

Linux Design Paradigmen

Tim Krämer

Universität Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg
7kraemer@informatik.uni-hamburg.de

Bernhard Stöckl

Universität Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg
1stoeckl@informatik.uni.hamburg.de

ABSTRACT

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

Im Weiteren werden GNU/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, welches 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 hat und was dieses Verhalten für die zukünftige Entwicklung bedeutet.

Author Keywords

GNU/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

GNU/Linux

Bei GNU/Linux besteht aus dem sog. Linux Kernel und den GNU Paketen. Während der Linux Kernel, bestehend aus hardwarenaher Software für Scheduling, Multitasking, Gerätetreiber, Speicherverwaltung, usw. besteht, handelt es sich bei den GNU Paketen um die darauf arbeitende Software, bestehend aus Shell, Coreutils, Compiler wie GCC, Bibliotheken wie glibc und Umsetzung sämtlicher

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.

Funktionen des POSIX System Application Program Interface (vgl. Stallmann2006). Während sich zumeist die kürzere Bezeichnung *Linux* durchgesetzt hat, wird beispielsweise durch das Debian-Projekt, Knoppix, usw. die Bezeichnung *GNU/Linux* verwendet. Der Leserlichkeit zu Gunsten wird im folgenden der kürzere Begriff Linux verwendet.

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 Designparadigmen.

Design Paradigmen

Solche Designparadigmen (engl. Design Patterns) dienen dazu Programmierern zu helfen den Entwicklungsprozess ihrer Software nach einer bestimmten Art und Weise zu strukturieren. Dies eine möglichst hohe Qualität der Software ermöglichen, auch wenn viele verschiedene Personen an dem selben Projekt arbeiten. Ebenso soll die Weiterentwicklung und generelle Anwendung einer Software dadurch erleichtert werden. Designparadigmen stellen immer Standards dar, welche es gilt ein- bzw. aufrecht zu erhalten.

Ein Beispiel hierfür sind die bekannten DIN-Normen, die am 22. Dezember 1917 vom Deutschen Institut für Normung (DIN) auf dem Normenausschuss der Deutschen Industrie eingeführt wurde (vgl. Bauer2006). 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 Designparadigmen in der sog. "Open Source Community" einen essentiellen Schritt dar, um sich auf dem Weltmarkt behaupten zu können. Auf Software bezogen, sorgen diese Standards nicht nur dafür, dass ein Entwickler die Software weiterentwickeln kann, auch wenn er initial nicht involviert war, sondern auch, dass eine Endanwenderin oder ein Endanwender sich sicher sein kann, nach einem Update

immer noch mit dem Produkt arbeiten zu können. Der besseren Lesbarkeit halber und in Anlehnung an einschlägige Literatur, werden die Anwenderinnen und Anwender im Folgenden kurz durch den englischen Begriff User bezeichnet, bei denen es sich selbstverständlich um jegliche Art von Geschlecht handeln kann.

RELEVANTE QUALITÄTSMERKMALE VON SOFTWARE IM BEZUG AUF DESIGN PARADIGMEN

Bei der Entwicklung von Software spielen die Softwarequalitätsattribute, wie Performanz, Sicherheit, Testbarkeit und Benutzbarkeit eine wichtige Rolle. Über die Ausprägung der einzelnen Qualitätsattribute im Rahmen von Designparadigmen wird häufig schon zu Beginn anhand von Anforderungen und Qualitätssicherung entschieden. Eine einmal festgelegte Struktur ist später nur mit hohem Aufwand veränderbar, deshalb müssen Softwareentwickler im Bereich der Mensch-Computer-Interaktion in der Lage sein, schon während der Planung des Projekts Entscheidungen über die Ausrichtung der Softwaresysteme im Hinblick auf die Anforderungen der User zu treffen. Für die Desktopoberflächen werden zunächst die Qualitätsattribute Usability und Performanz untersucht.

Usability

Usability aus dem Englischen bedeutet Brauch-/Nutz-/Bedienbarkeit und stellt im Kontext von Desktopoberflächen den Grad der 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 einer Software wird als sehr gut angesehen, wenn beispielsweise ein Enduser kaum Einarbeitungszeit in das System benötigt und unmittelbar mit dem System arbeiten kann. Wer schon einmal einem weniger technikaffinen Menschen geholfen hat einen PC zu bedienen, weiß dass ein komplexes System sehr schwer zu erklären sein kann.

