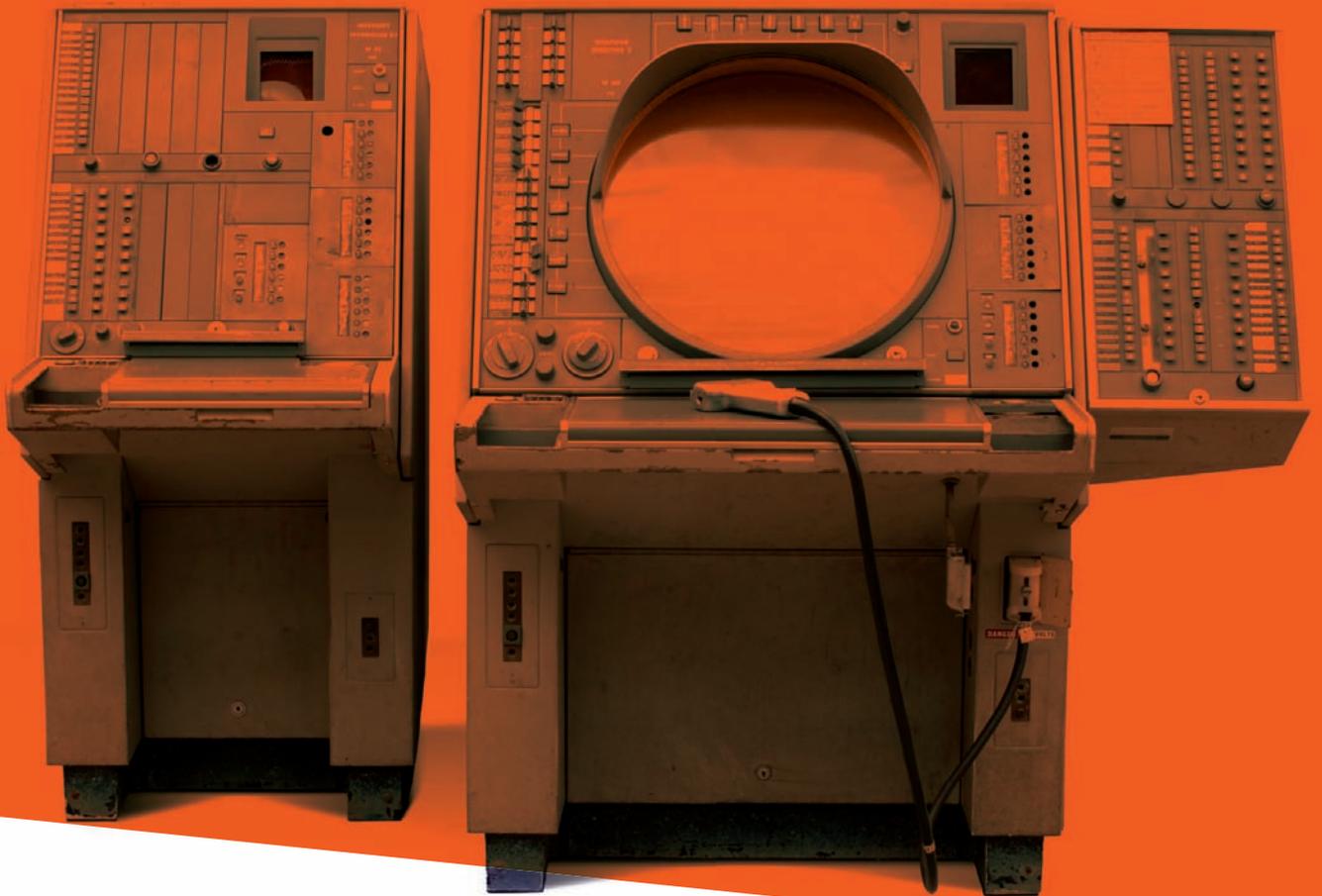


# Trying to be Cake

Feedbackmaschinen oder:



