

Beschreibung des Vorhabens - Projektanträge

Stefan HÖLTGEN (Berlin)

Computerphilologie: Technische Lektüren der BASIC-Programmierkultur

Beschreibung des Vorhabens

1 Ausgangslage

Ab Mitte der 1970er-Jahre diffundieren Programmiersprachen durch die Popularisierung der (Mikro)Computertechnik aus den Spezialisten- in die Privat- und Hobbyistenkulturen. Die Sprache BASIC nimmt hierbei eine besondere Stellung ein, weil sie in zahlreichen Hobbycomputern dieser Zeit implementiert ist. Hieraus ist über die Jahrzehnte eine einzigartige Programmierkultur erwachsen, die bislang noch nicht systematisch erforscht wurde. Diese soll das im Folgenden beschriebene Forschungsprojekt mit interdisziplinären Methoden für den Zeitraum von 1974 bis 1991 untersuchen.

Eine Besonderheit der Forschungslage zu Programmiersprachen im Allgemeinen und zu BASIC im Besonderen liegt in der Überschneidung von dokumentarischer, didaktischer, praktischer und analytischer Literatur. Mit der Erfindung von BASIC und der Entwicklung weiterer Dialekte wuchs der paratextuelle Korpus an Sekundärliteratur, die oft implizit und explizit auch analytische Stellungnahmen zur Programmierung und Programmiersprache enthält. Solche Literatur wird das Projekt berücksichtigen; der nachfolgende Forschungsstand beinhaltet *exemplarisch*¹ einige dieser Beiträge, konzentriert sich allerdings auf jüngere Publikationen dezidiert wissenschaftlicher Provenienz und benennt die sich daraus ergebenden Forschungslücken.²

1.1 Stand der Forschung und eigene Vorarbeiten

a) Historiografische Forschung

BASIC wird seit Ende der 1960er-Jahre [Sammet 1969:229-232] in eine fortlaufende Historiografie der Programmiersprachen [Wexelblat 1981; Bergin/Gibson 1996] integriert. Hierbei überwiegen allerdings noch dokumentarische Konferenzbeiträge der Entwickler von BASIC [Kurtz 1981] sowie überblicksartige Sprachbeschreibungen [Lee 1971]. Erst nach Ende der Hochzeit von BASIC setzt eine medien- und kulturwissenschaftliche Auseinandersetzung vor allem mit der Frühgeschichte der Programmiersprache [Rankin 2018:66-105; Montfort u.a. 2013:157-194] sowie mit einzelnen ihrer Dialekte (z.B. *Microsoft-BASIC* [Halvorson 2020], *BBC BASIC* [Gazzard 2016:45-70]) ein.

Diese Reflexionen orientieren sich an den Forschungsprogrammen der *Platform Studies* [ebd.], der *Software Studies* [Montfort u.a. 2013] oder an den Methodologien der Erfinder:innen-, Firmen- und Wirtschaftshistoriografien. Halvorson [2020:137] sieht eine beträchtliche Forschungslücke in der wissenschaftlichen Auseinandersetzung mit den Mikrocomputer-BASIC-Dialekten in Bezug auf deren Gemeinsamkeiten, Besonderheiten und Nutzungsweisen in privaten wie professionellen Kontexten. Eine bedeutsame Quelle historiografischer Forschung stellen die Arbei-

1 Die bibliografischen Angaben solcher Beiträge sind in *Kursivschrift* gesetzt.

2 Diese sollen durch das Projekt bearbeitet werden und sind in **Fettschrift** dargestellt.

ten der BASIC-Erfinder Thomas E. Kurtz und John G. Kemeny dar. Diese haben die Entwicklungsgeschichte ›ihrer‹ Sprache von den Anfängen bis in die 2000er-Jahre begleitet [Kemeny/Kurtz 1967; 1985; Kemeny 1984; Kurtz 1969; 1981; 2002; 2009]. **All diesen historiografischen Beiträgen ist die Makroperspektive auf die Sprache und ihrer Kultur gemein, die allerdings nicht durch mikroperspektivische Bezüge auf konkrete Sprachelemente begründet wird.**

b) Informatische Forschung

Die Informatik ergänzt diese Makroperspektive teilweise und diskutiert BASIC vor allem auf Basis seines Programmierparadigmas: Insbesondere die Forderung das seinerzeit neue Paradigma *strukturierter Programmierung* umzusetzen [Dahl u.a. 1972:72], erfüllt BASIC zunächst nicht, sondern wird als »Spaghetticode« [Lorenzo 2017:135; Halvorson 2020:143f.] exemplarischer Stellvertreter der sogenannten *GOTO-Kontroverse* (stellvertretend: [Wulf 1972]). Die Hardwarebezogenheit späterer BASIC-Dialekte liefert der deutschsprachigen Informatik-Didaktik Argumente für ein Abraten von BASIC als »Schulsprache« [FEoLL 1976:C1-C14]. Die Verbreitung der Homecomputer führt jedoch dazu, dass BASIC zunächst noch ›Schulsprache‹ bleibt [Keidel/Müller 1983]. **Auch hierbei bleiben Detailanalysen der Syntaxen, die Aufschluss über den Aufbau und die Operativität des jeweiligen BASIC-Dialektes geben, unberücksichtigt, obwohl diese das Programmierenlernen maßgeblich beeinflussen.**

Solche Syntaxbeschreibungen werden erst ab den 1980er-Jahren über die Entwicklung von BASIC-Compilern und -Interpretern zum Gegenstand hobbyistischer [Klein 1982] und nicht-curricularer, didaktischer Auseinandersetzungen [NDR 1984] – die Beschäftigung mit BASIC wechselt hier von der *Oberfläche* (der praktischen Anwendung) zur *Unterfläche* (der Tiefenstruktur formaler Sprachverarbeitung durch Computer). Die in diesem Kontext publizierte Literatur setzt sich mit den genannten fachinformatischen Diskursen allerdings nicht explizit auseinander. **Im Rahmen der jüngeren Forschungspublikationen zu BASIC werden ebenso nur cursorisch Beschreibungen von Übersetzerprogrammen vorgenommen [Berger 2011:105f.].**

