

Linux Design Paradigmen

Tim Krämer

University of Hamburg
Mittelweg 177, 20148 Hamburg
tkraemer@informatik.uni-hamburg.de

Bernhard Stöckl

University of Hamburg
Mittelweg 177, 20148 Hamburg
lstoeckl@informatik.uni.hamburg.de

ABSTRACT

Im Rahmen dieses Papers werden anhand von Benutzungsstatistiken, wie distrowatch.com, die populärsten Linux Desktop Oberflächen ausgewählt und hinsichtlich der Kriterien des Interaktions Designs untersucht. Insbesondere werden Usability Aspekte aufgezeigt und der Fokus beim Design der Benutzeroberflächen herausgearbeitet.

Im Weiteren werden Linux Desktops vorgestellt, die für spezielle Anwendergruppen entwickelt wurden. Ein Beispiel dafür ist XFCE, ein Desktop-System für ältere Menschen mit einer sehr einfachen Handhabung, was auch Leuten im fortgeschrittenen Alter einen Zugang zur neuen Technologie ermöglicht. Dieses Paper befasst sich mit den Fragen, welche Design Entscheidungen sich im Linux Desktop Bereich erfolgreich durchgesetzt haben, wieso diese zu einer breiten Akzeptanz geführt haben und was dieses Verhalten für die zukünftige Entwicklung bedeutet.

Author Keywords

Linux; Desktop Environments; Design Paradigma; Linux-Distributionen.; HCI.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Human Factors; Design; Measurement.

EINLEITUNG

Linux

Bekanntgeworden als offenes und freies Betriebssystem hat Linux in den letzten 10 Jahren viele Änderungen erlebt und platziert sich heute auf dem Desktop Markt neben kommerziellen Betriebssystemen wie Windows und MacOS. Dieses Paper betrachtet die aktuellen Desktop Oberflächen für Linux und untersucht im Speziellen die Design Paradigmen.

Design Paradigmen

Solche Design Paradigmen (engl. Design Patterns) dienen

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IS 2013 December 21, 2013, Hamburg, Germany.

Copyright 2013 by the Authors.

dazu, in unserem Fall Programmierern zu helfen den Entwicklungsprozess ihrer Software nach einer bestimmten Art und Weise zu strukturieren. Dies soll dabei helfen die Qualität der Software aufrecht zu erhalten auch wenn viele verschiedene Personen an demselben Projekt arbeiten. Ebenso soll die Weiterentwicklung und generelle Anwendung einer Software dadurch erleichtert werden. Design Paradigmen stellen immer Standards dar welche es gilt ein- bzw. aufrecht zu erhalten.

Ein Beispiel hierfür sind die allseits bekannten DIN-Normen, welche am 22. Dezember 1917 von dem Deutschen Institut für Normung (DIN) auf dem Normenausschuss der Deutschen Industrie eingeführt wurden. Wenn heute jemand von einem DIN A4 großem Blatt Papier spricht, weiß die Person gegenüber sofort wovon die Rede ist. Solche Paradigmen erlauben es uns in einer Gesellschaft, welche zunehmend globalisiert Produkte/Software produziert ein Endprodukt von hoher Qualität zu erstellen/entwickeln. Insbesondere stellen solche Design Patterns in der Open Source Community einen essentiellen Schritt dar um sich auf dem Weltmarkt behaupten zu können. Auf Linux bezogen sorgen diese Standards nicht nur dafür, dass ein Entwickler die Software weiterentwickeln kann, auch wenn er bisher nicht involviert war, sondern auch dass ein Enduser sich sicher sein kann, nach einem Update immer noch mit dem Produkt arbeiten zu können.

Von großer Bedeutung ist hier, dass die Design Paradigmen dabei helfen, höhere Akzeptanz beim Kunden zu schaffen, was notwendig ist um sich gegen andere (kommerzielle) Betriebssysteme durchzusetzen. Da sich Linux, als offenes Betriebssystem an dieser Stelle von der kommerziellen Konkurrenz unterscheidet, werden in diesem Paper die Design Entscheidungen und Entwicklungsrichtungen der populären Linux Desktops vorgestellt.

BEGRIFFSERLÄUTERUNGEN

Grafische Oberfläche

Mit Grafik ist in diesem Fall die Desktopumgebung und deren Aufbau gemeint. Die Anforderungen, welche den grafischen Teil des Design Paradigmas ausmachen befähigen den User die Funktionalitäten seines Systems zu nutzen. Es gibt Kommandozeilen-basierte Distributionen ohne Grafiken, diese können aber von dem Großteil der potentiellen Enduser, aufgrund von fehlendem Expertenwissen, nicht genutzt werden. Bei der Grafik

