UML-based Specification Environment

Presented by Wuliang Sun
Department of Computer Science
Colorado State University
sunwl@cs.colostate.edu
Introduction to the USE Tool

- Reason about the structure and behavior of a system
  - Check if system “snapshots” (i.e., object models) satisfy the properties (i.e., OCL invariants)
  - Check if an operation invocation violates the operation specification (i.e., OCL pre-/postconditions)
Introduction to the USE Tool

Input: a USE specification consisting of a class model and OCL invariants, a set of USE commands for creating object models and operation invocations

Output: results showing
- If object models satisfy the OCL invariants
- If operation invocations satisfy the OCL pre-/postconditions
An Example of USE Specification

```
model Employee

-- classes

class Person
attributes
    name : String
    age : Integer
    salary : Real
end

class University
attributes
    name : String
    location : String
operations
    fire(p : Person)
    pre firePre: self.employee->includes(p)
    post firePost: self.employee->excludes(p)
end

class Student < Person
end

class Professor < Person
end

-- associations

association WorksFor between Person[*] role employee
    University[0..1] role employer
end

-- constraints

context Professor

inv HigherSalaryThanStudent:
    Student.allInstances()->forAll(s:Student|s.salary < self.salary)
```
Specifying Classes in USE

USE Specification

class Person
attributes
  name : String
  age : Integer
  salary : Real
end

class Student < Person
end

class Professor < Person
end

Class Diagram

Class

Person

name : String
age : Integer
salary : Real

Attribute

Generalization

Student

Professor
Specifying Associations in USE

USE Specification

association WorksFor between
  Person[*] role employee
  University[0..1] role employer
end

Class Diagram

Association

Multiplicity

Association

End
Specifying Operations in USE

USE Specification

class University

attributes
  name : String
  location : String

operations
  fire(p : Person)
  pre  firePre: self.employee->includes(p)
  post firePost: self.employee->excludes(p)
end
Specifying Invariants in USE

**USE Specification**

```plaintext
constraints

class Professor

inv HigherSalaryThanStudent:
    Student.allInstances()->forAll(s:Student|s.salary < self.salary)
```

More details about the syntax can be found in section 2 of the USE documentation:

http://www.db.informatik.uni-bremen.de/projects/USE/use-documentation.pdf
Tool Download & Demonstration

1. Download the USE tool (version 2.4.0) from the following link:
   http://www.db.informatik.uni-bremen.de/projects/USE/#download

2. Unzip the tool in any location

3. Open use-2.4.0\bin folder, and run use.bat

4. Demonstration: import a USE specification

More details about the USE tool can be found in:
http://www.db.informatik.uni-bremen.de/projects/USE/
USE Commands

- Create objects
  - Syntax: `!create objectNameList: className`
  - Example: `!create Tom:Student`

- Destroy objects
  - Syntax: `!destroy objectName`
  - Example: `!destroy Tom`

- Insert links
  - Syntax: `!insert objectNameList into associationName`
  - Example: `!insert (Tom, CSU) into WorksFor`

- Delete links
  - Syntax: `!delete objectNameList from associationName`
  - Example: `!delete (Tom, CSU) from WorksFor`
USE Commands

- Set an attribute value of an object
  - Syntax: `!set objectName.attributeName:= Value`
  - Example: `!set Tom.age := 20`

- Invoke an operation
  - Syntax: `!openter objectName OpName(ParamList)`
  - Example: `!openter CSU fire(Tom)`

- Exit least recently invoked operation
  - Syntax: `!opexit ReturnValExpr`
  - Example: `!opexit`

More details about the commands can be found in section 5 of the USE documentation:
http://www.db.informatik.uni-bremen.de/projects/USE/use-documentation.pdf
Tool Demonstration

1. Create an object model
2. Check the OCL invariants
3. Invoke an operation
4. Check the operation pre-/postconditions

More details about the USE tool can be found in:
http://www.db.informatik.uni-bremen.de/projects/USE/