c) Linguistische und philologische Forschung zu Programmiersprachen

Die Untersuchung der Beziehungen zwischen Linguistik und Programmiersprachen ist immer noch ein »work in progress« [Rice u.a. 2014]. Dessen Anfänge reichen bis in die frühen 1980er-Jahre zurück und entstanden vor dem Hintergrund der Code-Qualitätssicherung [Misek-Falkoff 1982]. Aus jüngerer Zeit stammen Bemühungen eine Medienphilologie [Balke/Gaderer 2017] zu formulieren, welche die *Inhalte* von Medien aus einer die sprach- und literaturwissenschaftliche Linguistik übersteigenden philologischen Perspektive diskutiert. Computerphilologie stellt dabei zunächst Fragen nach den neuen Anforderungen, die Programmiersprachen für die Editionsphilologie [Hiller 2014], intermediale Translationen von Inhalten/Ästhetiken [Hess-Lüttich/Posner 1990] oder nach der Wirkung auf andere Schreibakte [Vee 2017] und Kommunikationskanäle darstellen. **Mit Programmiersprachen im Allgemeinen (als die ›philologischen Kanäle‹ digitaler Medien) und BASIC im Besonderen setzt sich Medienphilologie nicht auseinander.**

d) Medienwissenschaftliche Forschung

Bisherige Untersuchungen zur ›Homecomputer-Ära‹ leiten Diskurs- und Mentalitätsgeschichten zum BASIC-Programmieren aus zeitgenössischen Publikationen ab [Halvorson 2020:128; Ablinger 2019; Montfort u.a. 2013:169]. Zahlreiche Autor:innen konstatieren eine Interdependenz von Computer-Hardware- und -Software-Märkten mit den Märkten für die zugehörigen

publizistischen Erzeugnisse [Ehrmantraut 2019:138f.]. Diese sei vor allem im Bereich der Computerspiele-Programmierung in BASIC abzulesen [Rankin 2018:98,103]. Die Wechselwirkungen zwischen der Privatisierung der Computertechnik und dem Ruf nach Computer Literacy ab Anfang der 1980er-Jahre wird auch jenseits des »BBC Computer Literacy Project« [Gazzard 2016:4] konstatiert [Vee 2017:4-27; Brock 2019:23].

BASIC-Computerzeitschriften sind integraler Bestandteil kulturwissenschaftlicher Analysen der Homecomputerkultur [Methmacher 2017]. Solche Analysen wurden bislang allerdings nur an einzelnen Zeitschriften [Ablinger 2019] durchgeführt. Hierbei kam zumeist menschliches »Code Reading« [Monfort u.a. 2013:184-186; Spinellis 2003] zum Einsatz, da viele der historischen Publikationen noch gar nicht systematisch erfasst und maschinenlesbar aufbereitet sind. Die Beziehung zwischen Programmiersprachen, dem Archiv an Sourcecodes und den Programmierkulturen untersuchen die Software Studies, Critical Code Studies und (vor allem) die Platform Studies. **Diese konzentrieren sich hierbei vor allem auf den Vektor Mensch-Computer-Interaktion unter Marginalisierung der computertechnischen Abläufe. Letztere müssen für das Verständnis operativer formaler Sprachen jedoch berücksichtigt werden.**

e) Hobbyistische Forschung, graue und dunkelgraue Literatur

Zu BASIC existiert ein großer Korpus an *grauer Literatur*. Dieser könnte eine für die medienwissenschaftliche Analyse der Sprache zentrale Quellengattung darstellen [Halvorson 2020:133], würde er systematisch erschlossen. Derartige Publikation und Archivalien finden sich zumeist in Privatsammlungen, in Museumsdepots und Bibliotheksmagazinen sowie als Scans in elektronischen Internetarchiven oder Code-Repositories. BASIC-Sourcecodes als Texte (Programmlistings) oder in Form von Programmdateien werden für spezifische Plattformen von Hobbyist:innen online gesammelt und sind noch nicht systematisch oder nach wissenschaftlichen (präservatorischen) Kriterien erfasst [Stuermer/Abu-Tayeh 2016; Takhteyev/DuPont 2013]. Museen, wie das *Computerspielemuseum Berlin*, das *Oldenburger Computermuseum* oder Initiativen, wie die *Video Game History Foundation* (gamehistory.org), organisieren Forschungsbibliotheken, in denen auch Sourcecodes gesammelt werden. **Diese berücksichtigen jedoch aufgrund einer bislang noch fehlenden Systematik nur (in Büchern oder Zeitschriften) publizierte Schriften.**

Volatile/dynamische Quellen im Internet oder »dunkelgraue Literatur« (private Listing-Ausdrucke, handgeschriebene Programme, Glossierungen und Annotationen in Büchern und Zeitschriften) sind nicht systematisch erfassbar, aber aufgrund der vergleichsweise jungen BASIC-Programmierkultur oft online oder in Sammlungen und Nachlässen zu finden. Eine Auseinandersetzung mit derartigen Quellen findet bislang nicht statt, verspricht allerdings tiefe exemplarische Einblicke in hobbyistische Programmierpraktiken und -kulturen. Im Zuge der sukzessiven Musealisierung historischer Mikrocomputer-Hard- und -Software (Retrocomputing) gerät die private Programmierung auch in den Fokus neuerlicher Publikationen [Friedland 2019; Montfort u.a. 2013:147-156]). **Diese Entwicklung kann durch den hier vorgestellten Ansatz medienwissenschaftlich adäquat begleitet werden, indem dieser die zumeist kultur- und mentalitätsgeschichtlichen Analysen mit den medientechnischen Dispositiven flankiert.**

f) Forschungslücken

Das geplante Forschungsprojekt greift die genannten Forschungsansätze auf und will dabei folgende Forschungslücken schließen:

- 1 Auf Basis einer **medienwissenschaftlich-kritischen Lektüre der BASIC-Historiografien** soll ein **diskursarchäologischer Ansatz** entwickelt werden, der die Textarchive der BASIC-Programme und BASIC-Literatur in möglichst vielfältigen Erscheinungsformen konsultiert und damit die in der Forschungsliteratur unberücksichtigten technologischen und historischen Brüche sichtbar macht.
- 2 Den bisherigen, überblicksartigen BASIC-Historiografien soll eine **auf einzelne BASIC-Dialekte orientierte mikrohistoriografische Untersuchung** hinzugefügt und gegenübergestellt werden, um sowohl dialektale als auch stilistische Aspekte miteinander vergleichen zu können.
- 3 Zur Darstellung dieser spezifischen Programmierkultur soll eine **Erforschung der grauen und »dunkelgrauen« Literatur zu BASIC** einsetzen, um solche (quasi-informatischen) Quellen für eine kultur- und medienwissenschaftliche Analyse überschaubar und greifbar zu machen. Hierzu wird ein methodisches Instrumentarium zur Erfassung solcher Quellen entwickelt.
- 4 Eine auf BASIC fokussierte **Untersuchung der Oberflächen- und Tiefenstrukturen der Programmiersprache** soll die Sprache als Werkzeug und als Ausdruck spezifischer Programmierkulturen darstellen.
- 5 Exemplarische **Analysen längerer Sourcecodes**, die einen Einblick in die *Stilistik praktizierter Hobbyprogrammierung* geben, sollen sowohl anhand publizierter BASIC-Listings als auch an den in Assemblersprachen programmierten BASIC-Interpretern vorgenommen werden. (Letztere stellen das softwaretechnische Apriori der BASIC-Programmierung dar und müssen deshalb notwendig in die philologische Analyse einbezogen werden.)
- 6 Eine **experimentelle Analyse von BASIC-Dialekten und -Programmen im Vollzug** soll es ermöglichen, den Forschungsdiskurs um empirische Aussagen zur Operativität, Performanz und zu den Beschränkungen von Systemen zu ergänzen. **Operative Analysen**, wie sie in den Platform Studies [Bogost/Montfort 2009] und der »Alien Phenomenology« [Bogost 2012:85-111] vorgeschlagen werden, fehlen für die Untersuchung von BASIC-Interpretern und -Programmen bislang. Solche können nur **in Form computerarchäologischer Re-enactments** stattfinden.

1.2 Projektbezogenes Publikationsverzeichnis Ihrer Arbeiten

1.2.1 Veröffentlichte Arbeiten aus Publikationsorganen mit wissenschaftlicher Qualitätssicherung, Buchveröffentlichungen sowie bereits zur Veröffentlichung angenommene, aber noch nicht veröffentlichte Arbeiten.

- 1 Höltgen, S. (2020a): OPEN HISTORY. Archäologie der frühen Mikrocomputer und ihrer Programmierung (Diss.). Berlin: Humboldt-Universität. (Online: <http://txt3.de/open-history>)
- 2 Höltgen, S. (2020b): Fifty Years in Home Computing. The Digital Computer and its private Use(er)s. In: International Journal of Parallel, Emergent and Distributed Systems, Vol. 35(2), 2020, S. 170-184. DOI: 10.1080/17445760.2019.1597085.
- 3 Höltgen, S. (2020c): Hit any Key to Start Gaming. Programmieren und/als Spielen mit dem Heimcomputer. In: Blankenheim, Jörg (Hg.): Game Designers & Software Artists. Inszenierung und Selbstverständnis der Künstler/innen bei Electronic Arts, 1983-1988. Glückstadt: vwh, S. 61-66.
- 4 Höltgen, S. (2020d): GOTO MOON: Mondlandesimulationen als Computerspiele – Einblick und Eingriff ins Software-Archiv. In: PAIDIA. Zeitschrift für Computerspielforschung. 06.03.2020. (Online: <http://txt3.de/paidia4>)

- 5 Höltgen, S. (2020e): DaimoGraphien. Für ein Argumentieren jenseits des Diskursiven. In: Hiller, M./Höltgen, S. (Hgg.): Archäographien. Aspekte einer radikalen Medienarchäologie. Berlin: Schwabe, S. 295-307.
- 6 Höltgen, S. (2019): BASIC für Medienwissenschaftler. In: Höltgen, S. (Hg.): Medientechnisches Wissen, Band 2: Informatik, Programmieren, Kybernetik. Berlin/Boston: DeGruyter, S. 169-203.
- 7 Höltgen, S. (2018): Computer/Sprachen: ELIZA und BASIC – Urszenen des Homecomputings (und) künstlicher Intelligenz. In: Höltgen, S./Baranovska, M. (Hgg.): Hello, I'm Eliza. Fünfzig Jahre Gespräche mit Computern. Bochum: Projektverlag, S. 97-122.
- 8 Höltgen, S./van Treeck, J.-C. (2016): Time To Play. Zeit und Computerspiel. Glückstadt: vvh.
- 9 Höltgen, S. (2016a): GO BACK GOTO. Ein kurzer Überblick über die Entfernung der Schulinformatik von den Computern. In: Grundlagenstudien aus Kybernetik und Geisteswissenschaften, Band 57, Heft 3 (September 2016), S. 141–152.
- 10 Höltgen, S. (2016b): How to Do Things with Keys. (Assembly) Programming as (a Kind of) Gesture. In: Media Arts Performance #7: On Gestures (2016). (Online: <http://txt3.de/things-keys>)

[...]

2 Ziele und Arbeitsprogramm

2.1 Voraussichtliche Gesamtdauer des Projekts

Die Projektdauer ist auf 36 Monate angelegt. Innerhalb dieser Zeit sollen die nachgenannten Forschungsziele erreicht und in einer Monografie, in Vorträgen sowie einem peer-reviewten Zeitschriftenbeitrag gesichert werden. Das Projekt wird mit einer Ausstellung operativer Homecomputer mit den untersuchten BASIC-Spielen im *Heinz-Nixdorf-MuseumsForum (Paderborn)* beendet.

2.2 Ziele

Das Forschungsprojekt verfolgt zum einen eine **medienhistorische Agenda**: Es soll die Frage nach der Möglichkeit einer Geschichte der Programmiersprachen, exemplifiziert am Beispiel von BASIC, beantwortet werden. Hierbei wird berücksichtigt, dass die Betrachtung einer *Programmiersprache als medienkulturelles Objekt* stets deren technische, linguistische, praktische und kulturelle Kontexte berücksichtigen muss. Als besondere Mediengattung residieren Programmiersprachen in unterschiedlichen Mediensubstraten/-gattungen, was die Quellenforschung des Projektes berücksichtigen soll. Die **medientheoretische Agenda** zum anderen wird auf Basis einschlägiger Forschungsprogramme (Software/Platform/Critical Code Studies, Computer- und Medienarchäologie) in Verbindung mit philologischen Methoden *die neue Theorie der Computerphilologie* entwerfen. Diese wird sich zur Analyse neuer medienwissenschaftlicher Quellengattungen methodologisch zwischen Methoden des Code Reading und neueren Medienphilologien situieren. Zur Verwirklichung dieser Agenden sind **fünf Teilziele** zu erreichen, von denen die ersten drei die *Primärziele* und die nachfolgenden zwei die *Sekundärziele* darstellen:

2.2.1 Primärziele

1. Darstellung hobbyistischer BASIC-Programmierkultur als Konvergenzphänomen der Wechselwirkung von Computerplattformen, BASIC-Dialekten, Verwendungsweisen und Programmier-techniken unter Berücksichtigung der spezifischen Quellen und Kommunikationsmedien.
2. Theoretische Erschließung und Systematisierung neuer Quellen der medien- und kulturwissenschaftlichen Computerkultur-Forschung (Handbücher, Print-Outs, BASIC-Listings in Zeitschriften, Scribbles, ...).

