Forth Projectional Editing

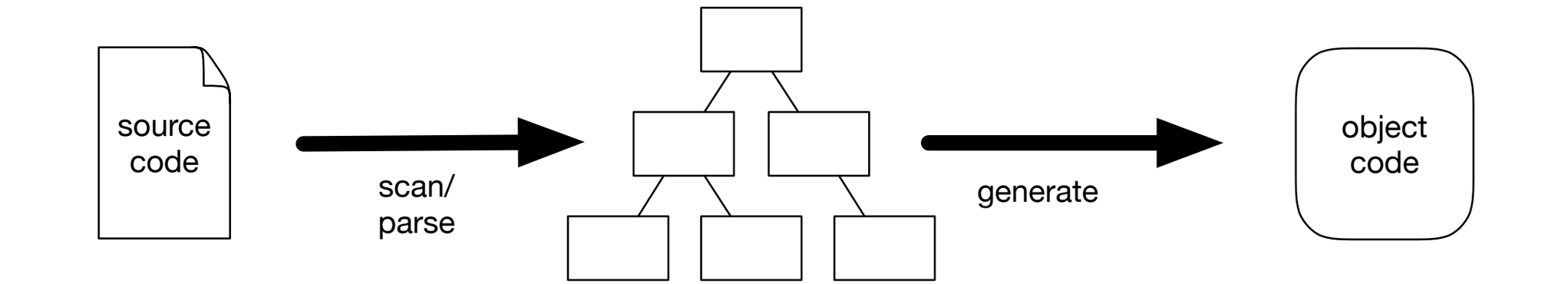
EuroForth 2019
Hamburg

Ulrich Hoffmann <uho@xlerb.de>

Overview

- Classical and Projectional Editing
- Forth Projectional Editing
  - hex-edit, stack-edit
  - other editors, other screen editors
  - related work
- Conclusion

