

# **Historische Innovationen der Informatik - Jef Raskin**

Stefan Uhlmann

28. Mai 2006

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Biographie . . . . .	3
<b>2</b>	<b>Macintosh</b>	<b>4</b>
2.1	Einleitung . . . . .	4
2.2	Die Geschichte des Macintosh . . . . .	4
2.2.1	Ein Rechner für die PITS . . . . .	4
2.2.2	Das Desktop Prinzip . . . . .	5
2.2.3	Der Mac im Wandel . . . . .	5
<b>3</b>	<b>The Humane Interface</b>	<b>6</b>
3.1	Inhalt . . . . .	6
3.2	Kritikpunkte . . . . .	6
3.3	Lösungsvorschläge . . . . .	7
3.3.1	Globales Dokument . . . . .	7
3.3.2	Leap Technologie . . . . .	7
3.3.3	Verbinden vieler Anwendungen . . . . .	7
3.3.4	Verstecktes Betriebssystem . . . . .	8
3.4	Diskussion der Umsetzung und Umsetzbarkeit der Konzepte . . . . .	8
3.5	The Raskin Center for Humane Interfaces . . . . .	9
<b>4</b>	<b>CanonCat</b>	<b>10</b>
4.1	SwyftWare, der Vorgänger . . . . .	10
4.2	Der Rechner Canon Cat . . . . .	10
4.2.1	Die technischen Details . . . . .	10
4.2.2	Die Software . . . . .	11
4.3	Gründe des Scheiterns . . . . .	11
<b>5</b>	<b>Zusammenfassung</b>	<b>12</b>

# 1 Einleitung

Diese Ausarbeitung zu dem Proseminar Historische Innovationen der Informatik befasst sich mit Jef Raskin. Raskin war ein Vorreiter der modernen Benutzerschnittstellen in den 80er Jahren, und hat maßgeblichen Anteil daran, dass heute die Bedürfnisse der Benutzer im Umgang mit Maschinen viel ernsthafter und gründlicher untersucht und beachtet werden. Durch seine Ideen hatte er großen Anteil am Erfolg des Apple Macintosh. Mit seinem Buch *The Humane Interface* stellt er eine im Vergleich zum heute Üblichen andere Möglichkeit zur Interfacegestaltung vor, wie sie sich nach seiner Meinung mittelfristig am Markt durchsetzen wird und die auf Konzepten beruht, die nachfolgend beschrieben werden.

## 1.1 Biographie

Jef Raskin wurde am 9. März 1943 in New York mit dem bürgerlichen Vornamen Jeff geboren, von welchem er aber später ein „f“ streichen ließ, da er es als redundant ansah. Noch in New York erreichte er 1964 und 65, Bachelorabschlüsse in Mathematik und Philosophie. Kurze Zeit später erhielt er den Mastertitel in Informatik an der Pennsylvania State University, wo er eine Abschlussarbeit über das sogenannte Quickdraw Graphic System schrieb, welches später bei Apple mit als Grundlage für die grafische Oberfläche des Macintosh dienen wird. Die nächsten vier Jahre von 1970 bis 74 verbringt er an der University of California, San Diego, als Assistant Professor, bevor er im Frühjahr 1978 als Mitarbeiter Nummer 31 bei der Firma Apple eingestellt wird. Er übernimmt hier schon bald das Macintosh-Projekt und stellt hierfür das Grundkonzept auf die Beine, bevor er schon drei Jahre später die Firma noch vor der Veröffentlichung des Rechners, nach Differenzen mit Steve Jobs, einem Gründungsmitglied, wieder verläßt. Er gründet nun seine eigene Firma, Information Appliances Inc., bei der er eine Software entwickelt, die für den Apple II auf den Markt kommt, und später, lizenziert von der Firma Canon als Komplettrechner veröffentlicht wird. Als dieses als Canon Cat getaufte Gerät aber im Jahre 1987 floppt, schließt er die Firma wieder und beginnt fortan als freier Berater und Autor seinen Lebensunterhalt zu verdienen. Im Jahr 2000 veröffentlichte er sein Buch „*The Humane Interface*“, in dem er ein zu den heute üblichen grafischen Oberflächen konträres Betriebssystem vorstellt und erläutert. Das Buch wurde später in viele Sprachen übersetzt und dient an manchen Universitäten als Lehrbuch. Das vorgestellte Betriebssystem wird an dem von ihm 2001 gegründeten Raskin Center for Humane Interfaces (RCHI) seit 2005 unter dem Namen Archy entwickelt. Raskin erliegt am 26. Februar 2005 einem Krebsleiden.

## 2 Macintosh

### 2.1 Einleitung

Jef Raskin wurde 1978 bei Apple eingestellt um die Dokumentationen für den Apple II zu schreiben. Im Frühjahr darauf, wurde ihm von Mike Markkula, dem damaligen Präsidenten, angeboten ein Projekt namens „Annie“, ein Spielecomputer für 500 \$, zu leiten. Raskin hatte aber schon seit Langem die Vorstellung von einem Rechner wie er ihn gerne entwickeln würde, und konnte Markkula letzten Endes auch davon überzeugen, das neue Projekt nach seinen Ideen gestalten zu dürfen. Raskin gab dem Projekt nun noch den Namen seiner Lieblingsapfelsorte McIntosh, wobei lediglich die Schreibweise aus lizenzrechtlichen Gründen verändert wurde, so dass der Name Macintosh im gleichen Spätsommer das erste Mal offiziell in Apple-Unterlagen auftauchte.

