Soundness Verification of Business Processes Specified in the Pi-Calculus

Frank Puhlmann
Hasso Plattner Institut
Potsdam, Germany

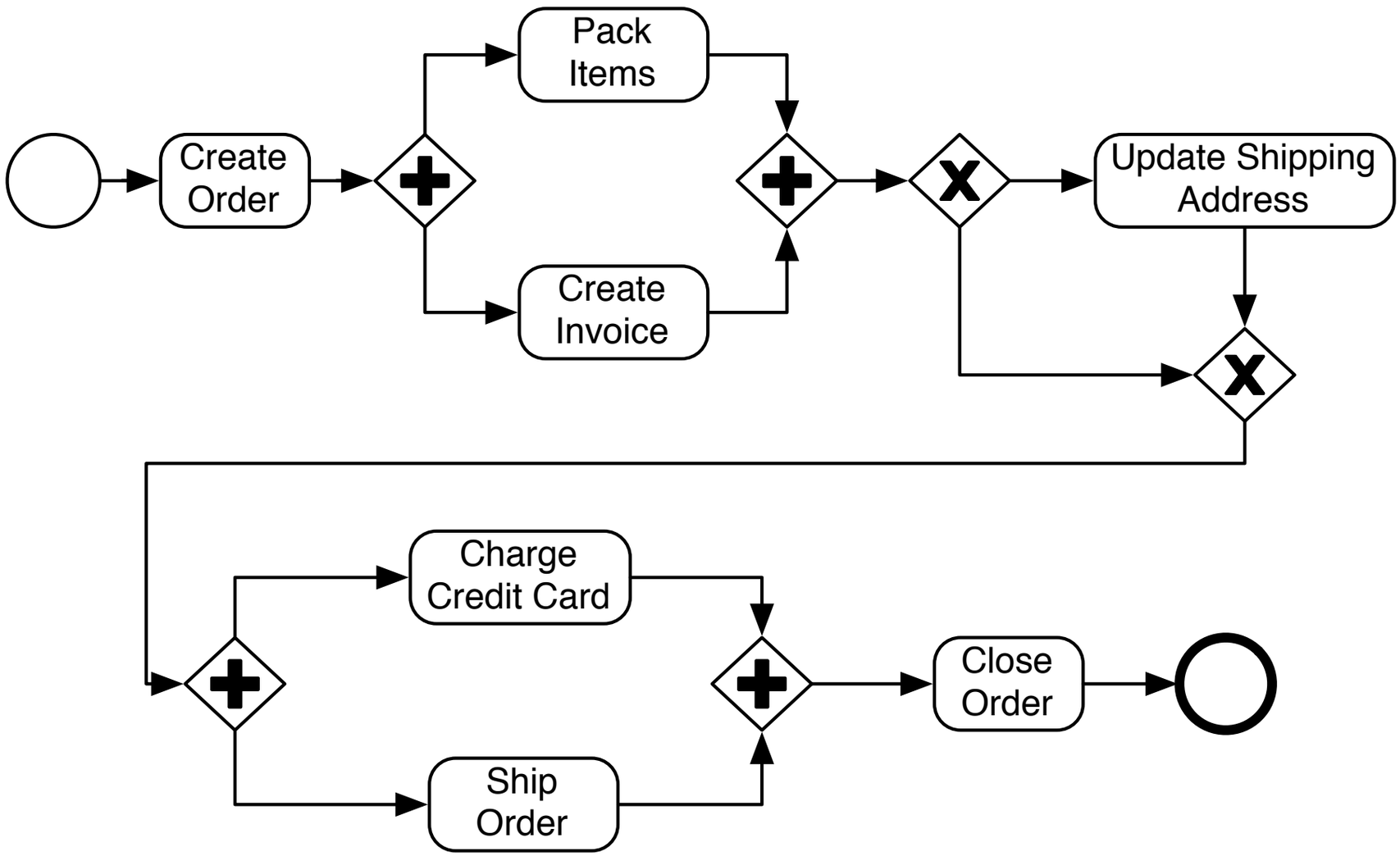
Content

- Formal Description and

  Verification of

  Business Rules for Process Models

- Warning: Simplified views!



