

---

Eine Dissertation der Universität Regensburg:

# System Administration Training in the Virtual Unix Lab

*An e-learning system with diagnosis via a domain specific language as  
base for an architecture for tutorial assistance and user adaption*

---

Autor: Hubert Feyrer <hubert@feyrer.de>

Erstbetreuer: Prof. Dr. Rainer Hammwöhner

Zweitbetreuer: Prof. Dr. Christian Wolff

Eingereicht am: 25. Januar 2008

Mündliche Prüfung: 11. November 2008



**Hubert Feyrer**

**System Administration Training  
in the Virtual Unix Lab**

An e-learning system with diagnosis via a domain specific  
language as base for an architecture for tutorial assistance  
and user adaption

Shaker Verlag  
Aachen 2009

**Bibliographic information published by the Deutsche Nationalbibliothek**

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: Eine Dissertation der Universität Regensburg, 2008

Copyright Shaker Verlag 2009

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8322-7874-8

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: [www.shaker.de](http://www.shaker.de) • e-mail: [info@shaker.de](mailto:info@shaker.de)

# Preface

Development of the basic training system was funded as part of the HWP-project of the German government, the system was implemented at the Computer Science department of the University of Applied Sciences (Fachhochschule, FH) Regensburg, Germany.

As the basic system used in this work was developed in German language, usage examples, exercises and screenshots are in that language. This work as a whole is written in English to match the intended audience. The male gender is used throughout the text for consistency and simplicity.

## Acknowledgements

I'd like to express words of gratitude to Prof. Jürgen Sauer as my longstanding mentor; Prof. Dr. Rainer Hammwöhner the principal advisor of this work, and Prof. Dr. Christian Wolff as co-principal advisor; The department of Computer Science at the University of Applied Sciences of Regensburg, especially the officiating dean Prof. Dr. Kucera and the former dean Prof. Dr. Schicker, kindly supported the work, and last but not least I'd like to thank my parents and friends for all their support.

Further thanks go to the NetBSD, R and PostgreSQL projects for their great and free software; the Virtual Unix Lab beta-tester Holger Amann, Sonja Öttl, Günter Schwarz, Holger Nösekabel, Stefan Zimmermann, DaNiel Ettle, and Michael Jobst; the students of the I5T semester in the summer semesters from 2004 to 2007, as well as to my proofreaders Verena Bäumler, Andreas Fassel, Sabine Salzl, Stefan Schumacher, Günter Schwarz, Matthew Sporleder, and Gabriele Steinberger.

Without them, this work would not exist in its current form today.

Hubert Feyrer  
Regensburg, November 24th, 2008



# Abstract

This work covers training of system administration by introducing a system called the Virtual Unix Lab, and illustrates advanced topics based on it. The work is divided into three parts.

In the first part, the goals of the Virtual Unix Lab is illustrated and compared to related works, followed by observations about education of system administration. General learning theories are observed and compared to an existing lecture on system administration, showing that there is demand for practical exercises in advanced topics.

The second part describes how diagnosis of the Virtual Unix Lab exercise results and feedback to the user are realized with the help of a domain specific language. After observing the fundamentals of domain specific languages, the design of diagnosis and feedback to the learner is presented, the Verification Unit Domain Specific Language (VUDSL) is described, and architecture and implementation within the existing Virtual Unix Lab are shown. An evaluation of the system was performed and shows that repeated exercises show improved performance of the students, and that the system is regarded as useful by students in general.

The third part adds tutoring and user adaption. Based on the fundamentals of tutoring and user adaption, an architecture for a tutoring component for the Virtual Unix Lab based on an overlay architecture is described. Aspects discussed include on-line diagnosis, feedback, assistance to the user, considerations for the user model, and impact on the user interface. User adaption is based on the user model built by the tutoring component. It observes structural and longitudinal consistency, and provides personalized feedback to the student. An architecture is described that fits in the overall Virtual Unix Lab architecture, and possible extensions for the VUDSL used for diagnosis and feedback are proposed.



