

Werkzeugunterstützung für die Überprüfung der Einhaltung von OCL-Geschäftsregeln in Java-Programmen

Ralf Wiebicke

28. August 2000

Übersicht

- Code Injector
- Scope von Invarianten
- Caching von Invarianten
- Ausblick

Code Injector - Probleme des einfachen Ansatzes

1. Post-Code an jeder return-Anweisung
2. Vorausberechnung des Rückgabeausdrucks
3. Namenskonflikte
4. kompletter Javaparser notwendig

Code Injector - Wrapping Methods

```
int someMethod(double x)
{
    // here comes the code.
}
```

```
int someMethod_wrappedbyocl(double x)
{
    // here comes the code.
}
```

```
int someMethod(double x)
{
    // pre method code
    int result=someMethod_wrappedbyocl(x);
    // post method code
    return result;
}
```

Code Injector - Wrapping Constructors

```
SomeClass(String x)
{
    // here comes the code
}
```

```
SomeClass(String x, Dummy)
{
    // here comes the code
}
```

```
SomeClass(String x)
{
    this(x, (Dummy)null);
    // post constructor code
}
```

Scope von Invarianten

Wann müssen die Invarianten erfüllt sein?

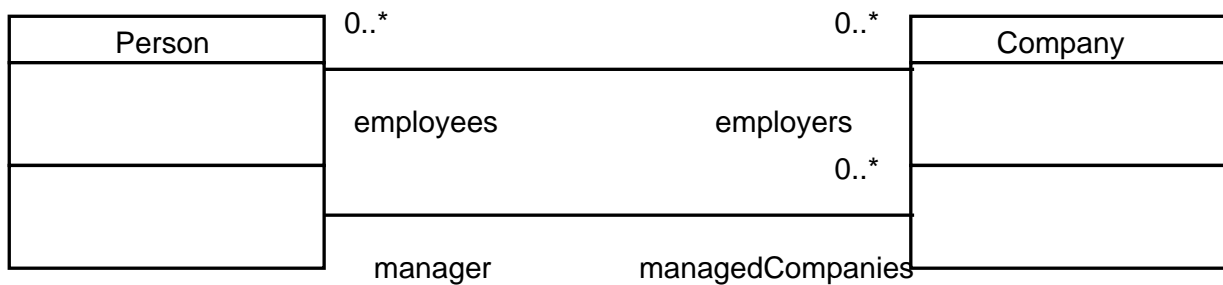
- Alle Methoden.
- Öffentliche Methoden.
- Ausgezeichnete Methoden.
- Explizite Prüfung.

Caching von Invarianten - Überwachung von Attributen

```
class Person
{
    int age;
    int age_oclbackup=age;

    private void checkForChangedFeatures()
    {
        if(age!=age_oclbackup)
        {
            age_oclbackup=age;
            // notify observers of age
        }
        // ... further attributes
    }
}
```

Caching von Invarianten - Beispiele



```
context Company inv:  
  manager.age >= 0
```

```
context Company inv:  
  employees.age >= 0
```

```
context Company inv:  
  employees.getAge() >= 0
```

```
context Company inv:  
  employees.getIncomeAfterTax(0.15) >= 5000
```


Ausblick

- Anwendung bei net-linx
- Scope von Invarianten
 - Unterstützung von ausgezeichneten Methoden
- Caching von Invarianten
 - Einschränkung der observierten Attribute im Typechecker
 - Observation von Methoden

Erweiterte Typinformation

```
class Company
{
    /**
        All persons employed by this company.
        @element-type Person
    */
    Collection employees;
}
```

```
class Bank
{
    /**
        Customers qualified by their
        account number.
        @element-type Person
        @key-type Integer
    */
    Map customers;
}
```

Erweiterte Typinformation - Statische Analyse

```
/**  
    All employed {@link Person persons}  
    of this company.  
    @see Person  
*/  
Collection employees;  
  
boolean isEmployee(Person);  
void addEmployee(Person);  

```