achtet man darauf möglichst verständliche Visualisierungen zu verwenden. Möchte man z.B. ein neues Textdokument erstellen, so ist es einfach für den User zu erkennen, dass wenn er ein Bild sieht welches geschriebene Zeilen auf einem Blatt Papier darstellt, sich dahinter wohl die gewünschte Funktionalität finden lässt. Diese Referenzen aus dem realen Leben helfen der Akzeptanz eines Systems. Je leichter man sich in der designten virtuellen Umgebung zurechtfinden kann umso besser ist das Design Paradigma, mit welchem es erstellt wurde. Leider erlauben unter anderem der heutige Konkurrenzkampf auf dem Weltmarkt und auch die persönlichen Präferenzen der Designer eine Standardisierung der Grafiken nicht. Hier bestimmen Faktoren wie Wiedererkennungswert und Identität welches Symbol verwendet wird. Durch die große Anzahl an Mitstreitern, insbesondere bei Linux-Distributionen, und dem Wunsch sich von anderen Desktop-Umgebungen zu unterscheiden, entstehen viele verschiedene Identitäten, die den Endnutzer verwirren und somit die Akzeptanz mindern. In vielen Blogs findet man zudem Diskussionen darüber ob ein beschriebenes Blatt als Symbol für einen Texteditor wirklich so selbstverständlich ist. Viele Diskussionsteilnehmer empfinden bereits das Allgemeinwissen als Expertenwissen und behaupten, dass lediglich ein Umdenken notwendig sei um andere Visualisierungen als optimal zu empfinden. Von dieser Meinung würden wir uns jedoch gerne distanzieren, da diese Ansicht sehr extrem ist. Es handelt sich hier um ein Feld der Mensch - Computer - Interaktion, in welchem es darum geht dem Menschen eine Interaktion mit dem Computer zu ermöglichen. Wir sagen damit, dass dem Menschen, welcher sich in der realen Welt befindet der Zugang zur virtuellen Welt ermöglicht werden soll. Somit muss auch Rücksicht auf "Standards" aus dem Umfeld des Users genommen werden. Ein Endnutzer muss unterstützt werden und dies geschieht wenig bis gar nicht, wenn dieser dazu gezwungen wird umzudenken. Heutzutage ist man umgeben von Computersystemen und die nächste Generation wächst noch stärker damit verbunden auf. Dennoch darf man nicht vergessen, dass diese Standards auch weiterhin existieren und man erst über ein Umdenken in der Visualisierung diskutieren sollte, sobald sich diese ändern.

Usability

Usability aus dem englischen heißt übersetzt Brauch-/Nutz-/Bedienbarkeit und stellt in diesem Kontext den Grad Verwendbarkeit für einen Nutzer dar. Es handelt sich hierbei um eine Eigenschaft die vor allem im Bereich Mensch-Computer-Interaktion von großer Bedeutung ist. Die Usability wird als sehr gut angesehen wenn zum Beispiel ein Enduser kaum bis gar keine Einarbeitungszeit in das System benötigt und sofort mit dem System arbeiten kann. Wer schon einmal einem nicht sehr technikaffinen Mitmenschen geholfen hat einen PC zu bedienen weiß, dass ein komplexes System sehr schwer zu erklären sein kann.

Möchte man einem Windows-User die Verwendung der Desktop-Umgebung Gnome 3 erklären so dürfte dies wohl etwas dauern bis Windows-gewöhnte User mit der neuen Umgebung zurecht kommen. Je besser die Bedienbarkeit jedoch ist, desto leichter fällt es neuen Benutzern sich umzugewöhnen. Daran sieht man auch sehr gut wie wichtig Design Paradigmen bei der Entwicklung sind. Ein Design zu wählen, welches nicht selbsterklärend ist, kann zu einer Ablehnungshaltung unter den Usern führen.

