Invasive Composition by Transformation Systems

Thomas Cleenewerck

STS workshop @ GPCE2004
Generative Programming

○ **Subject**
  ○ *software artifacts*

○ **Compositional**
  ○ *non-invasive composition*
  ○ *e.g. referencing, subclassing, mixins, ...*

○ **Transformational**
  ○ *Invasive composition*
  ○ *e.g. SOP, HyperJ + AspectJ, Concern-specific languages*
General Purpose Transformation Systems

- Template based (Jostraca, Apache, XSLT, etc.)
  - template is code parameterized by escaping vars
- Rule based (ASF+SDF, XSLT, etc.)
  - rule rewrite AST
- Attribute based (Attribute grammars, etc.)
  - Attributes attached to the productions are composed into one solution
Observation and Position

- **Observation**
  - Transformational generative techniques still implemented by ad-hoc generators

- **Position**
  - Invasive Composition by general purpose transformation systems is cumbersome and troublesome because of their fundamental underlying in-place substitution property.
    - Dependencies turn impl. into spaghetti
example: 1-n transformations consisting of one local result that is substituted by the matched term, and a number of nonlocal results that belong to different locations in the target program

1 class(
2     datamember("Wizard" "wizard")
3     datamember("Inputfield" "appname")
4     datamember("Inputfield" "appdir")
5     datamember("Key" "programpath")
)
6  method("main" (...) (stats(
7     assign( var("wizard") new("Wizard"))
8     assign( var("page") new("Page"))
9     assign( var("appname") new("InputField"))
10    assign( var("appdir") new("InputField"))
11    mcall( var("appname") setListener (... mcall( () "appdirdefvalue" ()) ... ))
12    mcall( var("appdir") setValue mcall( () "defvalue" () )
13    mcall( var("appname") setValue ("T") )
14    mcall( var("page") "add" (var("appname")))
15    mcall( var("page") "add" ( var("appdir" )))
16    mcall( var("wizard") "add" (var("page")))
17 )))
18  method("defvalue" () (stats(
19     assign( var("programpath") new("Key"))
20     return( ++( key("programpath") "/" appname) )))
21))
22

Research Question

- Extension of Transformation Systems with a basic suite of invasive capabilities.
  - What are these basic invasive capabilities
  - How to merge them orthogonally with the transformation systems
- Extended LTS and rule-based systems
  - mechanisms based on SOP: correspondence, combination and integration