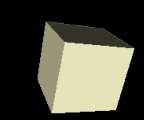


Software agent computing

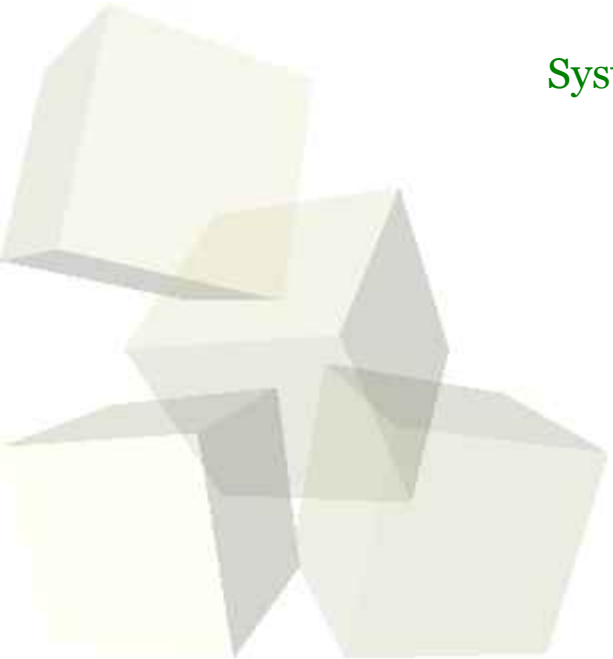
Homeworks
at Warsaw University of Technology