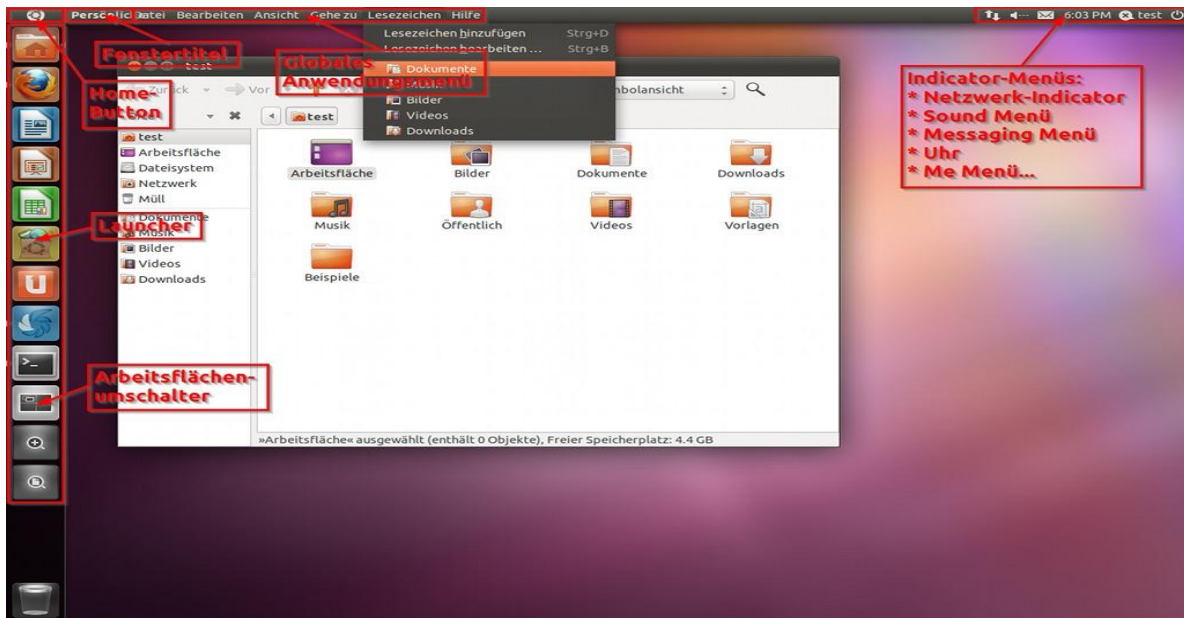
Diese Anforderung muss in vielen Aspekten des Betriebssystem erfüllt sein. Ein Beispiel ist Sicherheit in einem System welchen Linus Torvalds in einem Beitrag auf seinem Google+ - Account bezüglich OpenSuse stark kritisiert. Er bemängelt strengen Sicherheitsbestimmungen, die einen zwingen selbst bei den kleinen, alltäglichen Aufgaben das root - Passwort einzugeben. Dies wirft mehrere Probleme für User auf, einerseits muss dieser für einfache Prozesse über das root - Passwort verfügen, andererseits stört es den Arbeitsfluss auf Dauer erheblich. Auch wenn hierdurch mehr Sicherheit gewährleistet sein mag, sieht man dabei ein weiteres großes Problem: Kompromisse zu finden. User haben unterschiedliche Vorlieben und Ansichten. Eine verständliche Gegebenheit jedoch gleichzeitig eines der größten Probleme von Designern. Man muss jedem User ein möglichst ansprechendes Produkt liefern, im selben Moment aber auch auf wichtige 'Nebensächlichkeiten' wie Sicherheit achten. Hierbei sieht man wieder wie wichtig die Einhaltung von Design - Paradigmen ist. Sie können helfen, bereits bei der Entwicklung des Konzepts nicht zu viele Veränderungen vorzunehmen, die der Endnutzer nicht so schnell anzunehmen vermag.

Performanz

Ein System wird nach vielen Kriterien beurteilt aber kaum eines ist einem Endnutzer so wichtig wie die Performanz. Ein User möchte sich bei der Arbeit mit seinem Gerät nicht damit beschäftigen ob jetzt im Hintergrund eine gute Verschlüsselung der Daten erfolgt oder diese in einer guten Struktur abgelegt werden. Er möchte einen reibungslosen Umgang mit seinem System bei dem es nicht zu Verzögerungen kommt, zum Beispiel beim Starten eines Programms. Sollte sich der User mehr behindert als unterstützt fühlen so kann es sein, dass er sich für ein andere OS entscheidet. Damit so etwas nicht passiert kann, muss das Design Paradigma bereits bei der Erstellung solche Probleme möglichst vorweggreifen.

Desktop Umgebung

Mit der Desktop Umgebung eines Betriebssystems wird der Teil der Funktionalität bezeichnet, der um eine grafische Oberfläche ergänzt wurde. Im Gegensatz zur Kommandozeilen Umgebung wird dem User bei einem Desktop eine GUI (graphical user interface) zur Verfügung gestellt. Im Gegensatz zu Windows handelt es sich im Falle von Linux um flexibel gekoppelte Desktop Umgebungen



mit

Abbildung 1 Ubuntu Unity Desktop

mit Arbeitsumgebungen, wie GNOME, XFCE oder anderen.

Auf Grund dieser flexiblen Koppelung, erlaubt Linux dem User die unterschiedlichsten Kombinationen von Desktop Umgebungen. Je nach Wunsch des Users können außerdem noch individuelle Fenstermanager verwendet werden, die grundsätzlich für die Verwaltung der Fenster zuständig sind, in denen gestartete Programme angezeigt werden

LINUX DESIGN PARADIGMEN

Aufbauend auf dem Linux Kernel steht den Linux Distributionen eine gute Grundlage zur Verfügung, die eine Unterstützung von Eingabegeräten aller Art ermöglicht. Dies ist besonders für die Mensch Computer Interaktion interessant. Desktops können so konzipiert werden, dass sie

Hilfe spezieller Eingabemethoden auch besondere Anforderungen bei der Bedienung ermöglichen. Ein Beispiel dafür ist das OpenKinect-Projekt (<http://openkinect.org>), dass mit Hilfe von offener Bibliotheken (libfreenect) die Steuerung des Desktops durch Xbox Kinect Hardware ermöglicht.

Angewandten Design Paradigmen werden im Folgenden anhand von verschiedenen bekannten Linux-Distributionen näher aufgezeigt. Hierbei werden anhand von Abbildungen ihre speziellen Eigenheiten dargelegt.

