Zen and the art of Free Software

Know your users, know yourself

July 12, 2005

14 slides

Enrico Zini (enrico@debian.org)

Enrico Zini enrico@debian.org

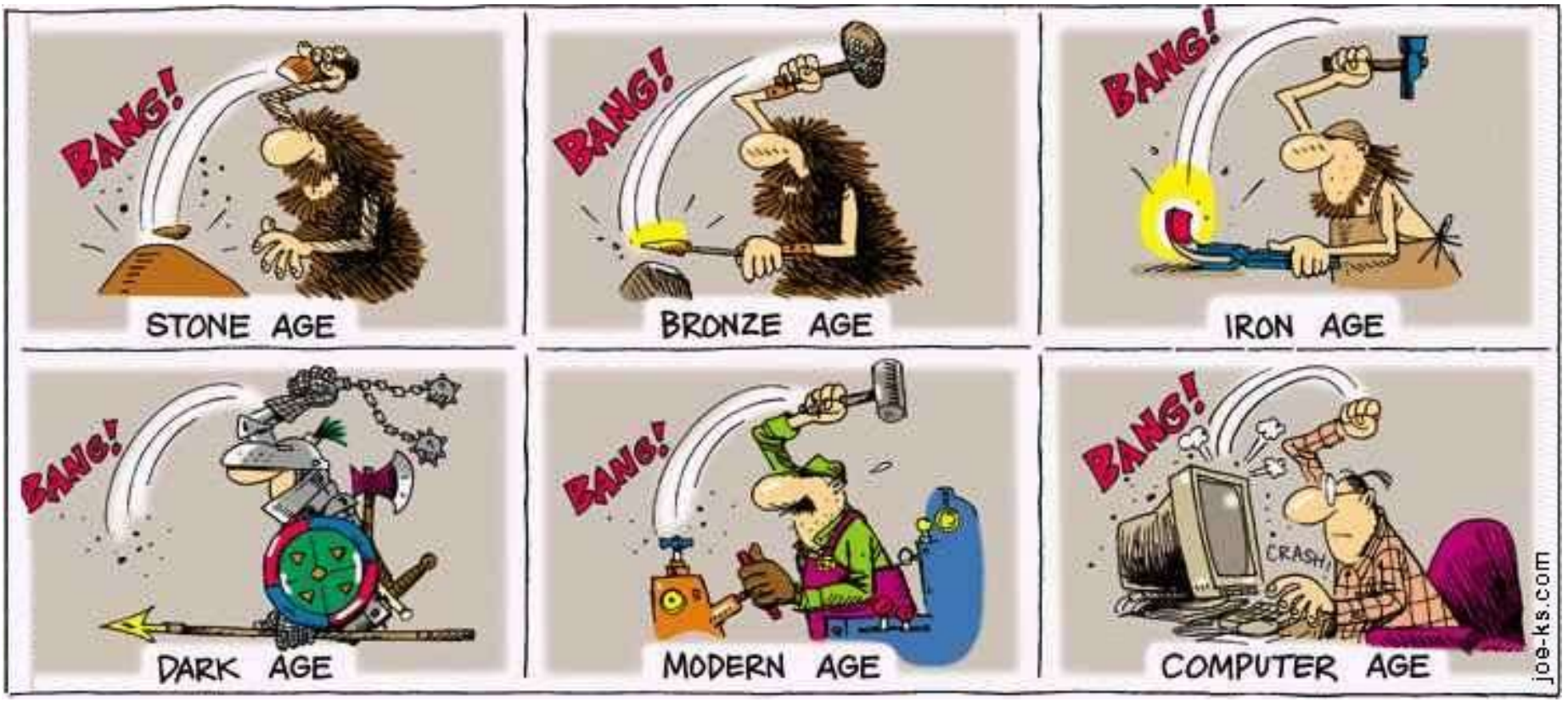
What I'll be talking about

- I will introduce 2 key concepts: "situation" and "frustration"
- I will explain how to do:
  - Social specifications
  - Social design
  - Social testing
  - Social debugging

...all of this in 45 minutes: get rrrrready!

Enrico Zini enrico@debian.org

Software shapes society



Enrico Zini enrico@debian.org

Software shapes identity

Quiz time!

```
X-Mailer: Microsoft Outlook, Build 10.0.2616

User-Agent: Mozilla Thunderbird 0.5 (X11/20040306)

User-Agent: Mutt/1.5.6i
```

Which of these persons would you trust more?