Business Process

Business Rules

- Examples:
  - It should be possible to select a different shipping address
  - Charge credit card must always be done before ship order
  - When the order is shipped, the process instance must always be closed

Invariant

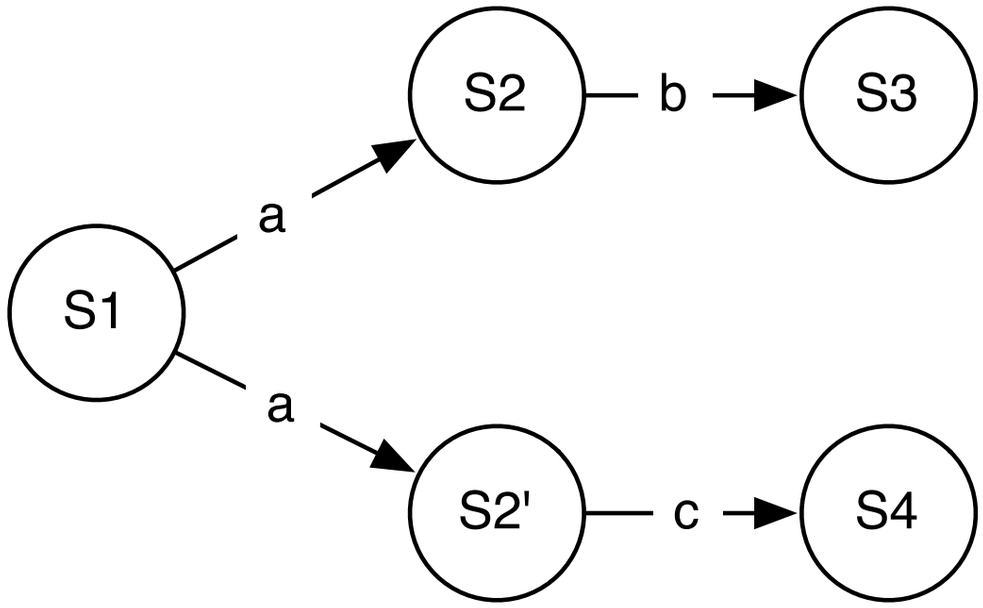
- Business rules represent invariants on process models
- They give an ordering of
  - Optional activities (can, might, should)
    - Technically: Exist quantifier
  - Required activities (must)
    - Technically: All quantifier

Formalized Business Processes

- Business processes can be represented as directed graphs, where each node has a certain semantics
- The semantics can be given by a transition system, such as
  - Process Algebra, e.g. CSP, CCS, Pi-Calculus
  - Petri nets, Abstract State Machines, etc.

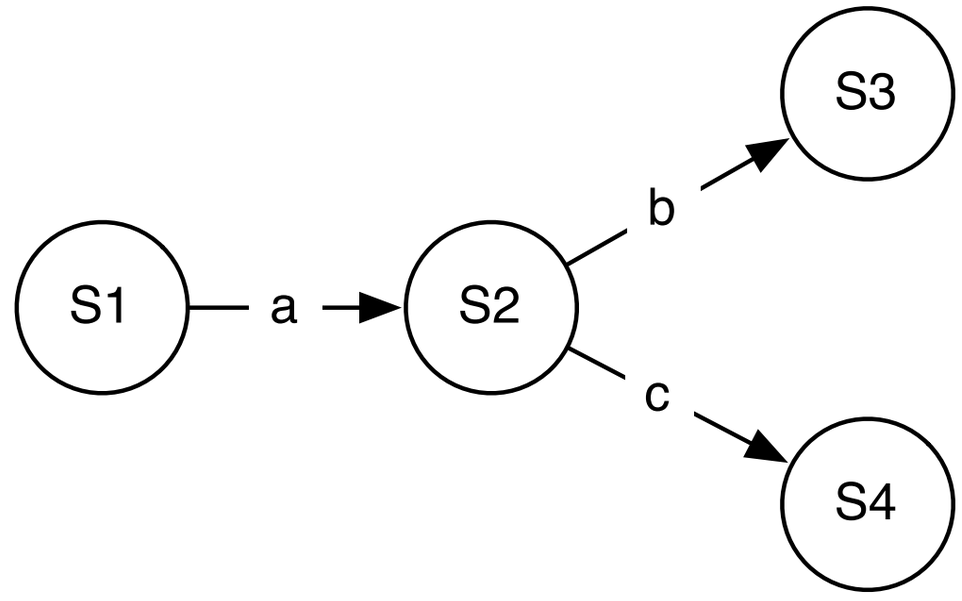
Simulations and Bisimulations

- Simulation: If a player A can do a move in his transition system, player B must be able to follow this move in her system
- Bisimulation: Like simulation, but both players can change the active role at each step