Ubuntu

Ubuntu dürfte die wohl am weitesten verbreitete Linux-Distribution sein. Sie bietet auch Neueinsteigern durch

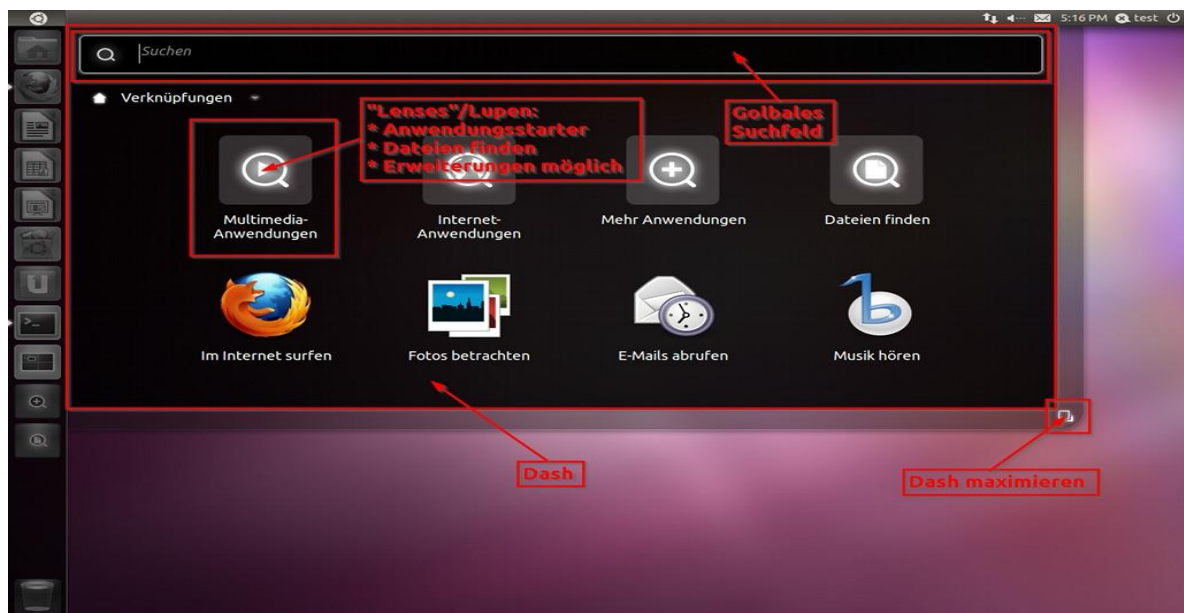


Abbildung 2 Ubuntu Unity Dash

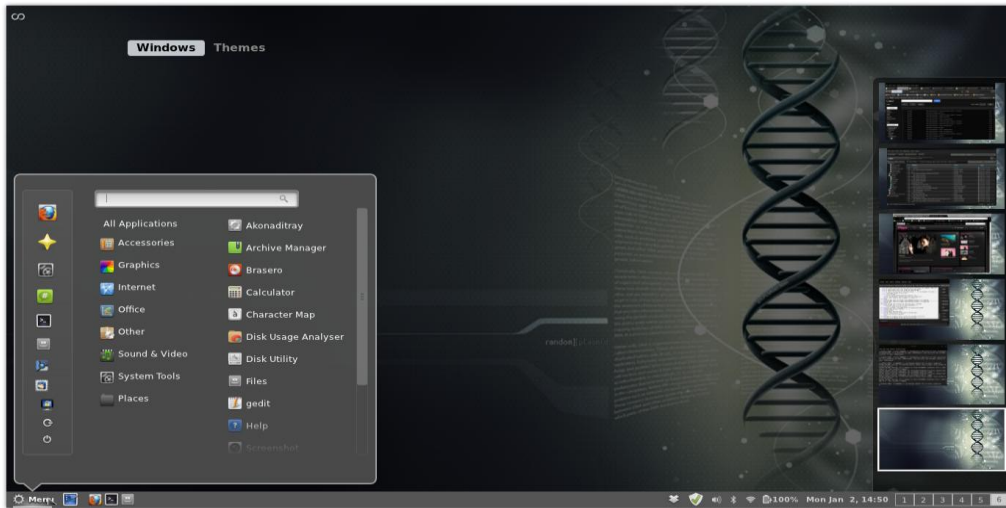


Abbildung 3 Linux Mint Desktop

bereits vorhandene Funktionalitäten die Möglichkeit schnell mit dem System zu arbeiten. Dies ist bei den meisten Distributionen nicht der Fall, um dem Enduser die Freiheit zu lassen, selbst zu entscheiden was er nutzen möchte. Diese Eigenschaft lässt sich auch wieder auf die verwendeten Design Paradigmen zurückführen.

