

EUD und Web 2.0

Dennis Koch

Fachgebiet End-User-Development

Prof. Dr. Volker Wulf

Institut für Wirtschaftsinformatik und neue Medien

Fachbereich 5, Wirtschaftswissenschaften

Universität Siegen

Gliederung

1. Einleitung
2. Was ist EUD?
3. Was ist Web 2.0?
4. Flickr
5. Wordpress
6. Pipes
7. Fazit
8. Referenzen

Einleitung

Das Internet in seiner derzeitigen Form erhält eine immer höhere Bedeutung in der Informationsbeschaffung sowie Erzeugung. Durch diese Tatsache wird es aber auch immer schwieriger auf geeignete Weise die immer größeren Informationsmengen zu verwalten und zu pflegen. Hinzu kommt, dass immer mehr Laien ohne Programmierkenntnisse aus ihrer Nutzerrolle entfliehen wollen und selber Informationen oder sogar Dienstleistungen im Internet anbieten wollen. Wie muss das Internet bzw. dessen Dienste also beschaffen sein, damit sie diesen Anforderungen genügen können?

Um diese Frage klären zu können werde ich das Web 2.0 im Ganzen sowie gängige Web 2.0 Anwendungen auf ihre Eigenschaften als EUD-Anwendung untersuchen. Beginnen werde ich allerdings mit den folgenden Begriffsdefinition.

Was ist EUD?

Nach der Definition von Lieberman et al. versteht man unter End-User-Development eine Ansammlung von Methoden, Techniken und Werkzeugen, die es einem unerfahrenen User ermöglichen Teile eines Softwaresysteme zu verändern, erweitern oder hinzuzufügen. Diese Definition wirkt auf den ersten Moment etwas unpräzise, allerdings ist eine genauere Definition kaum möglich, wenn man sich die unterschiedlichen Ausprägungen von EUD vor Augen führt. Angefangen bei einer Tabellenkalkulation mit ihren Möglichkeiten komplexe Berechnungsformulare zu erzeugen über visuelle Programmierungsumgebungen bis hin zu Softwaresystemen deren Eigenschaften von Grund auf verändert und neu zusammen gestellt werden können.

Man kann also inzwischen EUD-Eigenschaften in nahezu allen Softwareprojekten wiederfinden und sei es nur die Möglichkeit Programmelemente nach eigenen Wünschen mit Hilfe von selbst erstellten Skins einzufärben oder, wie in vielen Office-Suites der Fall, bestimmte Bedienelemente nach eigenen Vorlieben zu sortieren oder zu verstecken.

Nach Lieberman sollte ein gutes Softwaresystem möglichst viele Facetten von EUD vereinen und sich auf diese Weise an die persönlichen Fähigkeiten des Nutzers anpassen lassen. Er unterteilt die Schnittstellen zu EUD in zwei Teilbereiche. Erstens „Parameterisation or Customisation“ und zweitens „Program Creation and Modification“.

Unter „Parameterisation or Customisation“ versteht er lediglich die Möglichkeit Programme mit Hilfe von vorgefertigten, in der Regel eingeschränkten Optionen nach seinen Wünschen anzupassen. Meistens bezieht sich diese Art von EUD nur auf die Anordnung bzw. den Zugriff auf bestimmte Bedienelemente der Software

(oder auf vorgefertigte aber alternierende Wege zur Behandlung von Daten).

Der zweite Teilbereich ist nach Liebermanns Meinung der Interessantere, da der End-User dort einen viel größeren Einfluss auf den eigentlichen Programmablauf haben kann. Dieser Bereich beinhaltet das „Programming by Example“, bei dem das Softwaresystem aus bestimmten Interaktion des Benutzers eine Routine generiert. Desweiteren das „Incremental programming“ bei dem nur einzelne Teile eines Programms vom Benutzer implementiert werden müssen. Auf diese Weise wird es einfacher das Programm an seine Anforderungen anzupassen ohne dabei das komplette System verstehen zu müssen. Sowie das „Model-based development“, welches dem Benutzer erlaubt verschiedene Module einer Software miteinander zu verknüpfen, um so das gewünschte Ergebnis zu erreichen. Als letztes Beispiel nennt er die Möglichkeit eines Users seine Einstellungen zu speichern und anderen zu Verfügung zu stellen. Er bezeichnet es als eine erweiterte Form der Parameterisation.

