

Open Source Initiativen und Perspektiven für Bildung und Wirtschaft

Mag. (FH) Clemens Peter, Projektleiter OSIV 2004
Fachhochschule Vorarlberg

Dornbirn, am 16. Februar 2005



Übersicht

Open Source Initiative Vorarlberg OSIV 2004

- _ Einführung: Ziele, Aktivitäten
- _ Erfahrungen aus OSIV
- _ Erfahrungen allgemein und im Detail

Open Source Einsatz & Aktivitäten an der Fachhochschule Vorarlberg FHV

- _ Lehre
- _ Forschung
- _ IT Infrastruktur & Anwendungen
- _ Auswirkungen auf andere Fachbereiche

Ausblick und Perspektiven für Fachhochschulen (als Partner der Wirtschaft)

- _ Anbietergemeinschaften & Servicekooperationen
- _ Open Source Center



OSIV - Ziele

1. Unternehmen für Open Source Software, deren Hintergründe, Entwicklung und Anwendung zu sensibilisieren,
2. die Entwickler Vorarlbergs zu vernetzen und mit weltweiten Entwicklern in Austausch zu bringen und
3. den Markt für Open Source Dienstleister - im Bereich Beratung, Installation/Integration, Schulung, Support bis hin zur Anwendungsentwicklung - aufzubereiten



OSIV - Aktivitäten

1. Technologieforum

- _ Sensibilisierung, u.a. mittels Referenzbeispielen

2. Vortragsreihe

- _ Erfahrungsberichte & Austausch zwischen Unternehmen, öffentliche Verwaltung und Bildungseinrichtungen --> online unter <http://www.osiv.at>

3. Entwicklernetzwerke

- _ Kennenlernen der Softwareentwicklung in Communities – in intensive Arbeitsforen
- _ Weiterbildung & Know How Austausch
- _ Grundgedanke: lokales Arbeiten von kleinen Einheiten bei globaler Vernetzung und Austausch

4. Unternehmerworkshops

- _ Analyse von Betrieben, Evaluierung und produktneutrale Bewertung von Open Source Gestaltungspotentialen und Einsatzmöglichkeiten

5. Seminare

- _ Know How Transfer in 2 Kern-Seminaren: Linux Systemadministration und IT Security



OSIV – Erfahrungen aus der Initiative

1. Technologieforum

- _ Positives Feedback
- _ Wissen über Einsatz-Möglichkeiten fehlt
- _ kaum lokale Dienstleister für Service & Weiterentwicklungen bekannt

2. Vortragsreihe

- _ Großunternehmen setzen OSS in vielen Bereichen ein, z.T. die bessere Alternative
- _ kleinen Unternehmen fehlt Vertrauen in Leistungsfähigkeit und Servicesicherheit

3. Entwicklernetzwerke

- _ hohe Leistungen durch intensive Zusammenarbeit von hochmotivierten Entwicklern
- _ sehr umfassender Know-how Transfer in lokale Unternehmen
- _ FH Mitarbeiter und Studenten profitieren von „intensiven Weiterbildung-Seminaren“

4. Unternehmerworkshops

- _ Weniger mit OSS befasste Unternehmen zeigen Zurückhaltung, prüfen aber OSS Alternativen schon nach Erstberatung.



OSIV – Erfahrungen allgemein

- _ Vertrauen in Open Source Software zu gering
 - > wegen geringem Marketing von OSS Dienstleistern
 - > online weit verbreitet, UnternehmerInnen arbeiten aber (meist) offline
- _ Relevanz der strategischen IT Entscheidung unbewußt
 - > Bedeutung der raschen Weiterentwicklung durch die OS Communities und die fortlaufende Eingriffsmöglichkeit zu eigenen Anpassungen nicht bewußt
 - > Herstellerunabhängigkeit und IT Sicherheit weniger wichtig → Potenzial (Vergleich zu IT Sicherheit: 'MS Patch Day' alle 14 T vs. OSS Bugfixes am Nachmittag)
- _ Wissen über Einsatzmöglichkeiten fehlt teilweise
 - > Beispiel: SQL Ledger (alt) > Lx Office (neu) <http://www.lx-office.org>
- _ Servicestellen kaum bekannt bzw. z.T. tatsächlich fehlende, lokale Servicesicherheit
 - > durch OSS 'Freak'-Image und Negativwerbung wenig Aufmerksamkeit -> Informationsdefizit!



