A Sociology of the Free/Libre Open Source Software Innovation

Yuwei Lin

Department of Information Systems
Vrije Universiteit Amsterdam

Schedule

First session:

1. Technological innovation and society
2. Social worlds theory: A community of practices
3. Sense-making and negotiation of meanings
4. Cross-boundary collaboration
5. Community-based innovation: EMACS

Second session:

6. Hybrid collaboration between the FLOSS community and software firms

How to Understand the Relationships between Science, Technology and Society?

Qualitative Research Methods

- Ethnography on-line and in physical environment
- Interview
- Observation
- Documentary analysis
- Narrative/conversation analysis

Sociology of Weber

(Max Weber 1864 – 1920)

Humans are reflexive, meaning-creating beings who actively develop their self-image and social identity and negotiate their way through the social arrangements and relationship in which they find themselves. They do not act in isolation from each other but draw on common cultural, social and political resources. While these to a certain degree structure or shape their behaviour, what they do is never merely an act of conformism. It is also coloured by and oriented towards their own meanings and understandings.

(Bilton *et al*. 2002: 16)

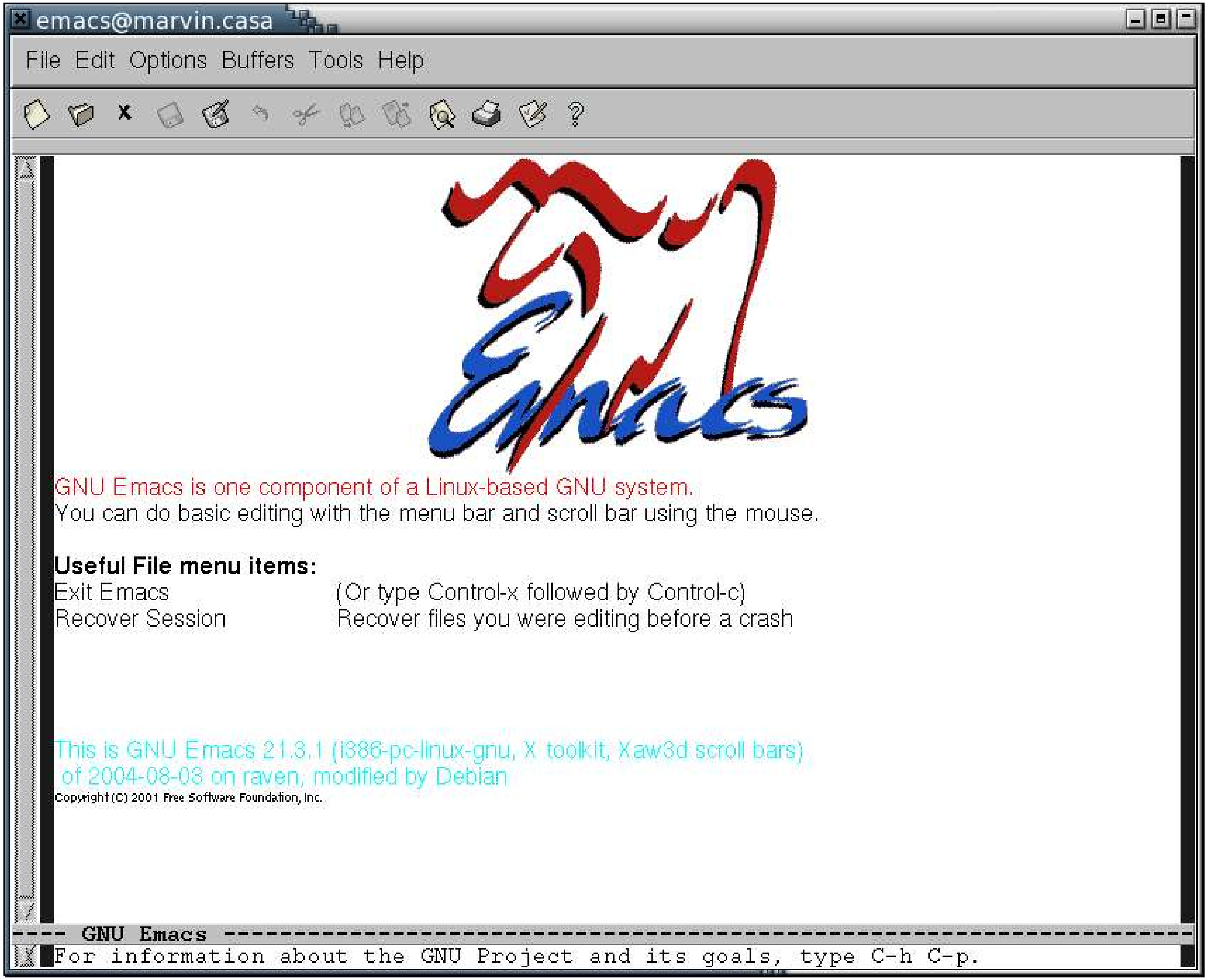
Shared Meanings

Social actors work out their lives with shared meanings, and in doing so, they make sense of what they and others are doing in their social lives.

They produce, reproduce and amend these meanings, understandings and knowledge by applying them and/or subtly refining them in practical circumstances and context: norms and values, rather than being absolute, substantial things from which social order is unambiguously constructed, can be better regarded as negotiable ideas.

(Bilton et al. 2002: 17)

EMACS (Editing MACroS)



GNU Emacs is one component of a Linux-based GNU system.
You can do basic editing with the menu bar and scroll bar using the mouse.

Useful File menu items:

```
Exit Emacs              (Or type Control-x followed by Control-c)
Recover Session         Recover files you were editing before a crash
```

This is GNU Emacs 21.3.1 (i386-pc-linux-gnu, X toolkit, Xaw3d scroll bars)
of 2004-08-03 on raven, modified by Debian
Copyright (C) 2001 Free Software Foundation, Inc.

```
---- GNU Emacs ------------------------------------------------------------
For information about the GNU Project and its goals, type C-h C-p.