# Contents (short)

## I Introduction

1	Problem domain and goal of the Virtual Unix Lab	3
2	Related works	11
3	Education of system administration	23

## II Diagnosis and feedback with a domain specific language

4	Basic design of the Virtual Unix Lab	53
5	Introduction of domain specific languages	69
6	Architecture and implementation of diagnosis and feedback with a domain specific language	81
7	Evaluation of the Virtual Unix Lab	129

## III Tutoring and user adaption

8	Introduction of tutoring and user adaption	179
9	Design of tutoring and user adaption	211

---

<b>10 Architecture of tutoring</b>	<b>221</b>
<b>11 Architecture of user adaption</b>	<b>247</b>
<b>12 Conclusion</b>	<b>271</b>
<b>List of figures</b>	<b>272</b>
<b>List of tables</b>	<b>278</b>
<b>Bibliography</b>	<b>280</b>
<b>A Example exercise components</b>	<b>311</b>
<b>B Database structure</b>	<b>345</b>
<b>C Evaluation data and code</b>	<b>351</b>
<b>D A theory of bugs — attempt of a reconstructive approach</b>	<b>383</b>
<b>E Analysis of exercises under tutorial and adaptive aspects</b>	<b>387</b>

# Contents

## I Introduction

<b>1 Problem domain and goal of the Virtual Unix Lab</b>	<b>3</b>
1.1 Problem domain of the Virtual Unix Lab . . . . .	3
1.2 The goal of the Virtual Unix Lab . . . . .	6
1.3 How this book is organized . . . . .	8
<b>2 Related works</b>	<b>11</b>
2.1 Computer science education . . . . .	11
2.2 System administration education . . . . .	12
2.3 Training systems for system administration . . . . .	13
2.3.1 Systems focused on education . . . . .	13
2.3.2 Systems focused on deployment . . . . .	14
2.3.3 Systems offering user-level access . . . . .	15
2.4 Domain specific languages . . . . .	16
2.5 Result verification, diagnosis and feedback . . . . .	17
2.6 Tutoring systems in Unix education . . . . .	18
2.7 Adaptive systems in Unix education . . . . .	19
2.8 Other virtual labs . . . . .	20
2.9 Virtualization & emulation . . . . .	20

<b>3</b>	<b>Education of system administration</b>	<b>23</b>
3.1	Fundamentals of education . . . . .	23
3.1.1	Psychology and learning theory . . . . .	23
3.1.2	Didactic realization, instruction theory and instructional design	27
3.1.3	Dimension of implementation and adaption . . . . .	31
3.1.4	Alternative learning-theoretical approaches . . . . .	32
3.1.5	Education – ideal progression and tools . . . . .	34
3.2	The “System Administration” class . . . . .	35
3.2.1	History and target audience . . . . .	35
3.2.2	Current curriculum . . . . .	36
3.2.3	Course layout . . . . .	40
3.2.4	Didactic instruments . . . . .	43
3.3	Analysis of the current situation . . . . .	46
3.4	Future directions . . . . .	48
<b>II</b>	<b>Diagnosis and feedback with a domain specific language</b>	
<b>4</b>	<b>Basic design of the Virtual Unix Lab</b>	<b>53</b>
4.1	A user-level walkthrough of the Virtual Unix Lab . . . . .	53
4.2	Hardware and network setup of the Virtual Unix Lab . . . . .	64
4.3	Software components of the Virtual Unix Lab . . . . .	65
<b>5</b>	<b>Introduction of domain specific languages</b>	<b>69</b>
5.1	Classification of languages . . . . .	69
5.2	Attributes of domain specific languages . . . . .	71