Bei der Bewertung von Usabilityaspekten von Desktopoberflächen, lohnt es sich zwischen unerfahrenen Usern und solchen zu unterscheiden, die ihre Desktopoberfläche nach längerer Zeit wechseln. Unerfahrene User besitzen noch keine festen Verknüpfungen zwischen grafischer Repräsentation und Funktionalität. Sie interpretieren das Gesehene neu und vergleichen mit bekannten Gegenständen aus der Realität und ihrer Umgebung. Erfahrene User einer anderen Oberfläche hingegen, vergleichen mit Ihren bisherigen Erfahrungen und bewerten die Usability der neuen Oberfläche anhand dessen, ob ihre Erwartungen erfüllt werden. Häufig wird hier die Ähnlichkeit zum vorherigen

System sehr stark gewichtet. Unterschieden werden kann in die drei Fälle:

- 1) gesuchte Funktionalität verbirgt sich hinter einem bekanntem oder ähnlichem Symbol -> hohe Akzeptanz zu erwarten
- 2) gesuchte Funktionalität verbirgt sich hinter einem unbekanntem, fremdem Symbol -> erneutes Lernen erforderlich
- 3) gesuchte Funktionalität verbirgt sich hinter einem bereits bekanntem, aber mit anderer Funktionalität verknüpftem Symbol -> negative Akzeptanz zu erwarten, sowie "überschreiben" von Wissen erforderlich

Möchte ein erfahrener Windows-User beispielsweise die Desktop-Umgebung Gnome 3 verwenden, so könnte die Umgewöhnungsphase einige Zeit dauern und auf weniger Akzeptanz treffen, als dies bei unerfahrenen Benutzern der Fall wäre. Je besser die Bedienbarkeit jedoch ist, desto leichter fällt es den Benutzern sich um zu gewöhnen. Je intensiver die Erfahrung mit einer Oberfläche ist, desto mehr freiwillige Akzeptanz für Umgewöhnung und Einarbeitungszeit sollte der User mitbringen.

An diesen Punkten zeigt sich, wie wichtig Designparadigmen bei der Entwicklung sind. Die Auswahl eines Designs und den benutzten Metaphern ist nicht trivial und sollte auf die Zielgruppe angepasst werden. Innovative Usability kann dazu beitragen, dass User ihre Oberfläche wechseln, aber auch zu einer Ablehnungshaltung unter den Usern führen.

Usability mit hohem Akzeptanzfaktor muss in vielen Bereichen des Betriebssystems erfüllt sein. Ein Beispiel sind die Sicherheitsrichtlinien eines Betriebssystems. Vor allem bei vernetzten Computersystemen sind Sicherheitskriterien, wie Vertraulichkeit, Integrität und Verfügbarkeit sehr wichtig, dennoch muss auch hier eine Balance zwischen Security und Usability des Systems gefunden werden, die möglichst viele User akzeptieren.

Dies ist auch heute noch ein großes Problem, bei dem sich die Meinungen einflussreicher Entwickler stark unterscheiden. So kritisierte Linus Torvalds im Frühjahr 2012 in einem Beitrag auf seinem Google+ - Account die strengen Sicherheitsbestimmungen von OpenSuse, die ihn zwangen, selbst bei kleinen, alltäglichen Aufgaben das root - Passwort einzugeben (Torvalds, L. (2012)). Dies werfe mehrere Probleme für User auf, einerseits müsse dieser für einfache Prozesse über das root - Passwort verfügen, andererseits störe es den Arbeitsfluss auf Dauer erheblich.

Auch wenn hierdurch mehr Sicherheit gewährleistet sein mag, sieht er dabei ein weiteres großes Problem: Kompromisse zu finden. User haben unterschiedliche Vorlieben und Ansichten. Eine verständliche Gegebenheit ist jedoch gleichzeitig eines der größten Probleme von

Usabilitydesignern. Jedem User ein möglichst ansprechendes Produkt liefern, im selben Moment aber auch auf andere Qualitätsmerkmale, wie Sicherheit zu achten, ist keine einfache Aufgabe. Auch hier zeigt sich wieder, wie wichtig die Einhaltung von Designparadigmen ist. Sie können helfen, bereits bei der Entwicklung des Konzepts auf bereits bekannte und akzeptierte Aspekte zurückzugreifen und die Erfahrung für die User so angenehm, wie möglich zu gestalten.