Was ist Web 2.0?

In dem Buch von Tim O'Reilly „What is Web 2.0?“ wird der zeitliche Beginn des Web 2.0 mit dem Platzen der Dot-Com-Blase im Herbst 2001 datiert. Die Bezeichnung entstand aus einer Diskussion zwischen Tim O'Reilly und Dale Dougherty in dem sie die Veränderungen, die durch das Platzen entstanden sind mit einem Schlagwort beschreiben wollten. Zu dieser Zeit entstanden immer mehr interessante Webanwendungen, die sich durch einige Eigenschaften von den „alten“ Web 1.0 Anwendungen unterschieden. Allerdings besitzt der Begriff Web 2.0 keine genaue Abgrenzung sondern stellt vielmehr eine Art Prinzipienansammlung dar, dessen Umsetzungen in unterschiedlichen Ausmaßen statt findet.

Das Hauptprinzip ist die Nutzung des Webs als Plattform. Daraus kristallisieren sich nun auf unterschiedliche Weise verschiedene Anforderungen an ein Unternehmen dessen Strategie es ist eine Web 2.0 Anwendung zu schaffen. Diese Anforderungen bezeichnet Tim O'Reilly als Kernkompetenzen. Seiner Meinung nach bestehen diese aus den folgenden 7 Punkten:

- (1) Ein Web 2.0 System sollte ein Service darstellen und nicht eine alleinstehende Software. Tim O'Reilly bezieht sich in diesem Punkt auf Google, dessen Strategie es niemals war ein Produkt fertig verpackt an den Endkunden zu verkaufen, sondern, entgegen der herkömmlichen Softwareentwicklung, einen Dienst anzubieten, der ständig und durchgehend verbessert wird, ohne dass dem Kunden ein bestimmter Releasezyklus, eine Software oder eine Hardware vorgeschrieben wird. Ebenso sollte, da die Software die hinter diesem Service steht niemals das Unternehmen verlassen muss, die Skalierbarkeit der Ressourcen möglichst kostengünstig sein.

- (2) Der eigentliche Wert einer Web 2.0 Anwendung bzw. des Unternehmens sollte auf der Kontrolle über einzigartige, schwer nachzubildende Datenquellen liegen, deren Wert proportional zur Nutzungshäufigkeit steigt [vgl. ...] Das bedeutet, im Fall von Googles Suchsystem liegt der eigentliche Wert in der Datenbank, dessen Inhalt sie nicht einmal selber erzeugen müssen. Die Macht eines Unternehmens besteht somit nicht mehr aus der Kontrolle einer bestimmten Software, sondern vielmehr aus der Kontrolle von Daten die nur mit Hilfe einer spezialisierten Software verwaltet werden kann.
- (3) Da die Unternehmen nicht mehr unbedingt selbst für den Inhalt sorgen müssen, müssen sie aber deren Benutzer genügend Vertrauen als Mitwirkender entgegenbringen. Als Beispiel dafür wird Wikipedia genannt, dessen kompletter Inhalt vollständig von den Benutzern generiert wird. Folglich bedeutet das aber auch, dass die Qualität einer solchen Anwendung nur mit der Anzahl der Benutzer steigen kann.
- (4) Durch die hohe Anzahl der Benutzer ergibt sich für die Unternehmen die Möglichkeit diese als eine Art kollektive Intelligenz zu nutzen, um ihr Angebot für eine noch breitere Masse optimieren zu können. So verweisen viele Online-Shops, allem voran Amazon, auf die aktuell beliebtesten Artikel oder schlagen Alternativen auf Basis der Wahl anderer Kunden vor. Diese Vorschläge ergeben sich ausschließlich aus der Interaktion der Kunden mit dieser Anwendung, sei es über ein Bewertungssystem oder nur aufgrund von Seitenaufrufen.
- (5) Die „kleinen“ Benutzer sollten also stärker in den Fokus eines Unternehmens rücken, da sie im Gegensatz zu den Großkunden aufgrund ihrer Masse im Endeffekt einen größeren Marktwert besitzen und Verträge zwischen einzelnen einfacher ausgehandelt werden können. Die gezielte Bildung von Communities hilft dabei schnell eine große Masse an Kunden zu erzeugen, da deren Mitglieder in der Regel unfreiwillig weitere Mitglieder werben und somit auch mögliche Einnahmen erhöhen.
- (6) Um diesen Effekt noch gezielter auszunutzen zu können sollte ein Webservice nicht nur auf einen bestimmten Gerätetyp spezialisiert werden. Beispielsweise wird inzwischen von vielen Unternehmen die Möglichkeit angeboten direkt von seinem Mobiltelefon Content ins Internet zu laden. Sei es als Blogeintrag, Positionsortung oder um einfach nur die Bilder aus der Kamera direkt in eine Bildergalerie zu laden. Dadurch wird der Kontakt des Benutzers mit dem Service erhöht und somit auch der Wert dieses Services.
- (7) Mit dem letzten Punkt beschreibt Tim O'Reilly auf welche Weise eine Web 2.0 Anwendung strukturell realisiert werden sollte, damit sie erfolgreich sein kann. Seiner Meinung nach sollten sie besonders einfach und leicht programmiert sein. Anhand Googles Mapservice, auf das ich später noch