Interaktion als Wiederholen durch  
Unterbrechen

Seit etwa 60 Jahren bedienen wir Computer. Dabei ist die Selbstverständlichkeit, mit der wir an den »Oberflächen« von PCs, Laptops oder Bankautomaten mit den Daten an den »Unterflächen«<sup>1</sup> dieser Rechenmaschinen umgehen, alles andere als natürlich. Und auch wenn die Informatik die Geschichte der Benutzeroberflächen als Fortschrittsgeschichte zu erzählen gewohnt ist, müssen wir, wenn wir verstehen wollen wie Ober- und Unterfläche zusammen kommen, vielleicht nicht auf die neusten und damit »besten« Interfaces schauen, sondern auf ihre Geschichte.

Eine momentan allgegenwärtige Aktualisierung des Interfacefortschritts ist der *Touchscreen*. Mit ihm werden, so könnte man meinen, Daten endlich begreifbar. Die Technik verkürzt den Abstand von Hand und Bild, von Geste und Berechnung auf ein Minimum und steht damit ganz im Zeichen eines Denkens, nach dem die Frage nach benutzbaren Schnittstellen vor allem eine Frage nach »natürlichen« und »intuitiven« Eingabemöglichkeiten ist. Nach dem ersten Einbrechen so »verkörperter« Interaktion in heimische Wohnzimmer durch *Wii* und zuletzt *Kinect* erreicht diese Zukunftsvision nun auf zahllosen Mobiltelefonen jede Hosentasche – verkürzt auf das unmittelbare Zeigen auf den Ort, an dem Daten zu Bildern werden. Darstellung und Manipulation von Information kommen von nun an endlich räumlich und zeitlich zusammen. Andererseits kehrt Interaktion mit dieser Entwicklung an ihre Ursprünge zurück.

Einer dieser Ursprünge ist ein gescheitertes Forschungsprojekt. Das MIT erhält 1943 den Auftrag, einen neuartigen Flugsimulator zu entwickeln: einen universellen Flugsimulator, der gleichermaßen existierende und nicht-existierende Flugzeuge simulieren kann und der so dem Training an aktuellen Maschinen und dem Test zukünftiger dienen soll. Dieser *Aircraft Stability and Control Analyzer (ASCA)* wird nie gebaut werden.<sup>2</sup>

## Wiederholen: Feedback

Bevor aber überhaupt die Idee eines solchen universellen Flugsimulators entstehen konnte, musste eine andere Entwicklung stattfinden. Ihr wichtigstes Merkmal ist der Siegeszug technischer Lösungen, die auf dem Zurückholen der Ausgabe eines Systems in ihren Eingang beruhen: der Siegeszug des Feedbacks in der Nachrichten- und Regelungstechnik der 1920er und 30er Jahre.<sup>3</sup>

»The word »feedback« is a 20th century neologism introduced in the 1920s by radio engineers to describe parasitic,

positive feeding back of the signal from the output of an amplifier to the input circuit.«<sup>4</sup>

Diese Bedeutung des Begriffs soll sich bald grundlegend verschieben. Die Nachrichtentechnik der Zeit kämpft mit dem Problem, dass ihre Röhrenverstärker das, was sie verstärken, zugleich auch verrauschen. Was als Klirrfaktor im Sound verstärkter Gitarren einen ästhetischen Mehrwert bedeuten kann, stellt für andere Anwendungen ein Problem dar: vor allem für die Übertragung gesprochener Sprache in den gerade entstehenden Telefonnetzen. Während die Forschung lange versucht, »bessere« Röhren (mit linearer Verstärkung) zu bauen, kommt der Ingenieur Harold Stephen Black in den *Bell Laboratories* auf eine andere Lösung. Er benutzt den Fehler der Röhren, um ihn zu beheben. Dies geschieht zum ersten Mal in den 1920er Jahren, als er einen Verstärker baut, in dem das Rauschen nach der Verstärkung isoliert wird, um es vom Signal vor der Verstärkung abzuziehen. Das Ausgangssignal des Verstärkers wird damit zu seinem Eingangssignal. Da es aber abgezogen wird, arbeitet dieser Ansatz mit negativem Feedback, das nicht »parasitic« sondern selbstkorrigierend wirkt. Das Wieder-Holen des Outputs in den Input eines Systems kann benutzt werden, um sein Verhalten zu stabilisieren, womit – das ist die entscheidende Konsequenz – Signale (wie die Sprache beim Telefonieren) unabhängig von ihrem Träger (wie den Strömen im Telefonnetz) übertragen und verstärkt werden können.