OSIV – Erfahrungen im Detail

- _ Open Source heißt nicht gleich Linux (-betriebssystem und -anwendungen)
 - > vor allem in der Wirtschaft: Spezialanwendungen für Windows und Windows Clients (Büros)
- _ Server > Linux/OSS, Client > Windows
 - > Server – Linux: speziell bei sicherheitskritischen Anwendungen & Webservices (Apache 67%)
 - > Client – Windows: Benutzerfreundlichkeit für unerfahrene User besser (USB plug & play)
- _ OSS auch am Mac
 - > seit Unix System-Grundlage ist (OS X), viele Anwendungen für den Mac
- _ „Open Source gut, offene Standards besser“
 - > offene Standards ermöglichen bessere Vernetzung in heterogener Softwarelandschaft – ganz gleich ob proprietär oder open source, software- & plattformunabhängig
- _ Open Source Entwicklung & Fortschritte heute ohne Community undenkbar
- _ OSS bietet höhere Code-Qualität durch übersichtliche Strukturierung & zahlreiche Kommentare
 - > begünstigt lernen anhand von OSS Code-Beispielen: --> siehe 4 Beispiele auf Folgeseiten



OSS Code-Beispiel 01: Polarviewer: wertet Trainingsdaten einer Pulsuhr aus, C# unter Linux in Monodevelop

PolarViewer - src/PolarViewer/Parser/PolarS710RawParser.cs - MonoDevelop

File Edit View Run Search Window Help

Classes

- IntPtr bindtextdomain (IntPtr, IntPtr)
- IntPtr gettext (IntPtr)
- IntPtr ngettext (IntPtr, IntPtr, int)
- IntPtr textdomain (IntPtr)
- string GetPluralString (string, string)
- string GetString (string)
- void Init (string, string)
- {} PolarViewer
 - {} Core
 - PVException
 - ctor ()
 - ctor (string)
 - ctor (string, Exception)
 - {} Data
 - {} Gui
 - {} Parser
 - ExerciseParser
 - byte[] ReadFileToByteArray (string)
 - Exercise ParseExercise (string)
 - string[] ReadFileToStringArray (string)
 - PolarHRMParser
 - PolarHRMParserTest
 - PolarS710RawParser
 - byte DecodeBCD (byte)
 - byte[] fileContent
 - char DecodeChar (byte)
 - Exercise ParseExercise (string)
 - HeartRateLimit DecodeHeartRateLimit (byte)
 - int GetProperIndex (int, int, byte)
 - short DecodeAltitude (byte, byte)
 - short DecodeTemperature (byte, byte)
 - PolarS710RawParserTest
 - PVMain
 - {} Util
 - UnitConverter

```
92 // when bit 7 of byte 12 is set, then the hour is in PM mode
93 if ((fileContent[12] & 0x80) == 0x80) {
94     dateHours += 12;
95 }
96
97 exercise.Date = new DateTime (dateYear, dateMonth, dateDay, dateHours, dateMinutes, dateSeconds);
98
99 // get duration of exercise (ignore 1/10th of seconds)
100 // short durationTenthSeconds = (short) (fileContent[15] >> 4);
101 byte durationSeconds = DecodeBCD (fileContent[16]);
102 byte durationMinutes = DecodeBCD (fileContent[17]);
103 byte durationHours = DecodeBCD (fileContent[18]);
104 exercise.Duration = (durationHours * 60 * 60) + (durationMinutes * 60) + durationSeconds;
105
106 // get heartrate data
107 exercise.HeartRateAVG = fileContent[19];
108 exercise.HeartRateMax = fileContent[20];
109
110 // get number of laps
111 byte numberOfLaps = DecodeBCD (fileContent[21]);
112
113 // get user ID
114 exercise.UserID = DecodeBCD (fileContent[24]);
115
116 // get unit format from bit 1 of byte 25
117 // => 0 = metric, 1 = english
118 bool fMetricUnits = (fileContent[25] & 0x02) == 0x00;
119
120 // decode recording mode (heartrate is always recorded)
121 // (not available on S610 files)
122 RecordingMode recMode = new RecordingMode ();
123 exercise.RecordingMode = recMode;
124
125 if (fS610 == false)
126 {
127     bool fBike2 = (fileContent[26] & 0x20) == 0x20;
128     bool fBike1 = (fileContent[26] & 0x10) == 0x10;
129     recMode.Power = (fileContent[26] & 0x08) == 0x08;
130     recMode.Cadence = (fileContent[26] & 0x04) == 0x04;
131     recMode.Altitude = (fileContent[26] & 0x02) == 0x02;
132 }
```

Task List

| ! | Line | Description | Path | File |
|---|------|-------------|------|------|
|---|------|-------------|------|------|