mal kurz eingehen werde, beschreibt er welche Vorteile ein solches Lightweight Programming Model hat und welche Auswirkungen dieses Prinzip auf ganze Geschäftsmodelle haben kann.

Wie man bereits erkennt, ist es zum Teil nicht möglich gleich mehrere dieser Prinzipien gleichzeitig zu erfüllen. Allerdings ist das auch nicht zwangsläufig erforderlich, da sich ein Unternehmen auch dadurch positionieren kann, indem es ausgewählte Merkmale besonders innovativ realisiert. Im Folgenden werden ich mich nun auf einige Anwendungen die man im Kontext von EUD ansiedeln kann untersuchen und einzelne Aspekte genauer analysieren.

Flickr

Flickr bezeichnet sich selber als wahrscheinlich beste Online-Fotoplattform der Welt. Sie wurde ursprünglich als Teil eines Internetrollenspiels entwickelt um den Spielern die Möglichkeit zu geben während des Spiels Bilder ins Internet zu laden und anderen zur Verfügung zu stellen. Das eigentliche Spiel rückte allerdings schnell in den Hintergrund und so entschieden Caterina Fake und Stewart Butterfield 2004 die Fotoplattform Flickr zu gründen. Seitdem ist sie unter anderem durch den Kauf von Yahoo zu eine der größten Community-Plattformen des Internets mit mehreren Tausend Seitenaufrufen pro Minute aufgestiegen.

Auf ihrer Website schreiben sie weiter, dass sie für ihre Community zwei Hauptziele verfolgen. Erstens möchten sie „[...] Leuten dabei helfen, ihre Fotos und Videos den Menschen zu zeigen, die ihnen wichtig sind“ und zweitens „[...] neue Möglichkeiten bieten, Ihre Fotos und Videos zu organisieren.“ Beide Ziele beschreiben deutliche Anwendungskonzepte von Web 2.0 und bieten großes Potenzial EUD Aspekte zu integrieren.