Player A

Player B



(Bi)-Simulation Example

Proving Invariants

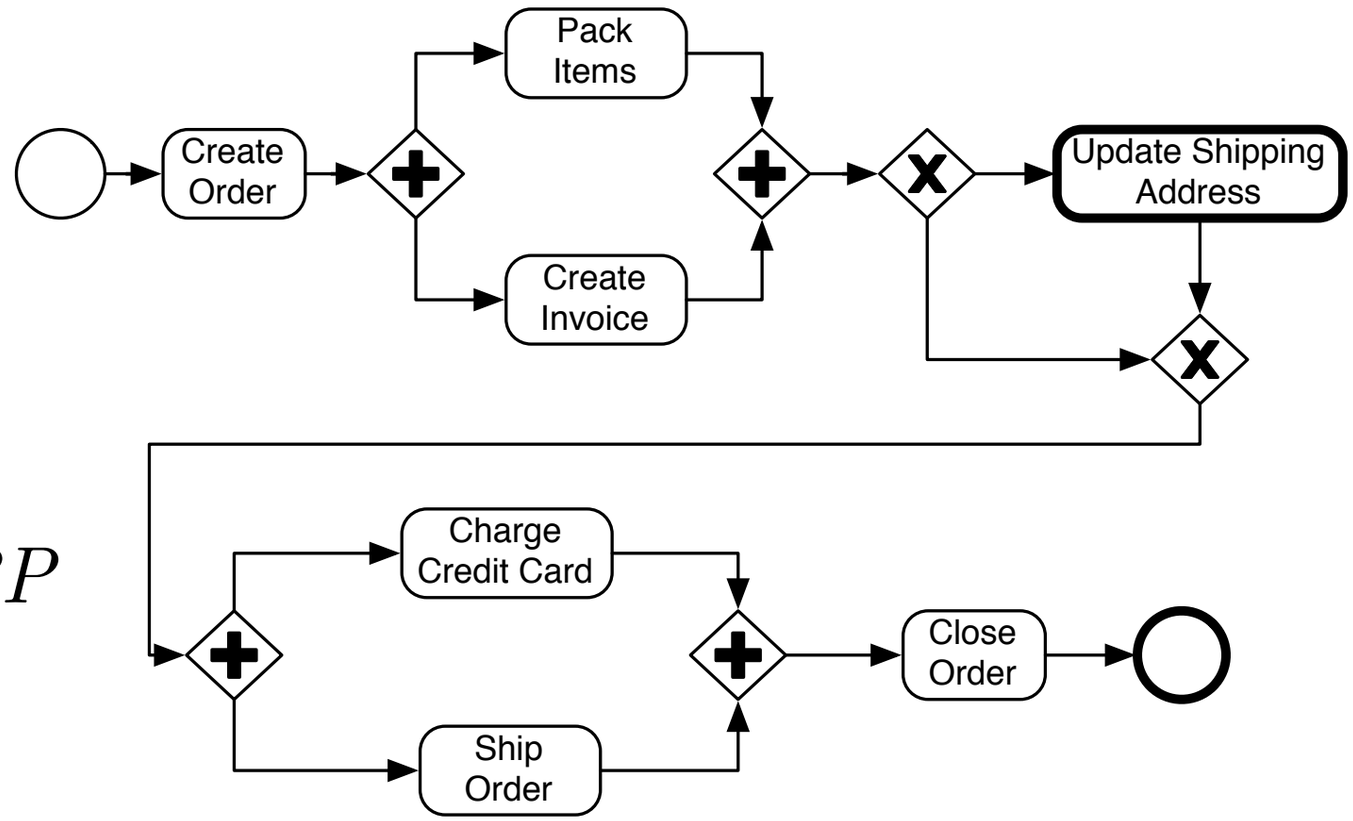
- Idea:
  - Describe invariant as "minimalistic" transition system
- Use simulation for invariants regarding optional activities
- Use bisimulation for invariants regarding required activities

