

Configuration Management

Markus Raab

Institute of Information Systems Engineering, TU Wien

2.3.2018



Language

- Slides are in English
- Papers are in English
- Book is in English

Language of the Talk?

Task

Hands up if you prefer German.

Unanimous preference of German required, otherwise English.

Motivation

- 1 Motivation
- 2 Content Overview
 - Terminology
 - Requirements
 - Topics
- 3 Organisation
 - Preliminaries
 - Grades
 - Assignments

Misconfiguration

- configuration is a user interface for both developers and system administrators
- *misconfigurations* [1, 8, 10, 11] are a major cause of system failures [4, 5, 9]
- much time needed to fix misconfigurations [3, 4, 7, 11]

No-Futz

- Holland et al. [2] defined *futz*ing to denote “*tinkering or fiddling experimentally with something.*”
- With ***no-futz computing*** Holland et al. [2] mean “*that futzing should be allowed, but should never be required.*”
- currently configuration is error-prone and under-specified, *futz*ing is often required

Examples

Not every misconfiguration involves big companies, cloud, and huge amounts of money:

- No internet access because resolv.conf symlink broken.
- KDE crash because of ulimit setting.
- Out-of-service of computers during exam.

First Assignment

- Have you already experienced misconfiguration?
- Did you read about misconfiguration in the news?

Task

Discuss with your neighbor and tell us the best stories.

Content Overview

- 1 Motivation
- 2 Content Overview
 - Terminology
 - Requirements
 - Topics
- 3 Organisation
 - Preliminaries
 - Grades
 - Assignments

Terminology

Definition

A **configuration setting**, or **setting** in short, fulfills these properties:

- 1 It is provided by the execution environment.
- 2 It is *consumed* by an application.
- 3 It consists of a key, a configuration value, and potentially *metadata*. The **configuration value**, or **value** in short, influences the application's behavior.
- 4 It can be *produced* by the maintainer, user, or system administrator of the software.

Requirement

A configuration library must be able to integrate (legacy) systems and must fully support (legacy) configuration files.

Requirement

Validation of configuration settings must happen systematically before the application is even started.

Requirements

Task

Discuss about requirements a configuration framework should fulfil.

Topic: *sources of configuration*

- semi-structured data
- configuration file formats
- command-line arguments
- environment variables

Topic: *design and architecture of configuration and configuration access*

- architectural decisions
- introspection
- code generation
- variability

Topic: *reduction of misconfiguration and configuration duplicates*
Misconfiguration obviously needs configuration. We will discuss the ideas of:

- complexity reduction
- when configuration is needed (decisions, user interface)
- configuration-less systems (auto-detection)
- how duplication can be avoided (generation of artifacts)
- testability (generation of test cases)

Further topics:

- *context-awareness* (context-oriented programming)
- *avoidance of dependences*
- *cascading configuration*
- *strategies for validation and modularization*
- *documentation of configuration*

Task

Break.

Topic: *sound, complete and early detection of misconfiguration*

- points in time for configuration access and validation
- validation techniques
- constraints

Topic: *configuration as user interface*

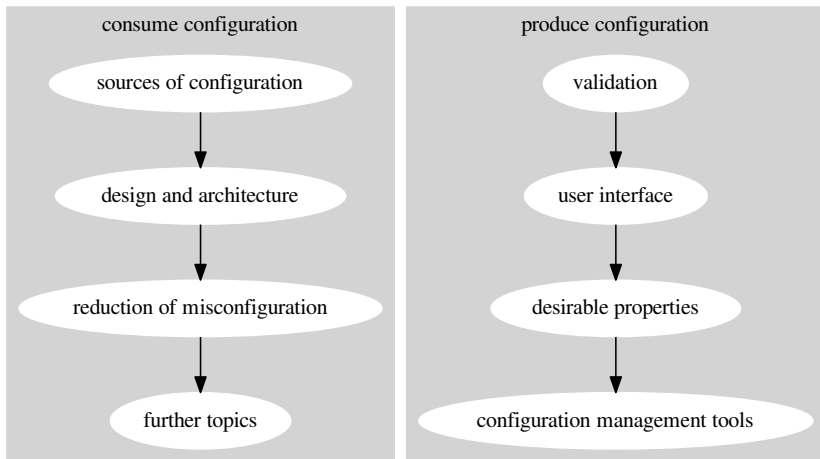
- How system administrators work.
- Which user interfaces exist.
- How to specify configuration.