### 2.2 Die Geschichte des Macintosh

Zu der Zeit als begonnen wurde den Macintosh zu entwickeln, war der breite Markt der Personal Computer, sehr unkomfortabel in der Bedienung. Nutzer mussten sich mit Kommandozeilen und vermeintlich hässlichen, winzigen Zeichensätzen zufrieden geben. Als einzige Ausnahme ist der Xerox Alto zu nennen, welcher im Palo Alto Research Center der Firma entwickelt wurde, in dem Raskin in den frühen 70er Jahren häufig ein und aus ging. Er musste sich deshalb auch oft gegen Vorwürfe erwehren, wonach er den Macintosh nur von den Xerox Forschern abgekupfert hätte, welche er aber konsequent abtritt und von einer Parallelentwicklung mit gegenseitiger Inspiration sprach.

#### 2.2.1 Ein Rechner für die PITS

„PITS“ ist eine Abkürzung, die bei Apple seit Jef Raskin zu einem geflügelten Wort avancierte, sie steht für „Person in the street“ und vereinheitlicht die Zielgruppe des Macintosh, der gewöhnliche Arbeiter, der unerfahren im Umgang mit moderner Technik ist. Für diese Zielgruppe sollte der Rechner so einfach gestaltet und zu bedienen sein, dass es ohne Studium des Handbuches möglich war ihn zu bedienen. Man musste also weg von der unübersichtlichen Kommandozeileneingabe die für Computerneulinge eine Hürde darstellte. Zur Lösung dieses Problems entwickelte das Macintosh-Team, QuickDraw auf Grundlage von Raskins Abschlussarbeit von 1967. Hier traten neben Raskin vor allem Bill Atkinson, ein ehemaliger Student Raskins, und Andy Hertzfeld in Erscheinung. Dieses 2D-Anzeigesystem ermöglichte es durch Bitmapping Grafiken an jeder Stelle des Bildschirms frei und in beliebiger Größe darzustellen. Der Weg war also geebnet um die nächste Vision von Jef Raskin einzuarbeiten, so war er es satt, dass jeder Rechner nur die Zeichen darstellen konnte, die der Hardware Character Generator in der Lage war zu produzieren. Er wollte verschiedene Schriftzeichen für verschiedene Anwendungen, so sollte ein Musiker doch auf seinem Rechner ebenso Noten darstellen können, wie ein Physiker das griechische Alphabet. Diese Maßnahme ermöglichte natürlich auch viel leichter die Internationalisierung des Rechners, schließlich musste die Hardware nicht mehr verändert werden, sondern lediglich eine andere Software aufgespielt werden um den Rechner im Ausland mit der passenden Sprache zu verkaufen, wobei dies anfangs kaum bis gar nicht genutzt wurde. Um der „PITS“ nun aber auch die Möglichkeit zu geben, diesen Rechner zu erwerben musste er natürlich auch besonders günstig im Vergleich zu anderen sein. So wurde ein Preisziel von 500 \$ ausgegeben, welches sich zwar als nicht haltbar erwies, aber mitverantwortlich für eine Eigenheit des Macintosh-Projektes war. So war es damals üblich seine Rechner mit der modernsten Hardware auszustatten und damit um jedes bisschen Leistung zu kämpfen. Unter Raskin wurde der Macintosh jedoch nach ganz anderen Gesichtspunkten entwickelt, so standen die Bedürfnisse, welche die Nutzer im Umgang mit dem Gerät hatten, im Vordergrund gegenüber schnelleren Ladezeiten. Daher wurde der Macintosh in

der frühen Konzeptphase mit einer schon veralteten CPU und einem auch für diese Zeit extrem eng bemessenen Speicher von 64KB geplant. Der Monitor sollte winzige 5 Zoll groß sein. Für Raskin war es zum Beispiel eher von Bedeutung, dass niemand jemals genötigt sein sollte das Gehäuse des Macintosh zu öffnen. Es wurden also Schnittstellen für Geräte, oder auch Speichererweiterungen nach außen gelegt. Es sollte sogar völlig auf externe Kabel verzichtet werden, sprich alle Bauteile sollten in einem Gehäuse verbaut sein, was dann in der fertigen Fassung aber zumindest für Tastatur und Maus nicht mehr galt.

### **2.2.2 Das Desktop Prinzip**

Die wohl bedeutendste Neuerung des Macintosh, abgesehen vom schon erwähnten Xerox Alto, war wohl das Desktop Prinzip, wie es sich bis heute maßgeblich im Heimcomputerbereich durchsetzen konnte. Es wurden die heute üblichen Icons eingeführt, die auf einem Hintergrund frei verschiebbar sind. Dieser soll den heimischen Schreibtisch repräsentieren. Öffnete man einen Ordner, so wurde ein Fenster auf dem Desktop geöffnet welches die einzelnen Dokumente enthielt. Diese konnten dann noch einzeln geöffnet werden, wie wenn man eine Akte auf seinem Schreibtisch aufschlägt und die enthaltenen Schriftstücke und Bilder durchgeht, wobei aber kein so verzweigtes Verzeichnissystem wie heute in Windows zur Anwendung kam, alle Daten konnten nur auf der selben Ebene abgespeichert werden. Wollte man ein Dokument oder eine Datei löschen, so wurde sie in einen Papierkorb verschoben wo man sie jedoch bis zum endgültigen Leeren jederzeit wieder herausholen konnte. Diese ganzen Methoden und Eigenschaften kommen einem sehr bekannt vor wenn man sich das Prinzip anschaut, welches hinter Windows steht. Das heute ebenso übliche Click&Drag (Heute: Drag&Drop genannt) diente dazu die Ordner frei auf dem Desktop verschieben zu können, oder eben auch, sie in den Papierkorb zu verfrachten. Desweiteren wurden die heute überall bekannten Pull-Down Menüs eingeführt, die von einer Leiste am oberen Bildschirmrand heruntergeklappt werden konnten. Auch die heute dauernd aufpoppenden Dialogboxen sind auf den Macintosh zurückzuführen, welche dort Bill Atkinson bei einer Besprechung mit Raskin erfunden hat. Dieser gesamte symbolische Umgang mit den Daten, verbunden mit dem im Vergleich zum Xerox Rechner (ca.7000 \$) günstigen Preis von am Ende 2500 \$, half dem Mac wohl zu seinem durchschlagenden Erfolg.