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.

KOMPONENTEN VON DESKTOP OBERFLÄCHEN

Grafische Oberfläche

Mit "Grafik" ist in diesem Fall die Desktopumgebung und deren Aufbau gemeint. Die Anforderungen, die den grafischen Teil des Designparadigmas ausmachen, befähigen den User die Funktionalitäten seines Systems zu nutzen.

So gibt es kommandozeilenbasierte Distributionen ohne Grafiken, diese können aber von dem Großteil der potentiellen Enduser, aufgrund von fehlendem Expertenwissen, nicht ohne hohen Aufwand genutzt werden (vgl Ravasio et al (2004)). Bei der Gestaltung von grafischen Oberflächen achtet man darauf, möglichst verständliche Visualisierungen zu verwenden. Möchte der User beispielsweise ein neues Textdokument erstellen, ist die Verwendung einer Metapher geeignet. So ist es für den User schnell zu erkennen, dass sich hinter einem Symbol, welches geschriebene Zeilen auf einem Blatt Papier darstellt, die gewünschte Funktionalität finden lässt. Der Gebrauch von Metaphern helfen der Akzeptanz eines Systems, indem sie sich visuellen Referenzen aus dem Leben des Users bedienen.

Ein Designparadigma lässt sich also anhand der Leichtigkeit bewerten, mit der sich ein User in der gestalteten virtuellen Umgebung zurechtfinden kann. Der

Ansatz bestimmte Metaphern als Standard für die Repräsentation einer bestimmten Funktionalität festzulegen ist nahe liegend, jedoch durch den heutige Konkurrenzkampf auf dem Weltmarkt und auch die persönlichen Präferenzen der unterschiedlichen Designer nicht einfach. Hier bestimmen Faktoren wie Wiedererkennungswert und Identität welches Symbol verwendet wird. Aus dem Wunsch sich von anderen Desktop-Umgebungen zu unterscheiden, entstehen viele verschiedene Versionen, die den Endnutzer verwirren können und somit die globale Akzeptanz mindern. So lassen sich beispielsweise in vielen Blogs Diskussionen darüber finden, ob ein beschriebenes Blatt als Symbol für einen Texteditor tatsächlich selbstverständlich ist. Viele Diskussionsteilnehmer empfinden bereits das häufig vermutete Allgemeinwissen als Expertenwissen und sind der Meinung, dass lediglich ein Umdenken notwendig sei um andere Visualisierungen als optimal zu empfinden.

Aus der wissenschaftlichen Betrachtung erscheint es sinnvoll sich auf empirische Studien und Erfahrungswissen der letzten Jahrzehnte zu berufen. Da es sich bei diesem Thema um einen Teilbereich der Mensch-Computer-Interaktion handelt, bedient sich dieses Paper der Erkenntnisse dieses Fachgebiets und diese empfehlen, dass auch weiterhin versucht wird visuelle Repräsentationen so zu gestalten, dass sie sich möglichst an erprobten Metaphern und Erwartungen des Users halten. Die Repräsentation von realen Gegenständen aus dem Leben des Users ist dafür eine erfolgreiche Methode. Ein Endnutzer muss unterstützt werden und dies geschieht wenig bis gar nicht, wenn dieser dazu gezwungen wird umzudenken. Heutzutage sind im Besonderen junge Menschen alltäglich von Computersystemen umgeben, und die nächste Generation wächst noch stärker damit verbunden auf. Daraus entstehen dem rapiden technischen Wandel zu Grunde liegende Artefakte, so ist vielen Kindern heute nicht mehr bekannt was eine Diskette ist, dennoch wurde die Metapher des "Speicherns" häufig gelernt und akzeptiert. Es darf nicht vergessen werden, dass auch Quasistandards veralten können und eine Diskussion über eine Anpassung und Modernisierung der verwendeten Visualisierung trotzdem sinnvoll sein kann.

Desktop Umgebung

Mit der Desktopumgebung 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 Kontrast zu Windows handelt es sich im Falle von Linux um flexibel gekoppelte Desktop Umgebungen 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.