---

5.3	Design patterns . . . . .	72
5.4	Choosing an implementation languages . . . . .	77
<b>6</b>	<b>Architecture and implementation of diagnosis and feedback with a domain specific language</b>	<b>81</b>
6.1	Requirements of exercise verification . . . . .	81
6.2	Roadmap of implementation . . . . .	83
6.2.1	Stepwise refinement . . . . .	83
6.2.2	Exercise phases . . . . .	84
6.2.3	What and how to verify . . . . .	85
6.3	Step 0: Basic design . . . . .	86
6.4	Step I: Instructions and checks not coupled . . . . .	87
6.4.1	Components . . . . .	87
6.4.2	Integration and interaction . . . . .	92
6.4.3	Summary and suggested improvements . . . . .	97
6.5	Step II: Instructions and checks coupled . . . . .	97
6.5.1	Improved check primitives . . . . .	97
6.5.2	Coupling of exercise text and checks . . . . .	102
6.5.2.1	Options . . . . .	103
6.5.2.2	Data structure representation . . . . .	105
6.5.2.3	Forming a domain specific language . . . . .	106
6.5.3	Giving feedback . . . . .	107
6.5.4	Creating a system front-end with check scripts . . . . .	110
6.5.5	Integration and interaction . . . . .	114
6.5.6	Summary of step II . . . . .	122
6.6	The Verification Unit Domain Specific Language (VUDSL) . . . . .	122

---

6.7	Conclusion of diagnosis and feedback with a domain specific language	124
6.8	Future Perspectives . . . . .	125
<b>7</b>	<b>Evaluation of the Virtual Unix Lab</b>	<b>129</b>
7.1	What to evaluate . . . . .	129
7.2	Analysis of data gathered during student exercises . . . . .	131
7.2.1	Methodology of the data analysis . . . . .	131
7.2.2	Number of exercises taken and repeated . . . . .	132
7.2.3	Performance of repeated exercises . . . . .	133
7.2.4	Results of selected exercise topics . . . . .	136
7.2.5	Exercise duration . . . . .	146
7.2.6	Exercise time . . . . .	149
7.2.7	Summary . . . . .	150
7.3	Analysis of the user questionnaire . . . . .	155
7.3.1	Methodology of the questionnaire analysis . . . . .	155
7.3.1.1	Aspects evaluated by the questionnaire . . . . .	156
7.3.1.2	Design and implementation of the questionnaire . . . . .	156
7.3.1.3	Evaluation methods . . . . .	157
7.3.2	Evaluation of user acceptance . . . . .	158
7.3.2.1	Questionnaire results . . . . .	158
7.3.2.2	Interpretation of the questionnaire results . . . . .	159
7.3.3	Evaluation of the course of exercises . . . . .	159
7.3.3.1	Questionnaire results . . . . .	159
7.3.3.2	Interpretation of the questionnaire results . . . . .	160
7.3.4	Evaluation of the use of learning material . . . . .	161
7.3.4.1	Impact of learning materials in general . . . . .	161

7.3.4.2	Impact of learning materials during Virtual Unix Lab exercises . . . . .	163
7.3.4.3	Impact of the “SA” lecture for exercises in the Virtual Unix Lab . . . . .	165
7.3.4.4	Impact of the “SA” lecture notes for exercises in the Virtual Unix Lab . . . . .	166
7.3.4.5	Interpretation of the questionnaire results . . . . .	167
7.3.5	Evaluation of the target audience . . . . .	168
7.3.5.1	Questionnaire results . . . . .	169
7.3.5.2	Interpretation of the questionnaire results . . . . .	170
7.3.6	Summary . . . . .	170
7.4	Other aspects to evaluate . . . . .	172
7.5	Conclusion of the evaluation . . . . .	175

### III Tutoring and user adaption

<b>8</b>	<b>Introduction of tutoring and user adaption</b>	<b>179</b>
8.1	Fundamentals of tutoring . . . . .	179
8.1.1	Approaching tutoring . . . . .	180
8.1.2	The teaching model . . . . .	181
8.1.2.1	Teaching and didactic operations . . . . .	182
8.1.2.2	Methods for plan recognition and assistance . . . . .	184
8.1.2.2.1	Classical approaches . . . . .	184
8.1.2.2.2	Cognitive approach . . . . .	185
8.1.2.2.3	Linguistic approach . . . . .	186
8.1.2.2.4	Artificial intelligence . . . . .	186
8.1.2.2.5	Semantic networks and ontologies . . . . .	188