### **2.2.3 Der Mac im Wandel**

Vieles veränderte sich während der Entwicklungszeit des Macintosh, so wurden zum Beispiel der Drucker und auch der Monitor, welcher dann durch einen Fernsehanschluss ersetzt werden sollte, zeitweise aus Kostengründen gestrichen, aber dann doch wieder hinzugefügt. Das zweite Eingabegerät neben der Tastatur war zu Beginn eine Art Lightpen, um dann durch einen Joystick ersetzt zu werden und letzten Endes als Maus die Marktreife zu erlangen. Auch der Inhalt des begrenzten ROM-Speichers wechselte oft. Die größten Veränderungen in dem Projekt wurden aber durchgeführt als Steve Jobs eine leitende Funktion in dem Projekt einnahm. Das Prinzip der Entwicklung nach Gesichtspunkten der menschlichen Bedürfnisse wurde wieder in ein technologisches Wettrüsten mit der Konkurrenz umgewandelt. Die damit verbundene Preissteigerung von geschätzten 1500 \$ auf 2500 \$ ist scheinbar allein Jobs zu verdanken, der vom parallel laufenden LISA-Projekt unfreiwillig ins Macintosh-Lager überwechselte. Nach einiger Zeit ist es nun so weit und Raskin fühlt sich in seinem eigenen Projekt nicht mehr verstanden und konnte seine Konzepte nicht mehr ausreichend durchsetzen. Er kündigt im Frühjahr 1982 nachdem Jobs ihn soweit aus dem Projekt verdrängen wollte, dass er nur noch die Dokumentationen schreiben sollte. So verließ Jef Raskin das Projekt schon zwei Jahre vor der Veröffentlichung des ersten Macintosh. Diese erfolgte übrigens äußerst fulminant in Szene gesetzt durch einen legendären Werbespot der nur ein einziges Mal während des Superbowl gezeigt wurde und in dem der große Bruder IBM symbolisch zerschlagen wurde.

## 3 The Humane Interface

Ich möchte an dieser Stelle dem historischen Verlauf etwas vorgehen, und schon hier das Buch „The Humane Interface“ besprechen, da die Konzepte die hier vorgestellt wurden schon teilweise als Grundlage des 15 Jahre früher entwickelten CanonCat Rechners und komplett als Fundament des neuartigen System Archy zu sehen sind. Das Buch erschien im Jahr 2000 beim Addison-Wesley Verlag und liegt mittlerweile in vielen verschiedenen Sprachen vor. An manchen Universitäten vor allem in den Vereinigten Staaten dient es sogar schon als Lehrbuch.

### 3.1 Inhalt

In seinem Buch befasst sich Raskin umfassend mit Schnittstellen zwischen Mensch und Maschine. Dabei beschränkt er sich nicht nur auf den Bereich der Personal Computer, auch ein Fahrkartenautomat einer Bahngesellschaft wird unter anderem genau unter die Lupe genommen und kritisiert. Raskin versucht seine Kritik stets durch psychologische und wissenschaftliche Hintergründe zu fundieren und führt Beispiele an, wie die Aufmerksamkeitszentren von Benutzern gar nicht beachtet oder auch falsch angesprochen werden. Er stellt auch verschiedene Methoden zur Effizienzmessung vor, bei denen sein System besser als die, dem Vergleich zu Grunde liegenden grafischen Oberflächen, abschneidet. Er stellt aber auch Lösungsvorschläge vor, die nach seinem Befinden den Umgang mit Computern deutlich vereinfachen könnten. Als umfassender Lösungsvorschlag ist wohl das Softwaresystem „The Humane Environment“ zu sehen, welchem er einen Großteil seines Buches widmet und welches die Benutzeroberflächen wie wir sie heute kennen ablösen soll.

### 3.2 Kritikpunkte

Als grundlegendes Problem sieht Raskin die heutige Vorgehensweise in der Softwareentwicklung, bei der die Entwicklung der Benutzeroberfläche häufig ein gezwungenes Übel am Ende der restlichen Programmierphase darstellt. Dies führt für ihn zwangsläufig zu Problemen, da die Software so sehr auf Effizienz ausgelegt ist und von Grund auf maschinenfreundlich programmiert wurde, dass ein Aufsetzen einer auf den Menschen gemünzten Oberfläche oft gar nicht mehr möglich ist. In den heutigen Zeiten, in denen die Rechenleistung so hoch ist, sollte man eher von der anderen Seite her entwickeln und die Bedürfnisse der Nutzer weiter in den Vordergrund rücken. Ein weiterer Kritikpunkt ist die falsche Automatisierung der Benutzer wie sie am Beispiel des Löschens einer Datei gut zu veranschaulichen ist. Dadurch, dass man bei jedem Löschvorgang stets die gleiche Frage „Möchten sie die Datei X wirklich löschen ? J/N“ zu beantworten hat und diese auch immer mit „Ja“ quittiert, speichert man als Nutzer den Löschvorgang mit der Kombination Löschen + Ja im Gedächtnis ab und beachtet den Sinn der Nachfrage gar nicht mehr weiter. Die Sicherheitsfrage verliert also ihre Bedeutung und damit ihre Daseinsberechtigung, schließlich wird man gezwungen einen weiteren Arbeitsschritt zu tun, der aber keine Relevanz hat. Ein weiterer Punkt ist die Inkonsistenz in der Bedienung die für ihn bei den gängigen Systemen ein Problem darstellt. So erreicht man in einer Applikation mit einer bestimmten Aktion oder auch Tastenfolge etwas völlig anderes als in einer anderen vielleicht sogar Artverwandten. Dadurch wird der Nutzer verwirrt und gezwungen sich für jede Anwendung aufs Neue zu merken, wie er zu einem gewünschten Ergebnis kommt. Auch die Tatsache, dass es so viele verschiedene Wege gibt innerhalb einer Applikation zu einem Ergebnis zu kommen stört ihn, er hält es für überflüssig, dass es in einer Textverarbeitung drei Möglichkeiten gibt eine Textstelle in kursiver Schrift darzustellen. Dies potenziert sich natürlich noch durch die unterschiedlichen Wege überhaupt einen Text zu markieren. Auch die Navigation durch die eigenen Daten ist für ihn suboptimal gelöst, vertritt er doch die These, dass Daten stets nach ihren Inhalten und nicht nach äußeren Gesichtspunkten wie Dateiname, -größe oder -endung in vielfach verschachtelten Ordnerhierarchien gesucht werden sollten.

