Goldsmiths UNIVERSITY OF LONDON

Recent Developments in SBCL

or

an adventure in Unicode

Christophe Rhodes

(and a cast of many characters)

Goldsmiths UNIVERSITY OF LONDON

Outline

- Introduction: what is SBCL?
- Exposition: the 0.8 series
  - Second Subject: new features
- Development: Unicode and external-format
- Recapitulation: the road to 1.0
- Coda: questions and discussion

Goldsmiths UNIVERSITY OF LONDON

Introduction

- Steel Bank Common Lisp
  - “Carnegie made his fortune in the steel industry, controlling the most extensive and complete system of iron and steel industries ever managed by an individual.”
  - “[Mellon] joined his father's banking firm, T. Mellon & Sons, two years later and had the ownership of the bank transferred to him in 1882 at the age of 27.”

    -- Wikipædia

Goldsmiths UNIVERSITY OF LONDON

What is SBCL?

- ANSI Common Lisp environment
- Written in portable ANSI Common Lisp
- Liberally licensed
- Trivially buildable (on most Unixoids)

Goldsmiths UNIVERSITY OF LONDON

Exposition

- sbcl-0.9.0 released 2005-04-24 (today!)
- What has changed since 2003-05-25 (0.8.0)?
  - new platforms
  - new features
  - new users and developers
  - fewer bugs
  - better infrastructure

Goldsmiths UNIVERSITY OF LONDON

0.8: new stuff

- platforms
  - Darwin / PowerPC
  - Linux / x86-64
  - CLISP (as host compiler)

Goldsmiths UNIVERSITY OF LONDON

0.8: new stuff

- features
  - linkage tables
  - sampling profiler
  - package locks
  - compiler conditions
  - stepper
  - contribs (asdf-install,simple-streams,md5,...)
  - Unicode characters
  - external-format

Goldsmiths UNIVERSITY OF LONDON

0.8: new stuff

- compiler enhancements
  - modular (hardware) arithmetic
  - dynamic-extent
  - contrib for 64-bit arithmetic
  - loop analysis