Task List Output

Ready ln 51 col 1 ch 1 INS


```
File Edit View Search Tools Documents Help
workqueue.c x
90 /*
91 * Queue work on a workqueue. Return non-zero if it was successfully
92 * added.
93 *
94 * We queue the work to the CPU it was submitted, but there is no
95 * guarantee that it will be processed by that CPU.
96 */
97 int fastcall queue_work(struct workqueue_struct *wq, struct work_struct *work)
98 {
99     int ret = 0, cpu = get_cpu();
100
101     if (!test_and_set_bit(0, &work->pending)) {
102         if (unlikely(is_single_threaded(wq)))
103             cpu = 0;
104         BUG_ON(!list_empty(&work->entry));
105         __queue_work(wq->cpu_wq + cpu, work);
106         ret = 1;
107     }
108     put_cpu();
109     return ret;
110 }
111
112 static void delayed_work_timer_fn(unsigned long __data)
113 {
114     struct work_struct *work = (struct work_struct *)__data;
115     struct workqueue_struct *wq = work->wq_data;
116     int cpu = smp_processor_id();
117
118     if (unlikely(is_single_threaded(wq)))
119         cpu = 0;
120
121     __queue_work(wq->cpu_wq + cpu, work);
122 }
123
124 int fastcall queue_delayed_work(struct workqueue_struct *wq,
125                               struct work_struct *work, unsigned long delay)
126 {
127     int ret = 0;
128     struct timer_list *timer = &work->timer;
129
130     if (!test_and_set_bit(0, &work->pending)) {
131         BUG_ON(timer_pending(timer));
132         BUG_ON(!list_empty(&work->entry));
133     }
```

OSS Code-Beispiel 03: MP3 Player, Java in Eclipse

```
Java - Player.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Jython C/C++ Browsing C/C++ Team Synchronizing CVS Repository Exploring
Player.java
64
65 public boolean load(File file) {
66
67     try {
68         m_InputStream = AudioSystem.getAudioInputStream(file);
69         m_File = file;
70         m_Format = m_InputStream.getFormat();
71         m_FileFormat = AudioSystem.getAudioFileFormat(file);
72
73         // check for id3 tags
74         if (m_FileFormat.properties() != null) {
75             m_HasId3Tag = true;
76         }
77
78     } catch (UnsupportedAudioFileException e) {
79         System.out.println("unsupported format");
80         return false;
81
82     } catch (IOException e) {
83         System.out.println("ioexception");
84         return false;
85     }
86
87     AudioFormat srcFmt = m_Format;
88     // check for valid format, else extend it manually
89     DataLine.Info info = new DataLine.Info(SourceDataLine.class, srcFmt);
90
91     if (AudioSystem.isLineSupported(info) == false) {
92         // corrupted file - build own format information
93         AudioFormat destFmt = new AudioFormat(
94             AudioFormat.Encoding.PCM_SIGNED, srcFmt.getSampleRate(),
95             SAMPLESIZE, srcFmt.getChannels(), srcFmt.getChannels()
96                 * (SAMPLESIZE / 8), srcFmt.getSampleRate(), false);
97         m_InputStream = AudioSystem.getAudioInputStream(destFmt,
98             m_InputStream);
99     } else {
100         m_InputStream = AudioSystem.getAudioInputStream(m_Format,
101             m_InputStream);
102     }
103     m_Format = m_InputStream.getFormat();
104
105     // lets initialize the line here too
106     try {
107         m_Line = AudioSystem.getSourceDataLine(m_Format);
108     } catch (LineUnavailableException e) {
109         System.out.println("line not available");
110     }

```

```
File Edit Source Navigate Search Project Run Window Help
graph.py
4 class myGraph(Graph.Graph):
5     """ extended graph implementation """
6
7     closeList = []
8     openList = []
9
10    def PrintInfo(self):
11        """ prints some basic info """
12
13        print 'NrOfNodes: ' + str( self.number_of_nodes() )
14        print 'Nodes: ' + str( self.node_list() )
15        print 'NrOfEdges: ' + str( self.number_of_edges() )
16        print 'Edges: ' + str( self.edge_list() )
17
18        print 'Graph is connected: ' + str( self.connected() )
19
20
21    def DFS(self, node):
22        """ public caller for DFS """
23
24        self.closeList = []
25        self.openList = self.node_list()
26
27        # check for connectivity
28        if self.connected():
29            self._DFS(node)
30        else:
31            # search starting at specified pos
32            self._DFS(node)
33            # search remaining nodes
34            for n in self.openList:
35                self._DFS(n)
36
37        print self.closeList
38
39
40    def _DFS(self, node):
41        """ custom implementation of DFS """
42
43        self.closeList.append(node)
44        self.openList.remove(node)
45
46        for adjNode in self.out_nbrs(node):
47            if not adjNode in self.closeList:
48                self._DFS(adjNode)
```

