Design and Modeling Issues in AOSD: A Special Focus on Pointcuts

UNIVERSITÄT DUISBURG ESSEN

Dominik Stein
University of Duisburg-Essen
Institute for Computer Science and Business Information Systems
Data Management Systems and Knowledge Representation
Prof. Dr. Rainer Unland
Schützenbahn 70, D-45117 Essen, Germany
Phone: +49 (0)201 / 183 - 4536, Fax: +49 (0)201 / 183 - 4460
Email: dstein@cs.uni-essen.de

Motivation

Need of Modeling Pointcuts

- pointcut specifications in AOP tend to be very complex and are difficult to read
- no graphical means are around to facilitate their understanding
- such graphical means are needed for
  - teaching novices
  - communication among developers and users
  - documentation for maintainers and administrators

Objectives

Pointcut Models Should...

- graphically represent both sets of static join points and sets of dynamic join points
  - graphical notations should provide uniform means to designate sets of join points of either conception
- describe selection criteria on the direct context of (a set of) join points
  - such selection criteria must be fulfilled for a given join point in order to belong to the pointcut
- designate the parts of the crosscut environment being exposed to crosscutting behavior/structure
  - those parts can be referenced by aspects to perform its crosscutting task

Key Features

Join Point Designation Diagrams...

- describe selection patterns on join points
- may specify both structural and behavioral selection criteria
- may be used to designate both static and dynamic join points
- may specify which parts of the join point's direct context are to be exposed
- are based on UML to facilitate their understanding
- are provided with selection semantics on UML models
- go beyond conventional selection means of prevailing AOP techniques

Further Readings

References

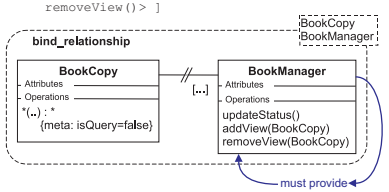
- http://dawis.informatik.uni-essen.de/site/staff/stein/
- Stein, D., Hanenberg, S., Unland, R., Modeling Pointcuts, Early Aspects Workshop, AOSD 2004
- Stein, D., Hanenberg, S., Unland, R., Issues on Representing Crosscutting Features, Workshop on Aspect-Oriented Modeling with UML, AOSD 2003
- Stein, D., Hanenberg, S., Unland, R., A UML-based Aspect-Oriented Design Notation, Proc. of AOSD 2002

Overview to «Join Point Designation Diagrams» («JPDD»)

Integration into Theme/UML

JPDDs detail bind relationships:

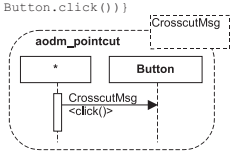
```
bind[ <BookCopy, {meta: isQuery=false}>,
   <BookManager, updateStatus(), addView(),
   removeView()> ]
```

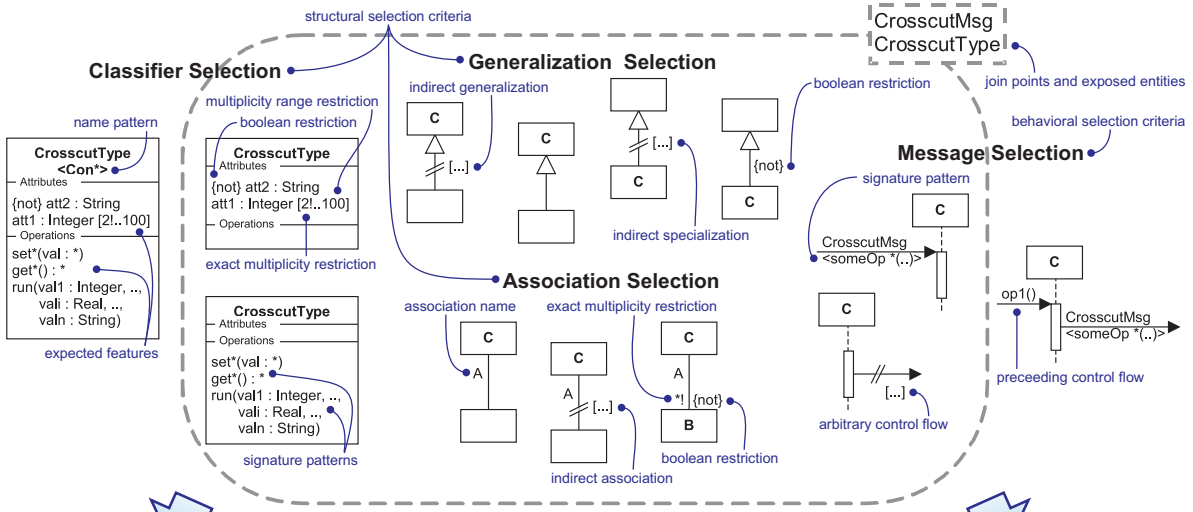


Integration into Our AODM

JPDDs detail stereotyped «pointcut» operations:

