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 compositon*
  - *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* ⇨

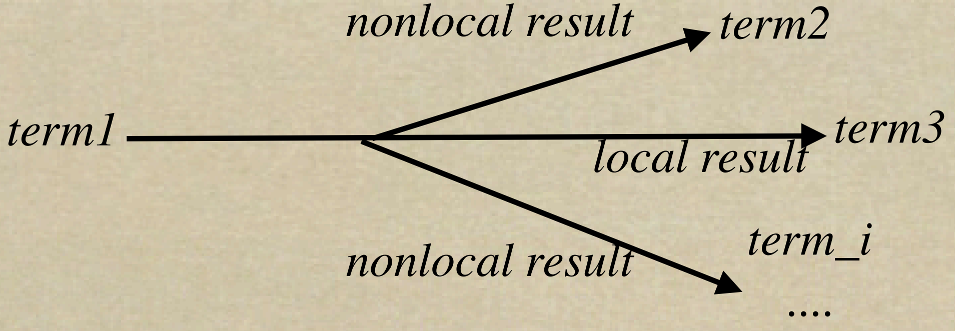
in-place substitution

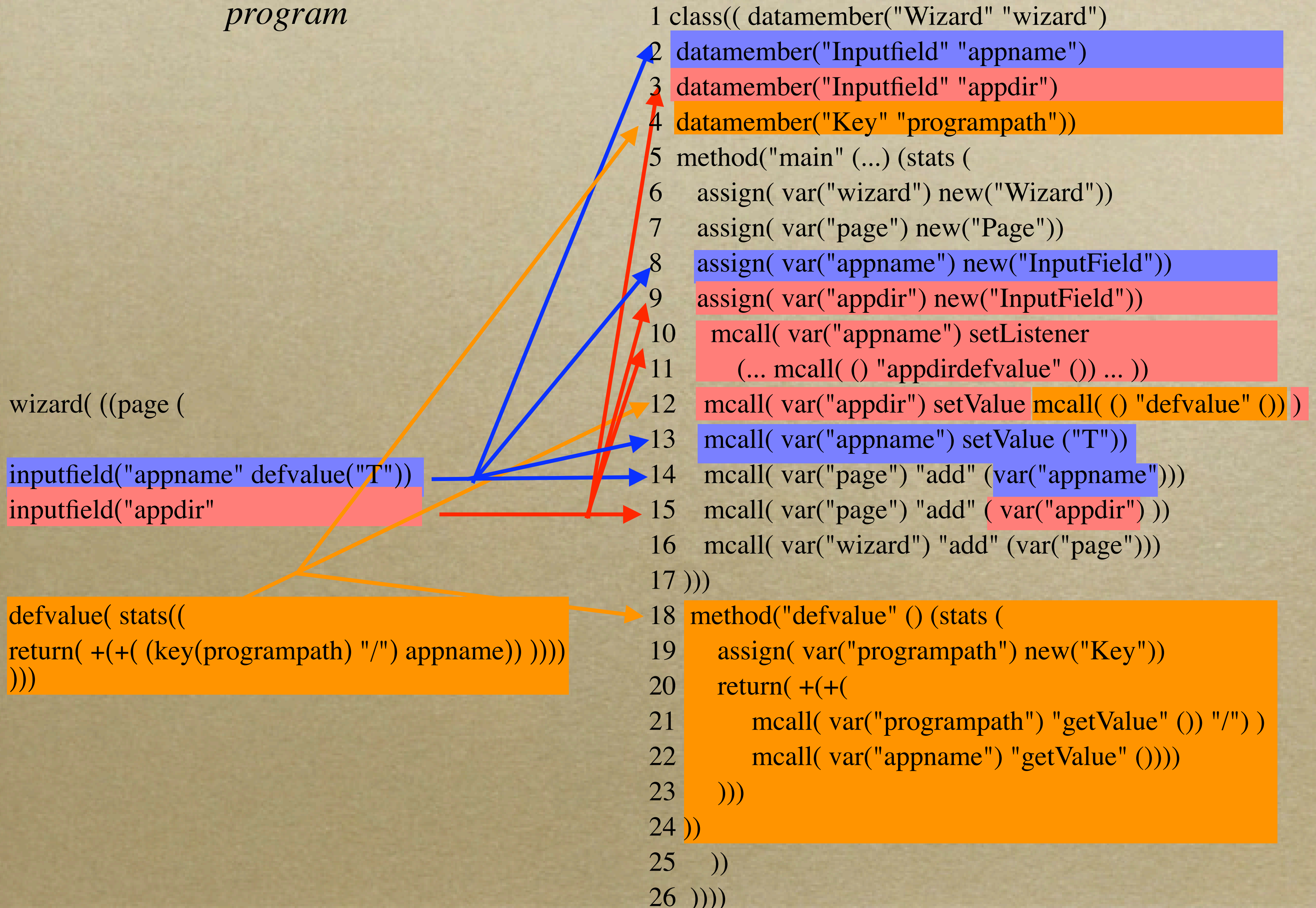
Observation and Position

- *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





```
wizard( ((page (

inputfield("appname" defvalue("T"))
inputfield("appdir"

defvalue( stats((
return( +(+( (key(programpath) "/") appname)) ))))
)))
```

```
class(( datamember("Wizard" "wizard")
 datamember("Inputfield" "appname")
 datamember("Inputfield" "appdir")
 datamember("Key" "programpath"))
 method("main" (...) (stats (
   assign( var("wizard") new("Wizard"))
   assign( var("page") new("Page"))
   assign( var("appname") new("InputField"))
   assign( var("appdir") new("InputField"))
    mcall( var("appname") setListener
       (... mcall( () "appdirdefvalue" ()) ... ))
   mcall( var("appdir") setValue mcall( () "defvalue" ()) )
   mcall( var("appname") setValue ("T"))
   mcall( var("page") "add" (var("appname")))
   mcall( var("page") "add" ( var("appdir") ))
   mcall( var("wizard") "add" (var("page")))
 )))
 method("defvalue" () (stats (
    assign( var("programpath") new("Key"))
    return( +(+(
        mcall( var("programpath") "getValue" ()) "/") )
        mcall( var("appname") "getValue" ())))
    )))
))
   ))
 ))))
```

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*