3. Weiterentwicklung der in [Höltgen 2020a] skizzierten Theorie und Methode der Computerphilologie auf Basis der analysierten Artefakte.

2.2.2 Sekundärziele

4. Entwurf einer *Soziolinguistik* und Taxonomie der BASIC-Dialekte und ihrer regionalen Proliferation auf Basis ihrer Syntax und Lexik.

5. Herleitung einer *Pragmatik und Stilistik* der Verwendungsweisen der ausgewählten BASIC-Dialekte (Plattform-/Dialekt-abhängig und Nutzergruppen-spezifisch) am Beispiel der Computer-spielprogrammierung.

2.2.3 Publikationen

Die Forschungsergebnisse des Projektes sollen in folgenden Formen veröffentlicht werden:

- 1 eine deutschsprachigen Monografie mit dem Umfang von circa 500 Seiten. Der Kapitel-aufbau geht aus der Tabelle 1 hervor,
- 2 ein englischsprachiger Beitrag in einer peer-reviewten Fachzeitschrift,
- 3 ein Konferenz-Vortrag,
- 4 eine abschließende Ausstellung.

Kapitel	Titel	Seiten
	Inhaltsverzeichnis, Vorwort, etc.	10
1	Einleitung	10
2	Methodenüberblick	20
3	Medientheoretische Dekonstruktion der BASIC-Geschichte	20
4	Die BASIC-Programmiersprache(n): Archäologie ausgewählter Dialekte anhand ihrer Oberflächen- und Tiefenstrukturen	100
5	Die BASIC-Programmierkultur(en): Medien, Diskurse und Kulturtechniken (an ausgewählten BASIC-Computerspielen)	100
6	Programmiersprachen und ihre Programmierung – eine computerphilologische Medientheorie	100
7	Zusammenfassung und Fazit	20
8	Anhang mit Sourcecodes	50
9	Anhang mit Abbildungen der Experimente	15
	Literaturverzeichnis, Schlagwortverzeichnis etc.	40

Tabelle1: Kapitelaufbau der geplanten Forschungspublikation

Um der Medienspezifität im Sinne der Computer- und Medienarchäologie gerecht zu werden, muss die **operative Analyse der Sourcecodes** auch auf den Originalplattformen stattfinden. Diese stellt das *Oldenburger Computermuseum* für das Projekt zur Verfügung. Zusätzlich zur schriftlich-diskursiven Sicherung der Forschungsergebnisse muss eine **operative Präsentation der Forschungsergebnisse** stattfinden, in der die Plattformen, BASIC-Dialekte und BASIC-Programme ›vergegenwärtigt‹ werden. Diese Ausstellung soll im *Heinz-Nixdorf-MuseumsForum* stattfinden und fotografisch und filmisch fixiert werden. Darüber hinaus sollen die **Sourcecodes und** (in Emulatoren oder auf Originalplattformen) **ausführbaren Objektcodes für das Re-enactment zur Verfügung gestellt** werden. Zu diesem Zweck wird am gastgebenden Institut

ein eigener Webserver für das Projekt aufgesetzt. Auf diesem können auch die Ergebnisse der Konferenz dokumentiert werden.

[...]

2.3.3 Theoretische Grundlagen

Die theoretische und methodologische Grundlage des geplanten Forschungsprojektes hat der Antragsteller in seiner Dissertation [Höltgen 2020a] als **Computerarchäologie** entworfen. Im Verlauf jener Forschungsarbeit haben sich die Bedeutung der (autodidaktischen) Hobbyprogrammierung sowie die Relevanz der Programmiersprache BASIC als zwei zentrale Aspekte der Computerkultur der 1970er- bis 1990er-Jahre offenbart. Die in diesem Rahmen untersuchten BASIC-Programme (z.B. [ebd.:87-92]) haben bereits die Wechselwirkungen zwischen BASIC-Dialekt, Computerplattform und Programmieretechniken angedeutet. Dies konnte allerdings ausschließlich am operativen System selbst nachvollzogen und demonstriert werden. Dabei stellten sich die *Demonstration* [ebd.:51,57; Höltgen 2020e] und das *Re-enactment* [ebd.:57f; Höltgen 2019] als zwei hinreichende Methoden medienwissenschaftlicher Forschung zum Computing heraus.

Insbesondere konnte die Beschäftigung mit operativen historischen Computern und darauf implementierten historischen wie aktuellen Programmen grundlegende Probleme der kanonischen Computergeschichtsschreibung offenbaren: Weder vermag eine allein am (chronologischen, ökonomischen, technischen, wissenschaftlichen) Fortschritt orientierte Historiografie die Brüche, Rekursionen und Anachronismen der Computergeschichte(n) angemessen zu erfassen; noch scheinen sich operative Computer überhaupt historiografisch ›fixieren‹ zu lassen. Vielmehr müssen sowohl deren medientechnische »Eigenzeitlichkeit« [Ernst 2013:15] als auch die Idiosynkrasien des jeweils konkreten operativen Computers Eingang in ihre Analyse und Beschreibung finden [Höltgen 2020a:34,139-141].

In einer solchen ›objektorientierten‹ Betrachtungsweise zeigen sich Computer als »aliens« [Bogost 2012], die als Archive und Agenten ihrer eigenen indexikalischen, symbolischen und ikonischen Zeichensysteme fungieren [Ernst 2002:143] und deren kultureller ›Sinn‹ sich erst in der operativen Verschränkung der Hardware-/Software-Plattformen mit ihren vielfältigen interaktiven Beziehungen zu Nutzer:innen (Programmierer:innen, Leser:innen, Debugger:innen, User:innen, Zuschauer:innen, ...) ergibt. Die aus dieser Problemlage entstandene Theorie und Methodologie soll nun im beantragten Forschungsprojekt auf die Untersuchung von BASIC angewendet und erweitert werden. Hierzu soll auf einen vielfältigen interdisziplinären ›Mix‹ von Methoden zurückgegriffen werden, der es ermöglicht, BASIC-Dialekte, -Programme und -Kulturen gegenstandsadäquat und in ihrer Beziehung zueinander darzustellen.