DESKTOP OBERFLÄCHEN FÜR LINUX

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 mit Hilfe spezieller Eingabemethoden auch besondere Anforderungen bei der Bedienung ermöglichen. Ein Beispiel dafür ist das OpenKinect-Projekt (<http://openkinect.org>), das mit Hilfe von offenen Bibliotheken (libfreenect) die Steuerung des Desktops durch Microsofts "XBOX Kinect" Hardware ermöglicht. Als Beispiel für die Integration von modernen Interaktionsgeräten, hat sich bei diesem Projekt eine Gruppe von Entwicklern damit beschäftigt offene Treiber für das proprietäre Gerät zu entwickeln. Mit Hilfe von bereits vorhandenen freien, opensource Bibliotheken konnte für den Linux Kernel schnell eine funktionsfähige Version veröffentlicht werden, die das Interesse vieler weiterer Entwickler geweckt hat.

Das OpenKinect-Projekt ist eines der vielzähligen Projekte, die durch das Engagement der Linux

Entwicklergemeinschaft entwickelt wurden. Häufig war dies nur durch die klare Struktur des Linux Kernels und der offen zugänglichen Dokumentation der Hardwarebibliotheken möglich.

Bei Desktopoberflächen handelt es sich um das gleiche Prinzip des sog. "community-driven-development". Die basisdemokratische Zusammensetzung solcher Communitys erschwert häufig das Durchsetzen von klaren Designparadigmen als Richtlinien. Um zu beurteilen, wie die Projekte dennoch erfolgreich entwickelt wurden und durch hohe Akzeptanzfaktoren eine hohe Popularität genießen, werden im Folgenden die populärsten Linux Desktopoberflächen vorgestellt. Ein besonderes Augenmerk wird auf die verwendeten Designparadigmen gerichtet und anhand von verschiedenen Linux-Distributionen erklärt. Hierbei werden anhand von Abbildungen ihre speziellen Eigenheiten aufgezeigt. DistroWatch.com bietet einen Anhaltspunkt dafür, welche Distributionen und Desktop-Oberflächen am häufigsten genutzt werden. Die dort berechnete Rangfolge stellt allerdings nicht unbedingt ein Abbild des tatsächlichen Marktanteils dar.

Im Dezember 2013 führen Linux Mint, Ubuntu, Debian, Mageia und Fedora die Rangliste an (vgl. <http://distrowatch.com/index.php?dataspan=2013>). Daraus ergibt sich eine große Liste von unterstützten Desktopoberflächen:

AfterStep, Awesome, Blackbox, Cinnamon, Enlightenment, Fluxbox, GNOME, IceWM, KDE, LXDE, MATE,