Viel zu oft gingen Daten durch das Vergessen solcher äußerer Merkmale durch den Nutzer verloren.

### **3.3 Lösungsvorschläge**

Nachfolgend möchte ich die Lösungsvorschläge die Raskin den vorhergehend schon geschilderten Kritikpunkten entgegengesetzt im Einzelnen etwas näher betrachten. Sie bleiben hierbei jedoch nur als Gesamtkonzept zu betrachten, da manche der Einzelaspekte ohne die restlichen Techniken nicht sinnvoll zu nutzen wären.

#### **3.3.1 Globales Dokument**

Mit diesem Ansatz möchte Raskin das übliche Dateisystem abschaffen, statt dessen wird ein von ihm so genanntes globales Dokument benutzt, in welches alle einzelnen Dokumente implementiert sind. Dieses Prinzip hat er bei der Canon Cat das erste Mal verwendet. Man kann sich dieses Dokument wie eine große Textdatei vorstellen, in der die einzelnen Dokumente durch spezielle Zeichen wie die Tilde getrennt werden. Man schreibt also alle seine Dokumente untereinander und trennt nicht mehr in einzelne Dateien. Datenverlust durch das Vergessen eines eilig gewählten Dateinamens soll so verhindert werden. Die Datensicherung soll ebenfalls durch eine theoretisch endlose „UNDO-Funktion“ verbessert werden mit der sich wirklich jeder Schritt im Einzelnen rückgängig machen und auch wiederherstellen lässt. In diesem Dokument wird sogar speichern überflüssig, es wird automatisch vom System übernommen. Dies geht soweit, dass man niemals Daten verliert, selbst wenn der Computer durch einen Stromausfall unterbrochen wird, findet man beim nächsten Start das globale Dokument in der Form vor, wie man es absichtlich oder unabsichtlich verlassen hat. Selbst der Cursor steht noch an exakt der Stelle wie vor dem Absturz.

#### **3.3.2 Leap Technologie**

In dem schon geschilderten globalen Dokument, navigiert man mittels der von Raskin erfundenen Leap-Technologie. Sie ähnelt vom Prinzip her einer Volltextsuche in Echtzeit wie man sie Heute beim Durchsuchen von Hilfebibliotheken findet. Durch die Aneinanderkettung der vielen Dokumente wird man früher oder später gezwungen sein sich schnell über große Strecken in dem globalen Dokument zu bewegen. Da dies mit den üblichen Methoden wie Scrollbar oder Cursortasten viel zu langsam beziehungsweise ungenau funktionieren würde, ließ sich Raskin seine eigene Technik einfallen. Auf der Tastatur gibt es zwei Tasten die für die Navigation im Dokument zuständig sind, eine für die Richtung Vorwärts, die andere für Rückwärts. Wird nun eine dieser Tasten gedrückt gehalten und dabei ein Zeichen oder auch eine Zeichenkette eingegeben, so springt der Cursor in die entsprechende Richtung, an die Stelle wo die Zeichenfolge das nächste Mal auftaucht. Möchte man aber zu einer gleichen Stelle nur noch weiter am Ende (bzw. am Anfang ) eines Dokumentes muss man nur noch die Kommandotaste so oft drücken bis die gesuchte Position erreicht ist. Da sich am Anfang jedes Abschnittes eine Tilde befindet, lässt sich so also leicht zwischen den Dokumenten navigieren. Auch die Suche nach Inhalten wurde so realisiert, da man durch das Suchen eines Begriffes sofort an die Stelle im Dokument springt an der er verwendet wird.

#### **3.3.3 Verbinden vieler Anwendungen**

Da er ja als großen Schwachpunkt der heutigen Systeme die inkonsistente Bedienung sieht, versucht er in seinem Konzept schon die Ursache für dieses Problem zu unterbinden. Für ihn entsteht Inkonsistenz ,wie oben schon erklärt, durch die vielen verschiedenen Anwendungen, in denen man häufig durch gleiche Aktionsfolgen zu völlig unterschiedlichen Ergebnissen kommt. Desweiteren wird der Nutzer durch die in jedem Programm, wenn auch

nur geringfügig anders sortierten und angeordneten Menüs irritiert. Daher möchte Jef Raskin viele wenn möglich sogar alle verschiedenen Anwendungen in eine einzige große Applikation verpacken. Dies ist grundlegend vergleichbar mit dem funktionsbasierten Editor des Macintosh oder besser mit heutigen Programmen wie zum Beispiel „Emacs“. Denn wie es für „Emacs“ verschiedene Plug Ins zum Installieren gibt um die vorhandenen Funktionen zu erweitern, so sollen Drittanbieter für Raskins System auch lediglich Funktionen zur Verfügung stellen, die sich der Benutzer dann für sein zu den schon vorhandenen ergänzend installieren kann. Es sollen also überhaupt keine eigenständigen Programme mit eigener Benutzeroberfläche mehr dafür entwickelt werden. Grundsätzlich soll diese große Anwendung aufgebaut sein wie ein Editor. In diesem ist das schon diskutierte globale Dokument geöffnet und man kann verschiedene Einträge durch die sogenannte Kommandotaste unterschiedlich interpretieren. Möchte man zum Beispiel eine Berechnung durchführen, so markiert man diese einfach und schreibt bei gedrückter Kommandotaste „Calc“ für das englische „calculate“ (berechnen). Die Rechnung wird nun als solche interpretiert und mit dem Ergebnis ersetzt. Nach diesem Schema soll es für alle Bedürfnisse spezielle Funktionen geben, welche mit Hilfe der Funktions- oder Kommandotaste aufgerufen werden.

