Executable Texts

SHOT, Washington, DC

stuart mawler

Virginia Tech, Northern Virginia Campus

2007/10/19

Background

- Usual view
  - Software running end product
    - e.g., Microsoft Word or PowerPoint
- Alternative view – Executable Texts
  - Software source code as document, manuscript, corpus, or text
  - Consumed among communities of programmers
  - Continual historical archive, written and re-written
    - Programmer acts like “Talmudic” scholar
- This presentation
  - Investigates two sub-communities
    - Linux Open Source Programmers working on the operating system kernel in C
    - Corporate programmers working on a core business application in COBOL

What I Found...

- “Real constraints”
  - The artifacts are impacted by practical considerations outside of the programming task
- Social roles of executable texts
  - What the common laborer is doing, rather than those in charge
    - The lives and intentions of actual people are encapsulated in the comments
      - Even when created for “capitalist” ends, the personalities of the programmers show through
- “Programming archaeology” / Micro-histories
  - Viewed through the modifications to a program or set of programs
  - Both “explicit” and “implicit”
  - A single program is a form of virtual archaeology, peeling back the layers of modifications by various people for various projects

Linguistic Requirements

- How to define a comment…
  - in COBOL (corporate program)

```
00892 *----SET ADDRESS OF MESSAGE MAIN HEADER.                          PRGMNBR1
00893 *                                                                 PRGMNBR1
```

  - in C++ (online programming guide)

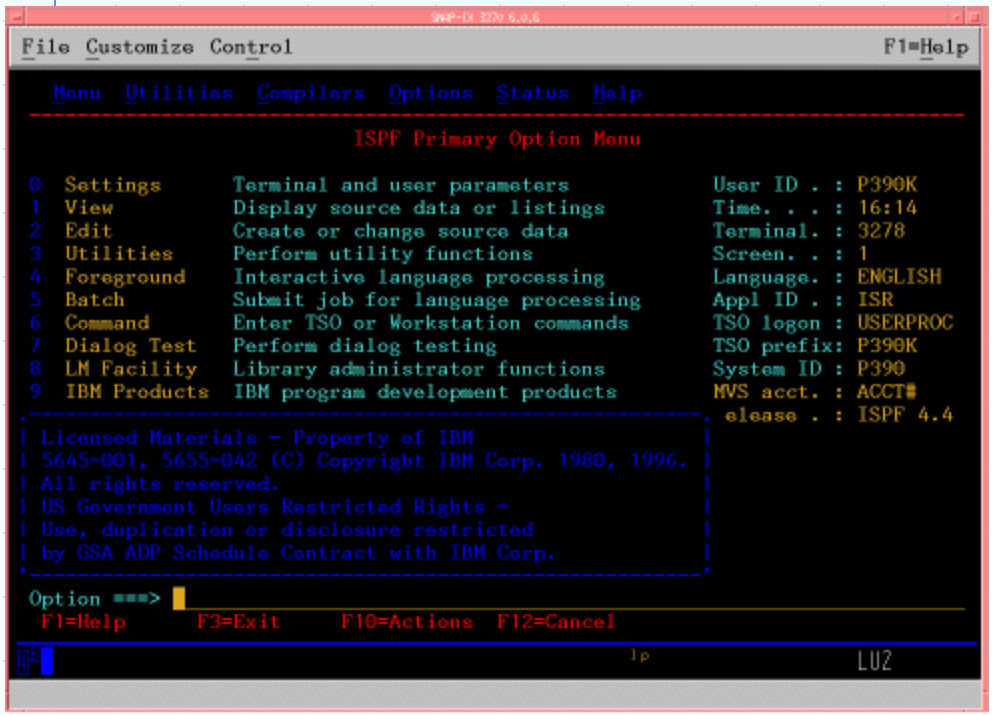
```
//============================================================//
//Development By : Jigar Mehta
//Date : [ & now() & ]
//============================================================//
```