Auch am MIT findet diese Entwicklung statt. In der Folge der Elektrifizierung der USA versucht das Electrical Engineering Department der Hochschule das Verhalten von Stromnetzen zu verstehen. Diese Arbeit beginnt damit, die Stromnetze als Miniaturnetze im Labor nachzubauen – womit dem Begriff »analog« zu Rechnen seine ursprüngliche Bedeutung gegeben wird: Die Miniaturnetze bilden strukturanaloge Nachbauten ihrer großen Vorbilder und »berechnen« deren Verhalten durch Ähnlichkeit. Bald aber entstehen Netze, deren Ähnlichkeit so allgemein gehalten ist, dass sie nicht länger nur Stromnetze berechnen können. Ein berühmtes Beispiel ist der *Product Integrator*, der das Integral des Produktes zweier Funktionen als elektrische Arbeit (die das Integral des Produktes aus Spannung und Strom ist) berechnet, indem er den Stromverbrauch einer Schaltung durch einen Wattstundenmeter misst. Dieses messende Rechnen wird von nun an die neue (und bis heute gültige) Definition von »analogem« Rechnen werden.

Das Rechnen mit solchen messenden Rechnern ist eine Performance. Am *Product Integrator* verfolgen menschliche

Bediener zwei Funktionskurven an einem Potentiometer, ihre Bewegung verursacht einen veränderlichen Stromverbrauch, den der Wattstundenmeter misst. Diese Messung äußert sich als Rotation. Sie muss, um als lesbares Rechenergebnis erhalten zu bleiben, aufgegriffen und auf einen Plotterarm übertragen werden. Und genau hier fällt plötzlich das analoge Rechnen am MIT und das Verstärken gesprochener Sprache in Telefonnetzen zusammen: die Bewegung des Wattstundenmeters ist eine Zahl,<sup>5</sup> benutzt man ihre Energie zum Antrieb eines Plotters wird sie (zum Teil) verbraucht und damit die Zahl verändert und das Messergebnis verfälscht. Analog Rechnen benötigt also effektive Mechanismen zur Verfolgung und Verstärkung veränderlicher Signale, genau wie die Verstärker der Bell Laboratories. Die Lösung heißt auch hier Feedback.

Beide Entwicklungen leiten einen Paradigmenwechsel in der Regelungs- und Nachrichtentechnik ein, dessen bekannteste Folge die Vision einer universellen Wissenschaft ist, die die Rolle von Feedback weiter radikalisiert: die Kybernetik.<sup>6</sup>

### Unterbrechen: Interrupt

Als der ASCA geplant wird ist diese Entwicklung bereits Geschichte. Die feedbackbasierten Analogrechner sind state-of-the-art und der berühmteste von ihnen, der *Differential Analyzer*, wird



Abb. 2 Im Juni 1947 ist der ASCA noch auf dem besten Weg ein Flugsimulator zu werden

nicht nur dem Namen nach zum Vorbild für den universellen Aircraft Stability and Control Analyzer. Das Entwicklerteam stellt sich den Flugsimulator als »a cockpit or control cabin connected, somehow, to an analog computer related to the design tradition of the electromechanical differential analyzer« vor.<sup>7</sup>