### **3.3.4 Verstecktes Betriebssystem**

Als letzten Punkt möchte ich den Verzicht auf ein Betriebssystem im herkömmlichen Sinne aufführen. Dies soll nicht bedeuten, dass auf ein Betriebssystem komplett verzichtet würde, es wird lediglich vor dem Benutzer „versteckt“. Man soll sich nicht um Treiberdetails kümmern müssen, oder dass immer die aktuellen Protokolle zur Netzwerkkommunikation installiert sind. Der Rechner soll dies von alleine regeln, falls Handlungsbedarf vorliegt. Der Nutzer soll nichts mit der Konfiguration und den technischen Details des Rechners zu tun bekommen, was wieder seinem Prinzip der einfachen Benutzbarkeit entspricht. Der Rechner soll dann beim Anschalten, möglichst ohne Bootvorgang, sofort im Multifunktionseditor starten und dort auch an der Stelle im globalen Dokument die Arbeit wieder aufnehmen, wo man sie zuvor verlassen hatte. Jef Raskin hat zu dieser Diskussion einmal gesagt, beziehungsweise in seinem Artikel „Recollections of the Macintosh Project“ geschrieben: „Someday operating systems will disappear from view, just as the details of the processor (thankfully) have.“. Er sieht also das Betriebssystem, welches mit der Hardware kommuniziert als grundsätzlich notwendig an, vergleicht es aber mit den Prozessoren der 70er und 80er Jahren, wo die Software noch speziell für die einzelnen Architekturen entwickelt wurden, von welchen heute ja abstrahiert wird.

## **3.4 Diskussion der Umsetzung und Umsetzbarkeit der Konzepte**

Wir haben hier nun einige teils mehr, teils weniger zum Heutigen konträre Konzepte gesehen. Im Nachfolgenden möchte ich diese noch auf ihre Umsetzbarkeit, beziehungsweise ihre schon vollbrachte Umsetzung hin untersuchen. So ist die von Raskin geforderte Suche nach Inhalten und nicht nur nach äußeren Dateimerkmale bei Apple, schon längere Zeit in die Realität umgesetzt und beim großen Konkurrenten Microsoft ist sie für die nächste Windows Generation ebenfalls angekündigt. Den Verzicht auf ein Dateisystem hat man auch bei Apple zumindest in einer abgeschwächten Form schon versucht, schließlich waren beim Macintosh alle Dateien nur auf der gleichen Ebene speicherbar, und Dateiendungen gab es ebenfalls nicht. Dass dieses Prinzip aber wieder verworfen wurde spricht nicht für dessen Umsetzbarkeit. Im Allgemeinen lässt sich zu dem Konzept des globalen Dokumentes sagen, dass es zu den Zeiten der Canon Cat mit ihren begrenzten Speichermöglichkeiten ja noch praktikabel gewesen sein mag, in der heutigen Zeit würde das Dokument aber riesige Ausmaße annehmen. So, dass es vermutlich sehr unübersichtlich würde. Außerdem hat sich die Nutzung der modernen PCs in eine völlig andere Richtung, nämlich die multimediale, weiterentwickelt und damit ein für den Privatanwender völlig neues Hauptanwendungsgebiet bekommen. Diese Sparte mit Bildern, Videos und Musikdaten und deren Bearbeitung



lässt sich für mich nicht mit einem globalen Dokument in Einklang bringen. Hingegen halte ich die Grundidee von einer multifunktionalen Anwendung für gut, wobei es wohl vermessen wäre zu behaupten wirklich alles in einer Umgebung unterbringen zu können. Man sieht ja auch am Beispiel des Programmes „Emacs“, dass das Prinzip eines umfassenden Editors, für den lediglich Plug Ins von anderen Anbietern entwickelt werden, durchaus erfolgreich umzusetzen ist. Ich sehe hier wieder Probleme in den Multimediaanwendungen oder dem schlichten Webbrowsing, welche so wenig mit dem Erstellen einer Datenbank zu tun haben und damit eine Verbindung wirklich aller Anwendungen utopisch erscheinen lassen. Aber eine reine Zusammenlegung aller Textverarbeitungs- und Tabellenkalkulationsprogramme beziehungsweise E-Mail und Kommunikationsprogramme sprich aller Büroanwendungen zu einer großen umfassenden Anwendung die wirklich leicht und konsistent zu bedienen ist würde ich für nicht abwegig halten. Verbunden mit der Leap Technologie, welche nach relativ kurzer Einarbeitungszeit wirklich sehr einfach und schnell zu benutzen ist, könnte ich mir gut vorstellen dass eine solche Anwendung bei professionellen, wie auch privaten Anwendern Anklang finden könnte. Durch das tastaturgesteuerte Navigieren im Dokument kann man wirklich sehr schnell zu den gewünschten Stellen springen. So habe ich beim Schreiben dieser Ausarbeitung rein interessehalber kurzzeitig auf eine sehr frühe Version des auf Raskins Grundlagen basierenden Archy Systems zurückgegriffen. Nachdem ich dann wieder auf einen gewöhnlichen Editor umgestiegen bin, habe ich die Sprungtasten noch häufig, natürlich vergebens, versucht anzuwenden und die Technik vermisst. Wenn ich zum Beispiel aus Versehen ein Substantiv klein schrieb, benötigte ich in Archy nur wenige Augenblicke um zu dem Beginn des Wortes zu springen, den Buchstaben zu ersetzen und wieder zurückzuspringen und weiterzuschreiben, während ich in dem gewöhnlichen Editor erst alle Zeichen wieder löschen oder mich mit dem Cursor einzeln durchklicken musste was wesentlich länger dauert. Für mich hat diese Technik etwas sehr Effizientes und auch Komfortables im Vergleich zu allen Texteditoren, die ich bisher kennengelernt habe. Als letztes bleibt noch das sozusagen versteckte Betriebssystem, von dem ich die Grundidee auch in gewissen Bereichen für angebracht halte. Gerade in Büros in denen die Nutzer ja wirklich nichts mit der Konfiguration ihres Arbeitsgerätes zu tun haben sollten, wäre so eine Umsetzung durchaus denkbar. Man kann ja auch feststellen dass die heutigen Betriebssysteme wie Windows immer mehr Tätigkeiten selbstständig einleiten und ausführen wollen und auch tun, so geht hier die Entwicklung definitiv in die von Raskin angedachte Richtung, wenn auch nicht in allen Teilbereichen. Abschließend kann man denke ich sagen, dass es schwer vorstellbar ist, seine Vision genau umzusetzen, jedoch einzelne Aspekte schon umgesetzt sind, noch umgesetzt werden oder zumindest vermutlich noch umgesetzt werden, da sie einfach als sinnvoll zu erachten sind.