In Abbildung 1 ist der Unity Desktop zu sehen, der für Ubuntu entwickelt wurde. Die wichtigsten Bestandteile der GUI sind neben dem Bildschirmfüllendem Desktop die Menüleiste, die Fenstertitel, globales Anwendungsmenü und das Indikatormenü enthält, sowie der Launcher, der einen schnellen Zugriff auf die meist benutzten Anwendungen des Users ermöglichen soll. Dieser Teil entspricht größtenteils dem bekannten Menüleisten-Konzept, dass in den meisten Desktop Oberflächen ähnlich aufgegriffen wird.

Abbildung 2 zeigt eine besondere Art der GUI, die Unity Dash. Diese wurde für Ubuntu konzipiert mit der Idee eine mächtige Suchleiste zu erstellen, die möglichst viele Menüs

obsolet macht und eine einfache Bedienung erlaubt (vgl. <http://www.omgubuntu.co.uk/2012/05/mark-shuttleworth-talks-ubuntu-12-10-netflix-laptops-and-more>). Mit Aussicht auf eine natürlichsprachliche Bedienung des Betriebssystems wird hier auf eine kontextsensitive Suche mit Zugriff auf installierte und installierbare Software, Dateisystem und ausgewählte Internet Ergebnisse gesetzt.

Mint

Andere Distributionen orientieren sich an den erfolgreichen kommerziellen Betriebssystemen und versuchen den Usern damit den Umstieg zu vereinfachen. Abbildung 3 zeigt die Desktop Umgebung eines aktuellen Linux Mint. Schnell ist die Ähnlichkeit zum Windows Desktop zu erkennen.

Linux Mint bietet außerdem eine Mischung aus Windows Startleiste und Launcher mit Suchleiste, wie auf Abbildung 4 sichtbar ist.