  - in C (Linux kernel)

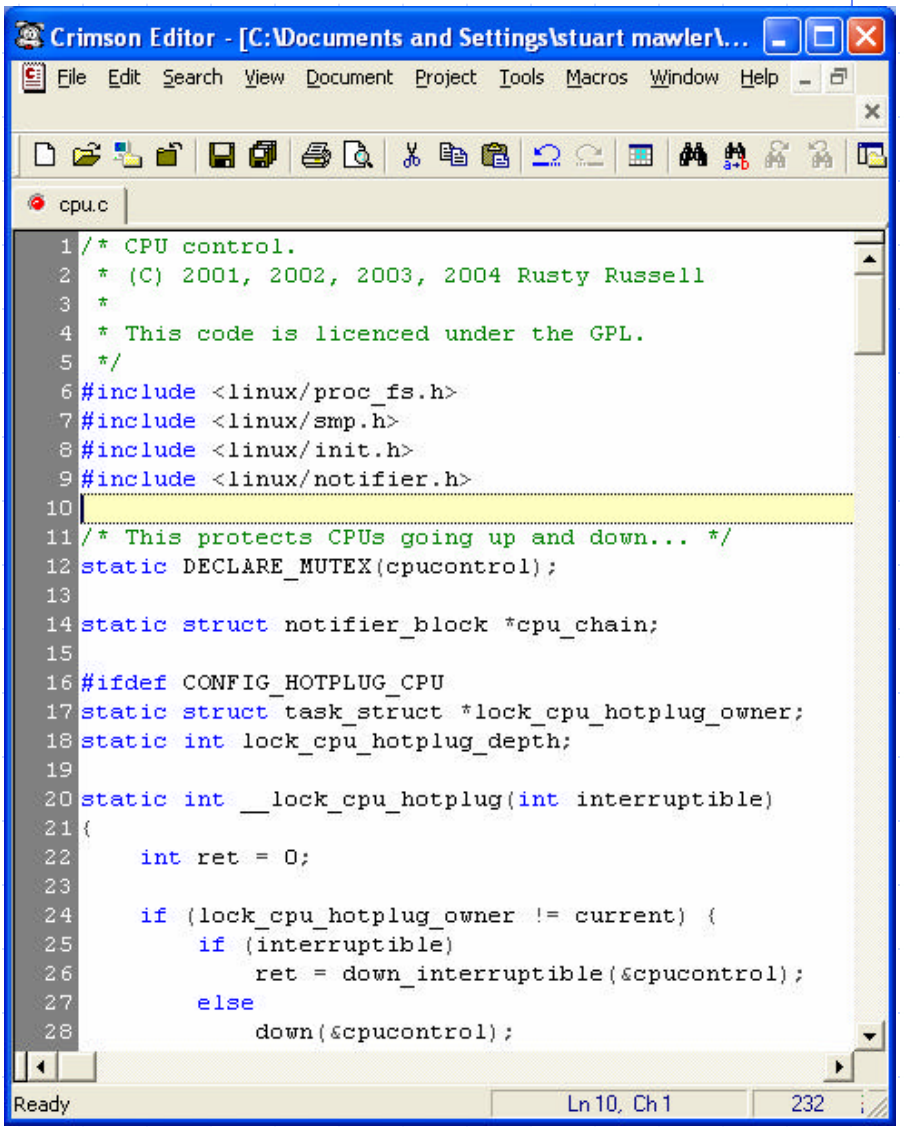
```
/* Arch-specific enabling code. */
```

Technical Environment

- Different visual constraints

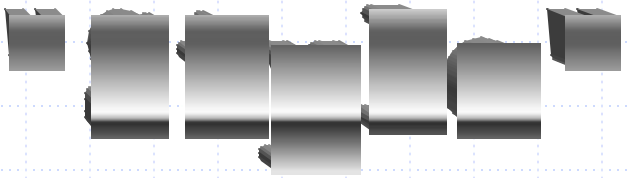


Source: www.dataconnection.com/sna/images/snapix3270.gif



Cultural Constraints and Norms

- Accepted practices
  - Normative purpose for the comment
    - Power structure driving the norms
  - Content
  - Visual Impact



