Artifacts
Combining Processes and Objects

Dirk Fahland
with contributions by Niels Lohmann
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
Outline

1. Activity-Centric Process Modeling + Data
2. Artifact-Centric Modeling
3. Processes with multiple instances
Activity-Centric Process Modeling

[by Niels Lohmann]
Activity-Centric Process Modeling

[by Niels Lohmann]
Activity-Centric Process Modeling

Disadvantages
- data objects scattered all over the model
- object behavior not expressed
- goal states of objects?

[by Niels Lohmann]
Outline

1. Activity-Centric Process Modeling + Data
2. Artifact-Centric Modeling
3. Processes with multiple instances
Process Objects

Debit

Cargo

Order

[by Niels Lohmann]
Object Life-Cycles

[Diagram of cargo life cycle]

INITIAL STATE

[Diagram of order life cycle]

EXECUTING AGENT

[Diagram of debit life cycle]

FINAL STATE

[by Niels Lohmann]
Locations and Message Exchange

[Diagram showing the process of a debit transaction with stages for filling, processing, and using, indicating initial location and location at the buyer, with communication channels.]
Shared Objects and Remote Access

**Order**

- unpaid
- register payment
  - Shipper
- paid
- undecided

- order beer
  - Buyer
  - beer ordered
- order wine
  - Buyer
  - wine ordered

**SHIPPER HAS REMOTE ACCESS**

[by Niels Lohmann]
Goal States

SPECIAL VIEW: SHOW ONLY FINAL STATES

MAKE THIS COMBINATION VALID

[by Niels Lohmann]
Artifact-Centric Process Model

[Diagram of Debit, Cargo, and Order processes with various actions and relationships between entities such as Buyer, Shipper, load, beer, wine, order beer, order wine, unpaid, paid, undecided, beer ordered, wine ordered, register payment, and send to buyer.]
1. Activity-Centric Process Modeling + Data
2. Artifact-Centric Modeling
3. Processes with multiple instances

- objects
- object life-cycles
- exchange/access objects
- policies and goal states

- object relations?
A process that needs multiple instances

complex relations between object instances
Complex relations between objects...

- ...lift to complex relations in object interactions
Cardinalities between actions

Each occurrence of **split** preceded by 0..n occurrences of **load**

Each occurrence of **split** followed by 1..n occurrence of **load**

**Policy**

Order:
- **create**
- **split**
- **notify**
- **bill**

Delivery tour:
- **load**
- **undeliv.**
- **retry**
- **finish**
Cardinalities between actions

order
- create
  - split
    - notify
  - bill

delivery tour
- * load
  - deliver
    - undeliv.
    - next
  - retry
    - finish
  - *
An example execution

model is incomplete
Correct execution: each package delivered
Conversations

order1 + delivery1 → package1
order1 + delivery1 → package1
order1 + delivery1 → package1

order2 + delivery2 → package2
order2 + delivery2 → package2
order2 + delivery2 → package2

order2 + delivery2 → package3
order2 + delivery2 → package3
order2 + delivery2 → package3

follow-up

*
Conversations → Refine Artifact Model

order → package → delivery

order1 + delivery1 → package1 → delivery2

order2 + delivery1 → package2 → delivery2

package3

follow-up
Describe Conversation Artifact

- a life-cycle
- a location
- some actions only at some locations
- some actions require specific participants → ports
Refine Process Model

order
  create
  split
  notify
  bill

delivery tour
  * load *
  1 deliver next
  1 undeliv. retry 1
  * finish *

package
  @order
  @delivery
  split
  load
  retry
  deliver
  undeliv.
  bill
  undeliv.
Refined Process Model

In each occurrence of **split**
1 instance of **order** and n instances of **package**
are synchronized
Take Home Points

object life-cycles

object relations

policies on object interaction

object exchange