Die Möglichkeiten zur Personalisierung der Weboberfläche sind bei Flickr allerdings nicht besonders stark ausgeprägt. Lediglich das Aussehen des Fotostreams lässt sich insofern anpassen, dass man eine Auswahl zwischen sechs verschiedenen vordefinierten Layouts besitzt. Diese beeinflussen die Größe und Anzahl der Bilder auf der ersten Seite, sowie die Sichtbarkeit einer Liste von Fotoalben oder Sammlungen. Erst beim Einstellen neuer Bilder fallen mehr Möglichkeiten zur Personalisierung des Websystems auf. So lassen sich bestimmte Eigenschaften mit Hilfe von voreingestellten Parametern bereits zum Zeitpunkt des Hochladens zuweisen. Diese Eigenschaften können fein granulare Zugriffsrechte oder die Auswertung und Darstellung von zusätzlichen Bildinformationen beeinflussen. Bilder können auf diese Weise bereits standardmäßig nur einer bestimmten Benutzergruppe zugänglich gemacht werden. Auch lassen sie sich über mögliche GPS-Daten innerhalb der EXIF-Informationen direkt auf einer Weltkarte positionieren.

Der Bereich Creation and Modification ist auf den ersten Blick nicht so sehr vergleichbar mit dem was Liberman et al. darunter verstehen. Im Internet ist es nur

schwer möglich einen Dienst, im Sinne der Veränderungen von Programmcode, nach seinen Wünschen zu gestalten. Statt dessen bietet Flickr allerdings eine öffentliche Programmschnittstelle (API) an, mit deren Hilfe man von außen auf Informationen von Flickr zugreifen, aber auch Daten an Flickr schicken kann.

Auf diese Weise existieren eine Menge Tools und Module von Drittanbietern, die über diese Api das augenscheinliche Verhalten von Flickr verändern. Zum Teil findet man bereits einfache Module dieser Form in einigen Gruppen und Profilen wieder. So existieren dort beispielsweise Statistiken über Gruppenmitglieder zusammen mit den zuletzt hoch geladenen Bildern.

Die Dokumentation der Api ist einfach zu verstehen und bietet zudem mehrere Anforderungs- und Antwortformate um die Integration in eine bestimmte Software einfacher zu gestalten. Anhand von REST¹ einem sehr einfachen Anforderungsformat demonstriere ich kurz die Interaktion mit der Flickr Api.

Um beispielsweise eine Liste der drei interessantesten Fotos des Tages zu erhalten wird der Api-Aufruf `flickr.interestingness.getList` benötigt. Verpackt in einer REST-Anforderung zusammen mit einem Erkennungsschlüssel sieht der Befehl dann wie folgt aus:

```
http://api.flickr.com/services/rest/?
method=flickr.interestingness.getList&per_page=3&api_key=xyz.
```

Als Antwort erscheint dann, wenn nicht anders angefordert, ein einfacher XML-Block indem die wichtigsten Informationen über die angeforderten Bilder enthalten sind:

```
<rsp stat="ok">
  <photos page="1" pages="167" perpage="3" total="500">
    <photo id="2592413708" owner="44124402241@N01"
      secret="da893e311d" server="3015" farm="4"
      title="Indecision II" ispublic="1" isfriend="0"
      isfamily="0"/>
    <photo id="2592497445" owner="47294773@N00"
      secret="da320e2fff" server="3062" farm="4"
      title="Baby Pileated Woodpeckers" ispublic="1"
      isfriend="0" isfamily="0"/>
    <photo id="2593053226" owner="7762644@N04"
      secret="fa271508de" server="3292" farm="4"
      title="Lotus Elise llls" ispublic="1"
      isfriend="0" isfamily="0"/>
  </photos>
</rsp>
```

Dieses XML-Dokument lässt sich nun relativ einfach entschlüsseln und anhand der gewonnen Informationen zum Beispiel die URL eines der Bilder rekonstruieren.

Neben dieser recht einfachen Anforderung gibt es durchaus komplexere Befehle

1 Representational State Transfer

mit denen man unter anderem seine Benutzerdaten editieren, oder nach bestimmten Bildern suchen kann. Wie aber bereits oben schon erwähnt hat man leider kaum Möglichkeiten diese Api innerhalb von Flickr für seine Zwecke zu nutzen. Entweder man ist dazu angehalten eine lokale Verwaltungssoftware zu programmieren, oder man ist auf die Dienste anderer Anbieter angewiesen.