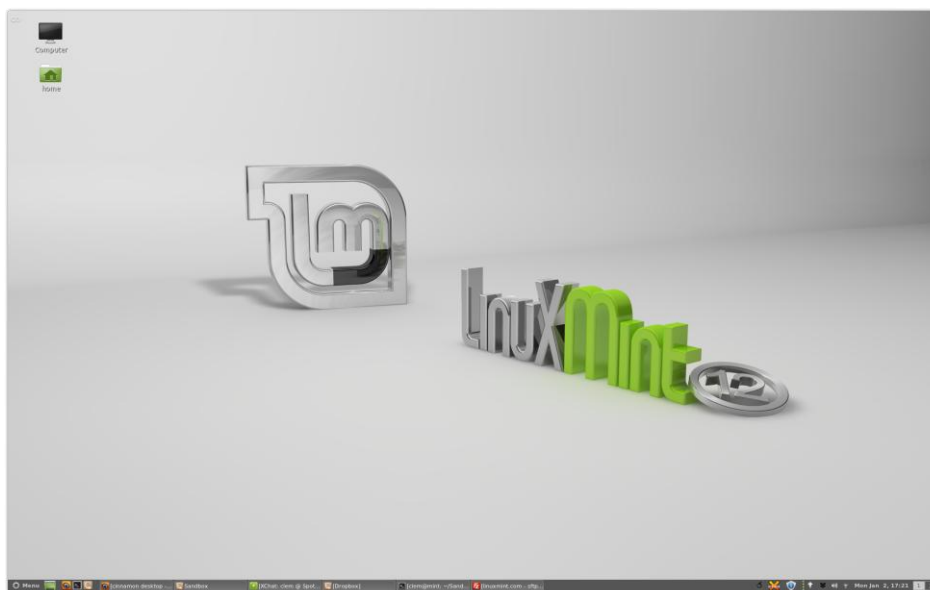


Abbildung 4 Linux Mint Startleiste

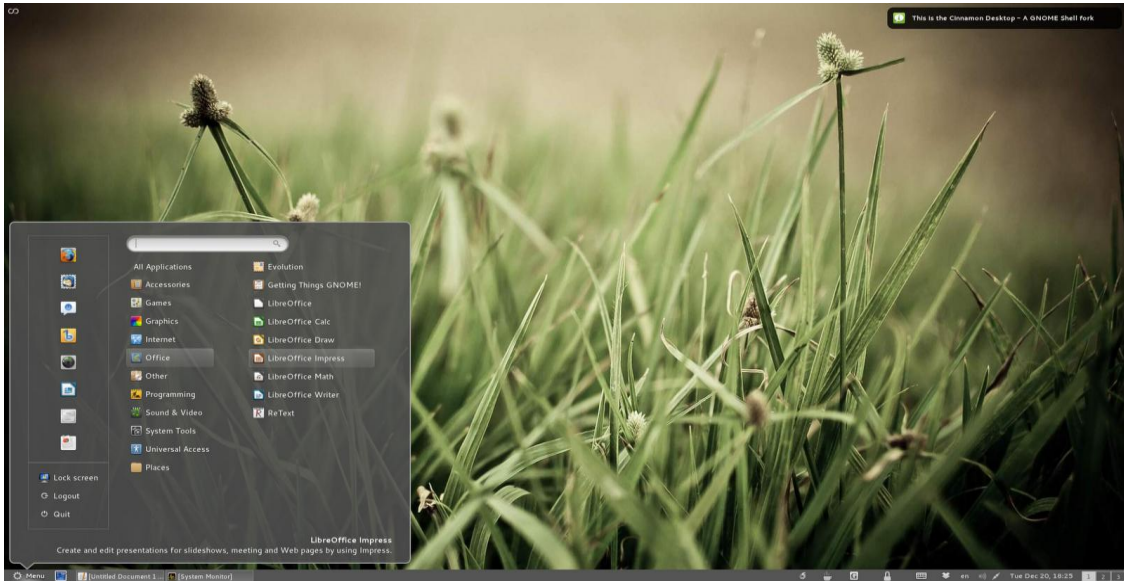


Abbildung 5 Gnome Cinnamon Desktop

Gnome

Ein Projekt bei dem das sogenannte 'forking' auftritt ist Gnome. Bei der Veröffentlichung von Gnome3 gab es viel Kritik am neuen Design-Konzept. Diese Änderung wurde von manchen so vehement abgelehnt, dass es zu einer Aufspaltung der Entwickler kam. Die unzufriedene Partei brachte nun eine Desktop-Umgebung heraus namens Cinnamon. Hierbei sieht man was passiert, wenn man bei einem community-driven Project, die Wünsche der mitentwickelnden Gemeinschaft nicht stark genug bei der Erstellung des Design Paradigma miteinbezieht.

Wie an Abbildung 5 (Gnome Cinnamon) und Abbildung 6 (Gnome 3) zu erkennen ist, verzichtet Gnome 3 komplett auf eine Taskleiste. Dies hat Vor- und Nachteile bezüglich

der Usability, welche allerdings wieder im Auge des Betrachters liegen.

XFCE

Hierbei handelt es sich, wie in Abbildung 7 zu sehen, um eine weniger aufwändige Oberfläche, welche versucht möglichst schlicht und einfach zu benutzen ist. Sie ist vollständig mit der Maus konfigurierbar und unterstützt Funktionalitäten wie Drag and Drop. Sie enthält ebenfalls schon grundsätzliche Funktionalitäten wie einen Datei-Manager, ist aber durchaus noch weiter anpassbar. Im Design Paradigma wurde starker Fokus auf hohe Performanz gelegt um auch auf älteren Maschinen oder solchen mit wenig Speicher lauffähig zu sein. Des Weiteren wurde auf die Usability geachtet um auch einfacher von

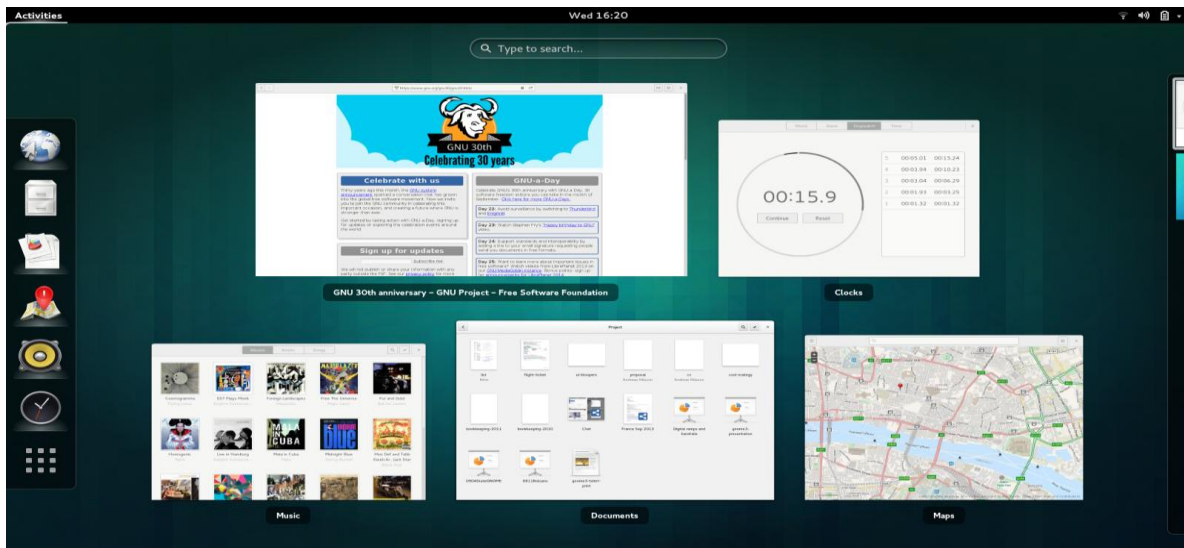


Abbildung 6Gnome 3 Desktop



Abbildung 7 XFCE Desktop

älteren Menschen genutzt werden zu können und somit auch bei einem speziellen Nutzerkreis auf höhere Akzeptanz zu stoßen.