Im Zentrum stehen dabei BASIC-Dialekte, die in Mikrocomputern, welche zwischen 1974 und 1991 veröffentlicht wurden, *fest installiert* waren. Diese Dialekte stehen in enger technischer Beziehung zur Hardware und haben (aufgrund der Käuferzielgruppe der Computerplattformen) vor allem private und hobbyistische Programmierer:innen hervorgebracht. An diese richteten sich die als Quellenmaterial zugrunde liegenden Zeitschriften, Sekundärliteraturen und graue Literaturen; aus ihnen gehen die Quellen der ›dunkelgrauen Literatur‹ hervor. Die ab Mitte der 1980er- bis Anfang der 1990er-Jahre publizierten strukturierten BASIC-Dialekte besitzen diese enge Plattformbezogenheit und die damit einhergehenden hobbyistische Nutzer:innen-Bindung nicht mehr.

2.3.4 Methoden

Um den Forschungsgegenstand adäquat sowohl als medien- und kulturwissenschaftliches als auch als philologisches und informatisches Objekt zu fassen, müssen diejenigen Methoden dieser Arbeitsfelder zu einer Schnittmenge vereinigt werden, die sich bislang mit der Analyse von Programmsourcecodes befasst haben. Das im Folgenden dargestellte, plurimethodische Set bildet diese Schnittmenge.

2.3.4.1 Informatische Methoden

Die Methoden des **Reverse Engineerings** von Software [Eilam 2005], etwa durch das **Disassemblieren und Dekompilieren** von Objektcodes, bilden die methodische Grundlage zur Quellenanalyse. Insbesondere die grundsätzliche **Bauweise von BASIC-Interpretern** [Gauthier/Ponto 1970; Klein 1982] als strukturelle Grundlage der Programmiersprachensyntax und der »synchronen« Code-Übersetzung kann hiermit nachvollzogen werden. Ähnlichkeiten verschiedener Dialekte in Hinblick auf einzelne Algorithmen ermöglichen Aussagen über die Verwandtschaftsbeziehungen von Interpretern/Dialekten. Probleme, die sich aus der »diachronen« Übersetzung durch **Versioning/Updates oder systemübergreifende Emulation von BASIC-Interpretern** ergeben, müssen in Hinblick auf ihre Übersetzungsweisen [Loebel 2014; Höltgen 2020a:179-186] betrachtet werden. Informatische Untersuchungen (unklarer) **Autorschaft einzelner Algorithmen** [Burrows 2020] werden durch philologische Methoden der Intertextualitäts- und Rezeptionsanalyse (s.u.) untersucht. Verfahren der Code-Auswertung/des **Code Reading** bilden die methodische Grundlage der BASIC-Sourcecode-Analysen [Spinellis 2003; Brock 2019]. Darüber hinaus sollen Erkenntnisse aus der **Code-Leseforschung** [Busjahn/Schulte 2011] Einblicke in die historischen hobbyistischen Programmier- und Programmlisting-Abtätigkeiten ermöglichen.

2.3.4.2 Philologische Methoden

Für das Projekt dienen die bestehenden computer- und medienphilologischen Diskurse als theoretischer Hintergrund, der durch ein Toolkit an literatur- und sprachwissenschaftlichen Methoden, auf die **Textsorte Sourcecode** übertragen, ergänzt werden soll. Hierfür spielen **sprachwissenschaftliche Aspekte** der Semiotik (Kontrollstrukturen) [Zemanek 1966], Textlinguistik (Anaphorik, Deixis von Programmmodulen), der Morphologie (von Befehls-Argument-Verbindungen) [Kurtz 1967:15; Marcus 1997:237-239], Stilistik [Burrows 2010] in **synchronischer und diachronischer Perspektive** eine Rolle. Hinzu kommen **literaturwissenschaftliche Methoden** der hermeneutischen Code-Interpretation [Marino 2006:4], der Intertextualitätsanalyse [Holtius 1993] sowie der Rezeptionstheorie [Warning 1979].

Während die sprachwissenschaftlichen Kategorien die spezifische **Stilistik** einzelner BASIC-Programme im Close Reading analysierbar machen sollen, werden mithilfe der literaturwissenschaftlichen Methoden die Adaptionen und Beziehungen als **Rezeption** verschiedener BASIC-Programme gedeutet und klassifiziert. Methodologische Vorarbeiten hierzu existieren bereits in [Höltgen 2020a:61-136]. Schon zeitgenössische Autor:innen haben sich der **Taxonomie** vor der Frage nach Ähnlichkeit und »Übersetzbarkeit« der verschiedenen BASIC-Dialekte angenommen. So enthält z. B. [Luther 1984] Hinweise zu einer **Dialektologie von BASIC** und schlägt synoptische **Vergleichsmethoden der Oberflächenstruktur** von BASIC-Programmen vor. Darüber hinaus geben die Standardisierungsbemühungen [Kurtz/Garland 1971] und das Entstehen plattformübergreifender BASIC-Dialekte [Wiegand u.a. 1984]) Hinweise auf eine **formalsprachliche Kommunikation unter den BASIC-Programmierer:innen**, die in Zeitschriften und Büchern bei BASIC-Spiele-Sourcecodes berücksichtigt wurde [Höltgen 2020d].

Mithilfe **soziolinguistischer** Methoden [Naz/Rice 2015] lassen sich Dialekte regional und historisch eingrenzen und in ihren Gemeinsamkeiten, Unterschieden und Einflüssen aufeinander darstellen. Diese Methoden sollen auf die Programmiersprache BASIC und ausgewählte Dialekte übertragen werden, um die Charakteristika spezifischer Programmierkulturen auch auf der formalsprachlichen Seite darzustellen.

2.3.4.3 Medien- und kulturwissenschaftliche Methoden

a. Medien- und Computerarchäologie

Medienarchäologie lenkt den Fokus von den Medieninhalten auf die medientechnischen Aprioris, welche diese Inhalte formatieren. Die Analyse derselben benötigt andere als hermeneutische Methoden: Die Eigenzeitlichkeit von Medientechnologie und die Operativität von Medienprozessen erfordern eine Notation in »nonhistorischer und nonhistoriographischer Weise« [Ernst 2013:13], die als **Archäographien** [Hiller/Höltgen 2020:10f.] sowohl die nicht-diskursivierbaren technischen Prozesse darstellbar macht, als auch im *Re-enactment* und der *Demonstration* die Eigenzeitlichkeit von Medienprozessen berücksichtigt. Die Methode des **Re-enactment** ermöglicht eine Untersuchung und Darstellung historischer Artefakte als ›Nachvollzug‹ in der Gegenwart und supplementiert die Analyse damit um performative und prozesshafte Elemente. Die **Demonstration** führt die technischen, vor allem zeitbasierten und zeitkritischen Eigenschaften operativer Medienartefakte vor. Für eine solche Untersuchung sollte daher historischen Computersystemen vor Emulatoren derselben der Vorzug gegeben werden. Die Artefakte der Programmiersprache BASIC, liegen sowohl in Form der Textsorte ›Quellcode‹ (Interpreter-Sourcecodes, BASIC-Programmlistings) als auch als ausführbare Programmdateien vor. Ersterer sollten nicht nur, letztere können nicht von Menschen, sondern (auch) von Maschinen ›gelesen‹ werden. Denn erst die **Implementierung der Programme in Computer** zeitigt spezifische Effekte, die archäographisch gesichert und präsentiert werden können. Die Computer bilden das technische Dispositiv, das zusammen und wechselseitig mit den es begleitenden Paratexten zur Programmierung die spezifische Programmierkultur, die das Projekt untersucht, hervorbringt und prägt.