Maciej Gawinecki

Systems Research Institute, Polish Academy of Sciences

maciej.gawinecki@ibspan.waw.pl

http://www.ibspan.waw.pl/~gawinec

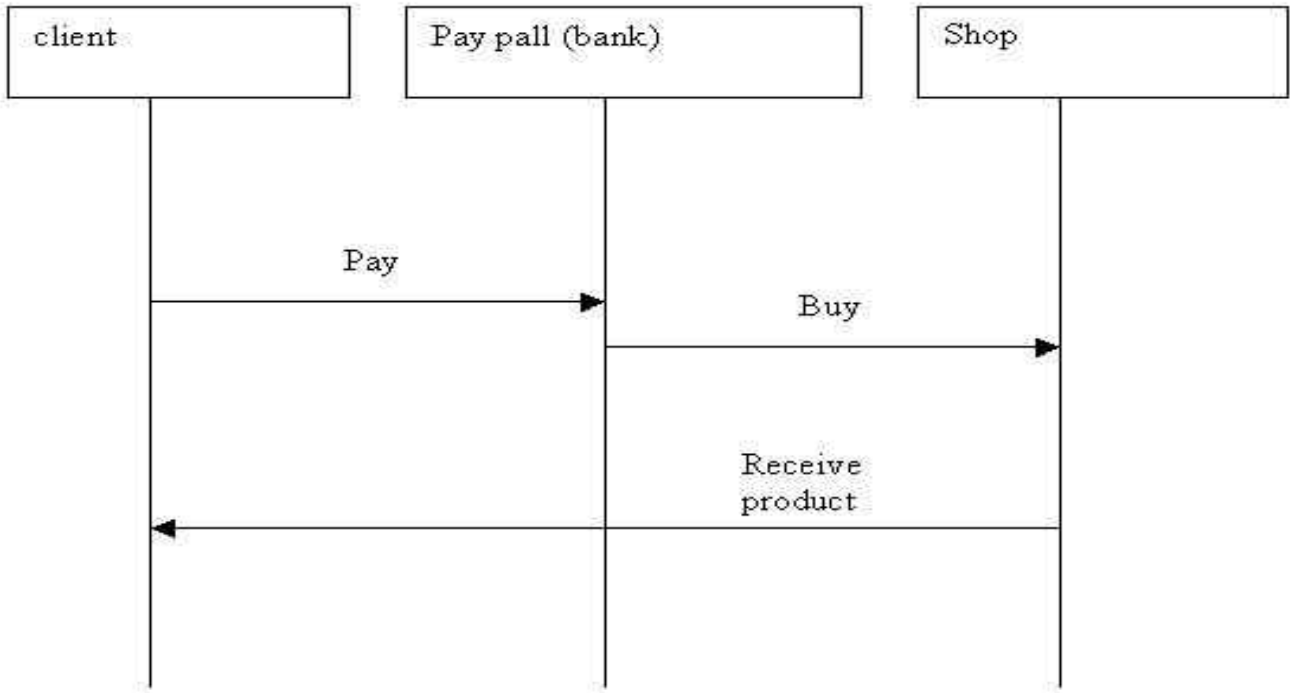


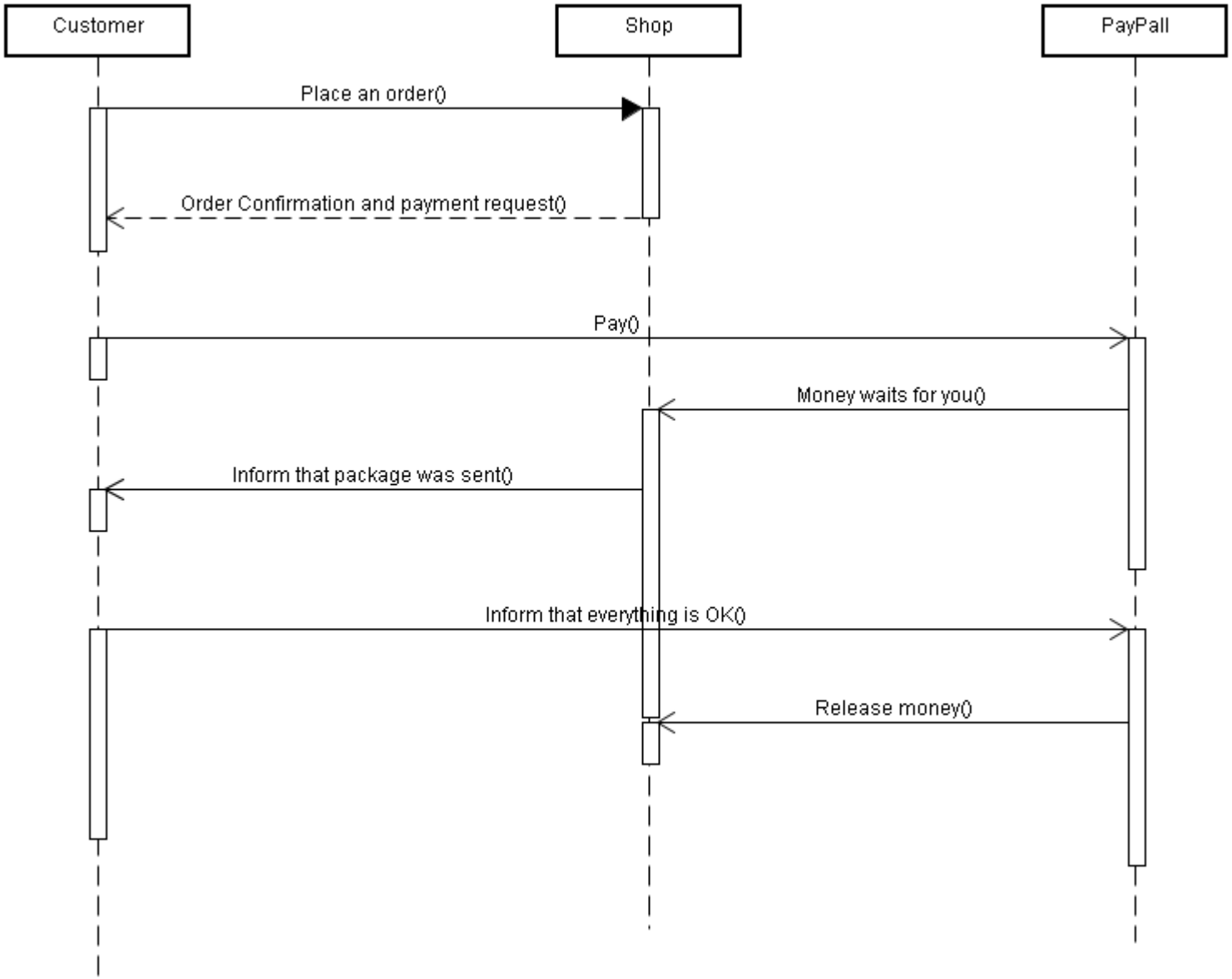
Aim

- Task:
  - Design protocol of eastablishing transaction between two agents:
    - Client
    - Shop
  - Third entity (Bank / CashHolder / PayPal) secures transaction sides.
  - Protocol should be secure for both: Client and Shop
- Conditions:
  - Individual work
- Outcome
  - Sequence diagram describing protocol
- Deadline: 10.04.2007

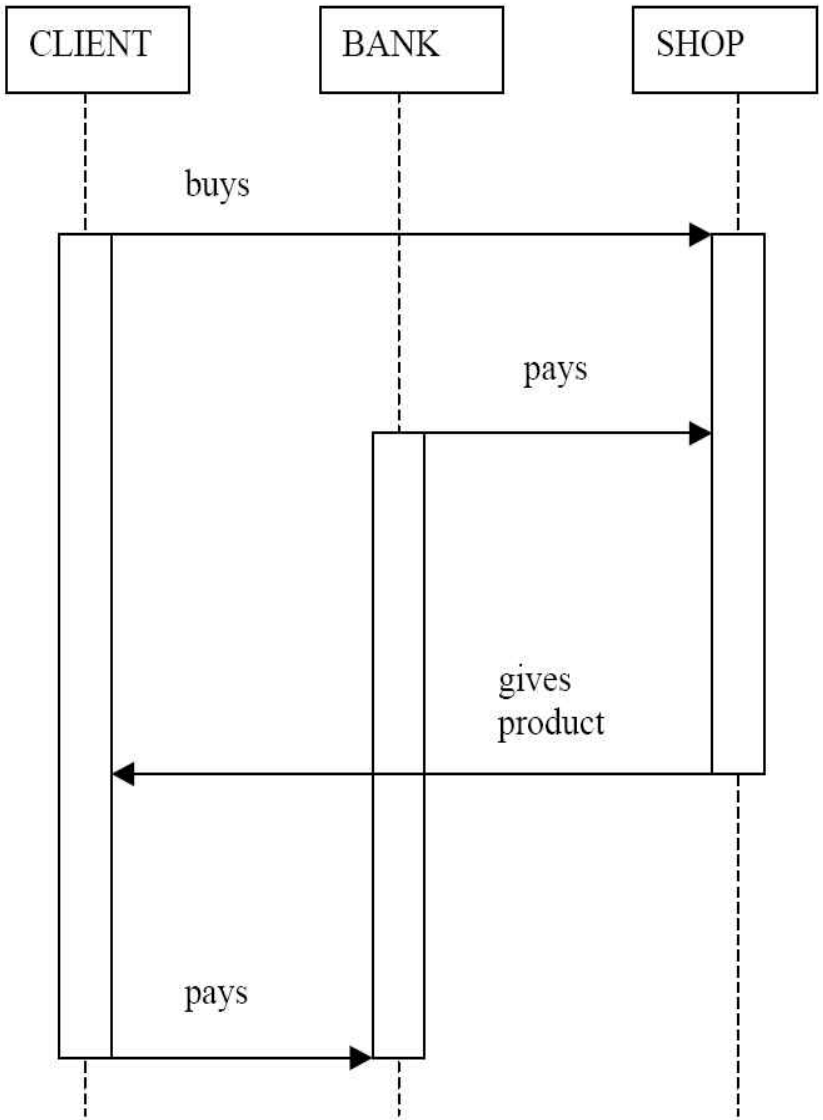
Why this is so important ?

- Example
  - Deployment of ATM in USA
- Documentation
  - UML's Sequence Diagram
    http://www-128.ibm.com/developerworks/rational/library/3101.html
  - Interaction Protocols Specifications
    http://www.fipa.org/repository/ips.php3