Purpose:
Identify "who" & "when"

Power:
Developed by a manager

Content:
Nothing about the code

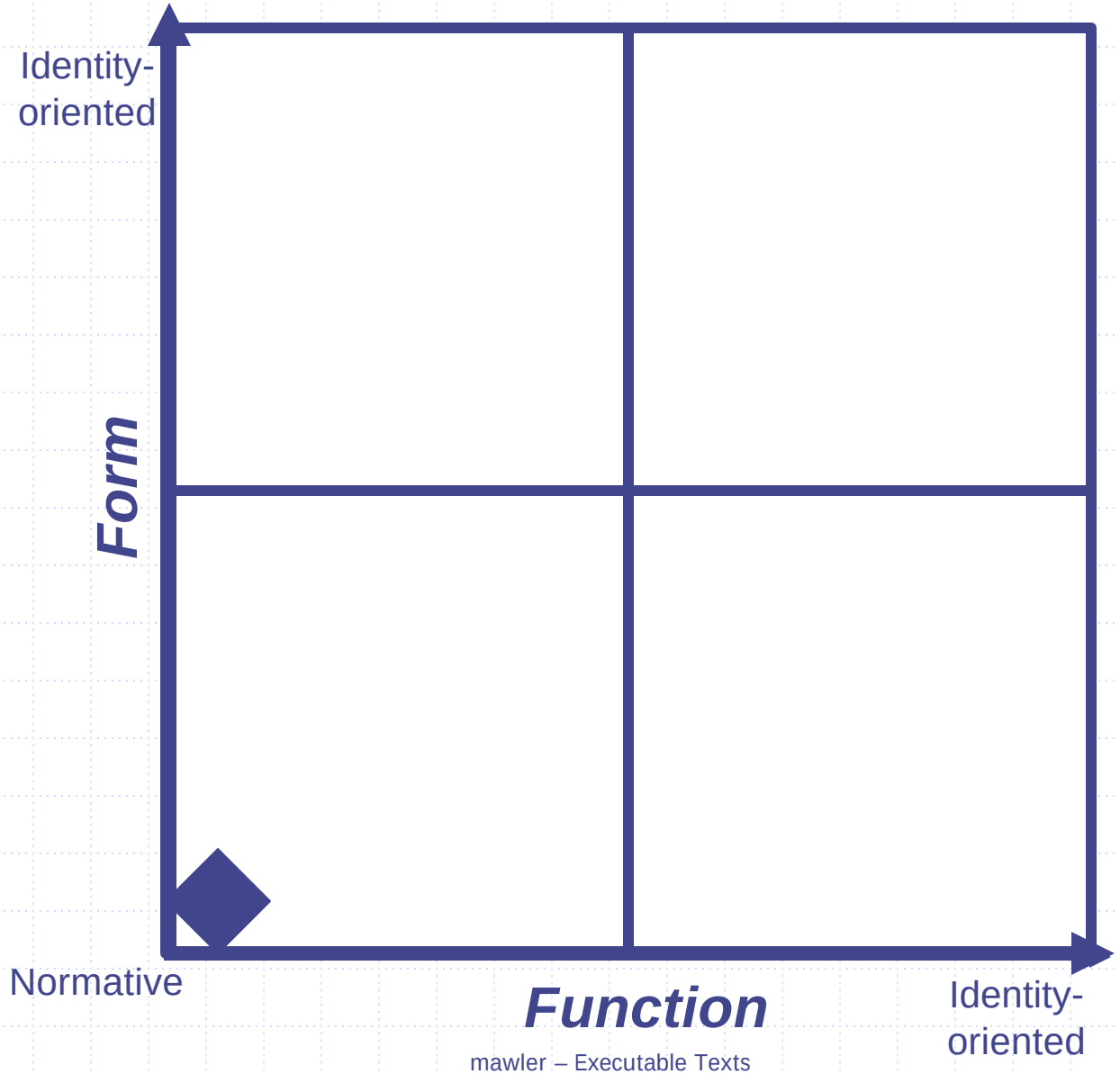
Visual:
Easy to see when scanning

```
//=========================================================//
//Development By : Jigar Mehta
//Date : [ & now() & ]
//=========================================================//
```

```
/* Arch-specific enabling code. */ (cpu.c)
```

```
#endif /*CONFIG_HOTPLUG_CPU*/ (cpu.c)
```