```

Why EMACS?

- EMACS provides a valuable illustration of the heterogeneous and contingent FLOSS innovation system
- A live proof of a GPL-ed product (ca. 30 years-old)
- It is widely deployed in multiple sites for various uses
- It shows a complex network between actors themselves and artefacts (tools)
- The EMACS story serves to illustrate how the innovation occurred within the FLOSS community, but one that is then adopted and deployed in other social contexts, including the commercial sector.
- It spans the period that saw the origins of the free software movement and the subsequent development of a broader FLOSS social world.

Background Information about EMACS

- 1976 Stallman and his colleagues wrote the editor EMACS to update the previous editor TECO on an ITS, the software running on the PDP10 mini-computer at the MIT AI Lab
- TECO allowed users to employ Macros (Command Strings)
- WYSIWYG
- Main figures: Richard Stallman (TMACS & EMACS)
  Carl Mikkelson (WYSIWYG for TECO)
  Guy Steele (Compiling Macros)
  David Moon (EMACS)
  Dan Weinreib (EMACS & EINE & ZWEI)
  Bernie Greenberg (Multics Emacs)
  James Gosling (Gosling Emacs)
  Jamie Zawinski(Lucis Emacs --> XEMacs)

Social Contract – earlier GPL

- Stallman started to employ a social contract that rendered the communal sharing on the basis of its distribution
- OTOH, users are able to modify and redistribute the code; OTOH, they are asked to report back the extensions
- to strengthen the functionality of EMACS and creating a reciprocal understanding of the written codes through sharing solutions
- to expand socio-technical network
- Such a network widens the range of 'digital epistemologies' (i.e. ways of odering and knowing problems) and their expressions in the innovation system

Extensibility & Customisation

XEmacs is more flexible than any other editor I know: it allows transparent multifiles editing; it is easily extendable -- I can modify or create modes if I need to; and has an enormous range of tools such as integrated source code control (CVS, clear case) and programming language specific syntax aware operations. For example: auto-completion, code-block matching, and block commenting.

(YLUG110104)

Artefacts and Habits

I prefer using emacs for complicated editing/coding jobs because of it's features. But I prefer it for simple and quick jobs like config file editing, hacking scripts, etc So it horses for courses.