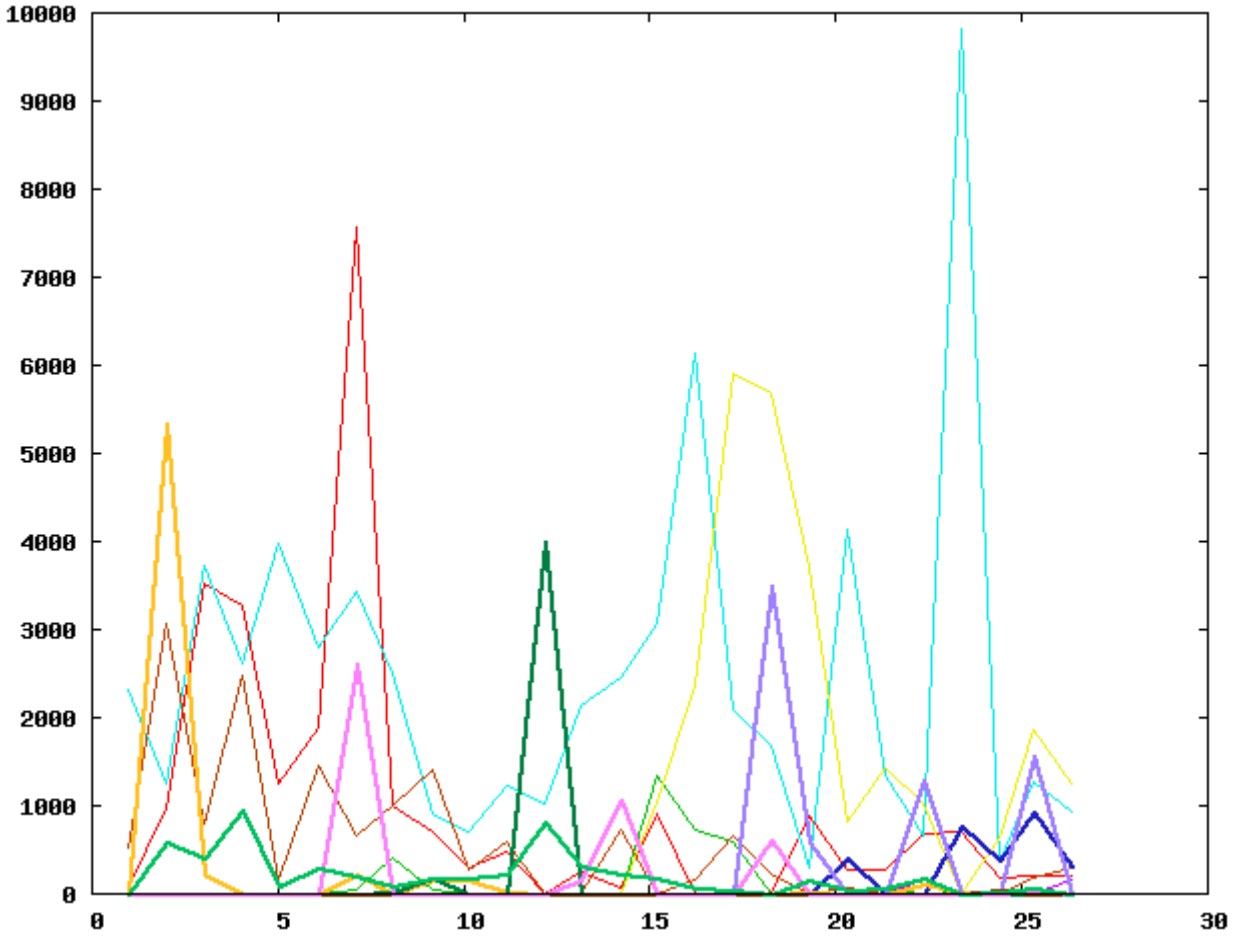
Goldsmiths UNIVERSITY OF LONDON

0.8: new stuff

- people
  - committers
  - users
  - development community

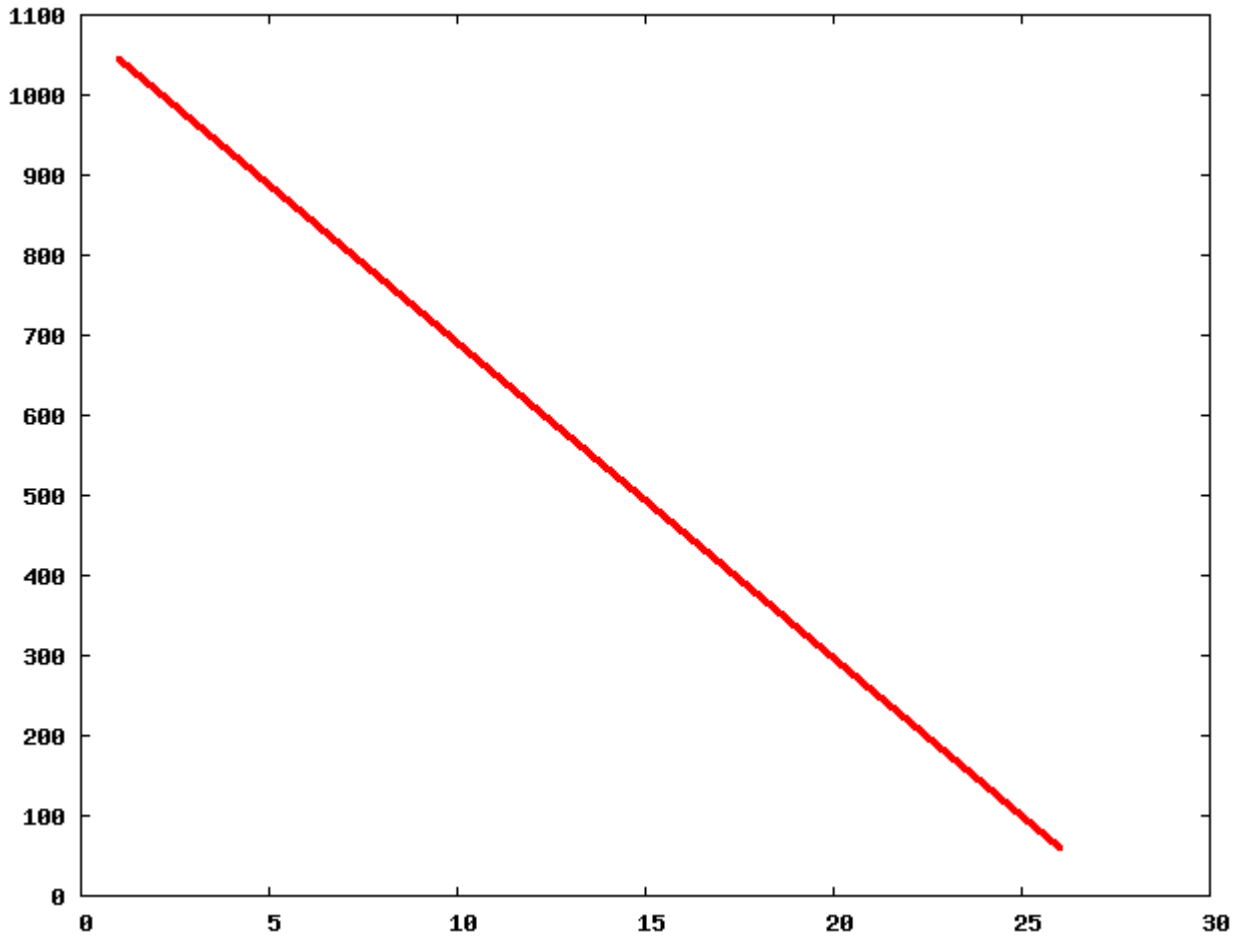
Goldsmiths UNIVERSITY OF LONDON

0.8: cvs activity



Goldsmiths UNIVERSITY OF LONDON

0.8: fewer bugs



Goldsmiths UNIVERSITY OF LONDON

0.8: better infrastructure

- automatic benchmarking
  - see: Grouping Common Lisp Benchmarks
- automatic building
  - self-tests
  - self-build
  - ansi-tests

Goldsmiths UNIVERSITY OF LONDON

Development

- Most user-visible features
  - Unicode characters
  - non-trivial external-format support