### **3.5 The Raskin Center for Humane Interfaces**

Raskin hat noch selbst dafür gesorgt, dass sein Konzept keine Idee bleibt, sondern auch wirklich in die Tat umgesetzt wird. Er gründete einige Zeit nachdem er seine Ideen in seinem Buch vorgestellt hatte, das Raskin Center for Humane Interfaces an dem sein System unter dem vorläufigen Namen Archy (Anspielung auf die Initialen des Forschungsinstitutes RCHI) entwickelt wird. Die Arbeiten an diesem Projekt wurden aber erst im Januar 2005, also einen Monat vor Raskins Ableben, durch eine 2 Millionen Dollar umfassende Spende von IBM ermöglicht und aufgenommen. Ausgerechnet die Firma, die noch 20 Jahre früher der große Konkurrent Raskins war, als er noch für Apple am Macintosh arbeitete. Nach dem Entwickeln einer grundsätzlich lauffähigen Version als normale Applikation für die gängigen Systeme wird das Projekt mittlerweile auf Open Source Basis weitergeführt. Die Entwickler müssen sich kommerziellen Aufgaben widmen um wieder Geld in die Kassen der Institution zu spülen. Sie sprechen dabei von „humanize Windows itself“, also einer Aufgabe die sich mit der Anpassung von Windows an ihre Denkweise beschäftigt. Raskin hat Archy übrigens kurz vor seinem Tod noch in „The Humane Environment“ umbenannt, wobei sich bis jetzt die alte Bezeichnung gehalten hat.

## 4 CanonCat

Nach diesem Sprung in der Zeit kommen wir nun wieder zurück in die Mitte der 80er, als Raskin Apple den Rücken kehrte und seine eigenen Wege ging. Er gründete schon bald nach seiner Kündigung seine eigene Firma mit dem Namen „Information Appliances Inc.“. Hier wollte er ein System genau nach seinen Vorstellungen entwickeln, welches nicht von den Vorstellungen und Marketingstrategien anderer Personen beeinflusst wird. Sein Macintosh war mittlerweile auf dem Markt und war mit seinen späteren Revisionen welche den geringen Speicher der ersten Version aufstockten, auch zu einem Verkaufsschlager avanciert. Der Markt an Privatanwendern wurde immer größer und die Benutzer fanden Gefallen an der mausgebundenen Steuerung und dem Desktopprinzip, welches durch die Portierung auf die IBM kompatiblen Rechner schon bald unter dem Namen Windows den Weltmarkt erobern sollte. Viele sagen dass die Canon Cat der Rechner wurde, der der Macintosh eigentlich hätte werden sollen. Raskin empfindet das selbst nicht ganz so, sondern sieht beide Rechner als Meilenstein in der Geschichte der Benutzerschnittstellen. Er sagte 1987 über die Canon Cat: „It is as advanced (in terms of human interface) over the Mac as the Mac was an advance in its day.“

### 4.1 SwyftWare, der Vorgänger

Raskin hatte schon in den Anfangstagen von Information Appliances Inc. und vermutlich auch schon viel früher, die Grundideen wie er sie später in dem Buch „The Humane Interface“ niederschrieb in seinem Kopf. Da er bei Apple nicht mehr genügend Freiräume für die Umsetzung dieser Ideen hatte und auch kein anderes Projekt im Gange war welches seinen Vorstellungen entsprach gründete er eben sein Eigenes. Begonnen wurde das, was später als Komplettrechner Canon Cat auf den Markt kommen sollte, als Software Projekt, welches auf dem Apple II entwickelt wurde. Ziel hierbei war es eine möglichst komfortable und effizient bedienbare Arbeitsumgebung zu entwickeln, mit der alle anfallenden Arbeiten zu erledigen sind. Als Besonderheit ist zu erwähnen, dass das Programm in einer schon lauffähigen Version von sich selbst implementiert wurde. Ab einem relativ frühen Zeitpunkt, als der Compiler für die Sprache Forth schon eingebaut war, begann das Team die Software in der schon funktionierenden Umgebung weiterzuentwickeln. Raskin selbst sprach noch vor kurzer Zeit davon, dass die SwyftWare wahrscheinlich das mit Abstand am besten getestete und funktionierende Programm seiner Zeit war, dadurch dass die Entwickler es ja schon ein bis zwei Jahre lang benutzten, bevor es auf den Markt kam. Er sagte auch dass bis zu seiner Lebzeit kein Bug aufgetreten wäre, von dem er etwas mitbekommen hätte. Da die Software dann ja so gut wurde, wollte Raskin sie auch auf den Markt bringen und dem Privatanwender zur Verfügung stellen. So wurde das SwyftWare genannte Programm für den Apple II auf den Markt gebracht, zuerst auf einer Erweiterungskarte („SwyftCard“ genannt), auf der lediglich die Software gespeichert war und später aus Kostengründen dann auch auf Diskette.

