

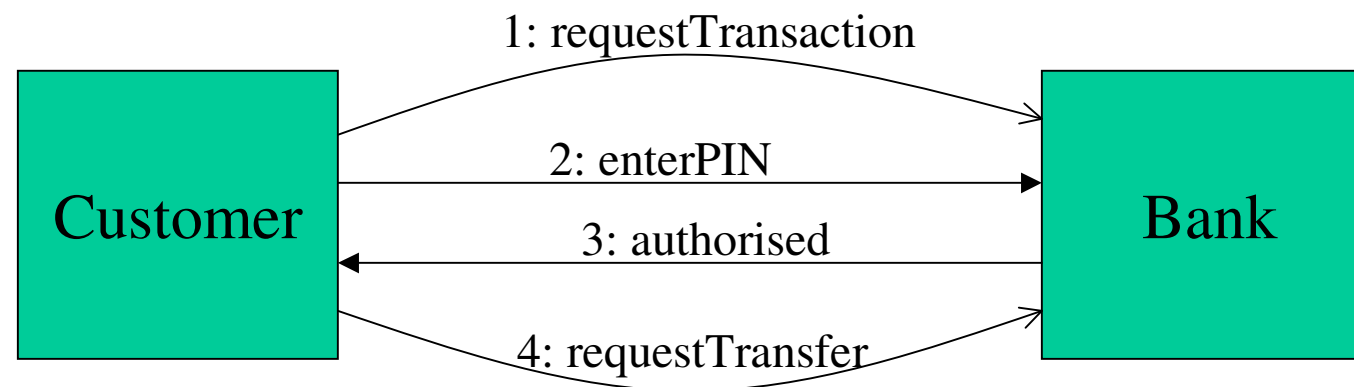
Web Services:  
From Conversation to Protocols to  
Composition

Bao Quoc Vo  
Intelligent Agent Groups  
RMIT University

# Conversation

- A Web service conversation involves maintaining some state during an interaction that involves multiple messages and/or participants.
  - From [Web Services Glossary \(2004-02-11\)](#)

# Conversations and protocols



- complements interface definition
- **protocols, choreography, conversation, orchestration, ...**
- Web services are loosely coupled => compatible interaction protocols?
- Deadlock/Unspecified inputs/Termination?

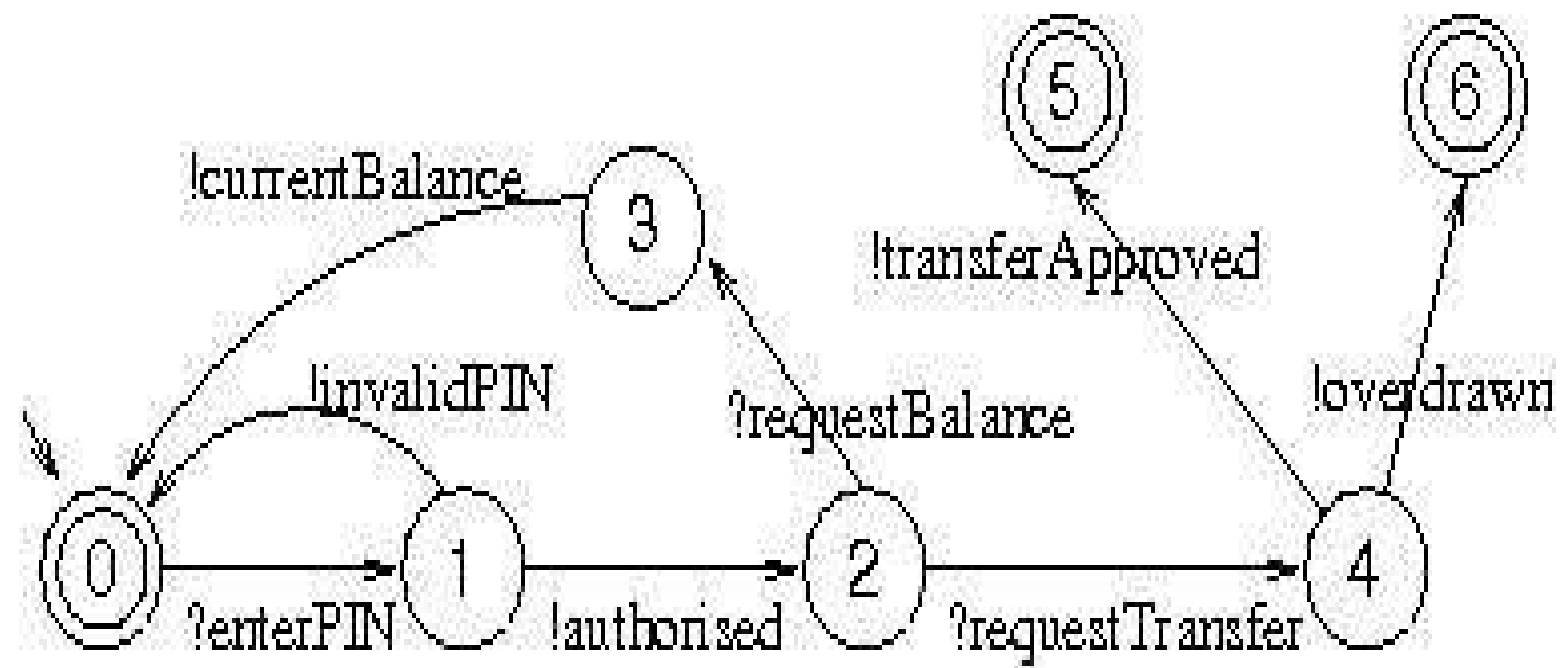
# Protocol models

- Perspective
  - Global (cooperative) vs. service-centric
  - The latter usually requires less constraints on protocol compatibility
- Formalism
  - Sequence diagrams
  - **FSMs**/statecharts
  - Process-based models
- Expressive power
  - Conditions, State management, Correlation, Transactions, Modularization, Exception handling, etc.

# Example

```
Service Banking {
  Interface {
    RECEIVE enterPIN(Account acc, EncryptedPIN PIN);
    RECEIVE requestTransfer(Account toAcc, float amount);
    RECEIVE requestBalance();
    SEND invalidPIN();
    SEND authorised();
    SEND overdrawn();
    SEND transactionApproved();
    SEND currentBalance(float balance); };
  Protocol {
    States { 0(init,final), 1, 2, 3, 4, 5(final), 6(final) };
    Transitions {
      0 : ?enterPIN      -> 1;
      1 : !invalidPIN    -> 0;
      1 : !authorised    -> 2;
      2 : ?requestTransfer -> 4;
      2 : ?requestBalance -> 3;
      3 : !currentBalance -> 0;
      4 : !transferApproved -> 5;
      4 : !overdrawn     -> 6; };
    };
};
```

# Example



# Service Composition

- A **composition system** consists of the external schema (i.e., messages and interaction protocol) of the **target service** plus the external schemata of the available component services.
- The **composition problem**: constructing the internal model of (the external schema of) the target service so that it employs only messages of the component services and satisfies the interaction protocols.
- **Well-defined compositions**: free of deadlocks and unspecified receptions.

## Asynchronous communication

- Under a synchronous semantics, the finite-state machines describing the protocols of the component services and the internal model of the composite service are required to advance atomically.
- To allow asynchronous communication, FIFO channels containing messages exchanged between different components of the system are introduced.

## Synthesising composition

- **Ontological reasoning**: mapping between component interfaces
- Constructing **execution tree**: exploring all possible execution traces given the finite state machines of the component services and the target service
- **error** nodes on the execution trees: corresponding to states in which a deadlock or unspecified reception might arise.