Invariant:
It should be possible to select a different shipping address

Invariant

$$\overline{i.update\_shipping\_address}.\mathbf{0} \lessapprox BP$$

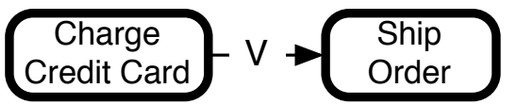
Business Process



Example 1

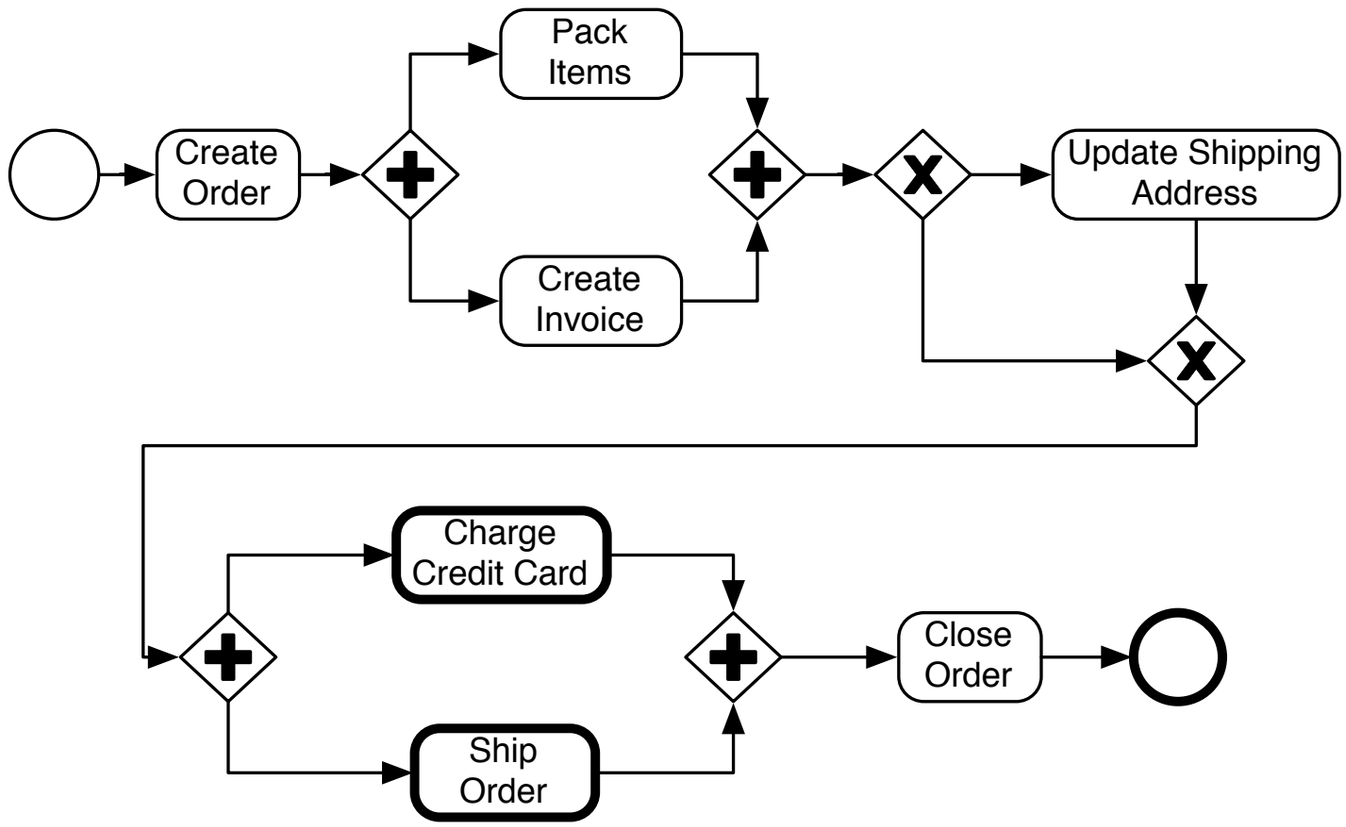
Invariant:
Charge credit card must always be done before ship order