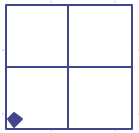
“Good” Comments



Good Comments – By Example

- Corporate Sample

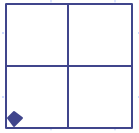
```
01827 *PSR3
01828 *     WHEN SUB PROMO IS PASSED, SET UP PGM TO UPDATE FF INFO IF
01829 *     SUBSEQUENT EDITS ARE COMPLETED ERROR FREE
01830 *         LOGIC TO UPDATE WAS PREVIOUS LOCATED HERE
01831 *         IT HAS BEEN MOVED TO THE 0840 PARAGRAPH
01832 *
```



  - Written in COBOL
  - Tells what is happening
  - What has been changed
  - How it was changed
  - Does not tell “who” or “when”
  - Does say “why”, subtly with “PSR3”
    - Shows much context, if you know the context...

- Good, not “historical”:

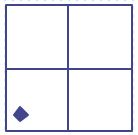
```
#endif /*CONFIG_HOTPLUG_CPU*/ (cpu.c)
```



Explicit History

- A snippet of the “change log” at the top of a COBOL program

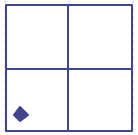
```
00013 ****************************************************************FOprogm1
00014 *                 CHANGE LOG                                     FOprogm1
00015 *   LEVEL       DATE        CHANGED BY     CHANGE                FOprogm1
00016 * -------  ----------   ------------   --------------------------FOprogm1
00017 *           8/10/94      B.K.           DELETED CHANGE LOG ENTRIESFOprogm1
00018 *  FO 2029  06-21-93     J.C.           ADD SECURITY TO AB CANCEL FOprogm1
00019 *                                       REASON                    FOprogm1
00020 *           07-27-93     R.P.           OBTAIN LANGUAGE LITERALS  FOprogm1
00021 *                                       FROM DECODE FILE INSTEAD OFOprogm1
00022 *                                       FROM HARD-CODED TABLE     FOprogm1
00023 *           08/02/93     B. H.          CHANGED CUSTOMER FIRST    FOprogm1
00024 *                                       TRANSACTION CODE VALUE    FOprogm1
00025 *                                       FROM 'PI' TO 'RP'.        FOprogm1
00026 *           08/20/93     P. S.          ADD CALL TO SUBROUTINE    FOprogm1
00027 *                                       FOsubrt1 WHEN BTN IS CHNGEFOprogm1
00028 *                                       TO &&& ACCOUNT.           FOprogm1
00029 *           10/02/93     R. P.          ADDED CODE TO UPDATE BTN  FOprogm1
00030 *                                       AND ANI FILES WHEN ACCT ISFOprogm1
00031 *                                       &&& CUSTOMER.             FOprogm1
00032 *           10/05/93     V. S.          ADDED LOGIC TO ALLOW UPDATFOprogm1
00033 *                                       ACCESS TO COLLECTION TYPESFOprogm1
```



