Web services UML modeling
Part II: Web services behavioral modeling

1st step: Model the system using UML.
2nd step: Annotate performance characteristics using the SPT Profile.
3rd step: Translate the UML-SPT diagrams into Petri nets.
4th step: Analyze the Petri nets to obtain performance results.
UML diagrams in our proposal

- UML use cases
- UML sequence diagram
- UML statecharts
- UML activity diagrams
- UML deployment diagram
- Version UML 1.4
The evolution of UML (from Bran Selic’s WOSP 2002 Tutorial)

**UML 1.1 (OMG Standard)**
- Rumbaugh
- Booch
- Jacobson
- Foundations of OO (Meyer, Stroustrup, Harel, Wirfs-Brock, Reenskaug,...)

**UML 1.3 (extensibility)**
- 1998

**UML 1.4 (action semantics)**
- 1997

**UML 1.4.1**
- 2001

**UML 2.0 (MDA)**
- 2002
- 2003-2004
Case study: POP3 protocol

- UML models of a basic mail client.
- Classes: server host, client host and user.
- Check mail feature:
  - Client establish TCP connection with the server.
  - Authorization phase (username and password).
  - Transaction phase (LIST command): download mails (text and with attach).
  - QUIT ends the interaction.
Use case diagram

- Scenario modeling technique. Introduced by Jacobson.
- A scenario models how a system behaves (works), not how to implement it.
- Each scenario is described with a sequence of actions.
- A means to understand the system: developers, final users and domain experts.
- **Elements**: actors, ellipses and relationships.
Use case diagram: Relationships

- **Communication.**
- **Generalization.**
- **Inclusion.**
  - The *base* use case **explicitly** incorporates the behavior of the *provider* use case.
- **Extension.**
  - The *base* use case **implicitly** incorporates the behavior of the *provider* use case.
  - Optional behavior.
Sequence diagram

- A SD models a “system interaction” as a set of messages exchanged among classes.
- Vertical dimension: Time; Horizontal dimension: Classes

Our proposal:
- Describe a use case.
- Useful to compute Response Time.
- Objects can reside in the same machine or in different ones.
  - Transmission time.
  - Message size.
- Probabilities as routing rates.
Sequence diagram (2)

- Check mail use case.
Clinical Decision Support System (CDSS) seq. diagram.
Electronic Patient Record (EPR).
Statechart diagram

- Model a class of the problem domain.
- Event-driven behavior: Reactive objects.
- Statechart basic modeling elements:
  - Initial pseudo-state.
  - Final state.
  - Simple state.
  - Transitions: event/class.event.
  - Activities in states.
- KEY: Communication among SC via events.
- KEY: Consistency with the SD.

User statechart.

- psClient
  - /CH.check_mail
  - UserMainState
    - DO: Thinking
    - /CH.exit_exec
- fsClient
Statechart diagram (2)

Client Host statechart.

- **psMClient**
  - check_mail
  - exit_exec

- **fsMClient**
  - text_message
  - attach_message

**Waiting4Entry**
- entry: SH.open_tcp_connection

**Authentication**
- entry: SH.username

**Quit**
- entry: SH.quit

**CheckPassword**
- entry: SH.password

**RetrieveMessage**
- entry: SH.retr

**DeleteMessage**
- entry: SH.dele

**Greetings**
- ok
- not new

**CheckMessages**
- ok
- new

**Greeting**
- ok
- messages_left
- not messages_left

**Retrieving**
- ok
- not new

**Quit**
- err
- ok
- not ok
Statechart diagram (3)

Server Host statechart.

- psPOP3Server
  - open_tcp_connection
    - / CH.greeting
  - Listening on TCP port 110
    - / CH.ok
  - Authorization
    - DO: Authorization
    - / CH.ok
  - Transaction
    - dele / CH.ok
    - / CH.text-_message
    - / CH.attach-_message
    - Sending
      - DO: read_message
      - / CH.ok
      - retr

DO:
- Authorization
- unlock_maildrop
- read_message
Detailed description of a statechart behavior: Execution model.

Statechart diagram (4)
Activity diagram

What happens with the *authorization activity* in the SH statechart?
Model it in great detail to get an accurate view of the problem.
Activity diagram is commonly used to *workflow* modeling → not in our approach.
We use the AD to *model activities in SCs*.

Special case of statechart, where:

- Most of the transitions are fired when the action/activity in the state ends → *no reactive behaviour*.
- Most of the states are action states.
- Automated transition firing.
Activity diagram (2)
Activity diagram (3)

- **Action states**
  - No time spent in computation.

- **Activity state**
  - Spent computation or waiting time.
  - Can be decomposed in others → may modeled with another AD
  - With entry and/or exit.

- **Transition**
  - Automatic firing

- **Bifurcations**
  - Alternative paths.

- **Object flow**
  - Objects involved in the diagram (create, destruct or modify).
Activity diagram (4)

Fork and Join

- Concurrent paths
- Join: each path waits until completions of all of them

Figura 19.6. División y unión.
Activity diagram (5)

Swimlanes

Pasajero

- Solicitar pasaje
- Selecionar vuelo
- Pagar pasaje

Vendedor

- Verificar existencia vuelo
- Informar alternativas y precios
- Solicitar pago
- Reservar plazas

Línea Aerea

- Dar detalles vuelo
- Confirmar plaza reservada

Dr. José Merseguer
Dpt. Informática e Ing. Sistemas (Univ. Zaragoza)
Deployment diagram

- Model the deployment of the software components in the hardware platform.
- Software component: configuration files, programs, tables in a DB, operating system, …
- Nodes: represent the hardware resources (CPUs, disks, printers, LAN, Internet, …)

Diagram:

- Name
- Service-1
- Service-2

Component

Nodo
Deployment diagram: Relationships

**Dependency**

Component-1

Component-2

**Communication**

Nodo <<Internet>> Nodo
Deployment diagram

POP3 deployment.

Deployment diagram:

- :clientHost
  - m:mailClient
  - :Internet

- :serverHost
  - s:ServerHost

Dr. José Merseguer
Dpt. Informática e Ing. Sistemas (Univ. Zaragoza)