Invariant



$$\overline{charge}.\overline{ship}.\mathbf{0} \not\approx BP$$

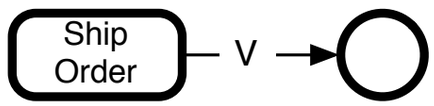
Business Process



Example 2

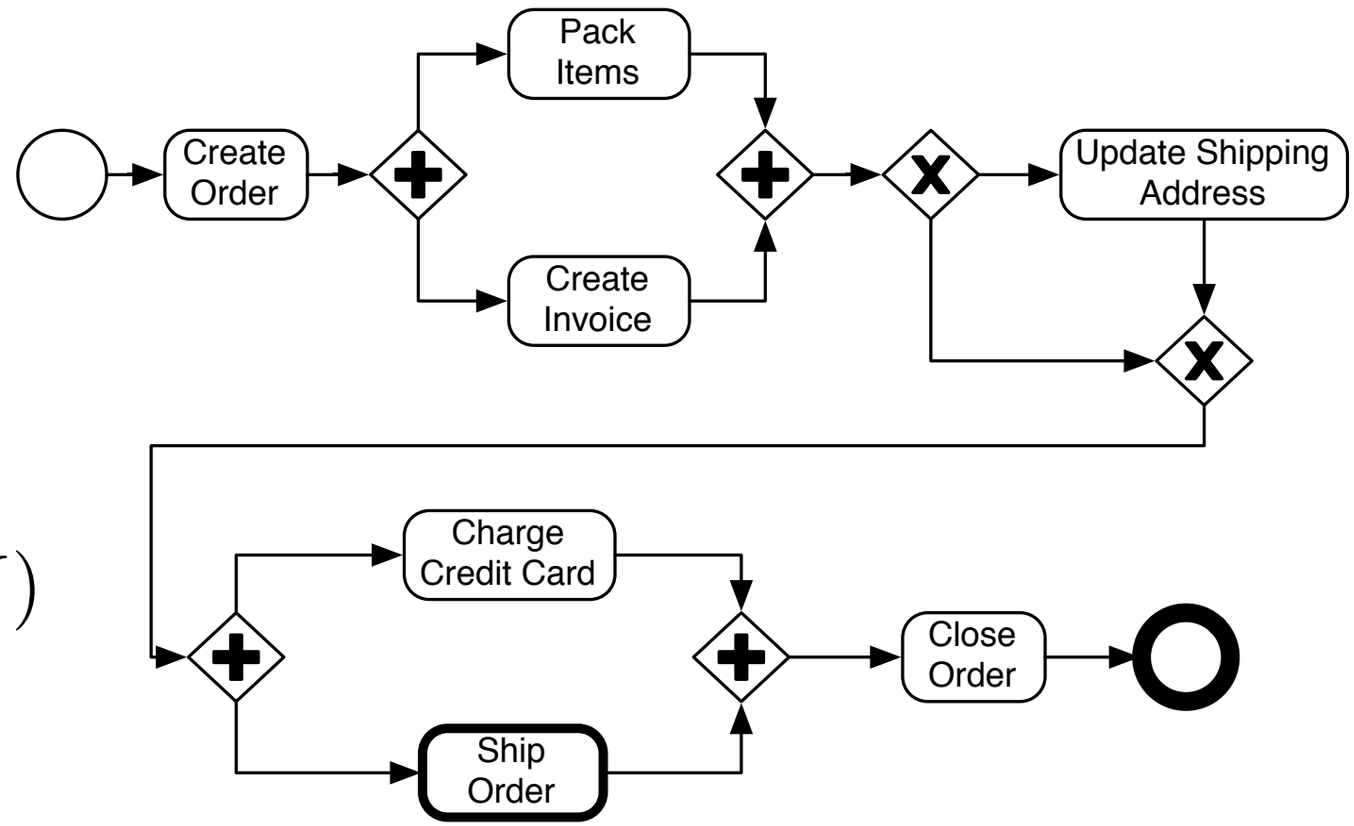
Invariant:
When the order is shipped,
the process instance must
always be closed
(Related to weak soundness)

Invariant



$$\tau.\overline{ship}.\mathbf{0} + \tau.\overline{s}.\overline{ship}.\mathbf{0} \not\approx (BP \mid X)$$

