DJNZ $+1 – Medienwissenschaft der Zeitverschwendung
Ein Mission Statement aus dem Signallabor

Dr. Stefan Höltgen (Wiss. Mitarbeiter am Lehrstuhl für Medientheorien)

0.

Ende 2008 bin ich von Bonn nach Berlin übersiedelt; meine Dissertationsschrift hatte ich an der Rheinischen Friedrich-Wilhelms-Universität eingereicht, die Disputation war auf März 2009 datiert. In Berlin wollte ich mein Forschungsthema methodes aus der Doktorarbeit ausbauen und auf einen anderen Gegenstand übertragen: Es sollte um die Frage gehen, wie sich die Wechselwirkungen zwischen der literarischen Fiktion (Science Fiction) des Computers und der technikhistorischen Entwicklung desselben im 20. Jahrhundert vollzogen haben könnte. Dazu habe ich mir zunächst einen Überblick über den Forschungsgegenstand (Computer und SF-Literatur) sowie über die Möglichkeiten, einen solchen in der akademischen Landschaft in Berlin zu situieren, verschafft.

Auf Empfehlung meines Doktorvaters Michael Wetzel bin ich im Wintersemester 2010/11 als Lehrbeauftragter an den Lehrstuhl für Medientheorien gekommen, um dort ein Seminar zu meinem Thema „Eine Medien-, Technik- und Kulturgeschichte des Roboters“ abzuhalten. In Gesprächen und Diskussionen im Anschluss an einen Gastvortrag im Kolloquium von Wolfgang Ernst sind mir dann erste Zweifel an der Originalität und Durchführbarkeit des Vorhabens gekommen: Auf welche Weise lässt sich eine diskursive Wechselwirkung adäquat untersuchen und welchen Erkenntniswert fügt man der Medienwissenschaft – über die bloße Neustrukturierung von Anekdoten hinaus – hinzu? Zu guter letzt: Läuft die Frage nicht bloß auf eine „Plausibilisierung“ von Fiktion im Rahmen einer technisch interessierten Literaturwissenschaft hinaus?

1.

Als mir die Stelle des Wissenschaftlichen Mitarbeiters am Fachgebiet Medienwissenschaft angeboten wurde, besann ich mich auf eine „alte Kompetenz“ nämlich die Computerprogrammierung, die ausdrücklich in Forschung und Lehre erwünscht war, und stellte Überlegungen zu einem neuen Projekt an: „Die Archäologie des frühen Mikrocomputers und seiner Programmierung“ sollte nun in meinem Fokus und der Verknüpfung von Forschung und Lehre stehen. Zwei Publikationen, die ich bereits im Vorfeld für das vorher angedachte Projekt ausgewertet hatte, haben mir dazu eine alternative Perspektive eröffnet: Claus Pias' Sammelband „Die Zukünfte des Computers“ und das von Friedrich Kittler, Georg Tholen und Norbert Bolz herausgegebene „Der Computer als Medium“. Bemerkenswert erschien mir: In beiden Büchern geht es zwar auch um konkrete Computer, vor allem aber um „den Computer“ als universelle Turingmaschine – einen Computer also, der gar nicht existiert und deshalb weder Zukünfte haben noch ein Medium sein kann.

Die Computer sollten es vielmehr sein, die ich in meiner Arbeit fokussieren wollte: Wie stellen sich Eigenschaften wie die Entwicklung von Programmiersprachen, Medienkonversion, Schnittstellendesign, Softwareentwicklung und andere Aspekte aus der Perspektive einer medienarchäologischen Untersuchung dar? Das zu diesem Zeitpunkt im wesentliche ungenutzte Signallabor sollte mir den notwendigen Raum geben, diesen Fragen praktisch nachzugehen. Dort begann ich eine Sammlung mit Mikrocomputern der 1970er-, -80er- und -90er-Jahre anzulegen, die sowohl der Forschung dienten als auch in der Lehre zur Verfügung stehen.