b. Platform Studies und Carpentry

Die **Platform Studies** versammeln verschiedene Tools, mit denen Computer-Systeme, die kultureller Produktion unterliegen, untersucht werden können. Hierzu zählen neben Hardware und Software auch Programmiersprachen – explizit auch BASIC-Dialekte und -Programme: »Programming languages such as BASIC can be thought of as platforms.« [Bogost/Montfort 2009:3]. Die Methode setzt auf verschiedenen »levels« von Plattformen an und perspektiviert »computation (at a fundamental level) with culture and creativity« [ebd.:4], um die Verbindung zwischen spezifischen Technologien und den Kulturen, aus denen sie hervorgegangen sind und in denen sie genutzt werden, »technically rigorous« [ebd.:9] zu analysieren. Hierbei gehen **Hardware-, Programmiersprachen- und Programmanalysen** ineinander über. **Im Projekt sollen diese technologischen Verbände als Plattformen analysiert werden.**

BASIC-Interpreter, die als ›Übersetzer‹ zwischen den ästhetischen Vorstellungen der BASIC-Spiel-Programmierer:innen und den Maschinen(sprachen) der Hardware-Level stehen, sieht Bogost »as actors« [Bogosts 2012:100], die mit Methoden der »**Carpentry**« [ebd.:85-111] in ihrer »alien presence« [ebd.:106] verstanden werden können. Hierzu sollen Praktiken, wie das **Abtippen von BASIC-Programmen** [ebd.:93] Theorie und Praxis miteinander verbinden: »carpentry entails making things that explain how things make their world. Like scientific experiments and engineering prototypes, the stuffs produced by carpentry are [...] themselves earnest

entries into philosophical discourse.« [ebd.:93] Diese **Praxis als Forschungsmethode bildet die Basis für die operative Analyse** des Forschungsgegenstandes.

c. Critical Code Studies

Zu Beginn des Personal-Computer-Zeitalters, als BASIC von Millionen Menschen [Halvorson 2020:159] programmiert wurde, waren die Grenzen zwischen dem, was die Menschen taten, zu dem, was die Computer taten, durchlässig: Formale Sprachen wurden gelesen, geschrieben, gesammelt und gedacht. Die **Critical Code Studies** setzen an diese Praktiken eine kritische Hermeneutik an, bei der **Code wie Literatur gelesen und erklärt** wird, um »computer code, program architecture, and documentation« [Marino 2006:4] in ihrem sozialhistorischen Kontext zu verstehen. Die sich aus einem solchen Kontext entwickelnde »Software Culture« [Fuller 2018:256] (bzw. »Programming Culture«) hat auch für die BASIC-Programmierung im fokussierten Zeitraum spezifische Techniken und Praktiken hervorgebracht.

Die Praxen des BASIC-Programmierens und des Computerspielens fallen in dieser Programmierkultur zusammen: Erst durch den Dialog-Betrieb des BASIC-Interpreters kann Programmieren zum interaktiven, spielerischen Experimentieren mit den Möglichkeiten von Hardware und Sprache werden. Diese Verschränkung ist produktiv [miss]verstehbar als »Gamification« und »eLearning« [Höltgen 2020a:248-253] und erklärt den Reiz des hobbyistischen Programmierens. Ebenso kann auch das Abtippen von BASIC-Listings und das Testen und Debuggen der Programme gedeutet werden, denn hierbei findet bereits eine mentale Antizipationen der Codeausführung statt [Busjahn/Schulte 2011:7]. Die damit verbundenen Praktiken des Entwurfs, Denkens, Abtippens und Tauschens von Code zwischen Computernutzer:innen sollen im zu untersuchenden Zeitraum nun auch für private Computernutzung/-programmierung [Petricek 2019] analysiert werden. Insbesondere durch die Lektüre annotierter BASIC-Listings und handschriftlicher notierter Programme als »dunkelgraue Quellen« soll eine **Methode für die Beschreibung einer historischen Computerkultur** erbringen.

2.3.5 Ausblick

Die Erkenntnisse über BASIC, seine philologischen Aspekte und Programmierkulturen lassen sich als **Softwaredialektforschung** auch auf andere Programmiersprachen und -kulturen übertragen. Hier können in einer Fortsetzung des Fortsetzungsprojektes vor allem jüngere semiprofessionelle und Amateur-Programmierkulturen untersucht werden, die im Privaten, in Online- oder Community-Umgebungen (Hackerspaces, Programming-Dojos, ...) praktizieren.

3 Literaturverzeichnis zum Stand der Forschung, zu den Zielen und dem Arbeitsprogramm

- Ablinger, F. (2019): Homecomputer. Zur Technik- und frühen Computerspielkultur anhand einer Zeitschrift der Jahre 1983 und 1984. (Reihe: Game Studies). Glückstadt: vwh.
- Balke, F./Gaderer, R. (2017): Medienphilologie. Konturen eines Paradigmas. Göttingen: Wallstein.
- Berger, U. (2011): uBasic - eine kleine plattformunabhängige Basic-Interpreter-Bibliothek. In: Chemnitzer Linux-Tage 2011, 19. und 20. März 2011, Tagungsband. Chemnitz: Universitätsverlag Chemnitz, S. 103-111.
- Bergin, T. J./Gibson, R. G. (Hgg.) (1996): History of Programming Languages II. Reading (Mass.) u.a.: Addison-Wesley.
- Bogost, I. (2012): Alien Phenomenology. Or What's Like To Be A Thing. Minneapolis/London: University of Minneapolis Press.
- Bogost, I./Montfort, N. (2009): Platform Studies: Frequently Questioned Answers. In: <http://txt3.de/basic1> (05.01.2021).
- Brock, K. (2019): Rhetorical {Code} Studies. Discovering Arguments in and around Code. Ann Arbor: University of Michigan Press.