Wordpress

Wordpress ist ein Opensource Blog, der es einem Nutzer ohne HTML Kenntnisse erlaubt komplexe Webseiten zu erstellen. Dabei orientiert es sich stark an dem Können der Nutzer. Entstanden ist es 2003 mit einer Hand voll Entwicklern und ist nun eines der größten self-hosted² Blogging Tools des Internets. Neben der für Opensource üblichen Community Version gibt es im weitesten Sinne auch eine kommerzielle Version des Blogsystems. Sie ist in manchen Punkten zum Teil systembedingt eingeschränkt lässt sich dafür aber leichter einrichten, da der Nutzer unter anderem nicht für die Konfiguration eines Servers verantwortlich ist.

Nachdem der Nutzer sich bei seinem Blog angemeldet hat, wird ihm eine Übersichtsseite präsentiert, auf der die wichtigsten Informationen zusammengefasst werden. Zu denen gehören Zugriffsstatistiken sowie wartende oder bereits freigeschaltete Kommentare. Auch befinden sich dort Links zu den wichtigsten Funktionen eines Blogs, dem Schreiben von Artikeln und Seiten.

Der Editor zum Erstellen von Artikeln unterscheidet sich kaum von dem eines einfachen lokalen Schreibprogramms. So bietet er die Möglichkeit einfache Textformatierungen ohne Kenntnisse von HTML oder CSS zu realisieren. Diese Art von Editor wird auch als WYSIWYG³-Editor bezeichnet, da der Nutzer seine Änderungen direkt sehen kann und er sich sicher sein kann, dass die spätere Anzeige sich nicht von der aktuellen unterscheiden wird.

Das Erstellen eines Artikels geschieht also wie bereits oben erwähnt ohne Kenntnisse der zugrunde liegenden Markup-Sprache. Allerdings ist es einem erfahreneren Benutzer bereits hier schon möglich auf einen HTML-Editor zurückzugreifen. In diesem Fall wird der Text zwar nicht entsprechend seiner Markierung formatiert, doch hat der Nutzer so eine bessere Kontrolle über die genutzten HTML-Elemente. Dieser Editor bietet dem Nutzer also in Abhängigkeit des Könnens und der Bedürfnisse unterschiedliche Grade von EUD-Aspekten. Die Art des Editor lässt sich zudem zu jeder Zeit umstellen. Auf diese Weise kann man im weitesten Sinne eine Art „Programming by Example“ betreiben.

Unabhängig vom Erstellen der Artikel kann der Benutzer aber auch die Darstellungsform der Website anpassen. So werden unterschiedliche, von anderen Usern bereitgestellte, so genannte Templates angeboten. Diese Templates

2 Self-hosted bedeutet der Nutzer muss selber den Server stellen und ist auch selbst dafür verantwortlich.

3 What You See Is What You Get

verändern das Aussehen der Website in einem viel stärkeren Maße als die vordefinierten Layouts von Flickr. Sie werden in einer übersichtlichen Tabelle zur Auswahl angeboten. Entscheidet man sich für ein Template wird der aktuelle Blog erst in einer Art Vorschau mit diesem Layout belegt bevor man dann seine Auswahl bestätigen kann. Liebermann würde das sicherlich als eine recht ausgereifte Form der Parameterisation bezeichnen.

Neben der Veränderung des Designs gibt es noch die Möglichkeit sogenannte Module in seinen Blog einzubinden und zu positionieren. Auch in diesem Fall bietet Wordpress bereits eine Auswahl an vordefinierten Modulen an. Diese können zum Beispiel eine automatisch generierte Liste der meistgelesenen Artikel sein. Viele Module basieren auch auf APIs anderer Webservices, wie der schon erwähnten API von Flickr.