8.1.2.2.6	Frames and scripts . . . . .	189
8.1.2.2.7	Bayesian networks . . . . .	190
8.1.2.3	Choosing a method . . . . .	190
8.1.3	The domain model . . . . .	191
8.1.4	The user model . . . . .	192
8.1.4.1	Theories of bugs . . . . .	193
8.1.4.2	Viewpoints . . . . .	194
8.1.4.3	Diagnosis . . . . .	195
8.1.4.3.1	Behavioral diagnosis . . . . .	196
8.1.4.3.2	Epistemic diagnosis . . . . .	197
8.1.4.3.2.1	Direct assignment of credit and blame	197
8.1.4.3.2.2	Structural consistency . . . . .	199
8.1.4.3.2.3	Longitudinal consistency . . . . .	199
8.1.4.3.3	Diagnostic data . . . . .	200
8.1.4.4	Feedback . . . . .	201
8.1.5	The user interface . . . . .	202
8.2	Fundamentals of user adaption . . . . .	203
8.2.1	The meaning of context . . . . .	207
8.2.2	Adaptive services and multiple agents . . . . .	208
8.2.3	Modeling techniques . . . . .	208
8.2.4	Adaptive axes . . . . .	210
<b>9</b>	<b>Design of tutoring and user adaption</b>	<b>211</b>
9.1	Goals of tutoring and user adaption . . . . .	211
9.2	Methodology of tutoring and user adaption . . . . .	212
9.3	The domain model . . . . .	212

---

9.3.1	Content decomposition . . . . .	213
9.3.2	Considerations for a theory of bugs . . . . .	215
9.3.2.1	Adjusting the domain model . . . . .	216
9.3.2.2	Analyzing existing exercise data . . . . .	216
9.3.2.3	Results and conclusion . . . . .	217
9.4	Software architecture . . . . .	219
<b>10</b>	<b>Architecture of tutoring</b>	<b>221</b>
10.1	Establishing the teaching model . . . . .	221
10.1.1	Selection criteria . . . . .	222
10.1.2	Classical approaches with overlay architecture . . . . .	222
10.1.3	Cognitive approach . . . . .	223
10.1.4	Linguistic approach . . . . .	224
10.1.5	Artificial Intelligence based approach . . . . .	225
10.1.6	Semantic networks and ontologies . . . . .	227
10.1.7	Frames and scripts . . . . .	228
10.1.8	Bayesian networks . . . . .	229
10.1.9	Comparison . . . . .	230
10.2	Using model tracing for diagnosis during the exercise . . . . .	232
10.3	Investigating on-line diagnosis . . . . .	232
10.4	Giving feedback and assistance . . . . .	235
10.4.1	Goal . . . . .	235
10.4.2	Assumptions . . . . .	236
10.4.3	Challenges . . . . .	236
10.4.4	Realization . . . . .	237
10.4.4.1	Contents . . . . .	237

10.4.4.2	Form of feedback . . . . .	238
10.4.5	Impact on organization of exercises and learning material . . . . .	239
10.5	Considerations for the user model . . . . .	240
10.6	Impact on the user interface . . . . .	241
10.6.1	Communication channels . . . . .	242
10.6.2	Analysis of the current user interface . . . . .	242
10.6.3	Blending information into the web-based user-interface . . . . .	245
10.7	Summary . . . . .	245
<b>11</b>	<b>Architecture of user adaption</b>	<b>247</b>
11.1	Establishing and maintaining the user model . . . . .	247
11.1.1	Initialization . . . . .	248
11.1.2	Clustering . . . . .	249
11.1.3	Observed data . . . . .	249
11.1.4	Updating the user model . . . . .	249
11.1.5	Accommodating plan recognition . . . . .	250
11.2	Adaptive axes . . . . .	251
11.3	Structural consistency . . . . .	252
11.3.1	Observing exercise velocity . . . . .	252
11.3.2	Observing mastered skills . . . . .	253
11.3.3	Observing help requests . . . . .	253
11.3.4	Adjusting the user model . . . . .	254
11.3.5	A metric for evaluation . . . . .	255
11.4	Longitudinal consistency . . . . .	256
11.4.1	Assumptions and methodology . . . . .	256
11.4.2	Descriptive analysis . . . . .	257