Als neu eingestellter Mitarbeiter habe ich begonnen für das Bachelor-Studium einen regelmäßigen Programmierkurs im Propädeutikum einzurichten, in welchem die Programmiersprache BASIC

unter je unterschiedlichen Fragestellungen erlernt werden sollte (Computerspiele, Grafikprogrammierung, Zeichenmanipulation, …) Zusätzlich begann ich einen wöchentlichen Kurs für Programmieren in Assembler anzubieten, um der Anforderung, die Computer bzw. Mikroprozessoren auf der (maschinellen) Ebene des Signal-/Symbolübergangs als Medien zu analysieren, gerecht zu werden. Der Kurs ist mittlerweile für sowohl Bachelor- als auch Master-Studenten im Rahmen der Projekt- und Praxismodule anrechenbar und mit mehr als 20 Teilnehmern pro Semester sehr gut angenommen.

Sowohl dieser Kurs als auch das Signallabor stehen aber nicht nur Mitarbeitern und Studierenden offen, sondern auch der interessierten Öffentlichkeit. Zu Ermöglichung dieser „Außenwirkung" fügte ich noch die 4- bis 6-wöchentlich stattfindende Computerspiele-Veranstaltung „Game Circuits – Operative Computerspielanalyse" hinzu, in der auf der Original-Hardware das „epistemische (Computer)Spiel" als Analyseobjekt spielend untersucht wird. Fragen, wie die nach Künstlicher Intelligenz, Multimedia-Fähigkeiten, Phänomenologie des Programmierfehlers, die Beziehung zwischen Code, Mensch und Maschine (beim Abtippen eines BASIC-Computerspiel-Listings) und anderes lassen sich an und mit Computerspielen besonders eindrücklich aufwerfen und beantworten. Die Konzentration auf die alten Plattformen ermöglicht hierzu bestmögliche Zugänglichkeit und Nachvollziehbarkeit des Hard-Software-Verbundes.

2.

Etliche der medientheoretischen Fragen, die am Lehrstuhl von Wolfgang Ernst eine zentrale Rolle spielen, lassen sich auf meinen Forschungsgegenstand applizieren. An einem Beispiel aus dem ersten Assembler-Kurs möchte ich dies vorführen. Dazu zuerst ein kleines Codefragment:

```
8000: LD B,9
8002: NOP
8003: DJNZ $+1
8005: RET
```

Das kleine Assemblerprogramm stellt eine klassische Zählschleife dar, bei der eine Anzahl von Takten mit NOPs (No Operation) „verschwendet" wird. Als Schleifenzähler dient das Register B, das in Adresse 8000 mit dem Dezimalwert 9 geladen wird – die Schleife soll also von 9 bis 0 rückwärts laufen. Dazu verhilft der komplexe Opcode DJNZ mit der relativen Sprungadresse $+1: Er dekrementiert B bei jedem Durchlauf um 1 und springt solange zur Adresse $+1, wie B noch nicht Null ist.

1. Welche Sprache(n) sprechen Computer: An DJNZ lässt sich zunächst zeigen, wie ein komplexe Programmierproblem auf Maschinenebene implementiert ist: Der Opcode DJNZ stellt ein im Z80-Mikroprozessor festverdrahtetes Mikroprogramm dar – hier fließen Software und Hardware in einer Struktur zusammen. Das Mikroprogramm – zumal in seiner hier vorliegenden materiellen Manifestation – stellt eine „Protosprache" dar, die zwar über ein eigenes Regelwerk verfügt, aber keine „parole" ausbildet – also von niemandem „gesprochen" bzw. programmiert wird als der Maschine selbst. Das Programm benutzt allerdings nicht nur Mikroprogramme und Assembler-Anweisungen, sondern auch einen „Pseudo-Opcode", der sich in „$+1" verbirgt. Das „$" ist ein Platzhalter für das Offset, das einer Adresse hinzugefügt wird. Diese Adresse liegt zur Verarbeitung des Befehls auf dem Programm Counter (s.u.), einem prozessorinternen Stapelspeicher. „$" wird erst beim Assemblieren des Programms mit einem konkreten Offset zwischen -126 und +127 ersetzt und dieser dann zur Laufzeit des Programms selbst zur Adresse auf dem Programmcounter addiert (bzw. davon subtrahiert). „Pseudo-Opcodes" schlagen die Brücke zwischen Maschinensprachen und höheren Programmiersprachen, indem sie den Programmierer von

der müßigen Aufgabe der Adressberechnung freistellen. Ihre Befehls- und Funktionsinventar ist nirgends festgeschrieben, sondern unterliegt zumeist einer tradierten Konvention, sich sich im Laufe der Entwicklung von Assemblern herausgebildet hat.