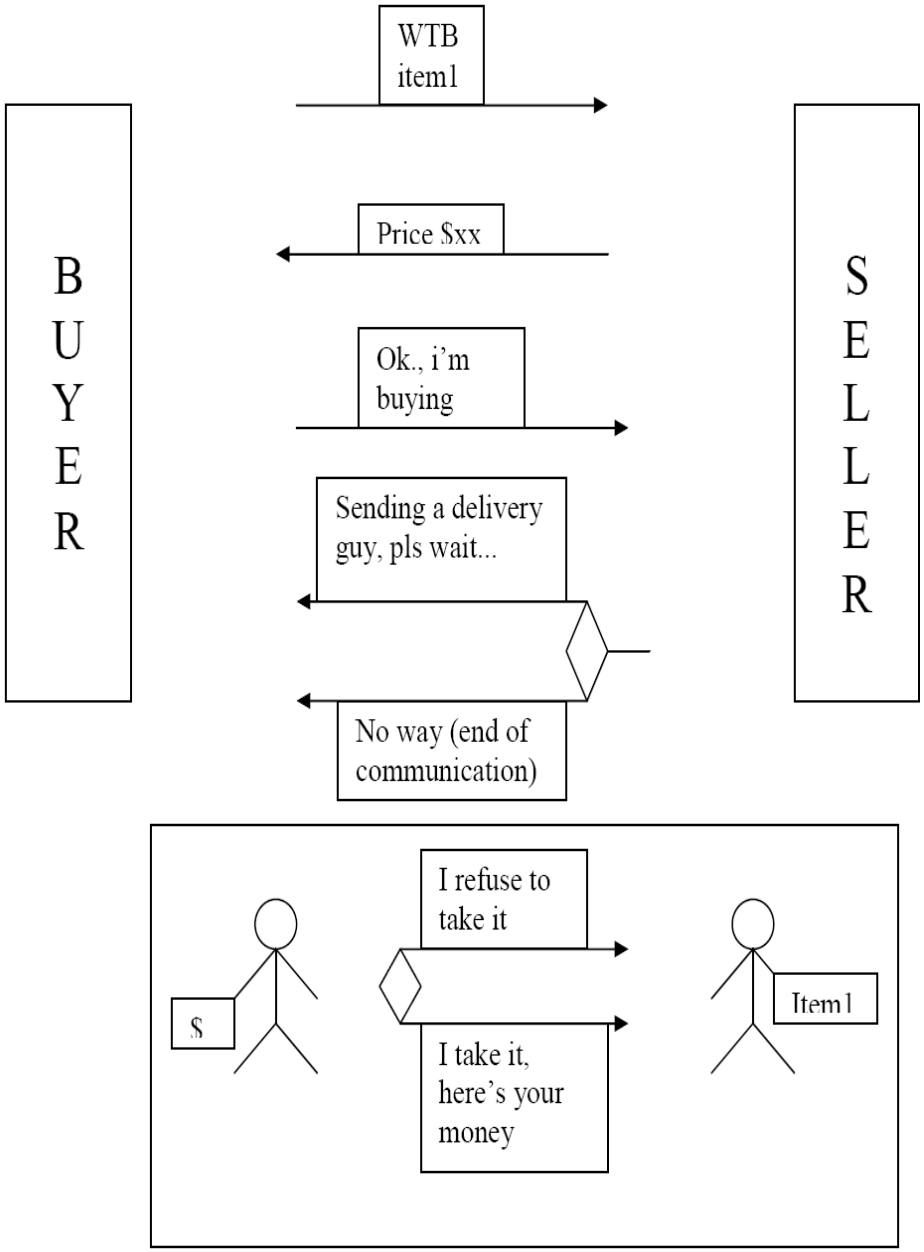

client
Pay pall (bank)
Shop
Pay
Buy
Receive
product

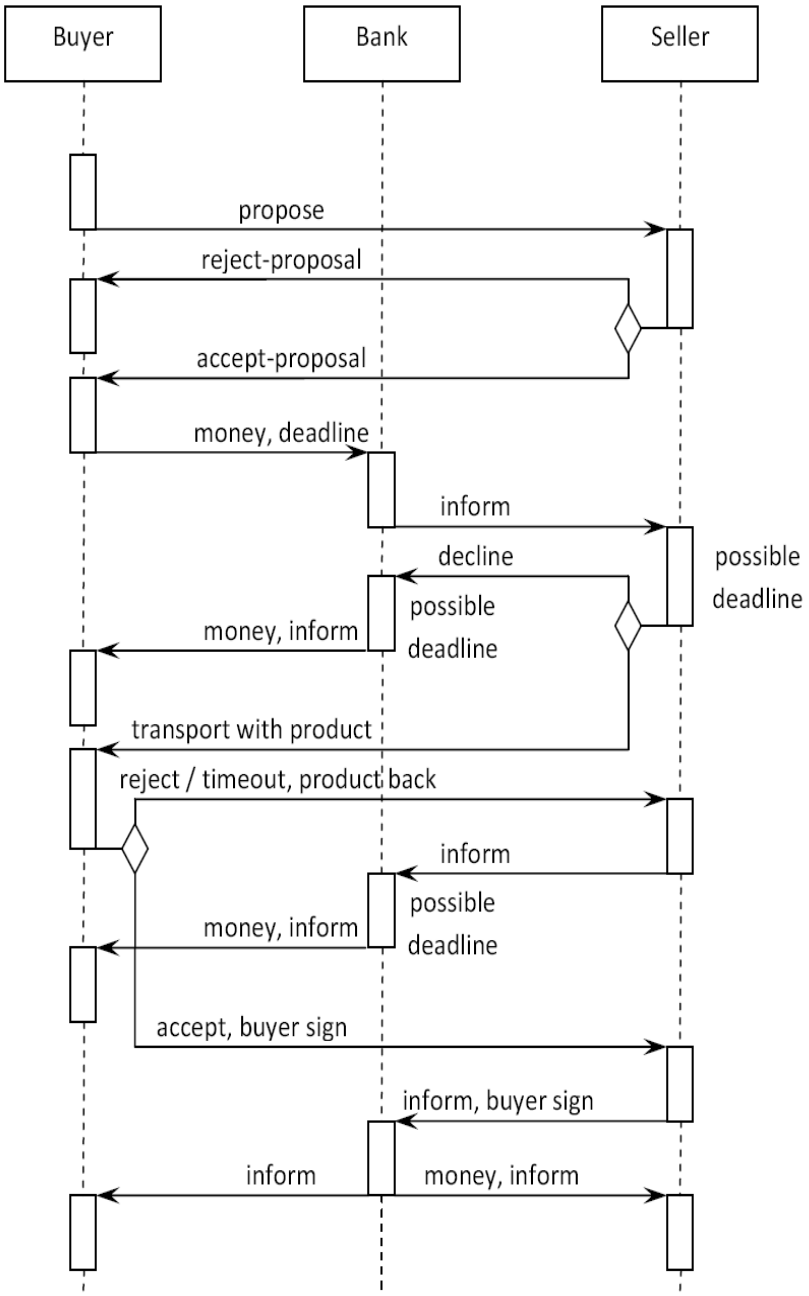

Customer
Shop
PayPall
Place an order()
Order Confirmation and payment request()
Pay()
Money waits for you()
Inform that package was sent()
Inform that everything is OK()
Release money()

sequence diagram



Buyer – Seller (cash on delivery, COD)





Notes / assumptions:

· Bank is trusted.

· Product checking operation is short.