OSS Code-Beispiel 04: Graph Tiefensuche, Python in Eclipse

FHV – OSS Einsatz in der Lehre

Allgemein

_ FHV lehrt und fordert Kenntnisse auf mehreren Plattformen (Windows, Linux, u.a.) - speziell Linux nicht nur im Serverbereich, sondern auch auf dem Client Pflicht

1. Semester

_ Betriebssystemprogrammierung findet gänzlich unter Linux statt

3. Semester

_ Programmieren III – Java Schwerpunkt, Eclipse als IDE im Einsatz

5./6. Semester

_ ICA I/II: Webapplikationen unter Linux, Einführung in Plone/Zope (Python)

_ im 6. Semester (Auslandssemester) kann das Betriebssystem frei gewählt werden

8. Semester

_ diverse Diplomarbeiten im Open Source Bereich (u.a. Zope/Plone)



FHV – OSS Einsatz in der Lehre – Details (1)

5./6. Semester – in ICA I/II:

Mozilla: <http://www.mozilla.org/>

AXIS: <http://ws.apache.org/axis/>

Apache HTTP-Server: <http://httpd.apache.org/>

Cocoon: <http://cocoon.apache.org/>

Tomcat: <http://jakarta.apache.org/tomcat/index.html>

Eclipse: <http://www.eclipse.org/>

PHP: <http://www.php.net/>

MySQL: <http://www.mysql.com/>

Xampp: <http://www.apachefriends.org/en/xampp.html> (install-toolkit: Apache, MySQL, PHP, Perl)

JBoss: <http://www.jboss.org/>

Xalan: <http://xml.apache.org/xalan-j/> (Apache XML Project)

Xerces: <http://xml.apache.org/xerces2-j/>

ZOPE: <http://www.zope.org/>

Plone: <http://www.plone.org>



FHV – OSS Einsatz in der Lehre – Details (2)

Diplomarbeiten – Themen:

- _ Unterstützung einer Verschlagwortung und Schlagwortsuche in einem Open Source Content Management System
- _ Versionierungs-Service für Information und Medien - für Zope
- _ Reinforcement Learning - Algorithmus/Framework - fürs LMS (Learning Management System) mit Zope/eduPlone
- _ Ein Routing Protokoll für mobile Ad-Hoc-Networks (MANETS) mit Hilfe von Potentialfeldern --> open standards !!
- _ Plattformübergreifende Desktopintegration des Zope Filemanagements mit Unterstützung von Metadaten Eingabe



FHV – OSS Einsatz in der Forschung

- _ Linux Cluster Projekt – Beowolf Cluster mit Libraries PVM und MPI
--> eventl. Anschluß an internationales GRID-Projekt - <https://gilda.ct.infn.it/>
- _ Linux Clients im Forschungszentrum Produkt- und Prozessentwicklung
--> u.a. für bestimmte Anwendungen (Integration in FH Information Services fehlt)
- _ Multicasting Projekt mit ausschließlich Open Source Netzwerk Analyse und Monitoring Tools
- _ Apple XServe Server mit Zope, 3 x Plone, Apache, MySQL, PostgreSQL für Forschung im Bereich der Mediengestaltung: Webservices mittels CMS, Streaming Audio & Video, u.a



FHV – OS IT Infrastruktur & Anwendungen

- _ (e)Learning Management System LMS auf Basis von Zope selbst entwickelt
- _ EDV-Räume mit SuSE Linux 9.2 und Windows XY (Dualboot)
- Einrichtung von OpenOffice, Mozilla, und anderen Standardanwendungen für den Einsatz auf allen Windows Desktops
- _ Groupspace mit ausgewählter OSS für Studenten und Mitarbeiter
- _ Plone Server zur Studienvertiefung in Informatik
 - > Praktika: Abgabesystem für Studentendarbeiten in Informatik mittels Plone
- Linux Terminal Server Projekt



Fachhochschule Vorarlberg – Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://lms.fh-vorarlberg.ac.at/fhv?mo=ects¤t=fhv/ects

Fachhochschule Vorarlberg
Clemens Peter (pd)