### 4.2 Der Rechner Canon Cat

Die Software sollte nachdem sie für den Apple II gerade von der Fachpresse mit viel Lob versehen wurde, auch als Komplettrechner auf den Markt kommen. Dieses Produkt, dass auch wieder von Raskin bei Information Appliances entwickelt wurde, lizenzierte sich der Drucker- und Bürotechnikhersteller Canon.

#### 4.2.1 Die technischen Details

Rein technisch gesehen war die Canon Cat kein Fortschritt gegenüber dem schon drei Jahre vorher veröffentlichten Macintosh. Raskin hatte wieder die Prämisse verfolgt, nicht am technisch Möglichen zu entwickeln. Vermutlich auch bedingt durch die damals schon recht

alte Entwicklungsumgebung Apple II wurde der Rechner mit nahezu gleichen Komponenten wie der Macintosh ausgestattet. So erhielt das Gerät die gleiche CPU, den Motorola 68000 Chip, und der Rechner war wieder ein Komplettrechner, der nun aber wirklich alles in einem Gehäuse unterbrachte. Er war mit einem 3,5 Zoll Diskettenlaufwerk und einem Arbeitsspeicher von 256KB ausgestattet, und hatte sonst keinen freien Speicher zum speichern von weiteren Daten, wie eine Festplatte zur Verfügung. Es war lediglich ein 256KB großer ROM-Chip (Read only Memory) verlötet, auf dem die Software gespeichert war. In diesem winzigen Platz brachten die Entwickler die komplette Benutzersoftware, sowie je ein Compiler für Forth, sowie die Assemblersprache des Motorola Prozessors und desweiteren noch ein 90000 Wörter umfassendes Wörterbuch unter. Ebenfalls enthalten war ein eingebautes Auto-Dial Modem.

#### **4.2.2 Die Software**

Die Software entspricht wie schon gesagt der SwyftWare, sie stellt die bislang erste und einzige Umsetzung der von Raskin in seinem Buch beschriebenen Konzepte dar. Das Konzept der Canon Cat, war von Beginn an auf ein mausloses Design ausgelegt, so kam auch die Leap Technologie hier zum Einsatz. Sie wurde schon hier durch die zwei Sprungtasten realisiert und erfüllte alle Aufgaben die ihre mausgesteuerten Konkurrenten auch bewältigen konnten, mit Ausnahme von mausgesteuertem Zeichnen natürlich. Dies ist aber wahrscheinlich auch nur daher möglich gewesen, da Raskin hier auch auf das Prinzip der einen multifunktionalen Anwendung zurückgriff, die als einzige Datei mit einem globalen Dokument arbeitete. Raskin führte sie später als Beleg für die Umsetzbarkeit seines Konzeptes an und sagte : „The Cat computer, which I developed for Canon, demonstrated that my alternate approach is implementable and both more productive and more pleasant than GUIs.“ Die Arbeitsoberfläche war stets die selbe und ähnelte einem Editor in dem eben dieses eine Dokument geöffnet war. Sie sieht den heutigen frühen Versionen des Archy Betriebssystems recht ähnlich. Es wurde auch auf ein weiterreichendes Dateisystem und eine Desktopumgebung und alles andere was bei anderen Systemen immer selbstverständlicher wurde, verzichtet. Große Vorteile gegenüber der Konkurrenz hatte die Canon Cat bei der Suche von Inhalten die noch bedeutend schneller funktionierte als beim vergleichsweise immer noch sehr schnellen Apple Finder den Raskin beim Macintosh mit geprägt hatte. Interessant ist es auch zu erwähnen, dass das globale Dokument durch die kleinen Speichermengen ungefähr 80 beschriebenen Seiten entsprach, was das Problem der Unübersichtlichkeit im Vergleich zum heutigen Archy natürlich klein hält. Auch die Umsetzung des verlustfreien Arbeitens durch den Rechner, der auch bei einem Stromausfall keine Daten verliert hat schon damals funktioniert. So setzte schon die Cat die Arbeit stets an dem Punkt fort an dem man sie verlassen hatte. Auch die Kommunikationsfähigkeit über ein Modem, welches zu jeder Zeit im Arbeitsbetrieb sofort zur Verfügung stand, war zu der Zeit noch nicht selbstverständlich und machte die Cat fortschrittlich. Der Rechner war in Sachen Arbeitseffizienz und Datensuche seiner Zeit weit voraus, was auch von Fachjournalisten oder auch anderen Spezialisten der Interfacegestaltung wie zum Beispiel Bruce Tognazzini (heute bei Sun Microsystems) anerkannt und gelobt wurde. Lediglich der breite Markt und damit der wichtigste Teil fand keinen Gefallen am Gesamtkonzept der Canon Cat.