Allerdings gewinnt zur gleichen Zeit ein völlig anderes Konzept »universeller« Maschinen an Bedeutung. Alan Turing löst 1937 in seinem berühmten Aufsatz *On computable numbers, with an application to the Entscheidungsproblem* nicht nur eine Grundfrage der Mathematik, auf dem Weg dorthin zeigt er auch, dass es eine theoretische Maschine gibt, die alle anderen (Rechen-)Maschinen sein kann. Die universelle Turingmaschine bildet das Rechnen als digitales Rechnen mit ganzen Zahlen auf einer Maschinenkonzeption nach, die nur diskrete Zustände und Übergänge zwischen ihnen kennt, und die sich damit sehr von den analogen, messenden Rechnern unterscheidet. Zur Zeit der Entwicklung des ASCA werden die ersten digitalen Rechner gebaut, die ersten Konferenzen zum Thema finden statt und Jay Forrester, der Leiter des Projektes, ist begeistert. »He liked what he saw, and the more he saw, the more he wanted to see.«<sup>8</sup> Für ihn ist klar: nur ein digitaler Hochgeschwindigkeitsrechner kann das leisten, was der universelle Flugsimulator können soll.

Er setzt durch, dass das Projekt auf digitales Rechnen umgestellt wird. Es wird daraufhin in *Whirlwind* umbenannt und hört langsam aber sicher auf, ein Flugsimulator zu werden. Die Ingenieure bemerken das lange nicht. All ihre Arbeit wird benötigt um die Probleme zu lösen, die der Versuch mit sich bringt, den bislang schnellsten Computer aller Zeiten zu bauen. Aber spätestens im Winter des Jahres 1948 lässt sich nicht länger ignorieren, dass *Whirlwind* kein Flugsimulator ist. In einem analogen Flugsimulator sind nämlich, wie schon beim *Product Integrator*, Zahlen und Bewegungen identisch. Die Frage, wie die berechnete Dynamik auf die Instrumente eines Cockpits zu



Abb. 3 1950 sind Cockpit und Controls Geschichte und *Whirlwind* ist ein universeller Rechner mit Legitimationsproblem

übertragen seien, wird damit zur bekannten Frage nach der Verfolgung und Verstärkung eines veränderlichen Signals durch Feedback. *Whirlwinds* Zahlen sind dagegen diskret. Um zu Bewegungen zu werden müssen sie übersetzt werden und zunächst weiß niemand wie: »The digital computer was too new.«<sup>9</sup> Das bereits fertige Cockpit funktioniert nicht mit dem digitalen Rechner – und wird verschrottet.

Übrig bleibt eine universelle Maschine (im Sinne der Turingmaschine), die offensichtlich nicht universell (was ihre wörtliche Anschlussfähigkeit an die Instrumente eines Cockpits anbelangt) genug ist, um als Flugsimulator dienen zu können: »Der Umstieg auf die zu diesem Zeitpunkt noch unsichtbare Digitaltechnik bedeutete nichts anderes, als wieder beim Blindflug anzufangen.«<sup>10</sup>

Das bleibt natürlich nicht unbemerkt. Das *Office of Naval Research*, das das Projekt finanziert, bezweifelt schließlich, dass dessen Fortführung weiter Sinn machen wird und der Mathematiker Warren Weaver fragt: »Was *Whirlwind* failing to be good biscuits by trying to be cake?«<sup>11</sup>

Die Rettung ist Geschichte: Aus dem Projekt wird das Luftraumüberwachungssystem *SAGE*. *Whirlwind* wird an ein Radarsystem gekoppelt und verfolgt von nun an Flugbewegungen und berechnet Abfangkurse – und bleibt damit seiner ursprünglichen Bestimmung insofern treu, als dass es im control loop der Luftabwehr wieder Teil eines Feedbacksystems ist, das sich zwischen feindlichen Flugzeugen, Tracking und Flugabwehr vollzieht.

