

"Kunst reflektiert den Code"

Computerprogramme sind für den britischen Webkünstler und Publizisten Matthew Fuller nicht nur eine Sprache der Maschinen: Sie enthalten auch die Kultur und Ideologie ihrer Gesellschaft

Interview VERENA DAUERER

taz: Software wird für Computer geschrieben. Trotzdem behaupten Sie, die Befehle, die eine solche Maschine verarbeitet, seien kulturell nicht neutral. Warum nicht?

Matthew Fuller: Unter dem Begriff des Werkzeugs versteht man ein Gerät, das die menschliche Gewandtheit erweitert, etwa so, wie ein Messer die Möglichkeiten der Hand verbessert, Dinge zu zerschneiden. Software ist dagegen ein dynamischer Prozess, in den man selbst involviert wird. In der Welt der Gerätschaften waren die Menschen von Objekten umgeben. Mit der Industrialisierung wurden sie von Maschinen umgeben und wurden durch sie ersetzt. Die Software jedoch verschmilzt auf eine gewisse Weise mit Computer und Mensch. Es geht mir darum, genau herauszufinden, wie dieses Vermischen stattfindet."

Sie unterscheiden zwischen sozialer, kritischer und spekulativer Software.

Kritische Software, die zum Beispiel geschützte Spiele knacken kann, will hauptsächlich ein Statement zu sich selbst abgeben. Sie enthüllt, wie etwas funktioniert, und ist damit gleichzeitig eine kritische Abhandlung darüber. Von sozialer Software gibt es zwei Ausrichtungen: einmal die freie Software, die eine direkte Beziehung zwischen den Programmierern und der sozialen Organisation beinhaltet, die die Programmierer erzeugt haben. Schließlich sind die Programmierer zunächst die Hauptuser. Der andere Zweig ist eine Beziehung zwischen Leuten, die Experten beim Programmieren sind, und Leuten, die sich ihre spezielle Software für ihre eigenen Aufgaben ausdenken. Zum Beispiel ist "Linker" von Mongrel ein kleines Programm, das Künstler benutzen können, aber auch Menschen, die aus dem Schulsystem ausgeschlossen wurden. Spekulative Software wie der "Webstalker" deckt gleichzeitig etwas über den Status der Software als kulturelle oder technische Form auf. Die Kunst lässt den Programmcode über seine eigene Ästhetik und technische Natur reflektieren, vergleichbar etwa damit, dass die interessantesten Romane zugleich die Struktur der Erzählung, des Alphabets und der Sprache kommentieren.

Sie sagen, dass Ideologien in eine Software eingewebt werden. Woran erkennt man das?

Das hängt von der Art der Programmierung ab und der Erfahrung des Entwicklers: ob er allein oder in einer Gruppe oder ob er in einer kommerziellen Firma arbeitet. Man kann die Software nicht für sich sehen, man muss auf ihre Verbindungen schauen und auf das gesamte System, in das es eingebettet ist. Wenn eine Firma mit SAP-Software arbeitet, wird sie im Grunde von den Algorithmen angetrieben, die nicht einmal mehr für die Direktoren erkennbar sind. Das macht die Software als Objekt der Kritik so spannend. Ein Betriebssystem hat beispielsweise vierzehn verschiedene Ebenen, dazu kommen die eigentlichen Objekte, mit denen gearbeitet wird, die Applikationen, die Netzwerke. Dann die Leute, die damit verbunden sind, dann die Gesellschaft, ihre Ästhetik, die Technik und Politik. Jeder Punkt liefert einen Ansatz, in die Software hineinzuschauen, sie nachzuvollziehen und aus einer bestimmten Ebene herauszufinden, welche Politik sie verfolgt."

Sie haben selbst an der Programmierung des "Stalker"-Browsers mitgewirkt. In welcher Beziehung steht ein Softwareentwickler zu seinen Programmen?

Das hängt von der Firma ab, in der die Software entsteht. Gerade haben wir auf dem Panel gehört, wie Software in Unternehmen entwickelt wird, in denen die Mitarbeiter überhaupt nicht das Ausmaß des gesamten Programms kennen und jeweils nur einen Teil davon zu bearbeiten haben. Das ist wie Fließbandarbeit. Bei der freien Software dagegen haben die Entwickler eine Vorstellung des Projekts und deshalb auch eine tiefere Beziehung dazu. Bei vielen der künstlerischen Arbeiten ist das genauso.

Immer häufiger wird in dieser Art der eigenverantwortlichen, kollektiven Entwicklung von Computerprogrammen eine Art Vorbild für die reale Politik gesehen. Ist das sinnvoll?

Open Source ist das Gegenteil von Regierungspolitik. Die Macht der Regierungen ist mittlerweile so eingeschränkt, dass sich Politiker selbst nur noch als Manager im Namen des Kapitalismus verstehen. Open Source liefert dagegen ein Modell, das in manchen Gebieten anwendbar sein könnte.

Die Transmediale von Berlin vergibt einen Preis für so genannte Softwarekunst. Welche Kriterien gibt es dafür?

Die sind nicht klar, und deshalb mag ich diese Art des Wettbewerbs nicht. Offenbar hat er es gar nicht nötig, darüber zu diskutieren. Aber wenn man einen solchen Preis vergibt, braucht man eindeutige Antworten. Man muss wenigstens einen Weg zeigen, auf dem die Leute vermitteln können, was Softwarekunst ist. Oder genauer: wie Künstler Software benutzen, auf welcher Ebene was dabei rauskommt. Viele Fragen sind offen. Gibt es eine Kunst auf der Ebene des Programmcodes selbst, oder liegt sie eher abgetrennt davon in der rein ästhetischen Funktion? Oder: Hat die Software einen politischen oder sozialen Auftrag?

Ein Beispiel dafür wäre wohl Ihr gegenwärtiges Projekt?

Richtig. Im Moment arbeite ich in einer Gruppe mit Graham Harwood und Ivan von Public Netbase in Wien an "TextFM". Wir wollten nichts Visuelles, sondern Software als Soundprozess. TextFM empfängt übers Telefon eine SMS, der Computer liest sie dann laut vor und sendet sie übers Radio. So wird ein komplett offener Radiosender geschaffen, dem jeder eine Nachricht, einen Dialog oder einen Slogan schicken kann. Wir haben es seit drei Monaten in einem Zelt in Wien installiert. Es ist interessant, wie der Gebrauch dabei völlig außer Kontrolle geriet und selbst zu einem sozialen Prozess geworden ist. "TextFM" ist freie Software, die jeder verändern kann."