Explicit History

- A snippet of the “change log” at the top of a Linux Kernel program: sched.c

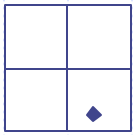
```
/*
 *  kernel/sched.c
 *
 *  Kernel scheduler and related syscalls
 *
 *  Copyright (C) 1991-2002  Linus Torvalds
 *
 *  1996-12-23  Modified by Dave Grothe to fix bugs in semaphores and
 *                  make semaphores SMP safe
 *  1998-11-19      Implemented schedule_timeout() and related stuff
 *                  by Andrea Arcangeli
 *  2002-01-04      New ultra-scalable O(1) scheduler by Ingo Molnar:
 *                  hybrid priority-list and round-robin design with
 *                  an array-switch method of distributing timeslices
 *                  and per-CPU runqueues.  Cleanups and useful suggestions
 *                  by Davide Libenzi, preemptible kernel bits by Robert Love.
 *  2003-09-03      Interactivity tuning by Con Kolivas.
 *  2004-04-02      Scheduler domains code by Nick Piggin
 */
```



Implicit History

- This example also tells something about programming style, since “structured” had to be “invented” at some point
  - After the “invention” of structured programming, all that went before it became “spaghetti” and this name also came to apply to improperly done code
    - Hence, the new methodology both retroactive and proactively made all other approaches “wrong”

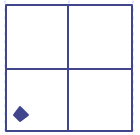
```
06073 * THAT CONCLUDES THE STRUCTURED COBOL PORTION OF THIS PGM...      FOprogm1
06074 * RETURN TO SPAGHETTI CODE!                                      FOprogm1
06075      GO TO 3159-CONTINUE-SPAGHETTI.                              FOprogm1
```



Implicit History

- Where “programming archaeology” comes into play:

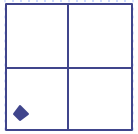
```
01188 ** MAC 07/17/95 METEOR 2QTR - BEGIN                          FOprogm1
01189            10  COM-FOSLSVC1-FLAG              PIC X(01).     FOprogm1
01190                88  GOTO-FOSLSVC1              VALUE 'Y'.     FOprogm1
01191 ** MAC 07/17/95 METEOR 2QTR - END                            FOprogm1
01192            10  COM-HOLD-CANCEL-REASON         PIC X(02).     FOprogm1
01193                88  COM-FRAUD-CANCEL-REASON    VALUES 'BA' 'BB'FOprogm1
01194                                                     'BD' 'BE'.FOprogm1
01195            10  COM-SKIP-FLAG-ST               PIC X.         FOprogm1
01196            10  SKIP-FLAG-SW-OTHER             PIC X.         FOprogm1
01197 ** 03/10/95 MC - BEGIN                                       FOprogm1
01198            10  COM-PA-TELCO-ID                PIC X(04).     FOprogm1
01199            10  COM-CX33-ANI-READ-KEY.                        FOprogm1
01200                15  COM-ANI-CUSTOMER-ID-T      PIC X(08).     FOprogm1
01201                15  COM-ANI-PHONE-NBR-T        PIC X(10).     FOprogm1
01202                15  COM-ANI-STATUS-T           PIC X(01).     FOprogm1
01203 ** 03/10/95 MC - END                                         FOprogm1
01204 ** MAC 07/17/95 METEOR 2QTR - BEGIN                          FOprogm1
01205            10  COM-ACN-PRD-DATE               PIC X(08).     FOprogm1
01206            10  COM-ACN-PRD-IND                PIC S9(09) COMP.FOprogm1
01207            10  COM-PREV-ACN-PRD-IND           PIC S9(09) COMP.FOprogm1
01208 ** MAC 07/17/95 METEOR 2QTR - END                            FOprogm1
01209 **         10  COM-CUST-CIC-CODE              PIC X(05).     FOprogm1
01210 ** JHK 11/10/01 SB LOCAL - BEGIN.                           FOprogm1
01211            10  COM-BUS-SEG-IND                PIC X(1).      FOprogm1
01212 ** JHK 11/10/01 SB LOCAL - END.                             FOprogm1
```