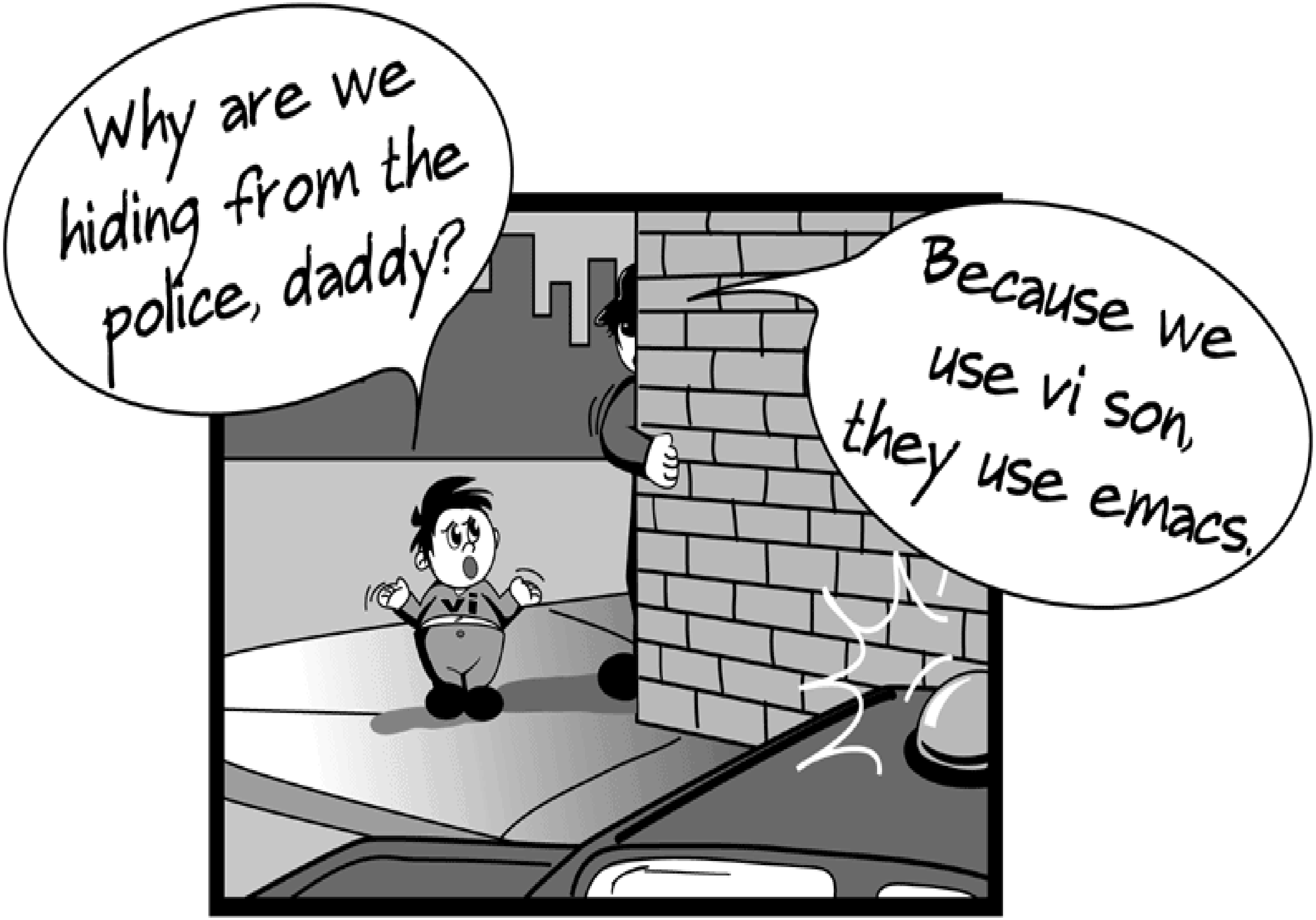
(JJ0104)

Because my fingers are friendly with all the commands I need!! I picked up EMACS first and haven't seen the need to change, I especially like the way it handles copy/paste/delete of rectangles within a text file.

(GH120104)

Material Culture and Habitus

Since EMACSen are used in many different ways and denotes various socio-cultural meanings, diverse projects have symbolised users’ habits and preferences (socially and technically). In adopting specific tools and participating in specific projects, users are attached to the artefacts. These artefacts grow to be norms to demarcate boundaries.


Why are we hiding from the police, daddy?
vi
Because we use vi son, they use emacs.