Topic: *desirable properties of configuration*

- self-description
- changeability
- idempotence
- round-tripping

Topic: *configuration management tools*

- Puppet
- CfEngine
- Nix
- others...



Elektra



- Elektra improves configuration management.
- Configuration management tools can use Elektra.
- Elektra implements what we discuss in this lecture.
- Elektra allows applications to fulfil the requirements we will discuss.
- Obviously you will also be able to apply the knowledge from this LVA without Elektra.
- Developed at TU Wien (<https://libelektra.org>).

In which topics are you interested?

Task (1)

Choose a partner for this task.

Task (2)

Go to stations and discuss topic with your partner.

Task (3)

Write down the most interesting topics. (Can be topics of stations or new topics.)

Organisation

- 1 Motivation
- 2 Content Overview
 - Terminology
 - Requirements
 - Topics
- 3 Organisation
 - Preliminaries
 - Grades
 - Assignments

Communication:

- TISS (forum and news)
`https://tiss.tuwien.ac.at/course/courseDetails.xhtml?courseNr=194030&semester=2018S`
- GitHub (private and public repo)
`https://git.libelektra.org`
- EMail `markus.raab@complang.tuwien.ac.at`
- before/after/during lectures

Task

Send me your GitHub name by email to get access to private repo.

Feedback:

- TISS anonymous feedback
- TISS LVA evaluation
- EMail `markus.raab@complang.tuwien.ac.at`
- before/after/during lectures

Previous Knowledge

- Obviously *no* prior knowledge about Configuration or Configuration Management necessary.
- If you already have experience, you can use it in your talk and assignments.
- Knowledge about software engineering and software requirements is beneficially.
- You should have an understanding of large-scale software construction.
- Programming skills is a must.

You will get a grade only if:

- You submitted your homework. (PR is not enough.)
- You participated in the team exercise.
- You gave your talk.

To get a positive grade:

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

Grade is calculated from:

30 %: homework

30 %: team exercise

10 %: talk

30 %: test

+ : extrapoints

Talk

You can give a talk about anything related to configuration management.

- 20 minutes.
- It must be about your experience.
- I.e., not only about study of literature.
- It is okay if the experience happened during this LVA.
- If you extensively use some tool, please share your knowledge.

Home Work

You can select your own task related to *use* configuration management. For example:

- Use Elektra or Puppet-Libelektra in/for a small application.
- Write a tutorial about how you did it.
- Fix usability bugs in Elektra, make it easier. (Extrapoints)

Task

Think about a talk and homework till next week. Write it down in the private repo (first come, first served).

Team Exercise

You can select your team (2-3 people) and your task related to *improve* configuration management. For example, you can improve Elektra:

- Write a validation plugin.
- Write support for a configuration file format.
- Make a configuration management tool use Elektra.
- Fix bugs. (Extrapoints)

Task

Talk with someone who is not your neighbor about a potential collaboration in the team exercise.

Lecture is every week. Time Line:

9.3.2018: TISS registration

16.3.2018: topic homework and talk

23.3.2018: teams found together

13.4.2018: homework submitted, topics of team exercise

18.5.2018: guest lecture

25.5.2018: team exercise submitted

22.6.2018: last corrections of team exercise

29.6.2018: test

Guest Lecture?

Title: Formal Foundations of Configuration Management for Program Analysis

Name: Jürgen Cito, MIT

Date/Time: 18.05.2018, 14:00 (c.t.), 1h

Questions?

Task

Please read TISS and register for the course.

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 and Gergö 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.
- [7] 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.
- [8] 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>.

- [9] Avishai Wool. A quantitative study of firewall configuration errors. *Computer*, 37(6):62–67, 2004.
- [10] 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>.
- [11] 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.