Goldsmiths UNIVERSITY OF LONDON

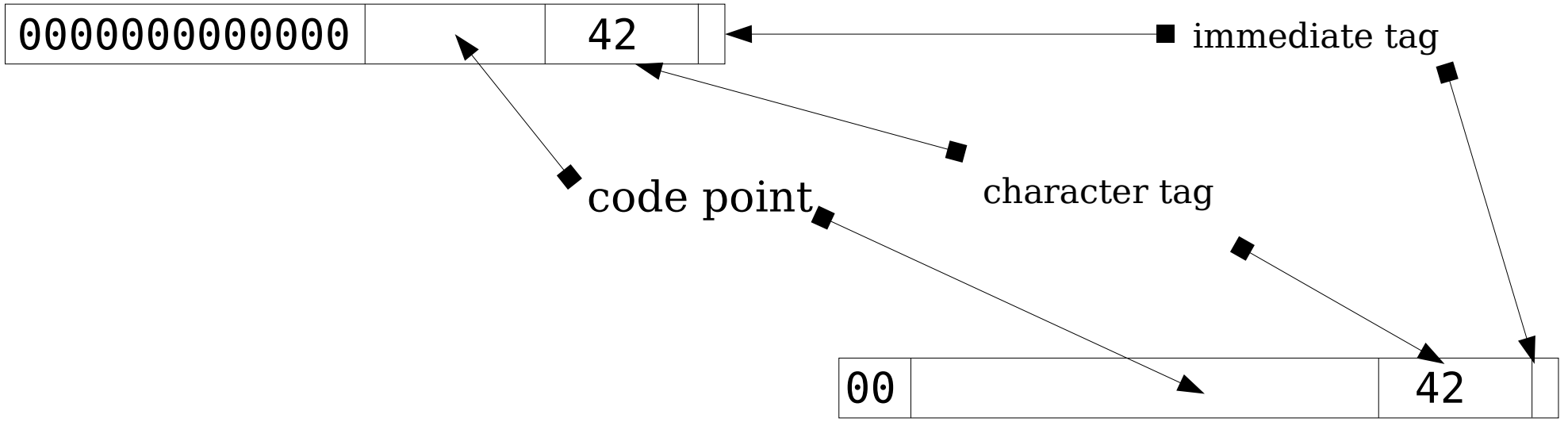
Unicode: characters

- What is a character?
  - glyph? grapheme? ligature?
  - integer, with interpretation (code point)
  - Unicode provides interpretation
    - code points between `0` and `#x110000`
    - examples
      - `#x27`: ' (APOSTROPHE)
      - `#xe9`: é (LATIN SMALL LETTER E WITH ACUTE)
      - `#x3bb`: λ (GREEK SMALL LETTER LAMDA)
      - `#x2206`: ∆ (INCREMENT)

Goldsmiths UNIVERSITY OF LONDON

Unicode: characters

- code-point range 0-127: ASCII
- code-point-range 128-255: Latin 1
- higher code points: other characters



Goldsmiths UNIVERSITY OF LONDON

Unicode: strings

- three possibilities
  - One string type, one representation
    - `(array-element-type string) => character`
    - each element takes 32 bits
  - One string type, many representations
    - `(array-element-type string) => character`
    - each element takes 8,16 or 32 bits
    - extra level of indirection
      - object identity in presence of `(setf char)`
      - garbage collection compacting support