Um aber überhaupt universeller Rechner und gleichzeitig regelndes Glied eines Feedbacksystems sein zu können, muss *Whirlwind* über die Idee der Turingmaschine hinausgehen. Denn dieses theoretische Modell sieht vor, dass zu Anfang einer Berechnung alle verfügbaren Daten vorhanden sind, abgearbeitet werden und in eine

Ausgabe münden (wenn die Maschine nicht in einem unendlichen Zyklus hängen bleibt). Dieses Modell des Rechnens ist eine bis heute gültige Präzisierung dessen, was wir Algorithmus nennen.

Feedback dagegen kennt diesen linearen Ablauf nicht. Hier stehen keineswegs am Anfang alle jemals benötigten Daten zur Verfügung, schlimmer noch: sie entstehen erst in Abhängigkeit von den Zwischenergebnissen des Rechnens. Der berechnete Abfangkurs wird an die eigene Luftabwehr übergeben und dadurch irgendwann der Kurs des Angreifers verändert, so dass der Abfangkurs erneut angepasst werden muss. Mit *Whirlwind* verlässt die Informatik damit schon zu ihrer Anfangszeit den sicheren Boden der eigenen Theorie, was diese Theorie bis heute noch nicht aufgeholt hat.<sup>12</sup>

Die Fähigkeit, zur Laufzeit auf Eingaben reagieren zu können, ist die entscheidende Neuerung *Whirlwinds*. Auch wenn sie für seine Entwickler völlig selbstverständlich ist, folgt sie doch einfach aus einer Ingenieurstradition, für die Rechnen und Steuern das gleiche ist. Um sie zu realisieren, muss das stetige Abarbeiten diskreter Schritte, das den Algorithmusbegriff kennzeichnet, unterbrochen werden.

Und hier beginnt *Whirlwind* mit der Kopplung, zu der die heutigen *Touchscreens* zurück kehren: Die Radardaten werden, wie damals üblich, auf Bildschirmen dargestellt, eine *Lightgun* ermöglicht es, Objekte darauf auszuwählen

– und über diese Auswahl in den Programmablauf des Computers einzugreifen. Bei *SAGE* werden so durch menschliche Bediener feindliche von freundlichen Flugbewegungen unterschieden. Diese überraschend zeitgenössisch wirkende Schnittstelle ist tatsächlich nichts als die direkte Konsequenz der Notwendigkeit, einen Computer beim Rechnen zu unterbrechen, wenn er interaktiv sein soll. Die *Lightgun* ist nämlich keine Gun, sie sendet keinen Lichtstrahl aus, sondern ist ein Sensor. Auf ein Objekt auf dem Bildschirm gerichtet, nimmt sie das Licht auf, das der Elektronenstrahl beim Auftreffen auf die Mattscheibe verursacht. In diesem Moment meldet sie das Licht zurück an den Rechner und unterbricht ihn so beim Zeichnen. Damit ist das, was gerade gezeichnet wurde, zwangsläufig das, was der Benutzer ausgewählt hat und worauf er zeigt. Es ist die Unterbrechung des Rechnens, die diese Auswahl im Rechenprozess wirksam werden lässt und es ermöglicht, dass der Computer »auf den langsamsten Systembestandteil Rücksicht nehmen konnte, nämlich den Benutzer der *Lightgun*.«<sup>13</sup>

*Whirlwind* löst das Unterbrechen noch durch Software.<sup>14</sup> Wenig später wird es aber als *Interrupt* zu einer Hardwarekomponente werden, die alle Computer besitzen und die nicht nur jeder Interaktion, sondern auch dem heute allgegenwärtigen Multitasking zu Grunde liegt.