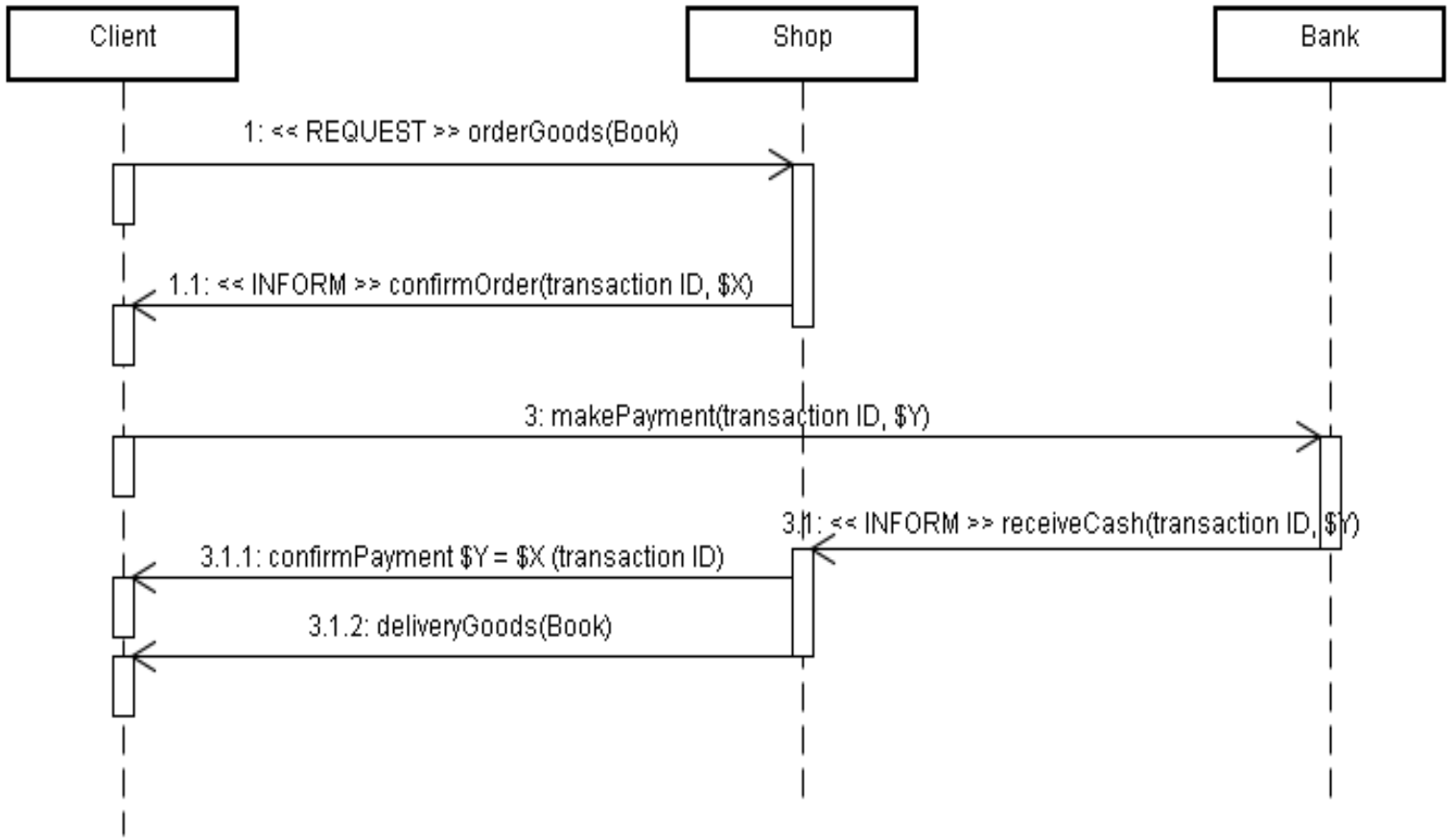
· Both buyer and seller know the deadline of transaction and have to accept or reject it.

· Seller has to deliver product and get buyer's sign before deadline or money is returned to buyer.

· Bank knows the deadline and when no buyer sign is delivered, or seller denial is received, money is returned to buyer.

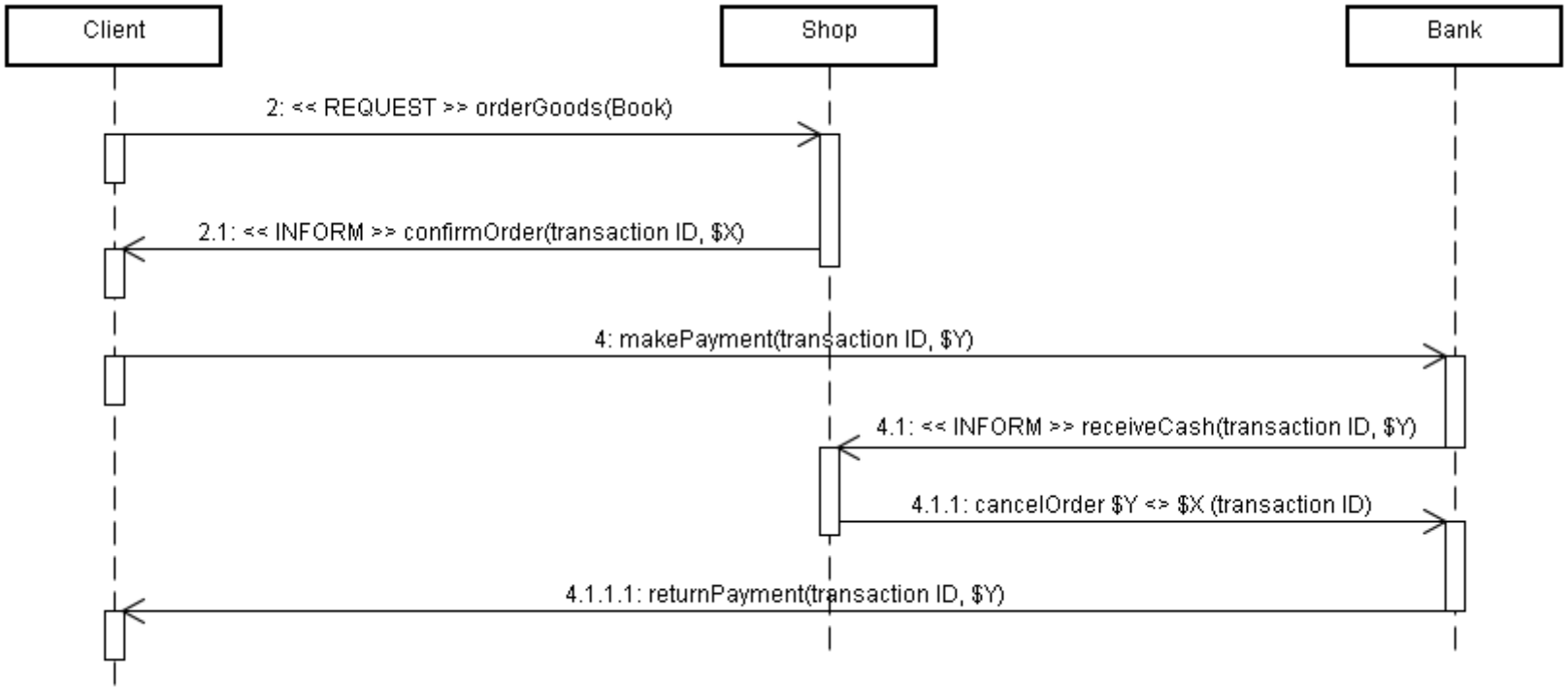
· Product check at deadline is understood as fault of seller. This problem could be solved by making buyer and
transport guy call bank to check deadline or extend it by short time, but it makes virtually no sense.