#### **4.3 Gründe des Scheiterns**

Nach nur einem halben Jahr und 20000 verkauften Exemplaren wurde die Cat von Canon wieder vom Markt genommen und der Verkauf eingestellt. Raskin sah mehrere Gründe für das Scheitern, so sagte er selbst dass die Canon Cat nur ein Schatten dessen war, was von ihm und seinem Team bei Information Appliances entwickelt wurde. Canon hat laut Raskin das Prinzip das hinter der Cat stand nie begriffen und durch daraus resultierende Fehlentscheidungen und Fehlvermarktungen große Schuld an deren Scheitern. So wäre der

Rechner so gut designt gewesen, dass er auch durch den damals modernen Einbau eines Bildschirmschoners den „green sticker“, ein Abzeichen für energiesparende Geräte in den USA, erhalten hätte, wäre dieser schon erfunden gewesen. Die Cat war nämlich für den Dauerbetrieb ausgelegt, so führt Raskin als weiteren Grund für das Unverständnis Canons an, dass der Rechner ohne Einschaltknopf konzipiert war, jedoch plötzlich einen eingebaut hatte als die ersten Modelle aus Asien zugesandt wurden. Für Raskin machte dieser kleine Knopf „Lichtjahre im Design“ aus, wie er später darüber schrieb. Canon hatte es verpasst Drittanbieter zu motivieren, Funktionen für das neue Konzept zu entwickeln. Desweiteren hat die Firma beschlossen den Rechner mit einem nicht grafikfähigen Drucker auszuliefern. Dies war natürlich tödlich für die grafikfähige Oberfläche mit verschiedenen Zeichensätzen, die vom Drucker nicht verarbeitet werden konnten. Der „What you see is what you get“ Effekt den Raskin erdacht hatte war damit völlig über den Haufen geworfen. Es gibt aber noch andere weiterreichende Geschichten zum Scheitern, nach denen Steve Jobs aus Rachegeleuten Canon gezwungen hätte die Cat einzustellen, damit Canon Anteile an seiner Firma Next erwerben durfte. Tognazzini sagte dazu :„The Canon Cat did not sell well, but this should be attributed to the hardware on which it ran, as well as Canon’s decision to target this ideal interface for professional writers almost exclusively to low-level clerical workers, who didn’t need its functionality and were confused by its „invisible“ interface.“ Die Canon Cat war wohl eine perfekte Arbeitsmaschine für jeden, der viel schrieb und dies unter Umständen auch noch in Verbindung mit anderen Rechner tun musste oder wollte.

## 5 Zusammenfassung

Jef Raskin hatte eine Vision, und die hatte er schon lange, denn schon zu seinen Universitätsjahren schrieb er erste Ideen zur Gestaltung von Interfaces nieder. Diese Vision hat sich aber im Laufe seines Lebens nicht mehr weiterentwickelt und wurde von ihm im Jahr 2000 in seinem Buch in der Form niedergeschrieben, wie er sie ähnlich wahrscheinlich auch schon in den frühen 80er Jahren benutzt hätte. Hierfür musste er auch immer wieder Kritik einstecken und sich Altmodigkeit vorwerfen lassen. Als er bei Apple den Macintosh entwickelte, versuchte er seine Prinzipien umzusetzen und tat dies auch so lange bis er die Firma nach Differenzen verließ. Wenige Jahre später entwickelte er die Canon Cat, welche wieder auf den gleichen Prinzipien aufbaute, die den Mac eigentlich hätten prägen sollen. Als diese zwar von der Fachpresse anerkannt und mit lobenden Tönen beschrieben wurde, scheiterte sie doch am freien Markt kläglich. Nun gab Raskin das entwickeln kompletter Systeme auf und arbeitete als freier Berater für Firmen wie HP, IBM und Ähnliche, wo er viele Projekte begleitete und deren Interfacegestaltung betreute. Über zehn Jahre nach der Canon Cat fasst er seine Prinzipien zu einem Buch zusammen, welches er unter dem Namen „The Humane Interface“ veröffentlicht. Er passt sich dabei in seiner Argumentation nicht für alle Kritiker ausreichend an die veränderten Umstände der Computerindustrie an. Trotz allem wird sein Buch an zahlreichen Universitäten als Lehrbuch verwendet. Auch wurden viele seiner Konzepte wie zum Beispiel die Datensuche nach Inhalten bei erfolgreichen Systemen aufgegriffen und angewandt. Man darf für die Zukunft gespannt sein ob die Umsetzung seiner Ideen in dem System Archy von Erfolg gekrönt sein wird, oder ob weiterhin nur einzelne Aspekte von anderen Firmen aufgegriffen und verwirklicht werden.

## Literatur

- [1] Homepage von Jef Raskin <<http://jef.raskincenter.org>>  
Insbesondere der Artikel „Holes in the History“  
<<http://jef.raskincenter.org/published/holes.html>>
- [2] Homepage des RCHI  
<http://rchi.raskincenter.org>>
- [3] Der Artikel „Recollections of the Macintosh“ von Raskin  
<<http://mxmora.best.vwh.net/JefRaskin.html>>
- [4] Artikel über die Canon Cat, Archy und Raskin selbst bei Wikipedia :
  - [4.1]<[http://de.wikipedia.org/wiki/Jef\\_Raskin](http://de.wikipedia.org/wiki/Jef_Raskin)>
  - [4.2]<[http://en.wikipedia.org/wiki/Jef\\_Raskin](http://en.wikipedia.org/wiki/Jef_Raskin)>
  - [4.3]<<http://de.wikipedia.org/wiki/Macintosh>>
  - [4.4]<[http://en.wikipedia.org/wiki/Apple\\_Macintosh](http://en.wikipedia.org/wiki/Apple_Macintosh)>
  - [4.5]<[http://en.wikipedia.org/wiki/Canon\\_Cat](http://en.wikipedia.org/wiki/Canon_Cat)>
  - [4.6]<<http://de.wikipedia.org/wiki/Archy>>
  - [4.7]<<http://de.wikipedia.org/wiki/Swyft>>
- [5] Die Geschichte von Apple  
<<http://www.apple-history.com/german.html>>
- [6] Jef Raskin: The Humane Interface  
Erschienen 2000 bei AddisonWesley, ISBN 3827317967