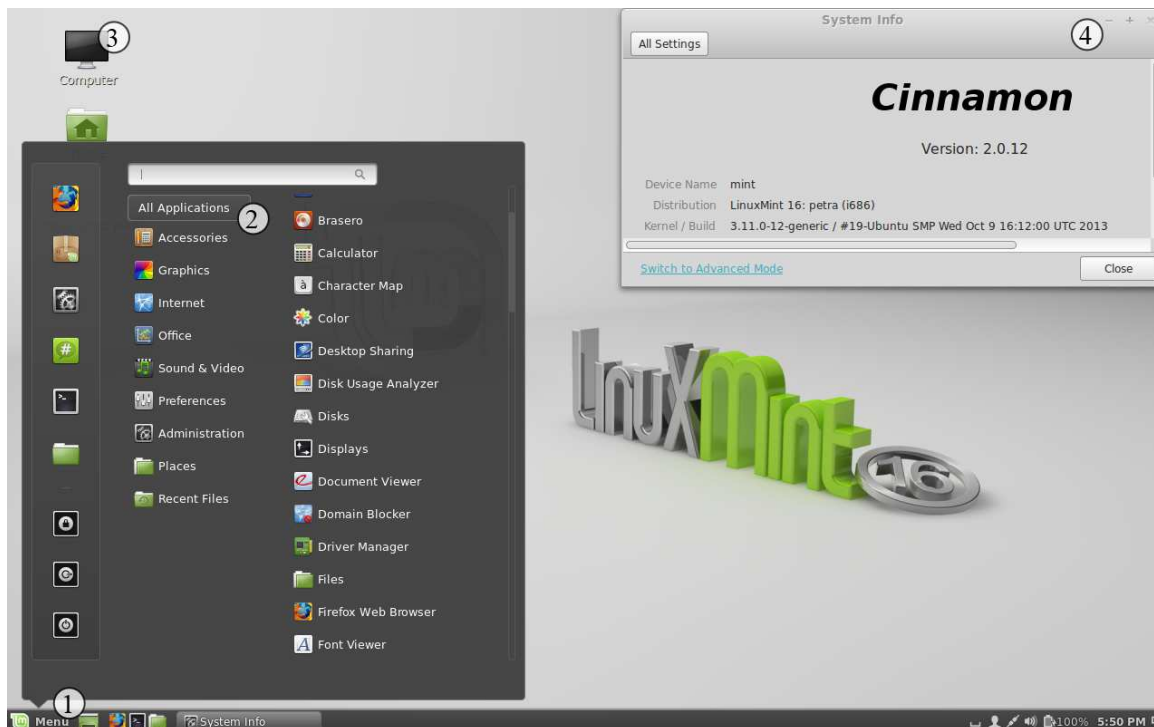


Abbildung 1 Linux Mint Desktop

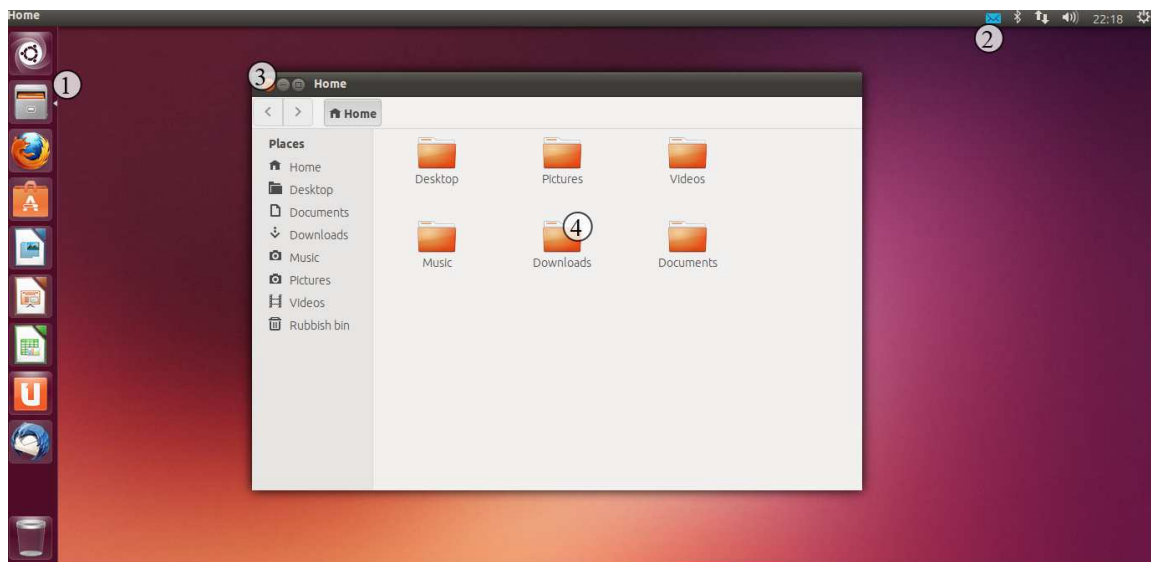


Abbildung 2 Ubuntu Unity Desktop

Openbox, Ratpoison, Razor-qt, Unity, WMaker, Xfce.

Eine Auswahl davon wird im kommenden Abschnitt vorgestellt.

Mint

Linux Mint ist eine der am weitesten verbreitetsten Linux-Distributionen für Desktoprechner. Bei oberflächlicher Betrachtung wird schnell klar, dass hier das Prinzip der bekannten Beidenelemente aufgegriffen wurde, so bietet Linux Mint eine Mischung aus Startleiste und Launcher mit Suchleiste, wie auf Abbildung 1 sichtbar ist. Schnell ist die Ähnlichkeit zum Windows Desktop zu erkennen.

Ubuntu

Ubuntu als eine weitere sehr verbreitete Linux-Distribution bietet auch Neueinsteigern durch 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 (vgl. Shuttleworth, M.(2012)).