· Another approach would be using bank with storage, which would wait for both money and product, check if everything is ok and forward money / product or return them. Again, it's too unrealistic.



order is performed:
transaction ID - order transaction identifier
$X - cost of order
$Y - how much client have paid

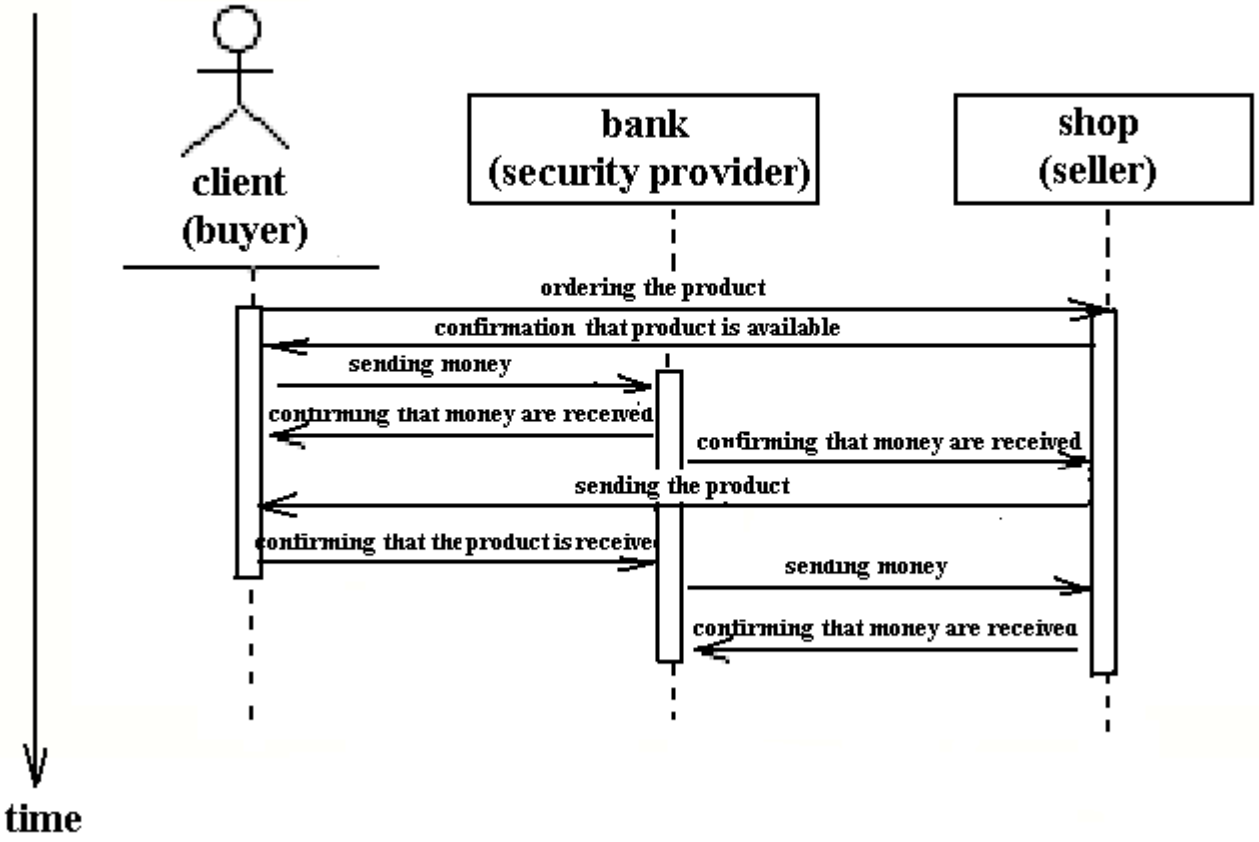
Paweł Olesiuk (cancel)



order is cancelled:
transaction ID - order transaction identifier
$X - cost of order
$Y - how much client have paid

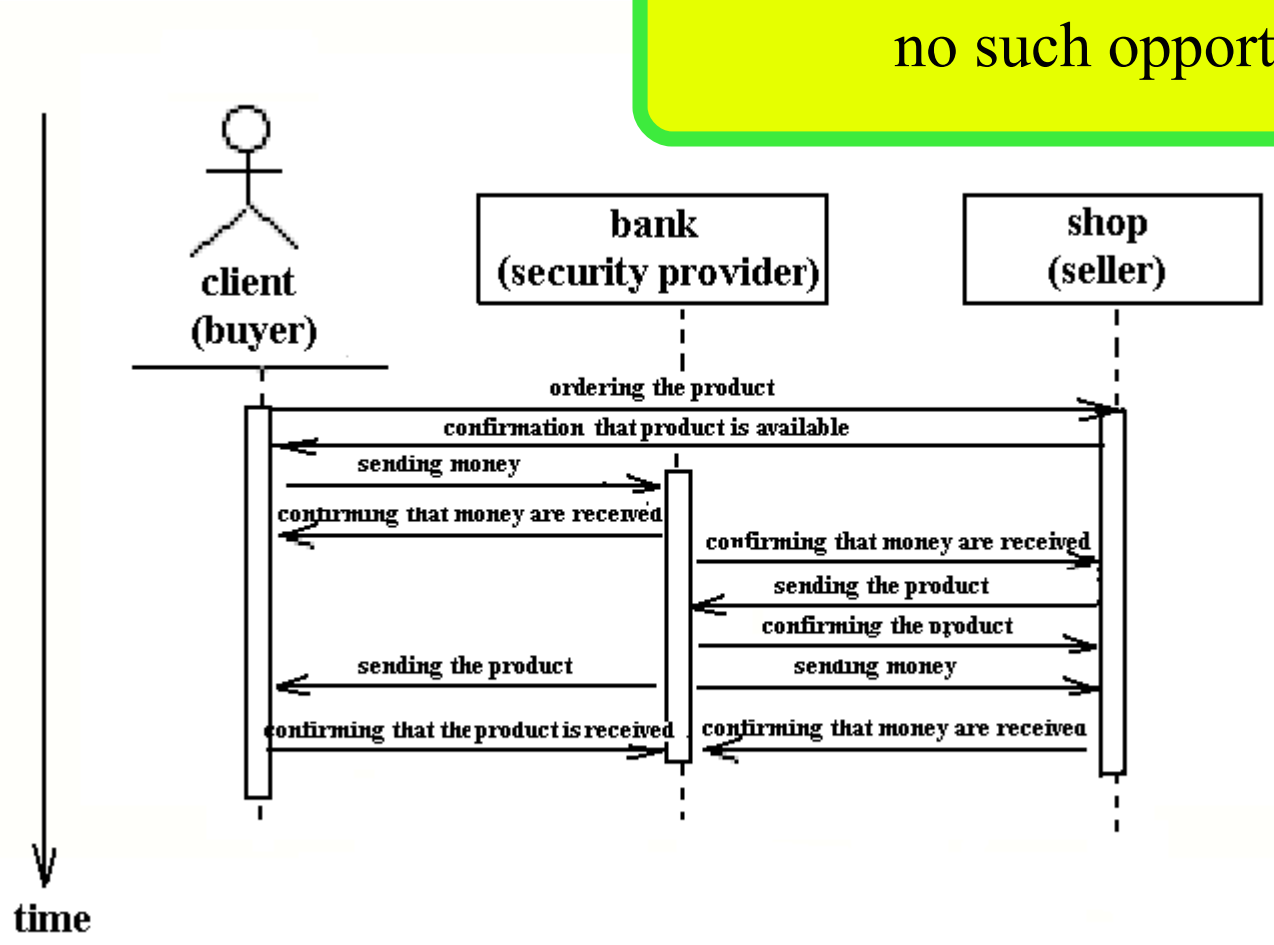
Kate Slezavina (1)