2. Welche Zeitverhältnisse herrschen im Computer: Im Phänomen der Schleife offenbaren sich unterschiedliche Facetten medientechnischer Zeitverarbeitung und -verwaltung: Computer zählen in Schleifen auf Maschinenebene in der Regel vom Schleifenstartwert rückwärts bis Null, während das Programm selbst doch (gemessen am Adressenwert des Program Counters) „vorwärts“ abläuft. Die dezidierte Anzahl von Schleifendurchläufen ermöglicht einen deterministischen Programmablauf (im Gegensatz zu einer Endlosschleife), bei der die Speicher des Computers sukzessive verändert werden. Darüber hinaus (bzw. darunter) laufen aber noch die internen Zeitprozesse des Computers ab: Taktzyklen, die je nach Größe des Opcodes eine reale Abarbeitungszeit des Programms nach sich ziehen. Der Sinn des obigen Programms liegt in der „Verschwendung“ solcher Zyklen. Das ist zum Beispiel notwendig, um den schnellen Prozessor mit langsamen Peripheriegeräten (einem Drucker, einem Diskettenlaufwerk aber auch dem Rasterstrahl des Monitors) zu synchronisieren und so die Zeitverhältnisse verschiedener technischer Medien aneinander anzupassen.
3. Was und wie rechnet ein Computer: Im Programm selbst offenbart sich eine computertypische Arithmetik: DJNZ stellt einen Dekrementierer dar, der stetig subtrahiert und dabei prüft, ob die Differenz bereits 0 ist. Diese Prüfung findet auf Basis einer Boole'schen Arithmetik mithilfe logischer Vergleichsoperatoren (im Beispiel: statt. Im konkreten Computer müssen also alle drei formalen Sprachen – Programmiersprache, Mathematik und Logik – aufeinandertreffen, damit der er „computieren“ kann.
4. Wie speichert ein Computer: Unterschiedliche Typen von Speichern werden vom Programm explizit und implizit genutzt: Das Prozessorregister B, das mit Hilfe von Flipflops 8 Bit parallel speichert. Im Hintergrund arbeitet der Program Counter, der ein Stapel- bzw-Kellerspeicher ist und dem Prozessor die Adresse des nächsten abzuarbeitenden Befehls übergibt. Letzteres ist eine unbedingte Notwendigkeit in Computern mit der von-Neumann-Architektur, die nicht zwischen Daten und Programmanweisungen unterscheiden können. Auf diesen Program Counter greift DJNZ implizit zu: Wenn die logische Prüfung wahr ist (Register B als bis 0 dekrementiert wurde), dann wird zur Adresse des Programmcounters „+1“ gesprungen. Hier werden reale Orte im Computer (im RAM-Speicher) mit Hilfe von Logik und Arithmetik erreichbar gemacht.
5. Der Computer als konkretes, technisches Medium: Das Programm hat also einen realen Ort (im Speicher), eine reale Zeit (in der es abläuft) und eine reale Größe (in Form gespeicherter Binärzustände). Dies lässt sich bis hinab zu jedem Befehl errechnen: DJNZ residiert in den RAM-Adressen 8003 (Opcode) und 8004 (Sprungziel), ist also 2 Byte groß, verbraucht beim Aufruf 8 oder 13 Takte (je nachdem, wie die logische Prüfung auf 0 ausfällt), dauert also bei einem mit 4 Mhz getakteten Z80-Prozessor 4 oder 6,5 Mikrosekunden. Dies lässt sich mit Hilfe von Oszilloskop und Logikanalysator an den Adress- und Datenbus-Leitungen des Mikroprozessors messend bestätigen.
6. Fehlerprogrammierung als Analyseverfahren: Leider funktioniert das obige Programm nicht bestimmungsgemäß, denn es enthält einen Programmierfehler: Die relative Adresse, die angesprungen werden soll, so lange die logische Prüfung auf 0 noch nicht wahr ist, sollte eigentlich zurück ($-3) nach 8002 springen, springt jedoch vorwärts ($+1) nach 8005 und beendet das Programm so nach nur einem Schleifendurchlauf. Der Fehler verursacht keinen Absturz sondern ein unerwartetes Programmverhalten und zählt damit zu den logischen oder semantischen Fehlern. Er offenbart, welche Schwierigkeiten im Verhältnis der sprachlichen Weltbezogenheit des Programmierers und der formalsprachlichen Übersetzung liegen können. Ein Verhältnis, das archetypisch für den Umgang mit dem Computer ist, das die Möglichkeiten und Notwendigkeiten von Medienkompetenzbildung verdeutlicht und nicht zuletzt auch emergente, kreative Impulse freisetzt: Wie verhält sich ein ehedem „richtiges“ Programm, wenn man es auf diese Weise modifiziert? Eine codebasierte, experimentelle