---

11.4.2.1	Interpolation vs. more data . . . . .	257
11.4.2.2	Detecting speed changes . . . . .	257
11.4.2.3	Observations for repeated exercises . . . . .	257
11.4.2.4	Speed and acceleration of progress . . . . .	258
11.4.2.5	Data model and storage . . . . .	258
11.4.2.6	Drawing conclusions from speed and acceleration . . . . .	258
11.4.3	Indicative analysis . . . . .	259
11.5	Personalizing feedback . . . . .	260
11.5.1	Adjusting of help contents . . . . .	260
11.5.2	Handling non-standard exercise progress . . . . .	261
11.5.3	Adjusting the system . . . . .	261
11.5.4	Preventing abuse of the help system . . . . .	262
11.6	Extending the VUDSL for user adaption . . . . .	262
11.6.1	VUDSL extensions for structural consistency . . . . .	263
11.6.2	VUDSL extensions for longitudinal consistency . . . . .	266
11.6.3	VUDSL extensions for personalized feedback . . . . .	267
11.6.4	Other VUDSL extensions . . . . .	267
11.7	Summary . . . . .	270
<b>12</b>	<b>Conclusion</b>	<b>271</b>
	<b>List of figures</b>	<b>272</b>
	<b>List of tables</b>	<b>278</b>
	<b>Bibliography</b>	<b>280</b>
<b>A</b>	<b>Example exercise components</b>	<b>311</b>

---

A.1	Exercise texts for users . . . . .	311
A.1.1	Network Information System (NIS) exercise . . . . .	311
A.1.2	Network File System (NFS) exercise . . . . .	313
A.2	Exercises including text and check data . . . . .	314
A.2.1	Network Information System (NIS) exercise . . . . .	314
A.2.2	Network File System (NFS) exercise . . . . .	318
A.3	The VUDSL processor: uebung2db . . . . .	321
A.4	Complete lists of checks used in exercises . . . . .	325
A.4.1	Network Information System (NIS) exercise . . . . .	325
A.4.2	Network File System (NFS) exercise . . . . .	326
A.5	List of check scripts and parameters . . . . .	327
A.6	Selected check scripts . . . . .	329
A.6.1	Step I . . . . .	329
A.6.1.1	netbsd-check-finger.sh . . . . .	329
A.6.1.2	netbsd-check-masterpw.sh . . . . .	330
A.6.1.3	netbsd-check-pkginstalled.sh . . . . .	330
A.6.1.4	netbsd-check-pw.pl . . . . .	330
A.6.1.5	netbsd-check-usershell2.sh . . . . .	331
A.6.1.6	check-program-output . . . . .	331
A.6.2	Step II . . . . .	333
A.6.2.1	admin-check-clearharddisk . . . . .	333
A.6.2.2	admin-check-makeimage . . . . .	333
A.6.2.3	check-file-contents . . . . .	336
A.6.2.4	unix-check-user-exists . . . . .	337
A.6.2.5	unix-check-user-shell . . . . .	338

---

A.6.2.6	unix-check-user-password . . . . .	340
A.6.2.7	unix-check-process-running . . . . .	341
A.6.2.8	netbsd-check-rcvar-set . . . . .	342
<b>B</b>	<b>Database structure</b>	<b>345</b>
B.1	Table: benutzer . . . . .	345
B.2	Table: rechner . . . . .	345
B.3	Table: images . . . . .	346
B.4	Table: uebungen . . . . .	346
B.4.1	Definition . . . . .	346
B.4.2	Example records . . . . .	346
B.5	Table: uebung_setup . . . . .	347
B.6	Table: uebungs_checks . . . . .	347
B.6.1	Definition . . . . .	347
B.6.2	Example records . . . . .	348
B.7	Table: buchungen . . . . .	348
B.7.1	Definition . . . . .	348
B.7.2	Example records . . . . .	348
B.8	Table: ergebnis_checks . . . . .	349
B.8.1	Definition . . . . .	349
B.8.2	Example records . . . . .	350
<b>C</b>	<b>Evaluation data and code</b>	<b>351</b>
C.1	Questionnaire: questions — raw format . . . . .	351
C.2	Questionnaire: questions and results . . . . .	353
C.3	Exercise results: selected SQL queries and results . . . . .	371

<b>D A theory of bugs — attempt of a reconstructive approach</b>	<b>383</b>
<b>E Analysis of exercises under tutorial and adaptive aspects</b>	<b>387</b>