Reflection in Oberon

DDipl.-Ing. Christoph Steindl

Johannes Kepler University Linz

Practical Computer Science

steindl@ssw.uni-linz.ac.at http://www.ssw.uni-linz.ac.at/Projects/Reflection.html

Motivation

- What are the field names of a record type *T* declared in module *M*?
- Which procedures are currently active? What are the names, types and values of their variables?
- Does the caller of the currently active procedure have a variable named “x”, and if so, what is its type and value?

Introspective Capabilities

- to look into itself (reflection)
- inspect other programs
- obtain information about the current run-time state
- do not change the state

Invocational Capabilities

- explicitly call (hidden) functionality of run-time system
    - create new objects
    - dynamic loading and linking
    - unloading of compiled code

Intercepting Capabilities

- change the behaviour of language primitives
  - object creation
  - object destruction
  - method dispatch
  - access to simple attributes

From Compile Time to Run Time

symbol table of the compiler

⬇

persistent reference information in object file

⬇

loaded reference information in memory

Objectives

- zero overhead during normal operation
  - no debug code
  - ignore symbol information normally
  - little memory consumption
- efficient access for typical application patterns
  - displaying exception information after trap
  - accessing local variables of a procedure
  - scanning types or global variables of a module

Iteration mechanism: “Rider”

```
Rider = RECORD
    name: ARRAY 32 OF CHAR;      (* of variable, type, procedure *)
    mode: SHORTINT;              (* “state” of rider *)
    form: SHORTINT;              (* type of element at current position of rider *)
    idx, off, len: LONGINT;
    mod: ARRAY 32 OF CHAR;       (* containing module *)
    level: INTEGER;
    PROCEDURE (VAR r: Rider) Next;
    PROCEDURE (VAR r: Rider) Zoom (VAR sub: Rider);
    PROCEDURE (VAR r: Rider) Read (VAR ch: CHAR);
    PROCEDURE (VAR r: Rider) Write (ch: CHAR);
    ...
END
```

Structure of Reference Information

frameRider.Zoom(varRider)

Stack = {Frame}.

Frame = {Variable}.

Variable = simpleVar | RecordVar | ArrayVar.

RecordVar = {Field}.

ArrayVar = {Elem}.

Field = Variable.

Elem = Variable.

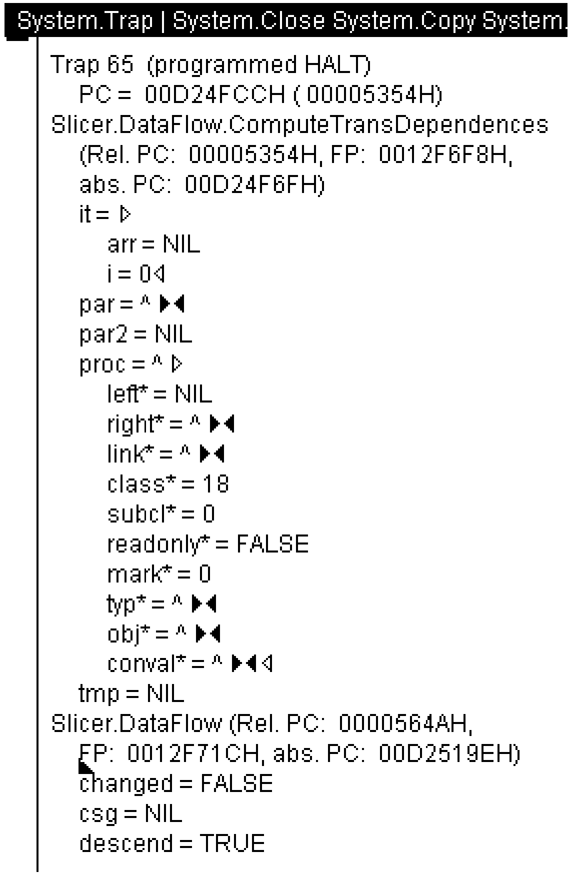
Proc = {Variable}.

How to Open a Rider?

- OpenStack <= current procedure on stack
- OpenVars <= global variables of a module
- OpenTypes <= types of a module
- OpenProcs <= procedures of a module
- OpenPtr <= information about an object
- OpenProc <= information about a procedure

Applications

- Post-mortem debugger (displaying trap information)
- Showing the global variables of a module (System.State)
- HeapInspector
- Embedded SQL
- Translation of OQL/ODL to Oberon



```
Trap 65  (programmed HALT)
   PC =  00D24FCCH ( 00005354H)
Slicer.DataFlow.ComputeTransDependences
   (Rel. PC:  00005354H, FP:  0012F6F8H,
   abs. PC:  00D24F6FH)
   it = ▷
      arr = NIL
      i = 0◁
   par = ^ ▶◀
   par2 = NIL
   proc = ^ ▷
      left* = NIL
      right* = ^ ▶◀
      link* = ^ ▶◀
      class* = 18
      subcl* = 0
      readonly* = FALSE
      mark* = 0
      typ* = ^ ▶◀
      obj* = ^ ▶◀
      conval* = ^ ▶◀◁
   tmp = NIL
Slicer.DataFlow (Rel. PC:  0000564AH,
   FP:  0012F71CH, abs. PC:  00D2519EH)
   changed = FALSE
   csg = NIL
   descend = TRUE
```

Recapitulation - Introspective Caps

- information about modules, types, variables, etc. is available
- reflective (self-inspection)
- information for loaded modules only, i.e. run-time state
- state may be changed => not only introspective, but also altering

Recapitulation Invocational Caps

- not main goal of this metaprogramming system
  - support already included in base system (dynamic object creation, dynamic loading and freeing of modules)

Recapitulation Intercepting Caps

- considered dangerous
- not directly supported, but:
  - object creation can easily be patched (Tracer)
  - object destruction - hooks possible (finalization)
  - method table entries can be patched
  - access to simple attributes - hard (data breakpoints)