Um also universelle Maschine im Sinne Turings sein zu können, und gleichzeitig dieses Konzept um die Fähigkeit zum Feedback zu erweitern – eine Fähigkeit, mit der durch die Abkehr von der analogen Rechen- und Regelungstechnik gerade im *Projekt Whirlwind* gebrochen worden war – führt *Whirlwind* fast unbemerkt einen neuen Maschinentyp ein. Dieser neue Maschinentyp erweitert die Turingmaschine um die Fähigkeit der ständigen Wieder-Holung ihrer eigenen Ausgaben und deren Einfluss auf die Umwelt. Sie ist die Grundlage aller Interaktionen – seien sie noch so altmodisch oder modern. Man könnte sie Feedbackmaschine nennen.

#### Lasse Scherffig

arbeitet am *Lab3*, Labor für Experimentelle Informatik der Kunsthochschule für Medien Köln. Er studierte Kognitionswissenschaften und Digitale Medien in Deutschland, den USA und der Schweiz. Zur Zeit arbeitet er an einer Dissertation über Feedbackmaschinen und ist in wissenschaftlichen und künstlerischen Projekten aktiv.

Abb. 1 *SAGE*-Bedienkonsole mit Bildschirm und Lightgun.

Copyright *Computer History Museum*. All Rights Reserved,  
<http://www.computerhistory.org/press/gallery/>

Abb. 2 Copyright *The MITRE Corporation*. All Rights Reserved,

[http://www.mitre.org/about/photo\\_archives/whirlwind\\_photo.html](http://www.mitre.org/about/photo_archives/whirlwind_photo.html)

Abb. 3 Copyright *The MITRE Corporation*. All Rights Reserved,

[http://www.mitre.org/about/photo\\_archives/whirlwind\\_photo.html](http://www.mitre.org/about/photo_archives/whirlwind_photo.html)

- 1 Frieder Nake: Surface, Interfac, Subface: Three Cases of Interaction and one Concept. In: Uwe Seifert, Jin Hyun Kim und Anthony Moore (Hrsg.): *Paradoxes of Interactivity*. Bielefeld (Transcript) 2008, S. 92–109.
- 2 Einen Umfassenden Überblick hierzu liefern Kent C. Redmond und Thomas M. Smith: *Project Whirlwind. The History of a Pioneer Computer*. Bedford (Digital Press) 1980.
- 3 Zum Folgenden vergleiche Stuart Bennett: *A History of Control Engineering 1930–1955*. Hitchin (Peter Peregrinus Ltd.) 1993 und David Mindell: *Between Human and Machine*. Baltimore, (Johns Hopkins University Press) 2002.
- 4 Stuart Bennett: A Brief History of Automatic Control. In: *IEEE Control Systems*. Volume 16 Issue 3, 1996, S. 17.
- 5 Vergleiche hierzu auch Claus Pias: *Computer Spiel Welten*. Weimar (Bauhaus-Universität Weimar) 2000, S. 45.
- 6 Lasse Scherffig: The Human Being as a Servo. Von feedback control zur Kybernetik. In: Stefan Fischer, Erik Maehle, Rüdiger Reischuk (Hg.): *Informantik 2009, Im Fokus das Leben, Lecture Notes in Informatics*. Bonn (GI) 2009, S. 766–776.
- 7 Redmond und Smith: *Project Whirlwind*. S. 32.
- 8 Redmond und Smith: *Project Whirlwind*. S. 27.
- 9 Redmond und Smith: *Project Whirlwind*. S. 49.
- 10 Pias: *Computer Spiel Welten*. S. 53.
- 11 Zitiert nach Kent C. Redmond und Thomas M. Smith: *Lessons from Project Whirlwind*. In: *IEEE Spectrum*, Volume 14 Number 10, 1977, S. 52.
- 12 Vergleiche zur Einführung in die anhaltende Diskussion Peter Wegner: *Why interaction is more powerful than algorithms*. In: *Communications of the ACM*, Volume 40 Issue 5, 1997.
- 13 Pias: *Computer Spiel Welten*, S. 57.
- 14 Mark Smotherman: A Sequencing-based Taxonomy of I/O Systems and Review of Historical Machines. In: *Computer Architecture News*, Volume 5, 1989.