KDE

Bei der Oberfläche KDE in Abbildung 8 hat man sich darauf konzentriert die Grafik und Usability auf ein hohes Niveau zu bringen. Der Desktop soll modern wirken und den User beim einrichten und benutzen Spaß bringen. Man merkt sofort die Ähnlichkeit zu den bekannten GUIs der Marktführer. Man hat zum Beispiel Standardmäßig eine Taskleiste am unteren Bildschirmrand, deren Aufteilung sehr an die Window-Startleiste erinnert. Dies ist mitunter ein Grund wieso diese Umgebung gut für Nutzer geeignet ist, die von Windows umsteigen. Dennoch ist KDE noch sehr stark veränderbar. Wie in anderen Distributionen kann man das Erscheinungsbild noch komplett verändern. Aufgrund ausgeprägt grafischen Möglichkeiten, ist die Performanz nicht ganz so hoch, sollte die Hardware nicht leistungsfähig genug sein.

FAZIT

Die verwendeten Design Paradigmen und ihre Entwicklung

sind zukunftsweisend und fortschrittbehindernd zugleich. Das mag widersprüchlich klingen aber es ist tatsächlich so, dass man zukünftig immer mehr auf den individuellen User zugehen möchte. Jedoch ebenso den Fortschritt dabei vorantreiben möchte. Hierbei birgt aber die „Offenheit“ von Linux-Distributionen ein großes Risiko. Forking, sorgt dafür, dass Linux speziell für die Community, welche es weiterentwickelt besser wird, im Allgemeinen aber immer mehr Splittergruppen entstehen. Daran können Distributionen zu Grunde gehen, da bei der Aufteilung der Gemeinschaft auch Kompetenzen sich verteilen. Es könnte des Weiteren dazu kommen, dass Linux-Distributionen wieder zu Umgebungen werden, die nur von speziellen Usern genutzt werden können. Hierdurch wäre sogar ein technologischer Rückschritt zu verzeichnen.

Einheitliches Denken und gemeinsame Ziele sind essentiell für ein erfolgreiches und zielorientiertes Projekt einer so großen Gemeinschaft.

Zielorientiert ist in diesem Kontext zu verstehen als der Wille fokussiert und ohne Umschweife dem Ziel, Weiterentwicklung der Desktop-Umgebung, entgegen zustreben. Dabei helfen besagte Standards. Sie verhindern das ausarten etwaiger Entwicklungsableger, die einem

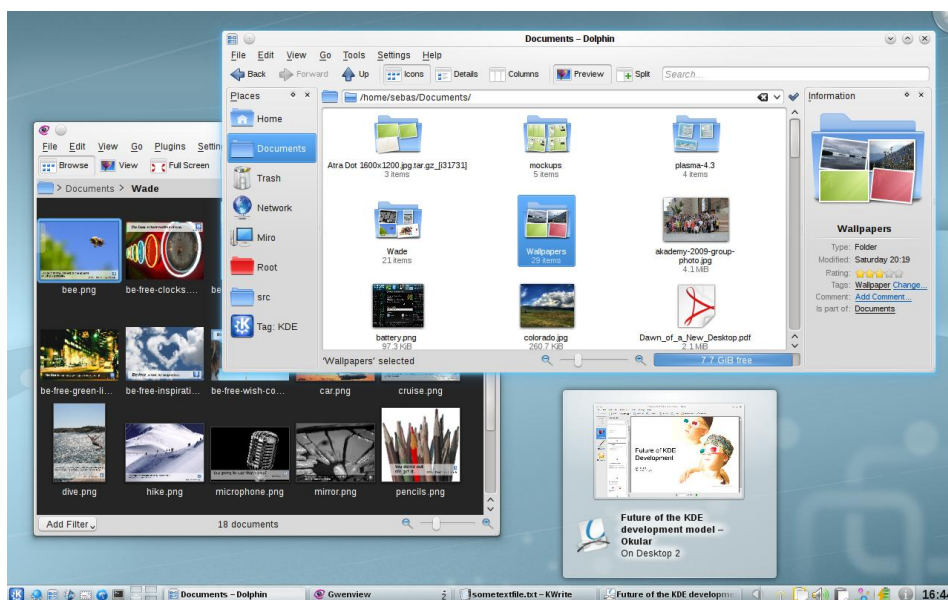


Abbildung 8 KDE Desktop

erfolgreichen Abschluss im Weg stehen würden. Ist von Beginn an geklärt, wer welche Aufgabe zu welchem Zeitpunkt erledigt haben muss, können auftretende Probleme weniger Einfluss auf das gesamte Projekt nehmen. Diese Aufgabe müssen von einem Design Paradigma mit übernommen werden.