In Abbildung 2 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 Anwendungs Menü 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.

Die Unity Dash wurde für Ubuntu konzipiert, mit der Idee eine mächtige Suchleiste zu erstellen, die möglichst viele Menüs obsolet macht und eine einfache Bedienung erlaubt (vgl. Snedon, J.-E. (2012)). 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.

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 3 (Gnome Cinnamon) und Abbildung 4 (Gnome 3) zu erkennen ist, verzichtet Gnome 3 komplett auf eine Taskleiste (Cherepii, S. (2013)). Dies hat Vor- und Nachteile bezüglich der Usability, welche allerdings wieder im Auge des Betrachters liegen.

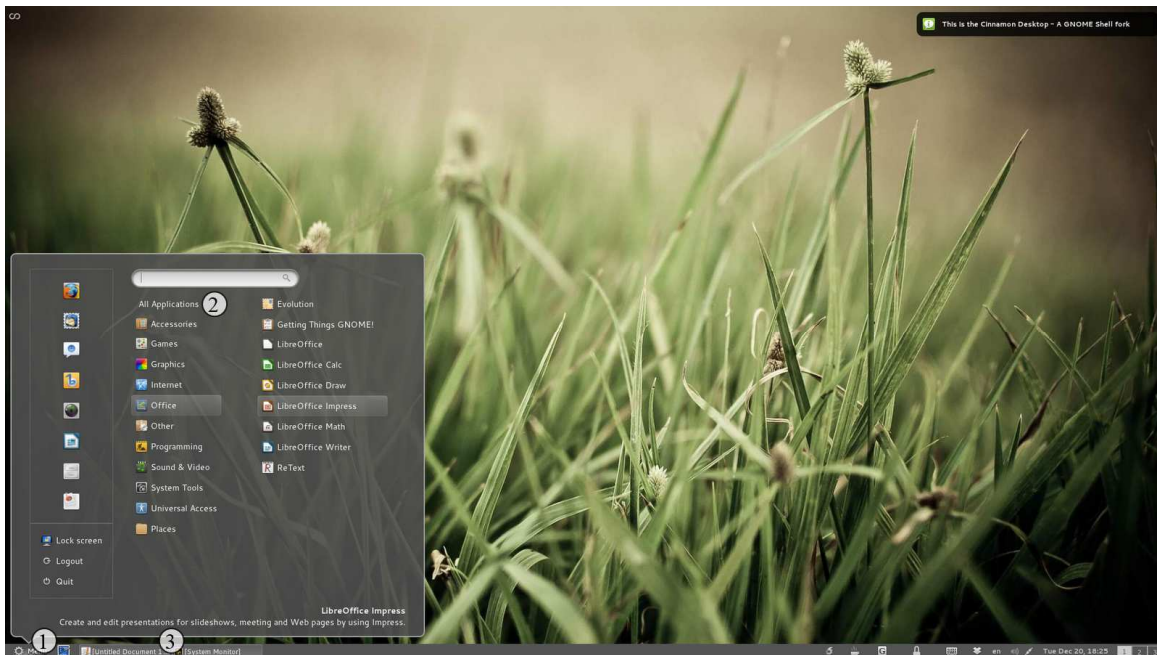


Abbildung 3 Gnome Cinnamon Desktop

XFCE

Hierbei handelt es sich, wie in Abbildung 5 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 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 6 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 oder auf Untergruppen von KDE zurückgreifen (siehe Serenity

(2010)). Aufgrund ausgeprägter grafischen Möglichkeiten, ist die Performanz nicht ganz so hoch, sollte die Hardware nicht leistungsfähig genug sein.