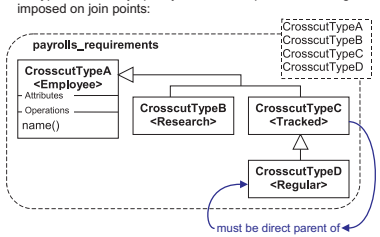
```
pointcut stateChanges(Subject s)
{base = target(s)
   && call(void Button.click())}
```





Application in MDSOC with Hyper/J

JPDDs reflect on the "declarative completeness" of hyperslices in Hyper/J, and may specify structural requirements being imposed on join points:



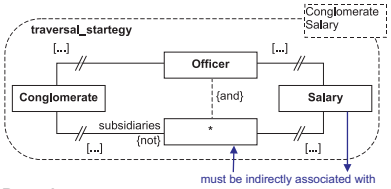
Remarks

- Pointcut models select sets of join points only
- Pointcut models do not state how to compose
- Pointcut models may be used to model concern mappings

Application in Adaptive Programming

JPDDs may model traversal strategies in AP:

```
*from* Conglomerat
  *bypassing* -> *,subsidiaries,*
  *via* Officer *to* Salary
```



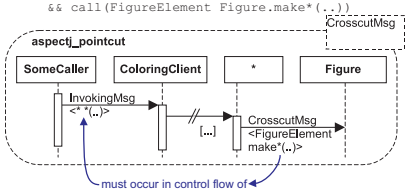
Remarks

- Construction vertices are depicted as associated classes
- Alternation vertices would be depicted as super-classes
- Repetition vertices would be depicted as multi-objects

Application in AOP with AspectJ

JPDDs may model pointcuts in AspectJ:

```
pointcut aspectj_pc():
  cflowbelow(call(* ColoringClient.*(..))
  && this(SomeCaller))
  && call(FigureElement Figure.make*(..))
```



Remarks

- Runtime-system-dependent join points, such as method executions, class and object initializations, must be indicated by special stereotypes.
- Program-text-based join points, such as within and withincode, currently have no graphical representation

Technical Perspective

Pointcut Models Are...

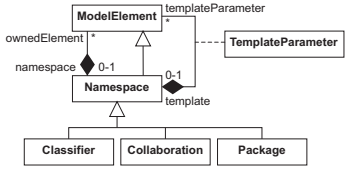
- a new diagram type
- based on UML's current version 1.5
- syntactically templates for namespaces or subtypes thereof (here, we use collaborations)
- semantically new to the UML (as they describe selection patterns rather than production patterns)

Pointcut Models May Specify...

- structural selection criteria on UML classifiers in a class-diagram-style
- behavioral selection criteria on UML messages in an interaction-diagram-style

Abstract Syntax

(Adopted from UML's meta-model)



Selection Semantics

- defined by means of special meta-operations on UML's meta-classes
- selection is initiated by a special meta-operation on UML template parameters (see below)
- selection is executed on namespaces (e.g. on models, packages, etc.)

```
context TemplateParameter::
matchingModelElements(Target : Namespace) :
Set(ModelElements)
post: result = Target.allContents->select(ME |
    if self.templateParameter.isOclKindOf(Classifier) then
        ME.oclIsKindOf(Classifier)    -- query UML Classifiers
        and ME.matchesClassifier (self.templateParameter)
        and ME.matchesRelationships
(self.templateParameter)
    else
    if self.templateParameter.isOclKindOf(Message) then
        ME.oclIsKindOf(Message)    -- query UML Messages
        and ME.matchesMessage (self.templateParameter)
    else
        false                                    -- empty set
    endif  endif)
```

Work Ahead

Remaining Issues

- application to further AOP techniques
- combination semantic of pointcut models
- specialization semantic for pointcut models
- implementation in prototype
- relating pointcut models to crosscutting assertions