An dieser Stelle sei auch noch erwähnt, dass ein solches Forking heut noch mehr als vor einigen Jahren den Untergang eines Betriebssystems bedeuten kann. In der heutigen Zeit wird immer mehr der Fokus auf die Interoperabilität aller zur Verfügung stehenden Geräten von Usern gelegt. Endnutzer möchten von ihrem Tablet eine kurz zuvor auf dem Handy eingegangene Email beantworten und dabei eine Datei anhängen, welche man auf dem Heim-PC erstellt hat. Das sollte so einfach und problemlos wie möglich sein. Sollten nun aber die einzelnen Betriebssysteme inkompatibel sein, so würde sich ein User zu seiner Erleichterung überlegen auf Geräte zu wechseln, welche die gewünschten Möglichkeiten anbieten. Gut gewählte Design Paradigmen können auch dabei helfen. Linux zeigt mit den angebotenen Desktop Environments viele Ansätze, wie man sie erstellen kann. Wie in diesem Paper erklärt sind sie abhängig von dem Fokus der Distribution. Somit würde die Design Paradigmen von XFCE auch in Zukunft nicht mit denen von Ubuntu zusammenführbar sein. Und somit bleibt es interessant was die nächsten Schritte in die Zukunft von Linux sein werden und wer sie gehen wird.

REFERENCES

- Adobe Acrobat Reader (Version 10.0) [Computer software]. Mountain View, CA: Adobe Systems.
Retrieved from: <http://get.adobe.com/reader/>
- Cherepii, S. (2013). *GNOME Shell Design*. Retrieved: December 21, 2013. from <https://wiki.gnome.org/Projects/GnomeShell/Design>.
- Snedon, J.-E. (2012). *Mark Shuttleworth on Ubuntu 12.10 Plans, Netflix & Windows 8*. Retrieved: December 21, 2013. from <http://www.omgubuntu.co.uk/2012/05/mark-shuttleworth-talks-ubuntu-12-10-netflix-laptops-and-more>.
- Serenity (2010). *Nepomuk in KDE –Theorie und Praxis*. Retrieved: December 21, 2013. from <http://beyondserenity.wordpress.com/2010/08/29/nepomuk-in-kde-theorie-und-praxis/>.
- Guedemann, C. (2011). *Die Macht der Design Paradigmen*. Retrieved: December 21, 2013. from http://www.webgate.biz/WGC/wgcblog.nsf/dx/Die_Macht_der_Design_Paradigmen.htm.
- Torvalds, L. (2012). *Venting*. Retrieved: December 21, 2013. from <https://plus.google.com/+LinusTorvalds/posts/1vyfmNCYpi5>.
- Koryavov, D. (2013). *Why GNU/Linux distros are still not rivals for Windows and OSX distros on desktops*. Retrieved: December 21, 2013. from <http://www.koryavov.net/2013/08/why-gnulinix-distros-still-not-rivals.html>.
- Olsen, C. (2011). *Coming soon to Linux Desktops*. In: *Linux J*. 2011(2011). Februar, Nr. 202. <http://dl.acm.org/citation.cfm?id=1938521.1938522>. – ISSN 1075–3583.
- Ravasio, P.; Schaer, S. G.; Krueger, H. (2004). In pursuit of desktop evolution: User problems and practices with modern desktop systems. In: *ACM Trans. Comput.-Hum. Interact.* 11 (2004), Juni, Nr. 2, 156–180. <http://dx.doi.org/10.1145/1005361.1005363>. – DOI 10.1145/1005361.1005363. – ISSN 1073–0516.
- Schwartz, M. (1995). *Guidelines for Bias-Free Writing*. Bloomington, IN: Indiana University Press.
- Shuttleworth, M. (2012). *Introducing the HUD. Say hello to the future*. Retrieved: December 21, 2013, from <http://www.markshuttleworth.com/archives/939>.

Abbildungen

- Abbildung 1. *Ubuntu Unity Desktop*. Retrieved: December 21, 2013. from http://linuxundich.de/wp-content/uploads/2011/04/unity_desktop.jpg.
- Abbildung 2. *Ubuntu Unity Dash*. Retrieved: December 21, 2013. from http://linuxundich.de/wp-content/uploads/2011/04/unity_desktop.jpg.
- Abbildung 3. *Linux Mint Desktop*. Retrieved: December 21, 2013. from <http://www.linuxmint.com/tmp/blog/1910/1.png>.
- Abbildung 4. *Linux Mint Startleiste*. Retrieved: December 21, 2013. from <http://www.linuxmint.com/tmp/blog/1910/2.png>.
- Abbildung 5. *Gnome Cinnamon Desktop*. Retrieved: December 21, 2013. from <http://1.bp.blogspot.com/-f9Lu5QC0ta8/TvDWCsU5bcl/AAAAAAAAAHFQ/jUjpO4Y7P78/s1600/cinnamon-gnome-shell-fork.png>.
- Abbildung 6. *Gnome 3 Desktop*. Retrieved: December 21, 2013. from <http://www.gnome.org/gnome-3/>.
- Abbildung 7. *XFCE Desktop*. Retrieved: December 21, 2013. from <http://www.xfce.org/>.
- Abbildung 8. *KDE Desktop*. Retrieved: December 21, 2013. from <http://www.kde.org/workspaces/plasmadesktop/screenshots/general-desktop.png>.