the Bank has the ability to store products as well as money



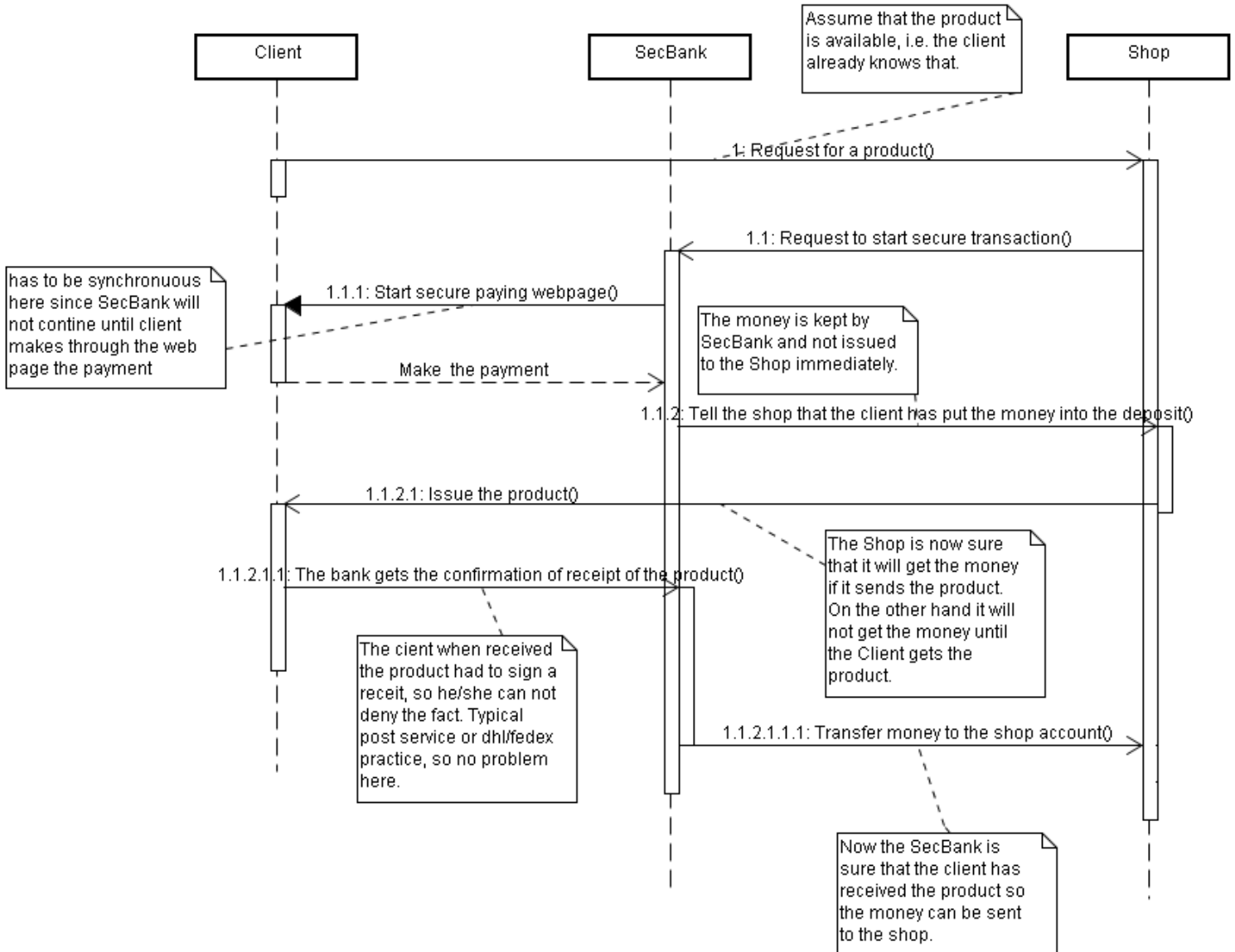
Kate Slezavina (2)

no such opportunity



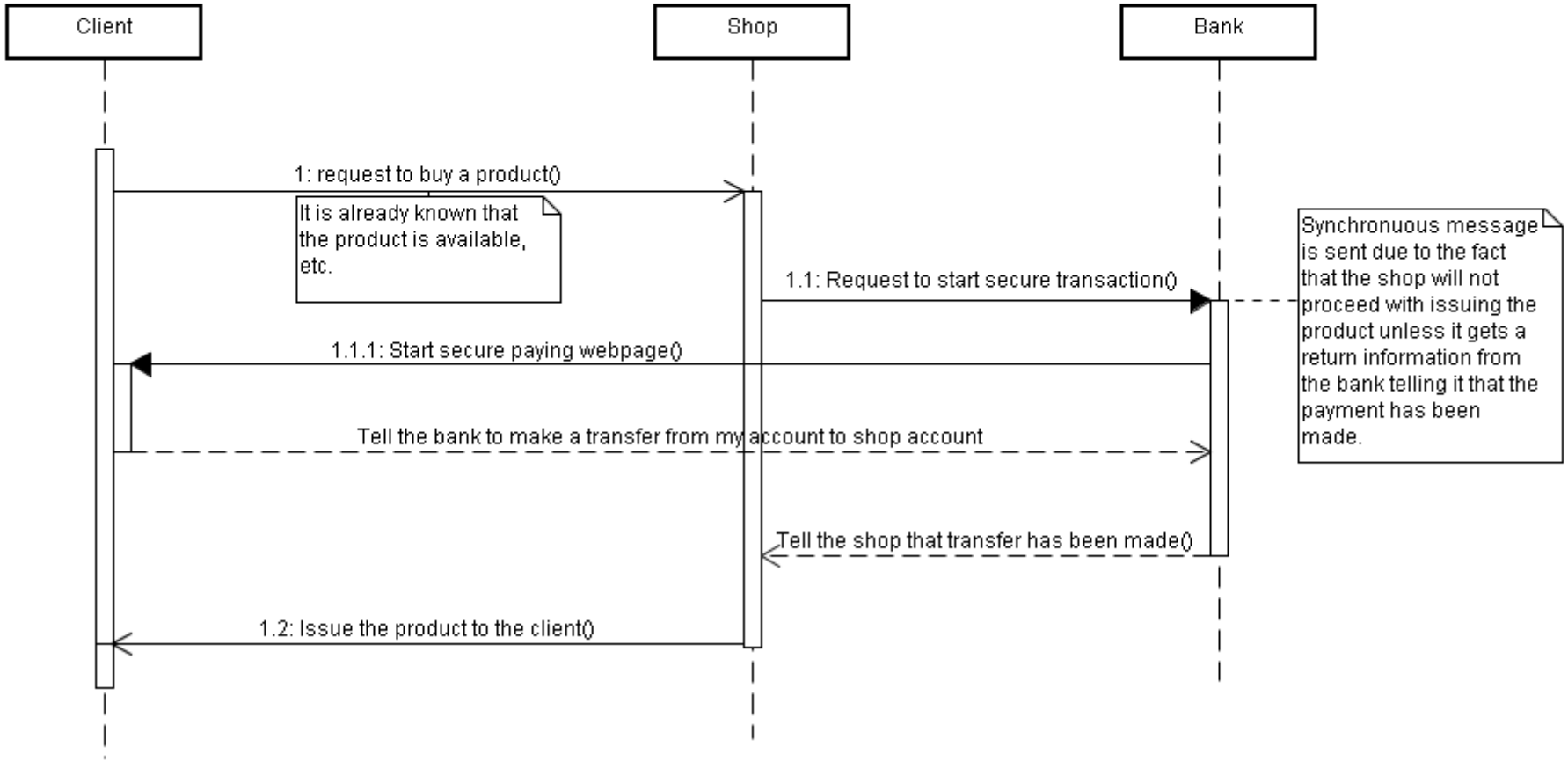
Sequence diagram for a transaction that is secure for both sides. The client is sure that he/she will receive the prodcut. The shop is sure that it will be paid. The third party, called it SecBank, assures that the following condition is satisfied.

