Open Source Community Through Comments in Code

Graduate Research Symposium
College of William & Mary, Williamsburg, VA

stuart mawler
2007/03/30
Key Argument

- Usual view
  - Software running end product
    - e.g., Microsoft Word or PowerPoint

- Alternative view
  - Software as document, manuscript, corpus, or text
  - Consumed among communities of programmers

- This Paper
  - Uncovers the social roles of these texts
  - Looks at a sub-community
    - Linux Open Source Programmers working on the operating system kernel
Context: STS Literature

- Programming
- Laboratory Science
- Executable Texts
- Inscription Devices

- More Controversial
- Less Controversial

- are the Products
- are the Communications vehicles

- are Communications vehicles
- are Products
Linguistic Requirements

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

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

- in C++ (online programming guide)

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

- in C (Linux kernel)

```c
/* 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

---

/** Arch-specific enabling code. */ (cpuc)

#ifndef CONFIG_HOTPLUG_CPU
    //Development By : Jigar Mehta
    //Date : [ & now() & ]
#endif

@endif /*CONFIG_HOTPLUG_CPU*/ (cpu.c)
Natural Language Dominance

- There are no “native” programming language speakers
  - Natural Language used to communicate even simple ideas
    - So much so, that it is used where it might be redundant

```c
/*
 * encode an unsigned long into a comp_t
 *
 * This routine has been adopted from the encode_comp_t() function in
 * the kern_acct.c file of the FreeBSD operating system. The encoding
 * is a 13-bit fraction with a 3-bit (base 8) exponent.
 */

#define MANTSIZE 13 /* 13 bit mantissa. */
#define EXPSIZE 3 /* Base 8 (3 bit) exponent. */
#define MAXFRACT ((1 << MANTSIZE) - 1) /* Maximum fractional value. */ (acct.c)
```
Social Representation via Languages

- Specific Languages as metaphors for the communities they represent
  - "Java" as a metaphor for the "cowboy" programmer up all night...
  - "COBOL" as boring business-y language (Common Ordinary Business-Oriented Language)
  - C as representative of arcana
Mapping Code Comments

/* Comments have both Form and Function, which exist along a continuum from Normative to Identity-oriented. */

/* Each comment can be plotted against these two axes. */
"Good" Comments

/* Comments have both Form and Function, which exist along a continuum from Normative to Identity-oriented.*/

/* Each comment can be plotted against these two axes.*/
/* Comments have both Form and Function, which exist along a continuum from Normative to Identity-oriented. */

/* Each comment can be plotted against these two axes. */
Good Comments

- How to Write Unmaintainable Code
  - Counter-normative
  - Shows how comments “should” be written

“Any fool can tell the truth, but it requires a man of some sense to know how to lie well.”
~ Samuel Butler (1835 - 1902)

“Incorrect documentation is often worse than no documentation.”
~ Bertrand Meyer
Constructing Social & Personal Identity

- The Commentator
  - Someone knows what comments are really doing...
The Texts Speak for Themselves...

- Letting individual personality come through

```c
/*
 * Select whether the frequency is to be controlled
 * and in which mode (PLL or FLL). Clamp to the operating
 * range. Ugly multiply/divide should be replaced someday.
 * /(time.c)
*/
```

- Allowing comments to act as a “dialog”

```c
if (IS_ERR(p)) { /* Should never happen since we send PATH_MAX */
    /* FIXME: can we save some information here? */
    audit_log_format(ab, "<too long>");
} else
    audit_log_untrustedstring(ab, p);
kfree(path);
}(acct.c)
```
“We”: Elevating the Discipline

- The “academic” lecture

```c
/*
 * If we're in an interrupt or softirq, we're done
 * (this also catches softirq-disabled code). We will
 * actually run the softirq once we return from
 * the irq or softirq.
 *
 * Otherwise we wake up ksoftirqd to make sure we
 * schedule the softirq soon.
 */(softirq.c)
```
“We”: Enforcing Boundaries

- **Boundaries with users**

```c
/* We only trust the superuser with rebooting the system. */
if (!capable(CAP_SYS_BOOT))
    return -EPERM;

/* For safety, we require “magic” arguments. */
if (magic1 != LINUX_REBOOT_MAGIC1 ||
    (magic2 != LINUX_REBOOT_MAGIC2 &&
     magic2 != LINUX_REBOOT_MAGIC2A &&
     magic2 != LINUX_REBOOT_MAGIC2B &&
     magic2 != LINUX_REBOOT_MAGIC2C))
    return -EINVAL;
```

- **With outsiders**

```c
/*
* setuid() is implemented like SysV with SAVED_IDS
*
* Note that SAVED_ID's is deficient in that a setuid root program
* like sendmail, for example, cannot set its uid to be a normal
* user and then switch back, because if you're root, setuid() sets
* the saved uid too. If you don't like this, blame the bright people
* in the POSIX committee and/or USG. Note that the BSD-style setreuid()
* will allow a root program to temporarily drop privileges and be able to
* regain them by swapping the real and effective uid.
*/
```
"We": Linking Programmer & System

- First: Anthropomorphize the System

  /* Some compilers disobey section attribute on statics when not initialized -- RR */ (softirq.c)

- Then becoming part of the system is “natural”...

  /*
   * We're trying to get all the cpus to the average_load, so we don't want to push ourselves above the average load, nor do we wish to reduce the max loaded cpu below the average load, as either of these actions would just result in more rebalancing later, and ping-pong tasks around. Thus we look for the minimum possible imbalance.
   * Negative imbalances (*we* are more loaded than anyone else) will be counted as no imbalance for these purposes -- we can't fix that by pulling tasks to us. Be careful of negative numbers as they'll appear as very large values with unsigned longs.
   */ (sched.c)
Conclusion

- Comments are both normative & identity-oriented
  - Structure (form & function)
    - Reflects & shapes programmer community
      - Collegial, cooperative, collective
    - Reflects levels of association with the machine
- Commenting is a critical element of
  - Group identity
  - Personal identity
- Pragmatic / business implications
- Future research
A Contrary View

- Does the “We” Construction really indicate collegiality?