Business Process



Example 3

Conclusions

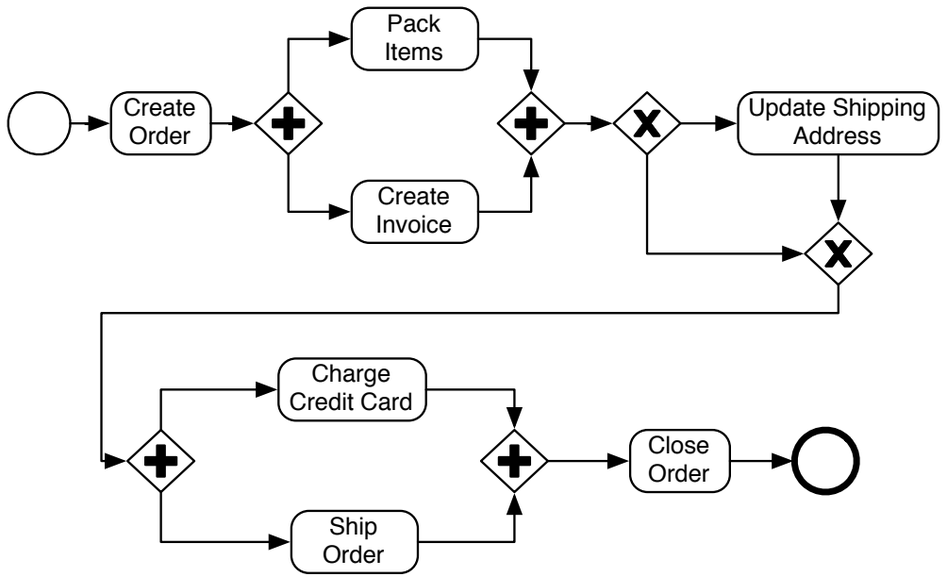
- Formal representation of invariants in terms of transition systems
- Application of simulation and bisimulation techniques to verify process models according to invariants
- The paper contains algorithms and extended invariants given in the pi-calculus
- Prototypical implementations available

But...

- Simulation and bisimulation have high computational efforts
- Works currently only for small process models
- Advanced invariants require link passing mobility for correlations
- Tool support is poor

Thank you!

```
agent N1=(^e771)(^e765)(^e772)(^e762)(^e753)(^e747)(^e764)(^e763)(^e746)(^e745)(^e731)
(^e730)(^e727)(^e729)(^e728)(^e773)(^e671)(N1_766(e771,e765,e772) | N1_754(e762,e753,e747)
| N1_748(e762,e764,e763) | N1_737(e747,e746,e745) | N1_732(e764,e765) |
N1_722(e731,e730,e745) | N1_717(e727,e729,e728) | N1_682(e729,e731) | N1_681(e728,e730) |
N1_538(e773) | N1_680(e772,e773) | N1_679(e763,e771) | N1_678(e746,e753) |
N1_677(e671,e727) | N1_534(e671))
agent N1_766(e771,e765,e772)=e771.e765.t.'e772.0
agent N1_754(e762,e753,e747)=(e753.N1_754_1(e762,e753,e747) +
e747.N1_754_1(e762,e753,e747))
agent N1_754_1(e762,e753,e747)=t.'e762.0
agent N1_748(e762,e764,e763)=e762.t.('e764.0 | 'e763.0)
agent N1_737(e747,e746,e745)=e745.N1_737_1(e747,e746,e745)
agent N1_737_1(e747,e746,e745)=(t.'e747.0 + t.'e746.0)
agent N1_732(e764,e765)=e764.t.'e765.0
agent N1_722(e731,e730,e745)=e731.e730.t.'e745.0
agent N1_717(e727,e729,e728)=e727.t.('e729.0 | 'e728.0)
agent N1_682(e729,e731)=e729.t.'e731.0
agent N1_681(e728,e730)=e728.t.'e730.0
agent N1_538(e773)=e773.t.0
agent N1_680(e772,e773)=e772.t.'e773.0
agent N1_679(e763,e771)=e763.t.'e771.0
agent N1_678(e746,e753)=e746.t.'e753.0
agent N1_677(e671,e727)=e671.t.'e727.0
agent N1_534(e671)=t.'e671.0
```



The Formalized Business Process