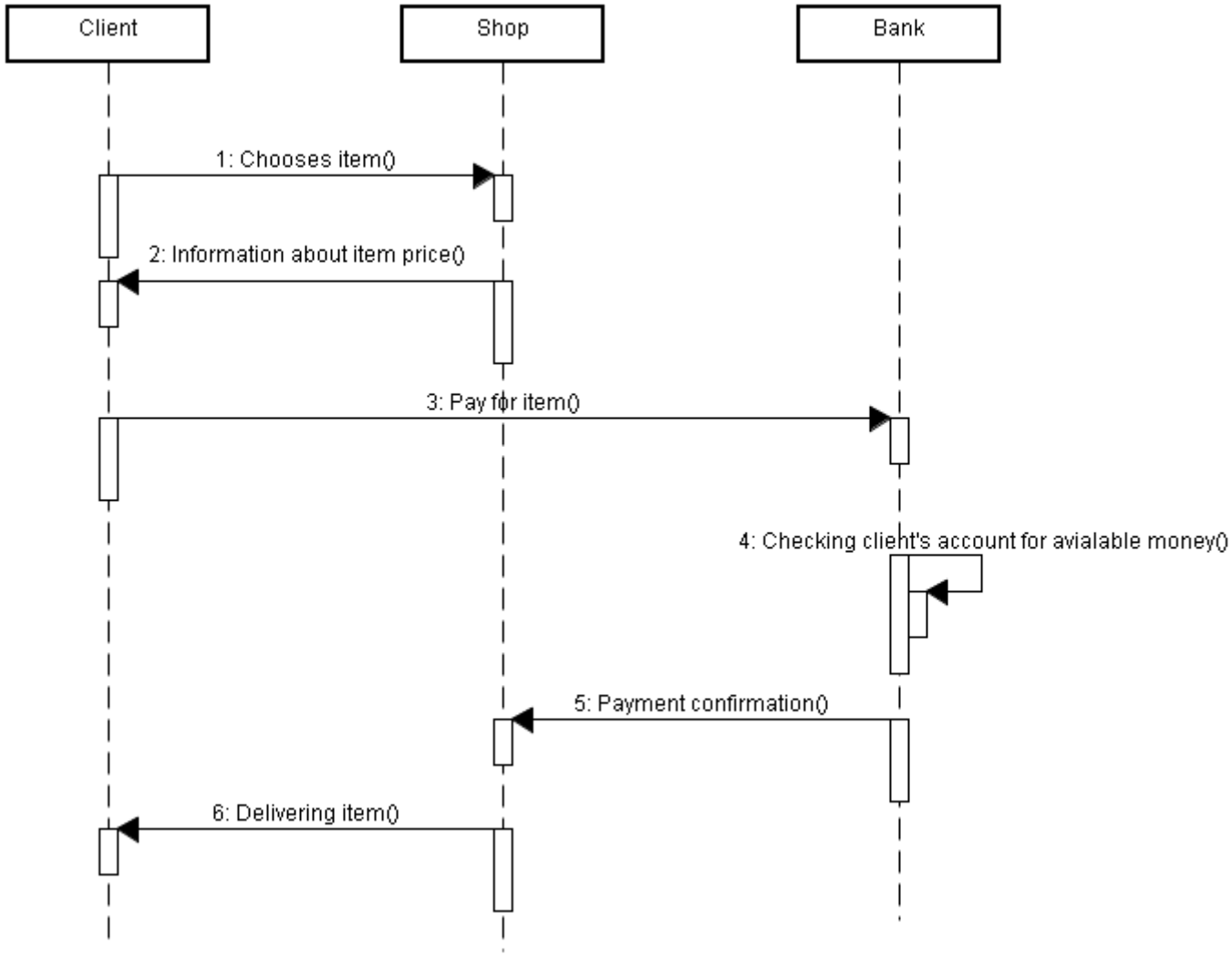
Both client and shop are secured during transaction.