Verlässt man nun die Oberfläche bzw. die kommerzielle Version von Wordpress, so hat man nahezu unbegrenzte Möglichkeiten den Blog nach seinen Bedürfnissen anzupassen. Die einfachste Anpassung besteht sicherlich darin ein eigenes Template zu erzeugen. Dafür ist im Prinzip nur die Kenntnis von HTML und CSS nötig. Zusammen mit ausführlichen Tutorials und Beispielen anderer Benutzer kann man auf diese Weise schnell ein individuelles Design realisieren. Dieses kann man dann ebenfalls wieder auf eines der zahlreichen Template-Plattformen mit anderen Benutzern teilen.

Aufwendiger wird es, wenn man eigene Module schreiben möchte, denn dafür ist neben den schon eben erwähnten Kenntnis zusätzlich grundlegendes Wissen der Programmiersprache PHP von Nöten. Aber auch dafür findet man, wie für ein Community-Projekt üblich, genügend Tutorials mit dessen Hilfe man seine Fertigkeiten verbessern kann. Auch bietet die Community ausreichend Hilfestellung Mechanismen zu erlernen um die erwähnten APIs in seinem Modul zu verwenden.

Die Integration und Entwicklung dieser Module kann also als eine Mischung aus Incremental programming und Model-based development verstanden werden.

Pipes

Pipes ist ein Online-Service, der es dem Nutzer erlaubt ohne große Programmierkenntnisse webbasierte Datenverarbeitung zu betreiben. Veröffentlicht wurde es Mitte Februar 2007 von Yahoo und erregte daraufhin große Euphorie unter allen Web 2.0 Anhängern.

Der Hauptanwendungsbereich dieses Services liegt darin Internet-Feeds wie RSS⁴ mit Hilfe von kleinen vorgefertigten Programmelementen zu bearbeiten. Der Name für diesen Service kommt aus der Unix-Systemumgebung. In Unix werden, ähnlich wie jetzt in Pipes, die Ein- und Ausgabeströme einzelner Programme mit

⁴ RDF Site Summary ist ein kurzes XML-Dokument welches Meta-Informationen über eine Website enthält.

Hilfe einer Pipe verbunden. Dadurch werden einem vielfältige Möglichkeiten zur Verarbeitung dieser Datenströme geboten. Das Besondere an diesem Webservice ist die intuitive Bedienung und die daraus resultierende niedrige Einstiegshürde.

Pipes bedient sich einer visuellen Programmiersprache dessen Look-And-Feel⁵ sich kaum von der einer lokalen Entwicklungsumgebung unterscheiden lässt. So bietet sie bereits unabhängig vom eigentlichen Programm die Möglichkeit einzelne Editorfelder wie zum Beispiel den Debugger temporär auszublenden.

Ein Programm, von Yahoo auch als Pipe bezeichnet, entsteht dadurch, dass man aus einem Katalog auf der linken Seite seiner Arbeitsumgebung das passende Modul auswählt und per Drag-And-Drop auf die Arbeitsfläche zieht. Dieser Katalog beinhaltet unterschiedliche Module um Daten zu importieren. Sie reichen von einfachen CSV-Dateien⁶ über RSS-Feeds und HTML-Seiten bis hin zu Ergebnissen von Flickr- oder Websuchanfragen. Die benötigten Parameter kann man entweder direkt im Modul oder aber auch mit Hilfe weiterer vorgefertigter Eingabeformularmodule angeben.

Danach entscheidet man was mit den gewonnen Daten geschehen soll. Pipes bietet auch dafür mehrere zum Teil sehr hilfreiche Module an. So befindet sich beispielsweise in der Kategorie „String“ ein Übersetzungsmodul. Es bietet dem Nutzer die Option eingehende Texte, beispielsweise die eines RSS-Feeds, in eine andere Sprache zu übersetzen. Wenn man nun also mindestens zwei Module auf seiner Arbeitsfläche liegen hat, kann man diese dann durch ein röhrenartiges Verbindungsstück miteinander verknüpfen und erhält dadurch bereits ein erstes lauffähiges Programm.