Medienarchäologie könnte auf genau diese Weise durch Fehler zu Erkenntnissen führen: Hätte man mit der obigen Schleife versucht, Computer und Peripherie synchronisieren, wäre es zu einem Moiré-Effekt gekommen, bei dem sich zwei Taktfrequenzen überlagern und so ganz unvorhergesehene Ergebnisse zeitigen.

Um solche Aspekte adäquat untersuchen zu können, muss der konkrete Computer operativ vorliegen und programmiert und messtechnisch analysiert werden. Die zahlreichen medientheoretischen Fragen, die sich an solche Einzelphänomene koppeln lassen, erfordern zuerst eine Kompetenzbildung – und mithin eine Programmierdidaktik, die sich Fragen nach „Zeitgemäßheit“ und „Berufsbezogenheit“ entzieht. Alte Plattformen und Programmiersprachen zu verwenden, kann helfen, solchen Ansprüche im Vorfeld zu begegnen.

Medienwissenschaft ist nicht Ingenieurswissenschaft oder Informatik. Sie ist aber auch nicht Technik*geschichte* oder *Geschichte* der Informatik, was – wie im obigen Beispiel – durch das beständige Flankieren der Arbeit an Hard- und Software mit medientheoretischen Überlegungen verdeutlicht werden muss. In Medienwissenschaft treffen diese Disziplinen vielmehr miteinander und mit Theorien aus Philosophie und Kulturwissenschaft konfrontiert. Von dort aus fragt Medienwissenschaft nach dem Einfluss und der Wirkung von Medientechnologien auf die Wissensgeschichte der „Mediengesellschaften“ (welche scheinbar ja selbst erst aus ihnen emergieren). Meine ursprünglich für das Forschungsprojekt angedachte Methodologie einer „Medienwissenschaft der Wechselwirkungen“ wird vom vorliegenden Projekt also am Ende doch noch „eingeholt“ – zwar nicht als Wechselwirkung zwischen Technik- und Literaturgeschichte des Computers, wohl als eine für die Wissensgeschichte und Medientechnologien der Computer.

3.

Dieser modus operandi, der eine enge Verzahnung von Forschung und Lehre notwendig macht, soll künftig noch weiter ausgebaut werden.

- Eine Zielfrage meines eigenen Forschungsprojektes ist die nach den Bedingungen und Möglichkeit eines „operativen Computermuseums“, in welchem Mediengeschichte durch praktische Medienarchäologie (Hacking) erschlossen und „der Computer“ als Medium erfahrbar wird, das einzig in seiner konkreten Ausgestaltungen und zur Laufzeit gänzlich verstanden werden kann.
- In Zusammenarbeit mit dem Museum für Kommunikation soll diese Frage „Wie funktioniert ein Computer?“ zu einer Ausstellung führen, die zentrale theoretische Anliegen der Medienwissenschaft in praktischen Anwendungen und Experimenten vor Augen führt und an eine Medienepistemologie des Computers rückkoppelt.
- Für 2016 ist eine Konferenz am Fachgebiet geplant, in der es um eine der Entstehungsgeschichten der künstlichen Intelligenz geht, wie sie 1966 im Programm ELIZA in Erscheinung trat. Die Medienwissenschaft bietet hierfür Methoden und Theorien, die eine bloß historische Würdigung oder Frage nach den (sozialen, technischen, …) Konsequenz dieser Erfindung überschreitet und ELIZA als Beispiel für zahlreiche medienwissenschaftliche Detailprobleme anführt.

Stefan Höltgen im Oktober 2013