Enrico Zini enrico@debian.org

Situation

Who are you?

I mean, here and now?

(thinking situated actions is like thinking life with runtime information)

Enrico Zini enrico@debian.org

Frustration

n. The feeling that accompanies an experience of being thwarted in attaining your goals

from WordNet 2.0 dict database

(how do you usually cope with frustration?)

Enrico Zini enrico@debian.org

Social specifications: users

Who are you developing for? You can design this using *personas*:

Persona: detailed description of your average, non existing user

(are you part of your users? try making a persona for yourself using your software)

Enrico Zini enrico@debian.org

Social specifications: goals

What do your users want to do with your software?

Your users have *goals* (everyone has, after all...):

Personal: "don't feel stupid", "have the computer do most of the work"

Work: "submit paper to the conference", "increase sales"

Practical: "enter the damn data", "look up the address in the directory"

False: "use few CPU cycles", "be a web application", "be easy to use"

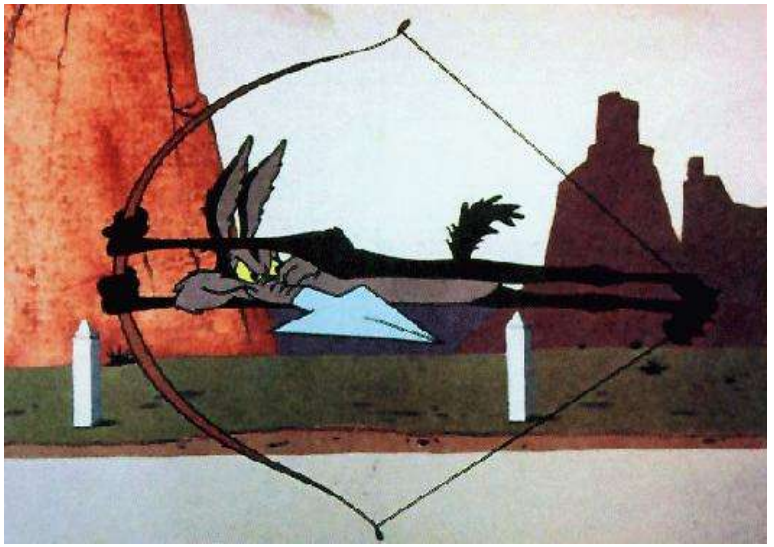
(small exercise: what are your goals at this talk?)

Enrico Zini enrico@debian.org

Social design: tasks

How are your users going to attain their goals?

- Take the description of your *persona*
- *Situate* him/her in the appropriate environment
- Take the list of *goals* you have
- What is the best way for that person to reach her goals with the minimum amount of *frustration*?



Welcome to the world of "*Task Analysis*"!

(with what tasks do you reach your goals? Is that how you would design it?)