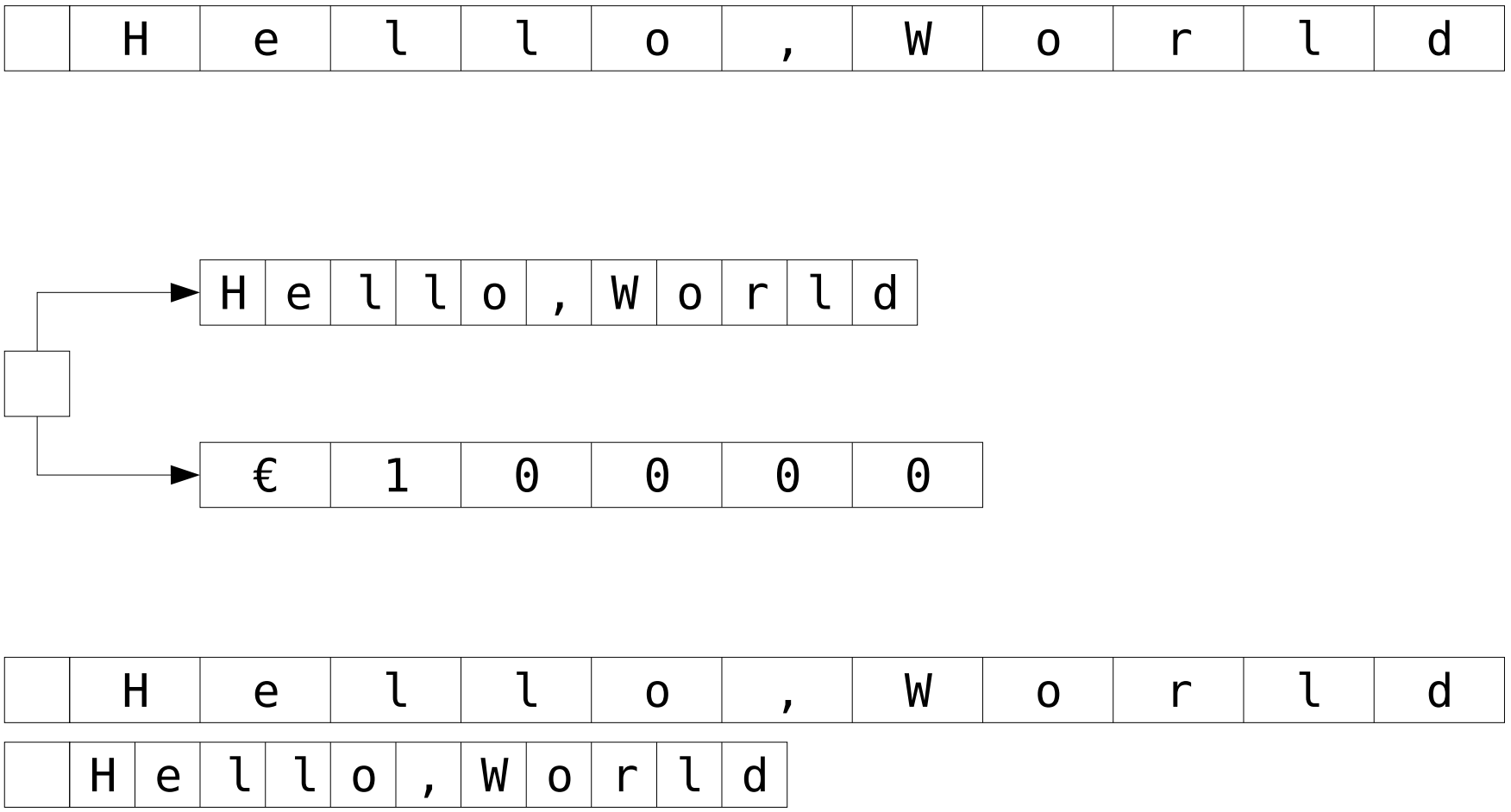
Goldsmiths UNIVERSITY OF LONDON

Unicode: strings

- three possibilities (cont'd...)
  - many string types, many representations
    - `(array-element-type string)` `=> base-char`
      `=> character`
    - each element takes 8 or 32 bits
    - confusing? you decide...

Goldsmiths UNIVERSITY OF LONDON

Unicode: strings



Goldsmiths UNIVERSITY OF LONDON

External-format

- Now we have characters and strings...
  - but what can we do with them?
    - upcase, downcase, capitalize...
    - char-code...
    - printing?
- Late 1960s: EBCDIC vs ASCII
- Late 1990s: Latin 1 vs Latin 9 (vs Shift-JIS...)
- Today: UTF-8 vs UCS-2 vs UTF-16

Goldsmiths UNIVERSITY OF LONDON

External-format

- More complex than just an encoding
  - character set (replacement / error behaviour)
  - line termination convention (`#\Newline`)
  - byte-order mark convention (Windows)
- Applicable elsewhere
  - pathnames / namestrings
  - FFI (`char *`, `char []` arguments / returns)
  - command line arguments
  - strings to octet vector conversions

Goldsmiths UNIVERSITY OF LONDON

External-format

- observations
  - external-format support required in today's world for `char-code-limit` greater than 128. (unless you're an isolationist American or Western European)
  - current implementation in SBCL far from complete
    - `:ascii`, `:latin-1`, `:latin-9`, `:ebcdic`, `:utf-8`
    - but no newline, BOM treatment

Goldsmiths UNIVERSITY OF LONDON

Recapitulation

- Wishlist
  - Unicode
    - character ↔ name translations
  - External-format
    - more external-format implementations
    - code unification with octets / FFI
    - filesystem interaction
    - bivalent streams

Goldsmiths UNIVERSITY OF LONDON

The Road to 1.0

- Documentation (more and better)
- Debugger API (enough for SLIME)
- Deletion of unmaintainable interfaces
- Stubborn bug squashing
- A few remaining features
  - callbacks
  - block compilation
  - your favourite feature here

Goldsmiths UNIVERSITY OF LONDON

Coda

- Thanks
  - CMUCL Hackers (25 years of history)
  - Eric Marsden, Teemu Kalvas, Robert Macomber
  - Edi and Arthur
- Any questions or suggestions?