Specifying Operations

Why operations are specified

Algorithmic methods

Non-algorithmic methods

The role of operation specifications

- Analysis perspective:
  - ✓ An operation specification is created at the point when the analyst's understanding of some aspect of an application domain can be fed back to users, ensuring that the proposals meet users' needs
  - ✓ Users must confirm the logic, or rules, of the behaviour
- From a design perspective an operation specification is a framework for a more detailed design specification, that later guides a programmer to a method.
- The designer and the programmer responsible for the class will be the main users of the specification, as they need to know what an operation is intended to do.
- Designers and programmers of other parts of the system also need to know about its effects on the other classes.
- Defining Operations should neither be begun too early, nor left too late

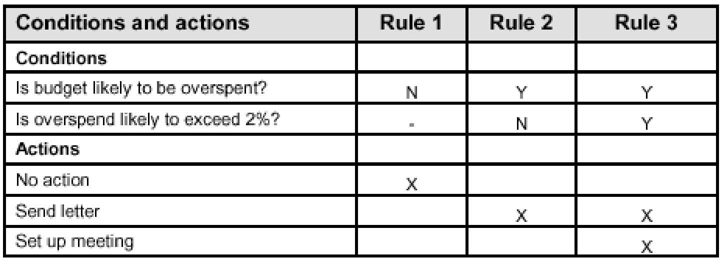
Specification by Contract

- Operations are defined primarily in terms of the services they deliver, and the ‘payment’ they receive
- We identify the nature of the service provided by the server object, and what must be provided by the client object in order to obtain the service
  - ✓ The intent or purpose of the operation
  - ✓ The return type
  - ✓ An appropriate description of the logic
  - ✓ Other operations called, whether in the same object or in other objects
  - ✓ Events transmitted to other objects
  - ✓ Attributes set during the operation’s execution
  - ✓ The response to exceptions
  - ✓ Any non-functional requirements that apply

Describing Operation Logic

- Non-algorithmic approaches: describing the logic of an operation as a black box
  - ✓ Decision table
  - ✓ Pre- and post conditions
- Algorithmic approaches: braking the internal logic of a process into small steps
  - ✓ Structured English
  - ✓ Pseudo-code
  - ✓ Activity Diagrams

Decision table



Conditions and actions	Rule 1	Rule 2	Rule 3
Conditions			
Is budget likely to be overspent?	N	Y	Y
Is overspend likely to exceed 2%?	-	N	Y
Actions			
No action	X		
Send letter		X	X
Set up meeting			X

Pre- and post conditions

- What conditions must be satisfied before an operation can take place?
- What are the conditions that can apply after an operation is completed?
  - ✓ Ex: `Campaign.assignStaff(creativeStaffObject.id)`
    - ✓ Pre-condition: *`creativeStaffObject`*`.id` is valid
    - ✓ Post-condition: a link is created between *`campaignObject`* and *`creativeStaffObject`*

Structured English

```
do while there are more staff in the list
      calculate staff bonus
      store bonus amount
      begin case
      case bonus > £250
              add name to 'star of the month' list
      case bonus < £25
               print warning letter
       end case
end do
```

Pseudo-code

- It differs from Structured english in that it is closer to the vocabulary and syntax of a specific programming language.
- Pseudo-code remains only a skeleton of a program, intended only to illustrate its logical structure without including full design and implementation details

Activity diagrams

- They are a part of UML notation set that can be used to specify the logic of procedurally complex operations
- We will discuss them later.

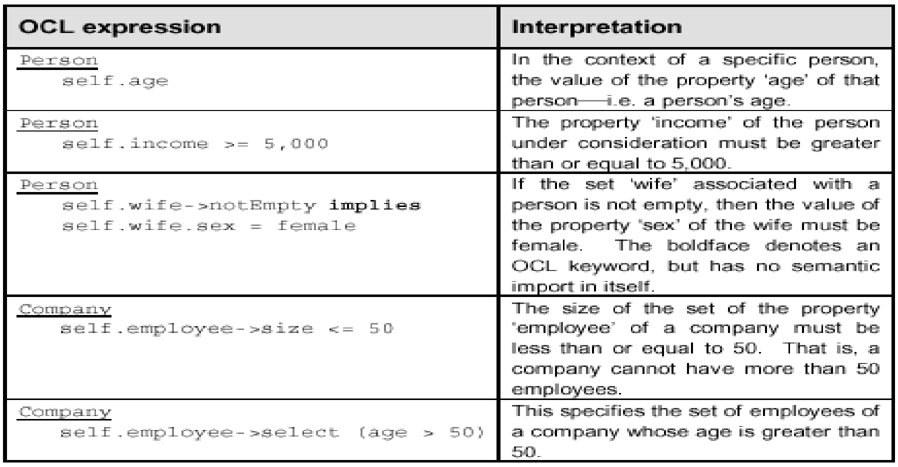
Object Constraint Language

- Some constraints can be adequately expressed in the graphical language (e.g., Multiplicity of an association).
- Some can not. For example, constraints within operation specifications (pre- and post-conditions)
- Where great precision is required, OCL provides a formal language
- OCL expressions are constructed from a collection of pre-defined elements and types
- The language has a precise grammar that enables the construction of unambiguous statements

OCL features

- A *context* that defines a domain within which the expression is valid (underlined for clarity).
- A *property* of that instance which is the context for the expression. Properties may include attributes, association-ends and query operations.
- An OCL *operation* that is applied to the property. Operations include the arithmetical operators (*,+,-,/) and logical operators (AND, OR, IMPLIES)

OCL features



OCL expression	Interpretation
Person self.age	In the context of a specific person, the value of the property 'age' of that person—i.e. a person's age.
Person self.income >= 5,000	The property 'income' of the person under consideration must be greater than or equal to 5,000.
Person self.wife->notEmpty **implies** self.wife.sex = female	If the set 'wife' associated with a person is not empty, then the value of the property 'sex' of the wife must be female. The boldface denotes an OCL keyword, but has no semantic import in itself.
Company self.employee->size <= 50	The size of the set of the property 'employee' of a company must be less than or equal to 50. That is, a company cannot have more than 50 employees.
Company self.employee->select (age > 50)	This specifies the set of employees of a company whose age is greater than 50.

Pre- and Post-conditions in OCL

```
Type::operation(parameter1:type, parameter2:type)return type
pre:     parameter1> …
         parameter2 = …
post:result= …
```

EX:

```
CreativeStaff::changeGrade(id:String, grade.id:String, gradeChangeDate:Date)
pre:     self.id->notEmpty
         self.grade.id->notEmpty
         self.gradeChangeDate >= today (assumes no retrospective
                                           changes)
post:    staffGrade[grade.id]->exists
         self.staffGrade->notEmpty
         self.staffGrade[grade.id].previousGrade->notEmpty
         self.staffGrade.gradeFinishDate = gradeChangeDate :Boolean
```

Additional Readings

- [Warmer99] Warmer, J. Kleppe, A. *The Object Constraint Language: Precise Modeling with UML* Addison-Wesley 1999
- [Yourdon89] Yourdon, E. *Modern Structured Analysis*. Prentice Hall.
- [Meyer97] Meyer, B. *Object Oriented Construction*. Prentice Hall.
- [Senn89] Senn, J. A. *Analysis & Design of Information Systems*. McGraw-Hill.