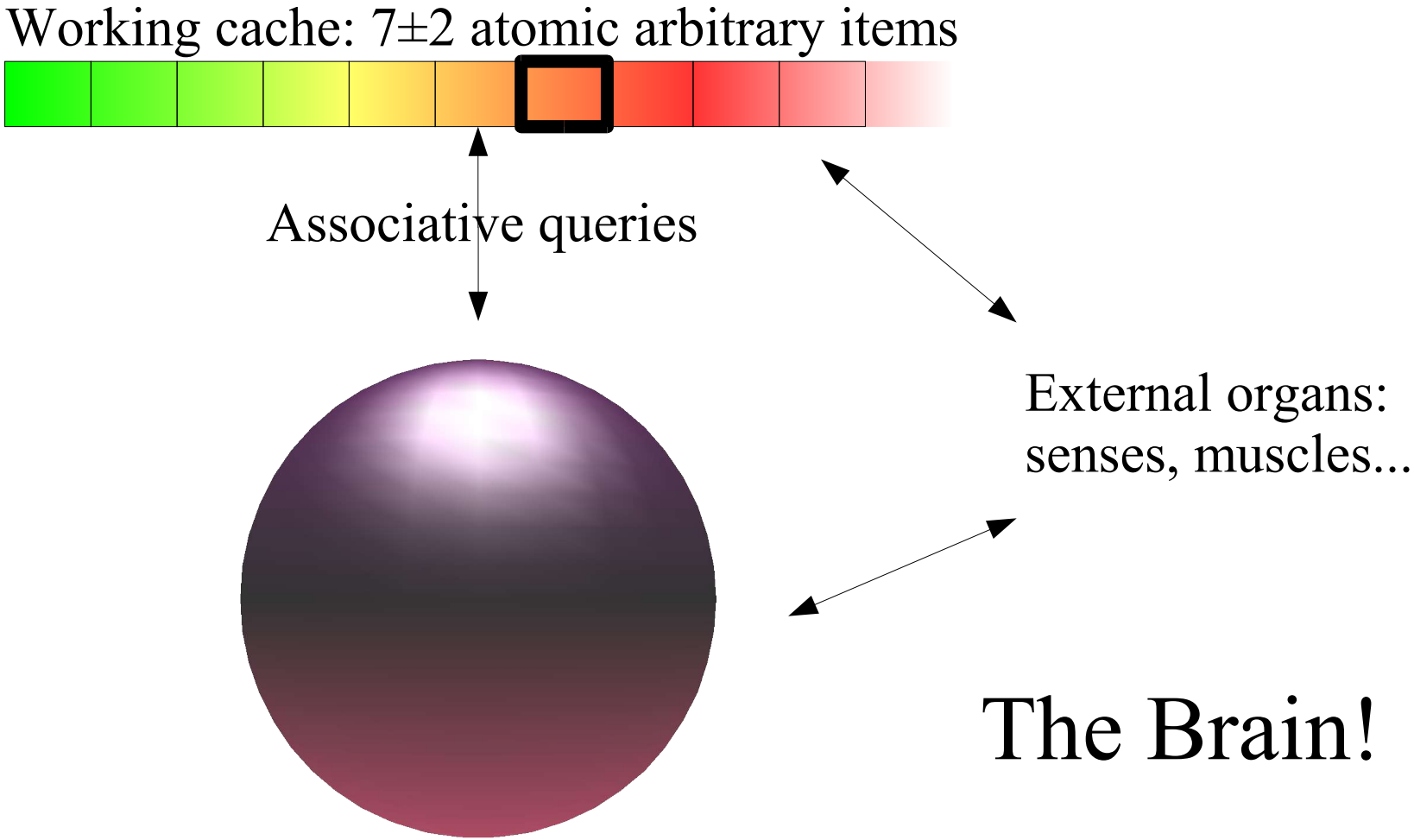
Social design: politeness

Cooper's rules of software politeness:

I. Polite Software Is Interested in Me
II. Polite Software Is Deferential to Me
III. Polite Software Is Forthcoming
IV. Polite Software Has Common Sense
V. Polite Software Anticipates My Needs
VI. Polite Software Is Responsive
VII. Polite Software Is Taciturn About Its Personal Problems
VIII. Polite Software Is Well-Informed
IX. Polite Software Is Perceptive
X. Polite Software Is Self-Confident
XI. Polite Software Stays Focused
XII. Polite Software Is Fudgable
XIII. Polite Software Gives Instant Gratification
XIV. Polite Software Is Trustworthy

(software is made to serve us: we may as well make it polite! How polite is yours?)

Social design: the magic number 7±2



(try to spot the magic number 7±2 in your everyday life!)

Enrico Zini enrico@debian.org

Social testing

Nielsen's Euristic Evaluation Technique:

I. Visibility of system status
II. Match between system and the real world
III. User control and freedom
IV. Consistency and standards
V. Error prevention
VI. Recognition rather than recall
VII. Flexibility and efficiency of use
VIII. Aesthetic and minimalist design
IX. Help users recognize, diagnose, and recover from errors
X. Help and documentation

(try the checklist on one of the interfaces you hate most!)

Social debugging

Flanagan Critical Incident Technique

Critical Incident: an interesting effective or ineffective behaviour

Debugging technique (like *gdb <program> core*):

- Describe what led up to the situation
- Exactly what did the person do that was especially effective or ineffective?
- What was the outcome or result of this action?
- Why was this action effective, or what more effective action might have been expected?

(now you know what to ask when someone writes you "this crap doesn't work!")

Enrico Zini enrico@debian.org

Wrapup

Many free software projects don't seem to have a direction: you've seen a possible way to give one to yours

Many free software projects don't care much about their users: you've seen how to work with them, and make them happy

Knowing how to make your users happy is very important: especially when you could be among them!

Conclusion

Happiness is very important!

It is the only road to Total World Domination!