Übersicht Support Englisch

Verwalten

- ECTS

ECTS - Übersicht Syllabi Verwaltung

| SS 2004 | WS 2004 | SS 2005 | WS 2005 | SS 2006 | |
|------------------------------|----------------|--------------------|-----------------|---------------|---------------------------------|
| Zu bearbeitende SLVAs | | | Meine SLVAs | | |
| <u>Id</u> | <u>Titel</u> | <u>Studiengang</u> | Semester | <u>Status</u> | <u>Verantwortliche Benutzer</u> |
| | | | | | |

Transferring data from lms.fh-vorarlberg.ac.at... 0

Entwicklungsbeispiel FHV: (e)Learning Management System LMS auf Basis von Zope selbst (weiter-)entwickelt

FHV – OS Einfluss auf andere Fachbereiche

_ OS Philosophie bei wissenschaftlichen Inhalten

--> Ziel: nicht nur Software, sondern auch wissenschaftliche Inhalte frei zur Verfügung stellen!
Vorteile: rascher Bekanntheitsgrad durch Verbreitung online und Bezugnahme anderer Wissenschaftler, dadurch bessere Reviews und Feedback und mehr eigene Motivation

_ Kompetenznetzwerk Mediengestaltung

--> media.coop – Open Content Publishing Plattform, integrierte Creative Commons Lizenzen für freie Inhalte, wird erstellt mittels freier Software (Plone, StreamOnTheFly, u.a.)
<http://www.media.coop>

_ Studiengang Mediengestaltung

--> offener Austausch von Inhalten, abgesichert durch CC Lizenzen, teil des Studienvertrages



Perspektiven – Anbietergemeinschaften (1)

Ziel

- _ Servicepartner & -vermittler für Unternehmen bei flexibler Zusammenarbeit mit OS Communities
 - > Unternehmen wollen EIN Ansprechpartner (vertraglich abgesichert), Entwickler wollen möglichst frei arbeiten
- _ Lösung einiger OS Probleme von Unternehmen – wie zB fehlende, lokale Servicesicherheit bzw. Servicestellen kaum bekannt, mangelndes Vertrauen, Wissen über Einsatzmöglichkeiten

Umsetzung: über Genossenschaft

- _ Modell und Rechtsform der Genossenschaft eignen sich gut, um Anbietergemeinschaften zu fördern: bietet Rahmen zur Mitbestimmung der Mitglieder sowie Identifikation mit gemeinsam Zielen – ohne Gewinnabsichten, was besonders OS Entwickler ansprechen sollte, dennoch professionelles unternehmerisches Auftreten
- _ Beispiel IMGEN „Internationale Medienverbund reg. Gen.m.b.H“

Perspektiven – Anbietergemeinschaften (2)

Chancen (für FHs)

- _ Angebot einer breiten Palette von Dienstleistungen mit mehr Servicequalität – FHs als Entwicklungspartner und Know How Träger
 - > Praktikas & Jobs für Studenten & Absolventen
- _ Anregung von OS Entwicklungen durch Koordination von Entwicklern für OS Projekte von KMUs
 - > OS Projekte für Studenten
- _ Zusammenschlüsse von Entwicklern und Dienstleistern bei größeren Ausschreibungen
 - > FHs als Forschungspartner



Perspektiven – Open Source Center

FHplus Antrag mit 3 Schwerpunkten

1 Grundlagen und OS Philosophie

> Aufarbeitung der Philosophie, Grundlagen und wirtschaftlichen Aspekte freier Software, Analyse und Prüfung von Geschäftsmodellen u.a.

2 Vermittlungs- und Vernetzungsplattform (analog zu Anbietergemeinschaften, aber online)

> sowohl bei Anfragen/Aufträgen, bei Forschungsvorhaben als auch für gemeinsame Serviceleistungen der dazu kooperierenden Unternehmen, Fachhochschulen und freien Entwicklern aus Online Communities

3 Eigene Softwareentwicklungen & Hosting

> Entwicklungen in 3 Bereichen und freies Hosting von OS Entwicklungen (Sourceforge an FH)



Ausblick

- _ OSIV Publikation – voraussichtlich im April 2005
- _ Open Source Center (eventl., Entscheidung März)
- _ Kooperationen für Forschungsprojekte mit Genossenschaft IMGEN und Unternehmenspartnern (aus OSIV)
- _ Kooperation mit OCG Arbeitskreis Open Source Software ?
- _ andere Vorschläge, Ideen??



Diese Präsentation steht unter eine Creative Commons Lizenz
<http://creativecommons.org/licenses/by-sa/2.0/at/>