Eingangs wurden ein paar Fragen gestellt die wir nun noch einmal aufgreifen wollen. Zuerst wollen wir versuchen zu beantworten welche Design-Entscheidungen sich im Linux Desktop Bereich erfolgreich durchgesetzt haben. Dies ist durchaus eine berechtigte Frage nach alledem was zuvor über die unterschiedlichen Ansätze genannt wurde. Jedoch ist die Antwort leider genauso wenig eindeutig, was an dem Aspekt 'community driven' liegt. Wir zeigten viele Oberflächen und nannten dazu auch verschiedene Zielgruppen und eben diese diversen Gruppen machen die Beantwortung dieser Frage so kompliziert. Die Vielzahl der Nutzer, deren spezielle Bedürfnisse befriedigt werden sollen reicht von Anfängern, Usern die einfach mal 'vorbeischaun' wollen, über Entwickler, Alltagsuser, Kommandozeilen-Fanatikern zu Menschen mit speziellen Bedürfnissen (Stichwort: barrierefreies Internet). Was man allerdings übergeordnet als Antwort geben kann ist, dass der User immer mehr zum unbewussten Designer eines Systems wird. Früher bewegten sich Entwickler in den Grenzen des technisch Möglichen und designten was umsetzbar war. Heutzutage sind Entwickler mehr wie Handwerker zu sehen, die das Wissen und das Geschick besitzen die Phantasien und Wünsche der Designer Wirklichkeit werden zu lassen. Der Designer ist nun der Endnutzer und somit kann man neben den speziellen, stark divergierenden Wünschen auch allgemeine, sich ähnelnde Entscheidungen benennen. Diese wären wiederum die Grundbedürfnisse hohe Performanz und Usability sowie erwartungskonformes Verhalten des Systems.

Eine weitere Frage war wieso diese Designentscheidungen zu einer breiten Akzeptanz führten. Dass die Paradigmenwahl so gut angenommen wurden, liegt wohl zum einen an den erfüllten, oben genannten Grundbedürfnissen aber zum anderen an psychologischen Aspekten, an welche man zuerst vielleicht gar nicht so denken würde. Der Mensch ist sehr anpassungsfähig, wenn er seine eigenen, für sich selbst höherpriorisierten Anliegen erfüllt sieht. Dies soll heißen, dass nicht alles optimal sein muss, damit man etwas annimmt. Somit ist auch diese Frage eine sehr differenziert zu beantwortende. Die entscheidenden Kriterien kann man unterteilen in die persönlichen, aus Erfahrungen und gegebenen Umständen (siehe, im Absatz zuvor genannte, Zielgruppen) resultierenden Sehnsüchte. Man darf nicht übersehen, dass der Mensch das Gerät auf welchem das Betriebssystem installiert ist heutzutage sehr in sein persönliches Leben integriert und sich auch immer stärker damit identifiziert. Deshalb sind auch allgemeine psychologische Faktoren zu nennen, welche eine Rolle bei der Akzeptanz spielen. Ein Beispiel hierfür ist das Gefühl 'besonders' zu sein, da zum Beispiel die Desktopoberfläche /k personalisiert k/ ist oder man sich in seinen Bedürfnissen verstanden fühlt. Ein Problem unserer heutigen Gesellschaft hält hier Einzug. Die Anonymisierung in der Masse der Konsumenten. Abschließend hierzu ist nun zu sagen, dass die breite Akzeptanz aus dem breiten Angebot und die Personalisierbarkeit der einzelnen Distributionen entsteht.

FAZIT

Die letzte Frage welche wir eingangs gestellt haben,

welchen Einfluss die gegenwärtige auf die kommende Entwicklung möglicherweise hat, möchten wir in unserem Fazit näher erläutern.

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 voranzutreiben versucht. 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 erfolgreichen Abschluss im Weg stehen würden. Ist von Beginn an geklärt, wer welche Aufgabe zu welchem Zeitpunkt erledigt haben muss, können auftretende