Implicit History

- More “programming archaeology”
  - Layers of edits

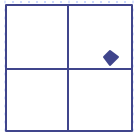
```
02871 ** MAC 07/17/95 METEOR 2QTR - BEGIN                              FOprogm1
02872      MOVE A-ACN-PRD-DATE OF RECORD                               FOprogm1
02873                           TO COM-ACN-PRD-DATE.                   FOprogm1
02874 *PS 07/07/97 COM-ACN-PRD-IND NO LONGER NEEDED, SET TO ZERO AND USEFOprogm1
02875 *AS FLAG TO INDICATE IF 500 OR PAGER ON FB-SERVICE (SEE GET-METEORFOprogm1
02876      MOVE 0                TO COM-ACN-PRD-IND.                   FOprogm1
02877 *    MOVE A-ACN-PRD-IND OF RECORD                                FOprogm1
02878 *                         TO COM-ACN-PRD-IND.                    FOprogm1
02879 *PS 07/07/97 END                                                 FOprogm1
02880      MOVE A-PREV-ACN-PRD-IND OF RECORD                           FOprogm1
02881                           TO COM-PREV-ACN-PRD-IND.               FOprogm1
02882 ** MAC 07/17/95 METEOR 2QTR - END                                FOprogm1
```



Implicit History

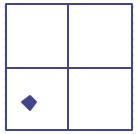
- Much less prevalent in Linux Kernel, an atypical example:

```
/*
 * This needs some heavy checking ...
 * I just haven't the stomach for it. I also don't fully
 * understand sessions/pgrp etc. Let somebody who does explain it.
 *
 * OK, I think I have the protection semantics right.... this is really
 * only important on a multi-user system anyway, to make sure one user
 * can't send a signal to a process owned by another.  -TYT, 12/12/91
 *
 * Auch. Had to add the 'did_exec' flag to conform completely to POSIX.
 * LBT 04.03.94
 */ (sys.c)
```