How is EMACS developed? - The Common Practice: Problem-Solving

Software like everything else is driven by a need to solve a particular problem. There are some developers who are innovative for the sake of being creative. I think commercial software is mainly driven by the market and the need to sell a product while open source software is driven mostly by the need of a solution.

(DY011202)

Situated Solutions

- In defining and solving a problem, one employs those materials that are available and negotiate or cooperate with actors within or across the boundary set by the problem itself.
- There is no standardised path to be followed; each resolution is a result of situated practices and knowledge.
- Each episode of problem-solving is materially-grounded, textured and situated.

Problem-Solving

- The more a problem is accessible, as noted above, the more diverse actors will be invited to participate in the innovation group centring on the problem.
- As there is no single perfect solution for a problem, multiple voices and silences should always be welcome.
- If a problem is presented in a perceivable/accessible way, it will encourage more participants to craft solutions.
- Well-defined problems in which the given information, operations, and goal are clearly specified will more likely to have solutions than ill-defined problems.
- An expert can articulate the enqueries more specifically and completely than could someone new to the domain.

Innovation Elements

- **T**eam Work and Communication
- **M**aterials & Social Resources
- **S**hared Interests
- **S**hared Goals & Trust

Teamwork and Communication

- **W**hen a new member of the team is recruited (formally or informally), s/he seems to share some common interests of the group but s/he also brings some differences into the group.
- **I**t appears that the more actors are involved in the team, the better innovation is mobilised, though less easy it becomes to maintain a consensus.
- **N**egotiations about problems become a crucial aspect of teamwork.
- **V**irtual and Face-to-Face Communications

Material Resources

A coherent set of new and redefined functions can be bound into a library so that the user can load them together conveniently. Libraries enable users to publish and share their extensions, which then become effectively part of the basic system. By this route, many people can contribute to the development of the system, for the most part without interfering with each other. This has led the EMACS system to become more powerful than any previous editor.

(Stallman 1998)

Social Resources

User customization helps in another subtler way, by making the whole user community into a breeding and testing ground for new ideas. Users think of small changes, try them, and give them to other users—if an idea becomes popular, it can be incorporated into the core system. When we poll users on suggested changes, they can respond on the basis of actual experience rather than thought experiments.

(Stallman 1998)

TACIT KNOWLEDGE & KNOWLEDGE-SHARING

Networking the Social and the Technical

> [E]ven though there was no organized political thought relating the way we shared software to the design of Emacs, I’m convinced that there was a connection between them, an unconscious connection perhaps. I think that it’s the nature of the way we lived at the AI Lab that led to Emacs and made it what it was.
>
> (Stallman 2003)

Shared Interests

The first and easiest way to find people who will immediately believe the statement, invest in the project, or buy the prototype is to tailor the object in such a way that it caters to these people’s **explicit interests**. As the name ‘inter-esse’ indicates, ‘interests’ are what lie *in between* actors and their goals, thus creating a tension that will make actors select only what, in their own eyes, helps them reach these goals amongst many possibilities.

(Latour 1987: 108-9)

Stallman's Thanksgiving Message

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for GNU's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.

(27 September 1983 on net.unix-wizard)

Meanings of Stallman's Message

- creation of a social network of crafting a new operating system
- enrollment of shared interests (post on the unix-wizard usergroup)
- uncertainty and risk
- democratic innovation (soft boundary)
    * projects split off into individual sub-networks because boundaries of the original networks are too elastic to concentrate on any central themes (everything is diverse).

Brief Summary

- How a software innovation network is created and developed in terms of classifications of problems, identifications of solutions and common interests.
- Actors and artefacts (tools) are brought together, interact and negotiate with each other to solve problems.
- Boundaries of networks, which determine access to innovation systems, vary in different contexts.
- A successful innovation, as discussed, is the one that manages to bring in as many actors and artefacts (tools) as possible by translating their interests to extend and mobilise the network as well as handles the uncertainties and risks emerging during the process.

Discussion

Why such a sociological analysis towards the FLOSS development important?

Exercise

Can you examine the development of a FLOSS project that you know of and identify its innovation network and elements from a sociological point of view?

e.g.

- Who are the actors (individuals and organisations) involved?
- What are the common practices?
- How do they communicate?

(4 people in a group, 10 minutes)