

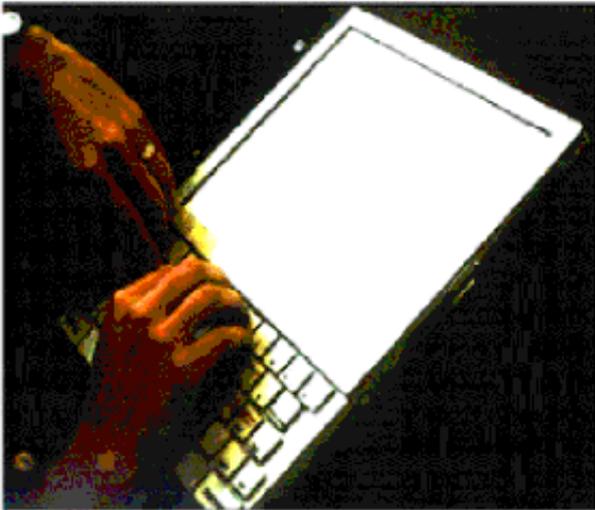
Nr. 2 im Sommersemester 2001

Squeak: Zurück in die Zukunft

Markus Denker

Eine Objektorientierte Programmiersprache, Bytecode-basiert, garbage collection, umfangreiche Klassenbibliothek für z.B. Collections und GUI-Programmierung. Nein, nicht Java: Ich spreche von 1978! Wie kann das sein?

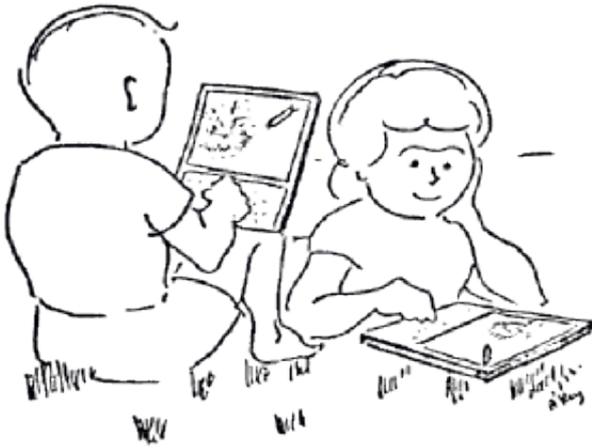
Die Geschichte beginnt so ca. 1968. Unser Held heißt Alan Kay, er ist Doktorant bei Ivan Sutherland an der Universität Utah.



Seine Idee war, einen Computer für Kinder zu bauen: Klein, leicht, tragbar. Einen echten "Personal Computer".

"Jeder wußte damals wie sich das Silizium entwickeln würde" Das Problem, so Alan, war die Software. Nach seinen Überlegungen sollte die Hardware ca. 1980 so weit sein, also blieben 10 Jahre, die entsprechende Software zu programmieren. 1970 wurde ihm eine Forschungsstelle am Xerox PARC angeboten, wo er mit einer kleinen Gruppe an diesem Problem arbeiten konnte.

Das Ziel, das es zu erreichen galt, nannte Alan "Dynabook", ein ca. 1 kg schweres, DIN A4-großes, flaches Gerät. Es sollte mehr ein Medium sein als Werkzeug, eine Art "interaktives Buch", das es Kindern jeden Alters helfen sollte, kreativ zu werden. Wichtig war auch, daß der Computer auch von den Kindern programmiert werden sollte. Die Anwender sollten nicht nur konsumieren, sondern selber aktiv werden und neuen Inhalt schaffen können.



Wenn die Zielgruppe aus Kindern besteht, die noch nicht einmal geboren sind, hat das einige interessante Konsequenzen: Rückwärtskompatibilität ist unnötig, stattdessen muß man vor allem eine einfache, leicht zu erlernende Oberfläche haben. Es wurde schnell klar, das eine die Oberfläche graphisch sein mußte.

Das Ergebnis war ein System names "Smalltalk": Ein integriertes System aus OO-Programmiersprache, Entwicklungsumgebung und Graphischer Oberfläche. Dieses System wurde ende der 70er einem gewissen Steve Jobs vorgeführt, es ist die erste moderne Graphische Benutzeroberfläche mit überlappenden Fenstern, wie man sie heute auf jedem PC findet.

