

Free/libre and Open Source Software

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“*Attribution-ShareAlike 4.0 International*” license.



Preliminaries

- 1 Preliminaries
- 2 Motivation
- 3 Elektra
- 4 Content Overview
- 5 Organisation
 - Preliminaries
 - Grades
 - Assignments

BigBlueButton

- used for weekly virtual meetings
- is FLOSS
- raise the hand immediately on any issues
- use “Real Name @GitHubName” as your name
- you can connect several times, e.g., phone and laptop
- on technical problems, try another browser, e.g., recent Firefox or Chromium

Language

Materials are in English:

- Slides are in English
- Papers are in English
- Videos are in English

Language of the Talk?

Task

- A English
- B Slightly Prefer English
- C Both are fine
- D Slightly Prefer German
- E German

Video

I am trying to keep meetings short with many breaks.
You are allowed to:

- stretch
- move
- eat
- look somewhere else
- leave your place

Task

But please turn video on.

Task

Break.

Free/libre and Open Source Software

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“Attribution-ShareAlike 4.0 International” license.



Motivation

- 1 Preliminaries
- 2 Motivation**
- 3 Elektra
- 4 Content Overview
- 5 Organisation
 - Preliminaries
 - Grades
 - Assignments

FLOSS

Free/libre and Open Source Software allows you to:

- ① Use
- ① Share
- ② Study
- ③ Improve

the software (binary and source) for any purpose without restrictions.

Implications

There are countless implications¹:

- ① none of your knowledge becomes irrelevant after changing employee
- ② people can give you money so that you improve FLOSS for them
- ③ you can do research on FLOSS without any restriction
- ④ you can modify FLOSS as you see fit

¹many of which we will discuss in the course

First Assignment

- Have you already used FLOSS?
- Did you already participate in FLOSS?
- Which (other) implications are relevant for you?

Task

Discuss in breakout room and tell your partner's story.

Task

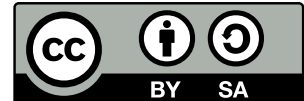
Break.

Free/libre and Open Source Software

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“Attribution-ShareAlike 4.0 International” license.



Elektra

- 1 Preliminaries
- 2 Motivation
- 3 Elektra**
- 4 Content Overview
- 5 Organisation
 - Preliminaries
 - Grades
 - Assignments

Elektra

- Elektra is one FLOSS initiative of what we discuss in this lecture.
- Elektra is developed at TU Wien (<https://libelektra.org>).



Misconfiguration

- *misconfigurations* [1, 9, 11, 12] are a major cause of system failures [4, 5, 10]
- much time is needed to fix misconfigurations [3, 4, 8, 12]

Elektra [6]

- ***configuration specification languages*** mitigate misconfigurations
- ELEKTRA is a framework implementing a modular ***configuration specification language*** for configuration settings
- ELEKTRA enables ***no-futz computing*** [2], i.e., error-prone “*tinkering or fiddling experimentally*” “*should be allowed, but should never be required*”

Goals

As written in doc/GOALS.md:

- Stability
- Simplicity
- Robustness
- Extensibility
- Performance

Task

Break.

Use Cases of Elektra

- Embedded systems
 - OpenWRT (distribution)
 - Broadcom (blue-ray devices)
 - Kapsch (cameras)
 - Toshiba (TVs)
- Server
 - Allianz (insurance)
 - TU Wien
 - puppet-libelektra
 - Other Universities
- Desktop
 - Oyranos
 - Redshift
 - LCDproc
 - KDE
 - GNOME

Goals

- Which use cases are specific to FLOSS?
- For which use cases exists more FLOSS?

Task

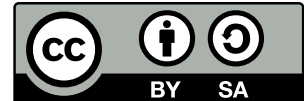
Think, Pair, Share.

Free/libre and Open Source Software

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“Attribution-ShareAlike 4.0 International” license.



Content Overview

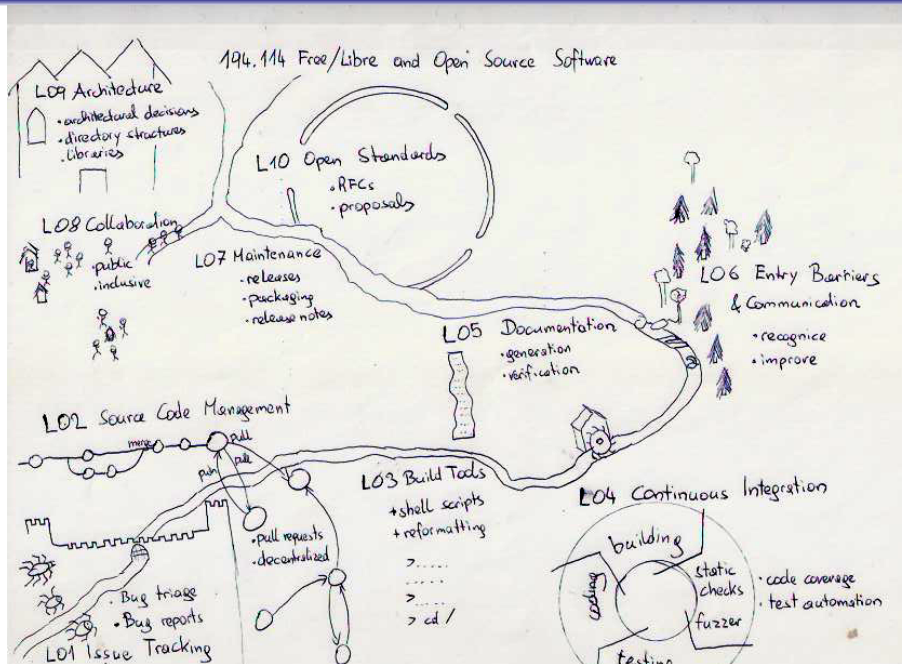
- 1 Preliminaries
- 2 Motivation
- 3 Elektra
- 4 Content Overview**
- 5 Organisation
 - Preliminaries
 - Grades
 - Assignments

learning outcomes:

- remember learning outcomes
- remember the topics

learning outcomes:

- participate in FLOSS initiatives,
- found new FLOSS initiatives,
- use FLOSS methods in your business context.