- Burrows, S. D. (2010): Source Code Authorship Attribution. (PhD Thesis). Melbourne: RMIT University, <http://txt3.de/basic2> (04.01.2021).
- Busjahn, T./Schulte, C. (2011): The Use of Code Reading in Teaching Programming. In: Proceedings of the 13th Koli Calling International Conference on Computing Education Research, S. 3-11.
- Dahl, O.-J./Dijkstra, E. W./Hoare, C. A. R. (1972): Structured Programming. London: Academic Press 1972.
- Ehrmanntraut, S. (2019): Wie Computer heimisch wurden. Zur Diskursgeschichte des Personal Computers. (Reihe: Edition Medienwissenschaft). Bielefeld: Transcript.
- Eilam, E. (2005): Reversing. Secrets of Reverse Engineering. Indianapolis: Wiley.
- Ernst, W. (2002): Archive im Übergang. In: v. Bismarck, B. u.a. (Hgg. für den Kunstraums der Universität Lüneburg): Interarchive. Archivarische Praktiken und Handlungsräume im zeitgenössischen Kunstfeld, Köln: Walter König, S. 137-146.
- Ernst, W. (2013): Signale aus der Vergangenheit. Eine kleine Geschichtskritik. München: Fink.
- FEoLL (1976): Arbeitskreis zum Design einer Programmiersprache für die Schule – Empfehlungen, Ergebnisse, Abschlussbericht und Stellungnahme zu BASIC. Paderborn: FEoLL.
- Friedland, G. (2019): Beginning Programming Using Retro Computing. Learn BASIC with a Commodore Emulator. O. O.: Apress.
- Fuller, M. (2018). Software Studies Methods. In: Sayers, J. (Hg.): The Routledge Companion to Media Studies and Digital Humanities. New York: Routledge, Taylor & Francis Group, S. 250-257.
- Gauthier, R. L./Ponto, S. D. (1970): Designing Systems Programs. Englewood Cliffs (New Jersey): Prentice-Hall.
- Gazzard, A. (2016): Now the Chips are Down. The BBC Micro. Reihe: Platform Studies. Cambridge/London: MIT Press.
- Halvorson, M. (2020): Code Nation: Personal Computing and the Learn to Program Movement in America. New York: ACM Books.
- Hess-Lüttich, E. W. B./Posner, R. (Hgg.) (1990): Code-Wechsel. Texte im Medienvergleich. Wiesbaden: Springer.
- Hiller, M. (2014): DISKURS/SIGNAL (II). Prolegomena zu einer Philologie digitaler Quelltexte. In: editio. Internationales Jahrbuch für Editionswissenschaft, Bd. 28. Berlin/Boston: DeGruyter, S. 193-212.
- Hiller, M./Höltgen, S. (Hgg.) (2020): Archäographien. Aspekte einer radikalen Medienarchäologie. Berlin: Schwabe.
- Höltgen, S. (2019): ArtWare. Re-enacting Cybernetic Art. In: Media Studies at Greenwich Blog, <http://txt3.de/basic4> und <http://txt3.de/basic3> (05.01.2021).
- Höltgen, S. (2020a): OPEN HISTORY. Archäologie der frühen Mikrocomputer und ihrer Programmierung (Diss.). Berlin: Humboldt-Universität. (Online: <http://txt3.de/open-history>)
- Höltgen, S. (2020d): GOTO MOON: Mondlandesimulationen als Computerspiele – Einblick und Eingriff ins Software-Archiv. In: PAIDIA. Zeitschrift für Computerspielforschung. 06.03.2020. (Online: <http://txt3.de/paidia4>)
- Höltgen, S. (2020e): DaimoGraphien. Für ein Argumentieren jenseits des Diskursiven. In: Ders./Hiller, M. (Hgg.): Archäographien. Aspekte einer radikalen Medienarchäologie. Berlin: Schwabe, S. 295-307.
- Holtius, S. (1993): Intertextualität. Aspekte einer rezeptionsorientierten Konzeption. Tübingen: Stauffenberg.
- Keidel, K./Müller, H. J. (1983): Informatik Einführung. Ein Lehr- und Arbeitsbuch. München: Bayerischer Schulbuch-Verlag.
- Kemeny, J. G. (1984): Oral history interview with John G. Kemeny, 1984 June 7. OCLC-Nr. 966255686, <http://txt3.de/basic5> (04.01.2021).
- Kemeny, J. G./Kurtz, T. E. (1967): The Dartmouth Time-Sharing Computing System. Final Report, June 1967. <http://txt3.de/basic6> (04.01.2021).
- Kemeny, J. G./Kurtz, T. E. (1985): Back to BASIC. The History, Corruption, and Future of the Language. Reading u.a.: Addison-Wesley.
- Kemeny, J. G./Kurtz, T. E. (1987): Structured BASIC Programming. New York u.a.: John Wiley & Sons.
- Klein, R.-D. (1982): Basic-Interpreter. Funktionsweise und Implementierung in 8080/Z-80-Computern. München: Franzis'.
- Kurtz, T. E. (1969): The many roles of computing on the campus. In: AFIPS '69 (Spring): Proceedings of the May 14-16, 1969, spring joint computer conference, May 1969, S. 649–656.
- Kurtz, T. E. (1981): BASIC Session. In: Wexelblat, R. L. (Hg.): History of Programming Languages. New York u.a.: Academic Press, S. 515-550.

