model

  - one artifact describes a conversation between several service instances

  - bears similarities to behavioral programming

- **Research Problems**

  - extend known problem by multi-instance dimension

Dirk **Fahland**
about.me/dirk.fahland

Artifacts
Processes with Multiple Instances

TU/e Technische Universiteit
Eindhoven
University of Technology

Where innovation starts