In which FLOSS topics are you interested?
(Can be other topics not mentioned.)

Task (1)

Discuss topics with your partner.

Task (2)

Write down the most interesting topics in the shared notes.

Free/libre and Open Source Software

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“Attribution-ShareAlike 4.0 International” license.



Organisation

- 1 Preliminaries
- 2 Motivation
- 3 Elektra
- 4 Content Overview
- 5 **Organisation**
 - Preliminaries
 - Grades
 - Assignments

Communication

- TUWEL
<https://tuwel.tuwien.ac.at/course/view.php?idnumber=194114-2021W>
- TISS <https://tiss.tuwien.ac.at/course/courseDetails.xhtml?courseNr=194114&semester=2021W>
- GitHub <https://git.libelektra.org>
- EMail markus.raab@complang.tuwien.ac.at
- before/after/during meetings

Inverted Classroom

Meetings are most Wednesday 09:00 c.t. - 11:00 (max.)

- always read/watch the material in advance
- TUWEL already contains materials for L01
- within meetings we will do recapitulations, discussions, etc.

Previous Knowledge

- Obviously *no* prior knowledge about FLOSS necessary.
(If you already have experience, you can use it in your presentation and assignments.)
- Understanding of software engineering and software requirements is preferable.
- Programming skills is a must.

Programming Languages

Elektra supports following programming languages:

- C¹
- C++¹
- Java¹
- Python¹
- Rust
- Go
- Lua
- Ruby

You can use either of these languages.

¹support on problems with the programming language by the lecturer

You will get a grade once you did the registration H0:

- Registration in TISS
- TUWEL Forum Entries
- Upload StudentID
- Create Issue with [FLOSS H0] in the title

Grade is calculated from following parts:

30 %: homework

30 %: teamwork

30 %: project

10 %: presentation

Furthermore, extrapoints can be earned in the lecture.

To get a positive grade:

- All parts must be done.
- All parts must be positive.

Presentation

- About anything related to FLOSS.
- It must be from your experience.
- Usually about your project in the end of the term (26.1.2022).

Deadlines

- if you make submissions earlier, you get feedback earlier
- dates are both in “semester schedule.pdf” and calender of TUWEL

There are up to three deadlines for each homework, teamwork or project:

- deadline for submission of the work
- deadline for review (review the submission of others)
- deadline for corrections (based on the feedback of submission)

Questions?

Task

Register for the course by doing H0 until 13. 10. 2021, 23:59.

Task

Any questions?

- [1] Mona Attariyan and Jason Flinn. Automating configuration troubleshooting with dynamic information flow analysis. In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, pages 1–11, Berkeley, CA, USA, 2010. USENIX Association.
- [2] David A. Holland, William Josephson, Kostas Magoutis, Margo I. Seltzer, Christopher A. Stein, and Ada Lim. Research issues in no-futz computing. In *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*, pages 106–110. IEEE, May 2001. doi: 10.1109/HOTOS.2001.990069.
- [3] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfiguration. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, pages 3–16, New York, NY, USA, 2002. ACM. ISBN 1-58113-570-X.
- [4] David Oppenheimer, Archana Ganapathi, and David A. Patterson. Why do Internet services fail, and what can be done about it? In *USENIX Symposium on Internet Technologies and Systems*, volume 67. Seattle, WA, 2003.

- [5] Soila Pertet and Priya Narasimhan. Causes of failure in web applications (cmu-pdl-05-109). *Parallel Data Laboratory*, page 48, 2005.
- [6] Markus Raab. Elektra: universal framework to access configuration parameters. *The Journal of Open Source Software*, 1(8):1–2, December 2016. doi: 10.21105/joss.00044. URL <http://dx.doi.org/10.21105/joss.00044>.
- [7] Markus Raab and Gergő Barany. Introducing context awareness in unmodified, context-unaware software. In *Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE*,, pages 218–225. INSTICC, ScitePress, 2017. ISBN 978-989-758-250-9. doi: 10.5220/0006326602180225.
- [8] Ariel Rabkin and Randy Katz. Static extraction of program configuration options. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 131–140. IEEE, 2011.

- [9] Ya-Yunn Su, Mona Attariyan, and Jason Flinn. Autobash: Improving configuration management with operating system causality analysis. pages 237–250, 2007. doi: 10.1145/1294261.1294284. URL <http://dx.doi.org/10.1145/1294261.1294284>.
- [10] Avishai Wool. A quantitative study of firewall configuration errors. *Computer*, 37(6):62–67, 2004.
- [11] Tianyin Xu and Yuanyuan Zhou. Systems approaches to tackling configuration errors: A survey. *ACM Comput. Surv.*, 47(4):70:1–70:41, July 2015. ISSN 0360-0300. doi: 10.1145/2791577. URL <http://dx.doi.org/10.1145/2791577>.
- [12] Zuoning Yin, Xiao Ma, Jing Zheng, Yuanyuan Zhou, Lakshmi N. Bairavasundaram, and Shankar Pasupathy. An empirical study on configuration errors in commercial and open source systems. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, SOSP '11*, pages 159–172, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0977-6. doi: 10.1145/2043556.2043572.