- Kurtz, T. E. (2002): Oral history interview with Thomas E. Kurtz. Dartmouth College Library, item 1402, <http://txt3.de/basic7> (04.01.2021).
- Kurtz, T. E. (2009): BASIC. In: Biancuzzi, F./Warden, S. (Hgg.): Visionäre der Programmierung. Die Sprachen und ihre Schöpfer. Köln u.a.: O'Reilly, S. 91-102.
- Kurtz, T. E./Garland, S. J. (1971): Towards Standardization of BASIC. In: Interface. Vol. 5, No. 5 (October 1971), S. 221f.
- Lee, J. A. N. (1971): The formal definition of the BASIC language. In: The Computer Journal, Vol. 15, No. 1, S. 37-41.
- Loebel, J.-M. (2014): Lost in Translation. Leistungsfähigkeit, Einsatz und Grenzen von Emulatoren bei der Langzeitbewahrung digitaler multimedialer Objekte am Beispiel von Computerspielen. Glückstadt: vwh.
- Lorenzo, M. J. (2017): Endless Loop. The History of the BASIC Programming Language. Philadelphia/Pittsburgh: SE Books.
- Luther, W.-D. (1984): Die große BASIC Referenztabelle der 51 Dialekte. Sprendlingen: Luther.
- Marcus, S. (1997): Contextual Grammars and Natural Languages. In: Rozenberg, G./Salomaa, A. (Hgg.): Handbook of Formal Languages. Vol. 2: Linear Modeling: Background and Application. Berlin/Heidelberg: Springer, S. 215-136.
- Marino, M. C. (2006): Critical Code Studies. In: Electronic Book Review, 12.04.2006, <http://txt3.de/basic8> (05.01.2021).
- Methmacher, M. (2017): Das Papier der digitalen Welt. Computerzeitschriften als »Akteure« im Netzwerk von (jugendlichen) Nutzern, Hardware und Software 1980-1995. (Dissertation, RWTH Aachen.)
- Misek-Falkoff, L. D. (1982): The New Field of »Software Linguistics«: An Early-Bird View. In: ACM SIGMETRICS Workshop on Software Metrics (1982), S. 35-50.
- Montfort, N./Baudoin, P./Bell, J./Bogost, I./Douglass, J./Marino, M. C./Mateas, M./Reas, C./Sample, M./Vawter, N. (2013): 10 PRINT CHR\$(205.5+RND(1)); : GOTO 10. (Reihe: Software Studies). Cambridge/London: MIT Press.
- Naz, F./Rice, J. E. (2015): Sociolinguistics and Programming. In: 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, 2015, S. 74-79.
- NDR (1984): Einführung in die Mikroelektronik. 26 Folgen. Hierzu: <http://txt3.de/ndrkc> (04.01.2021).
- Petricek, T. (2019): Cultures of Programming. Understanding the History of Programming Through Controversies and Technical Artifacts. In: <http://txt3.de/basic9> (01.01.2020)
- Rankin, J. L. (2018): A People's History of Computing in the United States. Cambridge/London: Harvard Press.
- Rice, J. E./Genee, I./Naz, F. (2014): Linking Linguistics and Programming: How to start? (Work in Progress). In: Psychology of Programming Interest Group, <http://txt3.de/basic10> (05.01.2021).
- Sammet, J. E. (1969): Programming Languages. History and Fundamentals. Englewood-Cliffs: Prentice-Hall.
- Spinellis, D. (2003): Code Reading. The Open Source Perspective. Boston u.a.: Addison-Wesley.
- Stuermer, M./Abu-Tayeh, G. (2016): Digital Preservation through Digital Sustainability. In: Swiss National Library (Hg.): iPRES 2016. Proceedings of the 13th International Conference on Digital Preservation. Bern: N. N., S. 18-21.
- Takhteyev, Y./DuPont, Q. (2013): Retrocomputing as Preservation and Remix. In: iConference 2013 Proceedings, S. 422-432.
- Vee, A. (2017): Coding Literacy. How Computer Programming Is Changing Writing. Cambridge/London: MIT Press.
- Warning, R. (1979): Rezeptionsästhetik als literaturwissenschaftliche Pragmatik. In: Ders. (Hg.): Rezeptionsästhetik. Theorie und Praxis. München: Fink (UTB), S. 9-41.
- Wexelblat, R. L. (Hg.) (1981): History of Programming Languages. New York u.a.: Academic Press.
- Wiegand, M./Fillinger, M. und H. (1984): Basicode. Die kompatible Computersprache. 14 Programme für über 25 Homecomputer. Ravensburg: Otto Maier Verlag.
- Wulf, W. A. (1972): A case against the GOTO. In: Hopkins, M. E. (1972): A Case for the GOTO. In: ACM '72: Proceedings of the ACM annual conference, Volume 2 (August 1972), S. 791-797.
- Zemanek, H. (1966): Semiotics and Programming Languages. In: Communications of the ACM, Vol. 9, No. 3 (March 1966), S. 139-143.

5.2 Umgang mit den im Projekt erzielten Forschungsdaten

Die Forschungsdaten werden in Form von schriftlichen Veröffentlichungen, Vorträgen (mit Zitiation von Forschungsdaten) und auf einer für das Forschungsprojekt einzurichtenden Internetseite veröffentlicht.

Die erzielten/gefundenen Forschungsdaten, soweit sie nicht bereits öffentlich zugänglich sind, sind:

1. Von Privatleuten erstellte Programmcodes (gedruckt und handschriftlich)
2. Von Privatleuten erstellte BASIC-Programme (ausführbare Dateien)
3. Fotos, Scans und Screenshot von (1) und (2)
4. Fotos und Filme der operativen Ausstellung am Projekt-Ende

(1) und (3) werden im Rahmen der Vorträge und Publikationen des Antragsstellers veröffentlicht. Dies erfolgt nach Rückfrage und mit schriftlicher Erlaubnis der Rechteinhaber:innen oder/und auf Wunsch vollständig anonymisiert.

(2) und (4) werden für die Veröffentlichung im Rahmen einer Ausstellung sowie auf einer projektbezogenen Internetseite veröffentlicht. Die Veröffentlichung von (2) findet ausschließlich mit ausdrücklicher Genehmigung der Rechteinhaber:innen und auf Wunsch anonymisiert statt; dies gilt auch für die Veröffentlichung von (4), wobei bei der Ausstellungsdokumentation entstehende Fotografien und Bilder keine Abbildung von Personen enthalten werden.

5.3 Weitere Angaben

Da die operative Präsentation der Forschungsdaten und -themen ein integraler Bestandteil der Forschungsdokumentation des Projektes ist, soll das Projekt mit einer operativen Ausstellung (auf Basis der o.g. Computer) der Forschungsergebnisse enden. Hierzu fanden bereits Gespräche mit dem *Heinz-Nixdorf-MuseumsForum* statt [s. Letter of Intent]. Das Depot des *Oldenburger Computermuseums* enthält zahlreiche operative Homecomputerplattformen, die für die praktische Forschungstätigkeit genutzt werden können [s. Letter of Intent].

Das Heimatinstitut stellt zudem die technische Infrastruktur (Webpace) für die Online-Präsentation der Forschungsdaten und -themen sowie der operativen Ausstellung (Online-Emulation) der Forschungsobjekte (BASIC-Programme) zur Verfügung.

Archivrecherchen finden an deutschen und internationalen Computermuseen und privaten -sammlungen statt. Die Kurator:innen und Sammler:innen sind dem Antragsteller bekannt und haben ihre Zustimmung zur Unterstützung erklärt.

[...]