Abbildung 4 Gnome 3 Desktop



Abbildung 5 XFCE Desktop

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

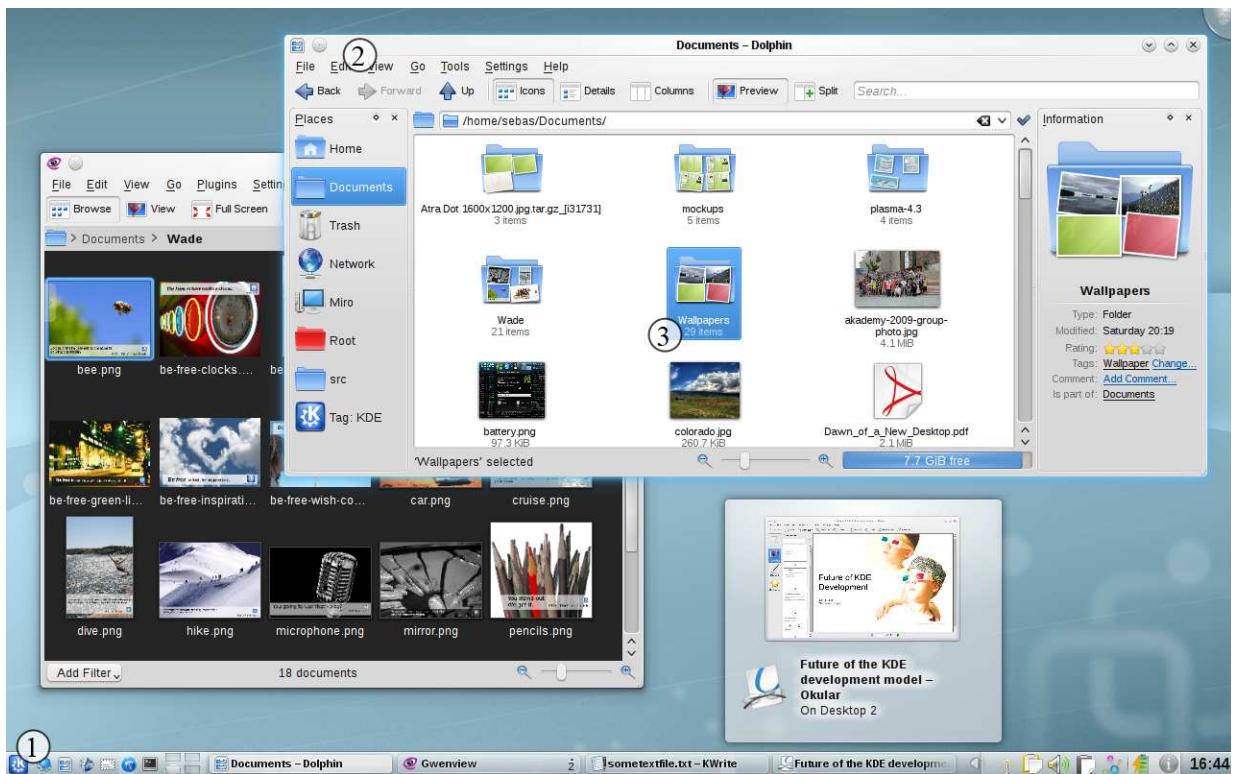


Abbildung 6 KDE Desktop

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 würden die Design Paradigmen von XFCE auch in Zukunft nicht mit denen von Ubuntu zusammenführbar sein. Auch hierbei zeigt sich, dass die Größte Stärke der Linux-Distributionen, ihre Vielfältigkeit auch gleichsam ihr größtes Problem ist (Koryavov, D. (2013)). 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/>.
- Stallman, R. (2006). GNU/Linux. Ist der Name Schall und Rauch? Retrieved: February 20, 2014 from: <http://www.gnu.org/gnu/why-gnu-linux.de.html>.
- Bauer, W. (2006). Die Norm regiert die Welt. Retrieved: December 21, 2013 from: http://www.focus.de/wissen/wissenschaft/focus-reportage_aid_117855.html.
- 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. *Linux Mint Desktop*. Retrieved: December 21, 2013. from <http://www.linuxmint.com/tmp/blog/1910/1.png>.
- Abbildung 2. *Ubuntu Unity Dash*. Retrieved: December 21, 2013. from http://linuxundich.de/wp-content/uploads/2011/04/unity_desktop.jpg.
- Abbildung 3. *Gnome Cinnamon Desktop*. Retrieved: December 21, 2013. from <http://1.bp.blogspot.com/-f9Lu5QC0ta8/TvDWCsU5bcI/AAAAAAAAAHFQ/jUjpO4Y7P78/s1600/cinnamon-gnome-shell-fork.png>.
- Abbildung 4. *Gnome 3 Desktop*. Retrieved: December 21, 2013. from <http://www.gnome.org/gnome-3/>.
- Abbildung 5. *XFCE Desktop*. Retrieved: December 21, 2013. from <http://www.xfce.org/>.
- Abbildung 6. *KDE Desktop*. Retrieved: December 21, 2013. from <http://www.kde.org/workspaces/plasmadesktop/screenshots/general-desktop.png>.