Die Ausgabe der Pipe kann die ganze Zeit problemlos im Debugger mitverfolgt werden, indem man die zu untersuchenden Module markiert. So erkennt man schnell, welche Parameter und Module noch falsch konfiguriert sind und bemerkt seine Fehler nicht erst beim Ausführen des Programms. Um Fehler aber von vornherein vorzubeugen und somit dem Nutzer nicht unnötige Frustrationen zu bereiten, werden zu allen Modulen einfach zu verstehende Kurzerklärungen bereitgestellt. Falls diese nicht ausreichen sollten, kann dann auf eine ausführlichere Beschreibung ausgewichen werden. Neben der Dokumentation besteht aber auch zu jedem Modul ein kleines Beispielprogramm anhand dessen man die Funktionsweise nachvollziehen kann.

Nachdem man also ein fertiges Programm erstellt hat, bietet Pipes einem zugleich auch mehrere Möglichkeiten dieses Programm weiter zu verwenden. So stehen gleich mehrere Exportmodule zu Verfügung, die die Einbindung in Socialnetworks wie MySpace erlaubt, aber auch Codeschnipsel, die eine Verwendung in privaten Homepages oder Blogs ermöglicht. Neben diesen Exportfunktionen wird auch eine Veröffentlichungsfunktion angeboten, die es dem Benutzer erlaubt sein Programm mit anderen Benutzern zu teilen und es auf

5 Look-And-Feel bezeichnet das Aussehen und Verhalten einer Software im Vergleich zu anderen Softwareangeboten.

6 Comma-Separated Values

diese Weise in größeren Projekten wiederzuverwenden.

Fazit

Wie man anhand dieser kleinen Auswahl an Beispielen erkennen kann bietet das World Wide Web viele Facetten von EUD. Nahezu jede Web 2.0 Anwendung ermöglicht einem wenigstens die Parametrisierung insofern, dass man das Erscheinungsbild von Teilen oder auch kompletten Webseiten verändern kann. Darüber hinaus kann das Angebot stark variieren. Häufig wird, wie im Beispiel von Flickr, aber auch Google Maps oder Facebook, eine API zur Verfügung gestellt, mit deren Hilfe man Teile des Services auch auf andere Plattformen portieren kann. Höhepunkt des eindeutig Web basierten EUD sind meiner Meinung nach Entwicklungsumgebungen wie Pipes. Sie nutzen die unter Web 2.0 verstandenen Prinzipien voll aus und bieten dem Nutzer daher alle Möglichkeiten seine Wünsche zu realisieren.

Sucht man nun allerdings die aus Liebermanns Sicht interessanteren Aspekte von EUD, wie beispielsweise das Incremental Programming wird es schwierig seiner Definition eines unbedarften Endbenutzers gerecht zu bleiben. Betrachtet man die Möglichkeiten von Wordpress so können diese erst dann umfassend ausgenutzt werden, wenn der End-User im Besitz eines vollwertigen Webservers und mit der Bedienung dessen vertraut ist. Die Frage die daraus allerdings resultiert ist, ob dieser Nutzer dann immer noch als unbedarft angesehen werden kann. Vielleicht sollte die Definition weniger technisch orientiert sein sondern stärker auf soziologische Aspekte eingehen.

Wenn ein Endbenutzer eine Software nach seinen Vorstellungen und aus seiner Sicht besonders einfach verändern kann, kann es als EUD aufgefasst werden. Unter dieser Prämisse lassen sich dann meiner Meinung nach auch die nötigen Veränderungen im Sourcecode von Wordpress oder anderer OpenSource Anwendungen unter dem Begriff EUD vereinen.

Referenzen

Lieberman, H., Paternò, F. and Wulf, V. (2005): End User Development, Springer.

O'Reilly (2005): What is Web 2.0,
<http://www.oreilly.de/artikel/web20.html> bzw.
http://www.distinguish.de/?page_id=36

Flickr:
<http://flickr.com/about/>
<http://www.flickr.com/services/api/>
http://www.usatoday.com/tech/products/2006-02-27-flickr_x.htm

Wordpress:
<http://wordpress.org/about/>
<http://themes.wordpress-deutschland.org/>

Pipes:
<http://pipes.yahoo.com/pipes/>