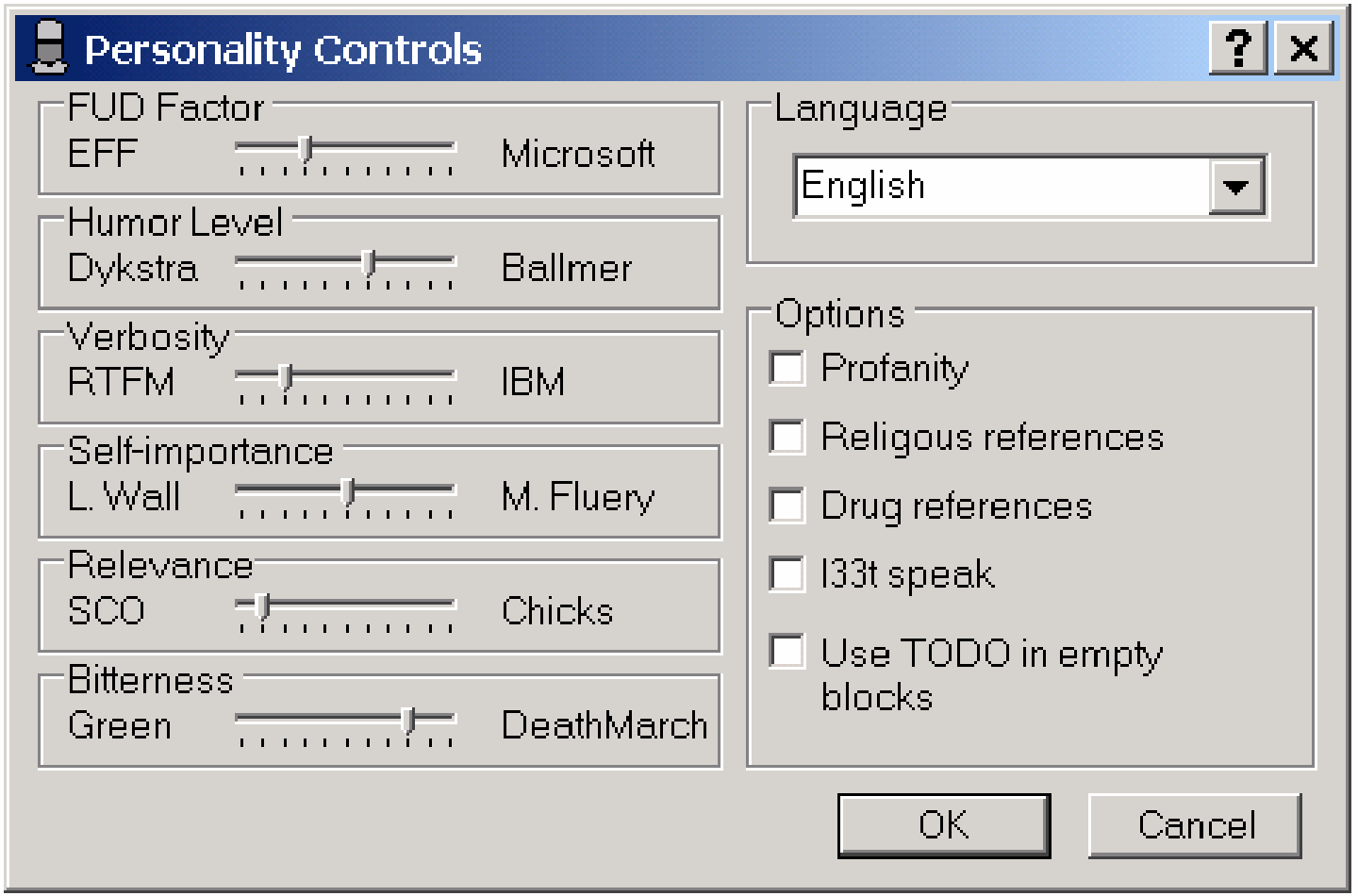
- This example is much more typical of longer comments

```
/*
 * 'User priority' is the nice value converted to something we
 * can work with better when scaling various scheduler parameters,
 * it's a [ 0 ... 39 ] range.
 */
#define USER_PRIO(p)                    ((p)-MAX_RT_PRIO)
#define TASK_USER_PRIO(p)    USER_PRIO((p)->static_prio)
#define MAX_USER_PRIO                   (USER_PRIO(MAX_PRIO))
(sched.c)
```



Technology History within Comments

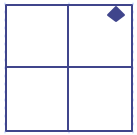
- Identity-orientated comments: informative in other ways
  - The Commentator – a satirical faux comment generator