Classical Editing

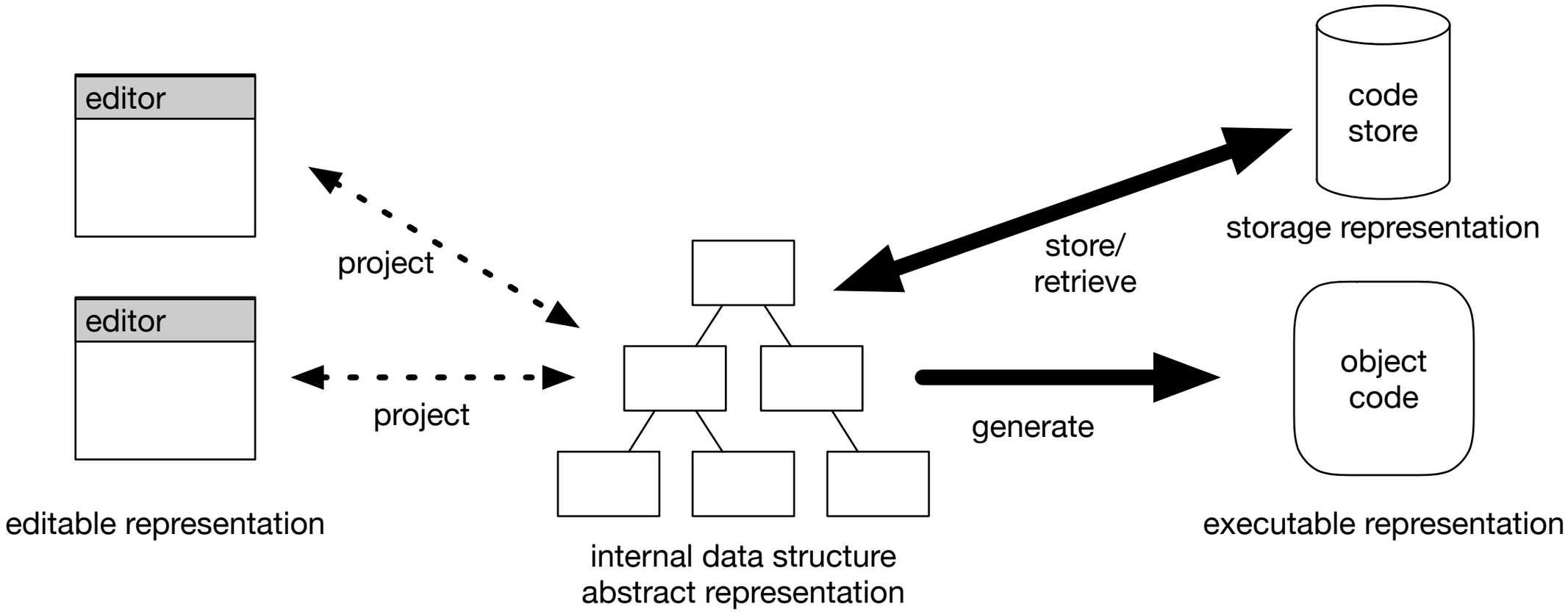


editable representation
storage representation

internal data structure
abstract representation

executable representation

Projectional Editing



Forth Projectional Editing

- What would be a suitable internal data structure for Forth programs?
- Forth progammers are
  - use to programming down to the metal
  - choose operators according to the type of data
  - represent data in memory by themselves

memory

Forth Projectional Editing memory

- Have editors that
- project memory to an appropriate editable form.
- allow humans to change the data.
- modify memory according to the changes.

Hex Edit

```
$ sf hexedit.fs

( hex-editor loaded. Usage: c-addr u HEX-EDIT  ) ok

Create conference  'E' c,  'u' c,  'r' c,  'o' c,  'F' c,  'o' c,  'r' c,  't' c,  'h' c,
conference 30 hex-edit

00003CB44  45 75 72 6F 46 6F 72 74 68 08 68 65 78 2D 65 64   EuroForth.hex-ed
00003CB54  69 74 63 65 2A 00 0F 00 4F 14 00 00 2A 00         itce*...O...*.
```

- Demo

Stack Edit

```
> gforth stackedit.fs
10 20 30 40 50 -1 stack-edit

  0: '?' $FFFFFFFFFFFFFFFF  #18446744073709551615                    -1
  1: '2'               $32                      #50                   50
  2: '('               $28                      #40                   40  42
  3: '.'               $1E                      #30                   30
  4: '.'               $14                      #20                   20
  5: '.'                $A                      #10                   10

up/down: select line   DEL   Ctrl-X, -C , -V    Forth words leaving one item
```

- Demo

Other Editors

- Variable Editor
  `BASE var-edit`
- User Area Editor
  `UP@ user-edit`
- Structure Editor
  BEGIN-STRUCTURE ... FIELD: ... END-STRUCTURE point
  `p1 point struct-edit`
- Wordlist Editor
  `FORTH-WORDLIST wl-edit`
- Word Definition Editor
  `' DUP word-edit`

Other Source Editors

- traditionally 64 x 16 screens in disk blocks
  "*It just seemed to be convenient at the time*" Charles Moore

- other projections possible and reasonable
  - Other screen sizes (80x25 or 4K)
  - Screen Editor with line terminators
  - Screen Editor with screen separators

Related Work

- Jetbrain's Meta Programming System (MPS)
- Jupiter Ace: word editor (*code is the source*)
- ForthOS: 80x25 screens
- Enth: CodeEd, line termiated source in 1KB blocks
- HolonForth: words in a data base, powerful editing views
- ColorForth: tokenizing on editing, editor works on tokens

Conclusion

- Projectional Editing can be applied to Forth but in a different way
- Memory is the Forth internal data structure
- Done before but not called that way
- hex-edit and stack-edit in Forth2012
- more editors on the way

Is the map the territory? You decide.

Forth is stacks, words, and blocks; start there.
Jeff Fox