Leider entwickelte sich Smalltalk immer mehr in Richtung einer professionellen Programmiersprache, und das 1980 vorgestellte und später kommerzialisierte "Smalltalk 80" wude dem ursprüngliche Ziel des Projektes nicht mehr gerecht. Aus Alan´s sicht war das Projekt gescheitert, trotzdem kann man aber sagen, daß Smalltalk eines der einflußreichen Software Projekte der letzten 30 Jahre war.



Alan wechselte von Xerox zu [Apple](#), wo er als "Apple Fellow" weiter an Software für Kinder forschte. 1995, am Beginn eines neuen Projektes, war seine Forschungsgruppe auf der Suche nach einer Entwicklungsumgebung, in der der Prototyp entwickelt werden sollte. Nach den Erfahrungen mit kommerziellen Smalltalk Systemen wollte man ein offeneres System, und daher wurde 1996 das "Squeak" Projekt gestartet mit dem Ziel, Apple Smalltalk unter einer Open Source Lizenz weiter zu entwickeln.

Squeak ist, in gewisser Weise, ein neuer Anlauf, die Ziele des Dynabook zu erreichen: Ein dynamisches Medium für Kinder jeden Alters, von der Vorschule bis zu den Erwachsenen, die Squeak selber in Squeak implementieren.

In den letzten 5 Jahren hat Squeak eine interessanten Funktionsumfang erreicht, vor allem im Multimedia-Bereich (3D-Graphik, Flash, Mpeg, etc..). Dabei ist das System immer noch relativ

klein und vorallem sehr portable geblieben: Es läuft auf ziemlich jedem System, vom Handheld bis zur Workstation, je nach vorhandener Umgebung direkt als Betriebssystem-Ersatz, als Applikation oder Browser-Plugin.

Trotzdem bleibt aber noch viel zu tun: Die Oberfläche ist noch sehr gewöhnungsbedürftig (bzw. noch nicht vorhanden), und ein Enduser-System, das die Lücke zwischen den Kindern und den "Gurus" füllt, ist gerade erst am entstehen.

Squeak ist ein faszinierendes System: Es ist vollständig objektorientiert. Zahlen, Klassen, Methoden, sogar die Stackframes sind Objekte. Ein "Edit-Compile-Run" Zyklus fehlt vollständig: Jede geänderte Methode wird sofort beim Speichern übersetzt: Es ist sogar möglich, an einem Programm zur Laufzeit Fehler zu beheben oder gar das laufende Programm zu erweitern.

Die Programmiersprache ist sehr einfach und versperrt nicht den Blick auf das wesentliche: Nicht ohne Grund sind wichtige Entwicklungen im Software- Engineering im Smalltalk-Umfeld entstanden, etwa **Extreme Programming** und **Design Patterns**.

Unter squeakland.org findet man einige Beispiele für typische Kinder-Projekte, dort kann man Squeak direkt mit Hilfe von Browser-Plugins für Windows und MacOS ausprobieren. Squeak.org ist die Website für den eher technisch interessierten, hier gibt's Squeak für jede Plattform, den Quellcode, und Informationen, wie man sich bei der Squeak Mailingliste anmeldet.

Squeak ist Open Source, und jeder kann einfach die Projekt-Mailingliste eintragen lassen und mithelfen. Ich jedenfalls habe dabei viel Spaß, man lernt viel und trifft wirklich interessante Leute.

Und bei all den Diskussionen um eine effizientere, kürzeren und Industrie-näheren Organisation des Informatik-Studiums sollte man nicht vergessen, daß eine sehr gute Sache ist, wenn das Studium die Möglichkeit bietet, sich auch abseits des Mainstream umzusehen. Ich jedenfalls habe das Gefühl, das meine Mitarbeit am Squeak Projekt zu den wichtigsten Erfahrungen meines Studiums zählt.



Home	zurück	auf	weiter	Email
------	--------	-----	--------	-------