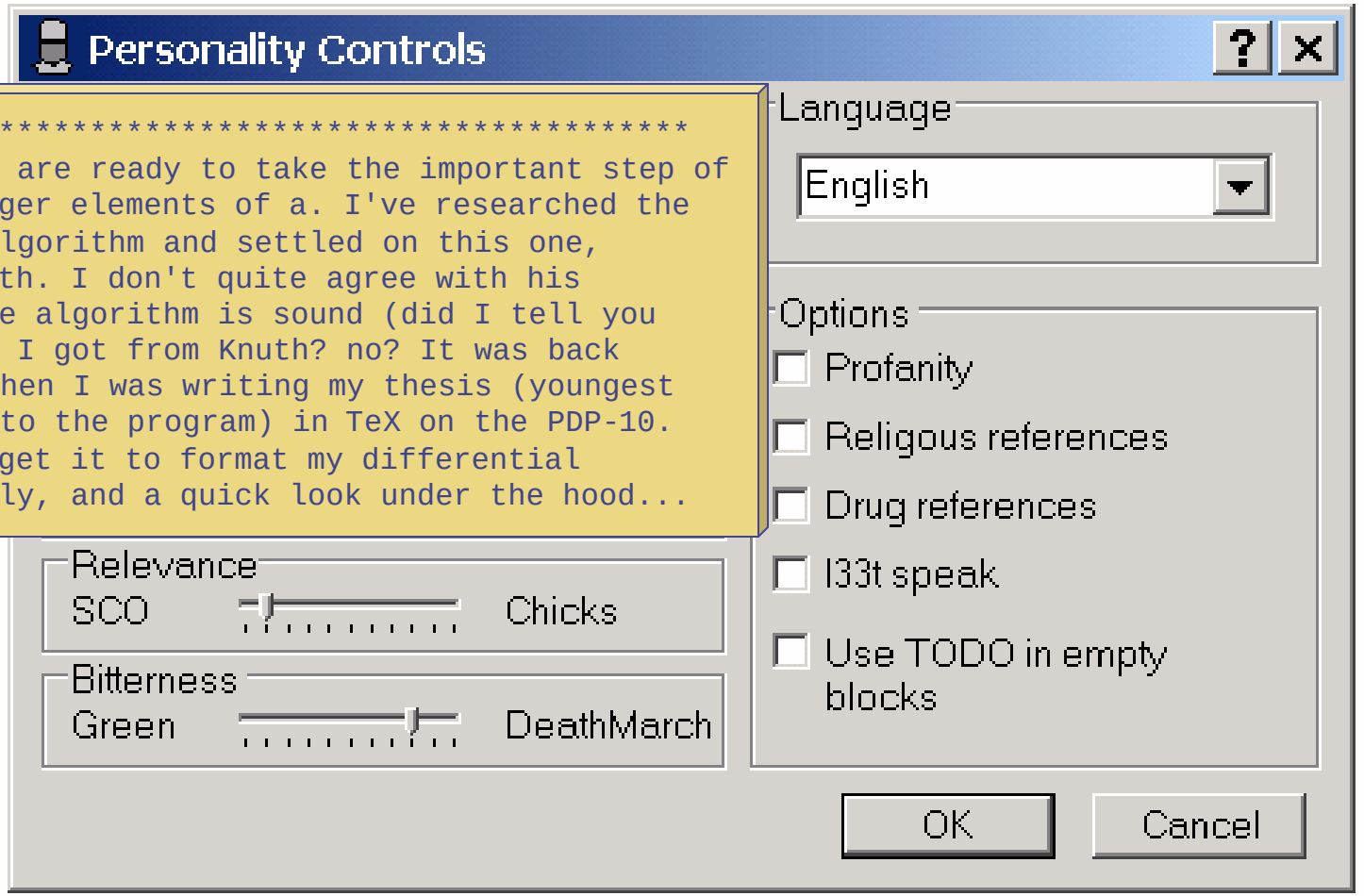


Technology History within Comments

- Identity-oriented comments: informative in other ways
  - The Commentator – a satirical faux comment generator

```
/*********************************************************
* okay, finally we are ready to take the important step of
* summing the integer elements of a. I've researched the
* most efficient algorithm and settled on this one,
* presented by Knuth. I don't quite agree with his
* reasoning but the algorithm is sound (did I tell you
* about the cheque I got from Knuth? no? It was back
* in my uni days when I was writing my thesis (youngest
* ever accepted into the program) in TeX on the PDP-10.
* I just couldn't get it to format my differential
* equations properly, and a quick look under the hood...
```





Conclusion

- Comments are both normative & identity-oriented
  - Structure (form & function)
    - Reflects & shapes programmer community & personal identity
    - Reflects levels of association with the machine
- Maintains historical continuity
  - Continuous historical commentary
  - Whether normative or identity-oriented
  - For both community norms and knowledge
- Much more prevalent in Corporate sample than in Linux kernel
  - Some normative differences
- Extremely valuable research archive for research purposes

History?

- "Lie in the comments"
  - Remember that comments are not tied to the code
    - They can be wrong