

A structurational model of leadership in virtual distributed groups

Kevin Crowston*, Hala Annabi[^],
and Robert Heckman*

*Syracuse University
School of Information Studies
Syracuse NY USA

[^]The Information School
University of Washington
Seattle, WA, USA

What is FLOSS?

- FLOSS = Free/Libre Open Source Software
- Software distributed under license that allows inspection, modification and redistribution of the source code
 - AKA free or libre software
- Examples: Linux, Apache, gcc, sendmail, X-windows, GNOME, GAIM, OpenOffice, etc.
 - ...as well as many lesser-known projects

Why FLOSS is interesting

- Mostly developed by distributed teams of volunteers coordinated via the Internet
 - Teams are largely self-organizing, without appointed leaders or clear indications of status
- Conway's law: Structure of the software reflects the structure of the team that develops it
- Successful FLOSS teams somehow overcome problems of distributed work

Our theory

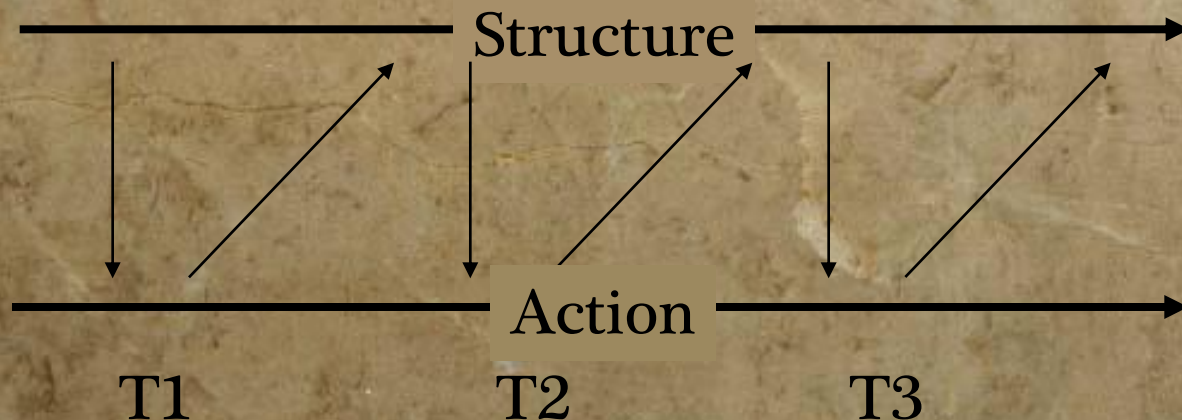
- Leadership in self-organizing distributed groups involves creating structures that guide group members' actions

Functional view of leadership

- Some behaviours serve leadership functions
 - Help group to achieve goals and perform effectively
 - One or more individuals may perform required leadership behaviours
- Example leadership behaviours
 - Task leadership behaviours
 - Organizing, coordinating and performing a task
 - Group maintenance leadership behaviours
 - Maintaining group morale, motivation and communication
 - Maintaining relations outside group

Structuration theory

- Premised on duality of structures
 - Rules and resources (structures) influence, guide or justify individual action
 - Actions taken reinforce or change structure
- Perspective provides insight into dynamics



Structures of signification

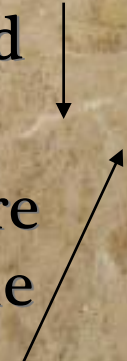
- Interpretive schemes create structures of signification
 - Shared mental models guide developers' actions ↓
- *Proposition 1:* Groups with practices that involve higher levels of socialization, conversation and narration will develop more highly developed shared mental models ↗
- *Proposition 2:* Group members who are more involved in socialization, conversation and narration will be recognized as leaders by other group members

FLOSS example (Apache)

“So long as you remembered to put in the #ifdef. Sometimes, people forget. With RCS, this is not a problem.

(A minor war story may be instructive, if only to let people know where I'm coming from. In the ai_httpd sources I've put up on ftp.ai.mit.edu, the nameserver cache is an option, so the code can be compiled at sites which don't do mmap(). My first cut at doing this left out an #ifdef around a line of modified code (the call to write_nameserver_cache in get_remote_host), meaning that while my modified server tested just fine, the base configuration could not be compiled after the patch. I fixed that, but this sort of human error is likely to happen again, and probably not just to me. If you really want accurate annotations as to what changed when, it's much better to have a machine do the work).”

Structures of domination

- Authoritative and allocative resources create structures of domination
 - Roles within project have differential access to code and documents, which guide and constrain actions
 - *Proposition 3:* Groups in which role definition functions are regularly performed will develop more clearly defined role structures.
 - *Proposition 4:* Group members who perform role definition functions such as task division and assignment will be recognized as leaders by fellow group members.
- 

FLOSS example (PLONE)

The normal flow is:

- 1. Author adds documentation*
- 2. Reviewer publishes documentation*
- 3. User reads documentation, has question/correction, adds comment*
- 4. Author gets email*
- 5. Author reads comment, corrects his article, removes the comment*

*(and if we had events, we could send a "thank you" mail here ;)
(also note that author can edit his content in-place after initial publication, no need for another workflow process.)*

- 6. The flow starts at (3) again.*

Structures of legitimation

- Norms create structures of legitimation
 - Explicit rules and implicit norms guide developers' actions
- *Proposition 5:* Groups with practices that involve high levels of collaborative, interactive problem solving, political negotiation, and experiential learning will develop clearer and more elaborate rules and norms
- *Proposition 6:* Members of a group who initiate the development of rules and norms, who implicitly or explicitly enforce rules and norms, and who socialize others in these rules and norms will be perceived as leaders by other members

FLOSS example (PLONE)

I assume that we will be pushing a lot of documentation in the next few weeks. I think it would be very helpful if documentation reviewers had a set of guidelines to follow for what to accept as-is, what to edit and publish, and what to reject. Things like

- o Short name format*
- o Descriptions*
- o Style/formatting of body text*
- o Version information*
- o Formatting*
- o Section organization*
- o Comments (when to add, when to remove)*

Perhaps the best thing would be to produce a checklist against which submitters and reviewers could gauge a piece of documentation. Hopefully, this should remove some ambiguity and resolve any disputes on what gets edited and what gets accepted.

I think it's important to do this sooner rather than later, as we want to establish PHC as a bonafide resource right from the outset. It doesn't have to be long or overly detailed, but it does have to be somewhat authoritative, which means that Alex or someone else core should produce the initial draft.

Conclusion

- Leadership in self-organizing distributed groups can be viewed as creating structures to guide actions of other members
- Currently applying this model in a “field” study of FLOSS development