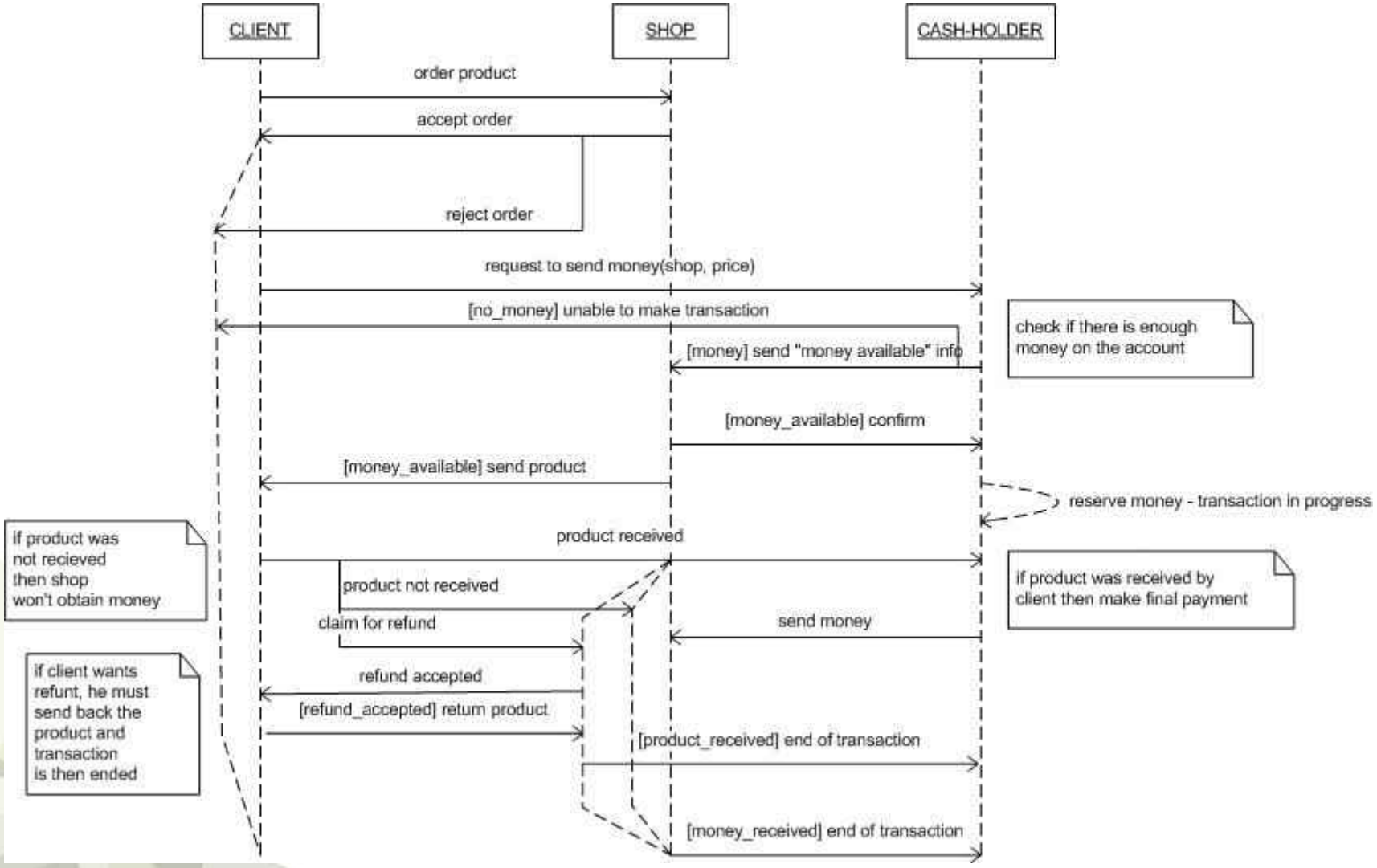


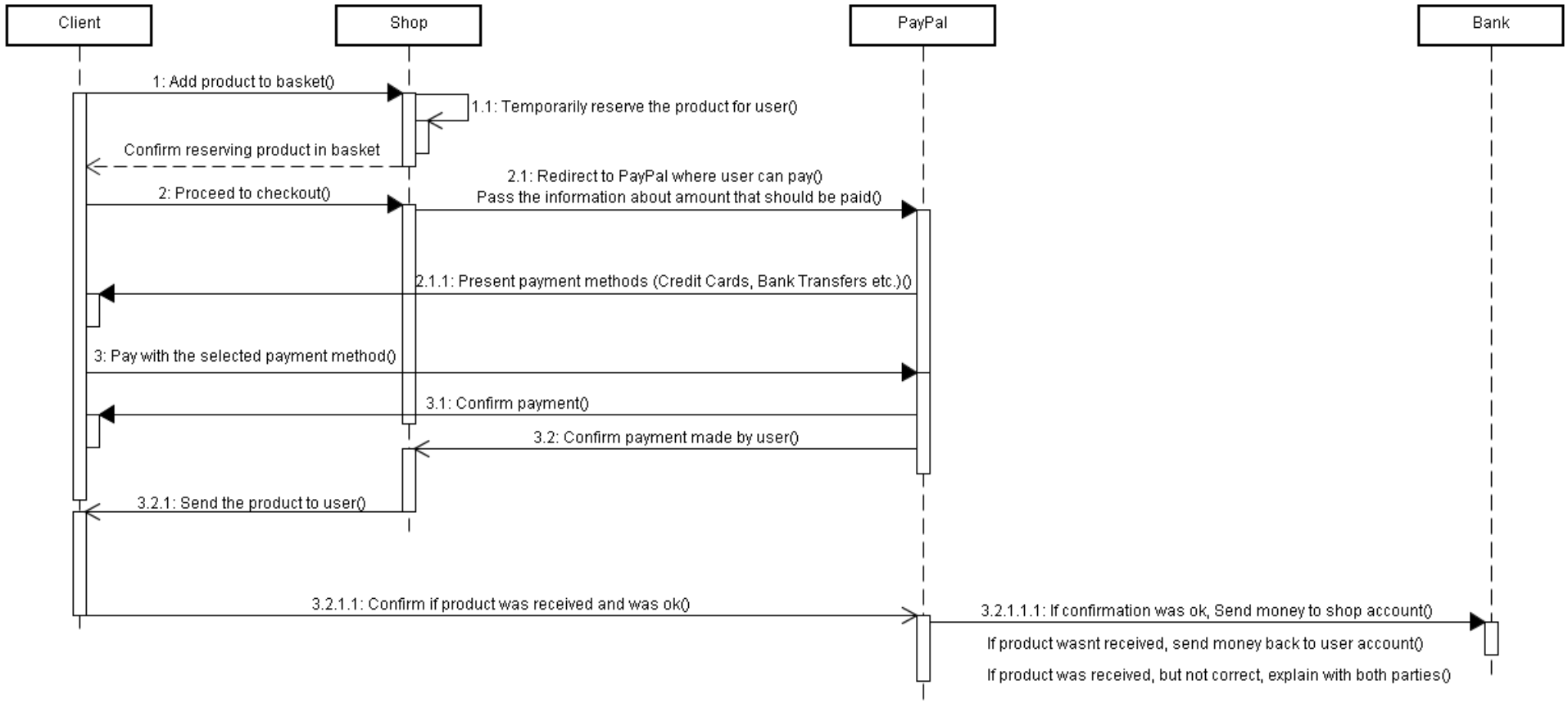
The sequence diagram for real life online shopping experience where the client first pays the money and after it the shop issues him/her the product. It is secure only for the shop. The client has no guarantee of receiving the product.

Like buying in Merlin on-line bookstore.




Client
Shop
Bank
1: Chooses item()
2: Information about item price()
3: Pay for item()
4: Checking client's account for avialable money()
5: Payment confirmation()
6: Delivering item()


CLIENT
SHOP
CASH-HOLDER
order product
accept order
reject order
request to send money(shop, price)
[no_money] unable to make transaction
[money] send "money available" info
check if there is enough money on the account
[money_available] confirm
[money_available] send product
reserve money - transaction in progress
product received
if product was not recieved then shop won't obtain money
product not received
claim for refund
send money
if product was received by client then make final payment
if client wants refunt, he must send back the product and transaction is then ended
refund accepted
[refund_accepted] return product
[product_received] end of transaction
[money_received] end of transaction



I wasn't able to use Interaction Operators (to be more specific 'alt' - alternative operator - I made a proper notes below message arrow)

Homework #1

- Task:
  - Design and implement transaction protocol
- Conditions:
  - Individual work
- Outcomes
  - Sequence diagram describing protocol
  - Ontology designed in Protege
  - Implemented JADE Agents:
    - Agents: BankAgent, ClientAgent, ShopAgent
    - Protocol implemented with use of FSMBehaviour class.
- Deadline: 23.04.2007