

## QUELLTEXTE IN NETZLITERATUR AUS ARCHIVARISCHER UND LITERATUR- WISSENSCHAFTLICHER PERSPEKTIVE

Wer mit einem derzeit gängigen Browser wie Firefox, Chrome oder Safari eine Webseite besucht, findet im Kontextmenü des Mauszeigers die Funktion »Seitenquelltext anzeigen«. Dieser Quelltext wird vom Server zum Browser übertragen, damit die Seite dort angezeigt werden kann. Es gibt keine Webseite ohne solchen Quelltext. Leser:innen können – mit der genannten Funktion – auf den Quelltext zugreifen und so die Funktionsweise einer Seite nachvollziehen. Für die archivarische Erhaltung einer Seite können die Quelltexte zusammen mit zusätzlichen Ressourcen der Seite, zum Beispiel Bilder, gespeichert werden. Neben Quelltexten, die auf diese Weise gelesen und gespeichert werden können, gibt es auch Quelltexte, die aus Sicht der Leser:innen im Verborgenen bleiben, weil sie beim Aufrufen einer Webseite nicht an den Browser übertragen, sondern nur auf dem Webserver ausgeführt werden. Solche Quelltexte sind nicht direkt zugänglich und können deshalb für die Erhaltung oder die Analyse einer Webseite nicht einfach gelesen oder gespeichert werden. Der fehlende Zugang zu diesen entfernten Quelltexten stellt die Erhaltung der Objekte vor Herausforderungen, weil hier wegen des fehlenden Direktzugriffs keine automatisierten Speicherverfahren eingesetzt werden können. Für den Fall, dass der serverseitige Quelltext gesichert werden kann, kommen noch dessen je spezifische Abhängigkeiten zu einer ausführenden Softwareumgebung hinzu, weil die Ausführungsumgebung ebenfalls gesichert werden muss, wenn der Quelltext mit seiner Funktionalität erhalten werden soll. Für Werke der Netzliteratur ist eine solche umfangreiche Erhaltung in der Regel wünschenswert. Denn ohne die Archivierung eines ausführbaren Quelltexts droht das Werk zu verschwinden.

Um die Bedeutung des Quelltextes für ein einzelnes Werk zu bestimmen, ist eine Funktionsanalyse erforderlich, die die Komponenten und die Grenzen des Objekts identifiziert.<sup>1</sup> Anhand von zwei Netzinstallationen, Kathrin

1 Dragan Espenschied und Klaus Rechart: 207.1 Fencing Apparently Infinite Objects, in: Open Science Framework (OSF), 2022, DOI: 10.17605/OSF.IO/6F2NM (<https://osf.io/6f2nm/>, Zugriff: 20. Januar 2023).

Passigs Twitterbot *Gomringador* (2018–) und Johannes Auers Webseite *free lutz!* (2005–), werden einige Problemstellungen herausgearbeitet, die sich aus der besonderen textuellen Beschaffenheit von Netzliteratur (und weiterführend von digitaler Literatur) ergeben.<sup>2</sup> Ein netzliterarisches Objekt besteht aus mehreren Komponenten, die jeweils abhängig sind von weiteren medientechnischen Komponenten. Es handelt sich um strukturell vernetzte Objekte. Die für die Erhaltung notwendige Begrenzung der Objekte geht deshalb stets mit einem Verlust einher, der nicht nur die ausgeschlossenen Elemente betrifft, sondern auch das zur Erhaltung objektivierte Objekt, weil der Ausschluss von Komponenten jeweils zurückwirkt auf das, was eingeschlossen wird. Oder anders formuliert: Es gibt keinen funktionsfähigen Twitterbot ohne Twitter – die erhaltbaren Elemente laufen ins Leere des nicht Erhaltbaren. Es kann jedoch den Twitterbot *Gomringador* als archivierten Twitterbot auch dann noch geben, wenn die Plattform Twitter möglicherweise längst verschwunden ist.

Im Folgenden geht es um ein Problem, das sich zunächst als medientechnisch bedingtes Abgrenzungsproblem formulieren lässt. Der Fokus liegt dabei auf der Rolle und Funktion von Quelltexten. Für die beiden ausgewählten Gegenstände wird die je spezifische Funktion der Quelltexte exemplarisch herausgearbeitet, verbunden mit dem Ziel, anhand dieser Beispiele einige Punkte zu verdeutlichen, die auch für andere Objekte gelten können. Dabei rücken Quelltexte als Texte in den Blick, die selbst literarisch sein können, ohne dabei den Unterschied zwischen Programmcode und gerenderten Formaten einzuebnen. Programmcode ist vor diesem Hintergrund also operativer Text, dessen Status im Verhältnis zu den weiteren technischen, ästhetischen und performativen Aspekten eines Gegenstands jeweils neu zu bestimmen ist.<sup>3</sup> Für eine solche Verhältnisbestimmung ist zunächst eine Bestimmung der einzelnen Text- und Medienelemente nötig sowie eine Beschreibung oder Setzung von Objektgrenzen, die schließlich die Archivierbarkeit einzelner Elemente und, je nach Objektverbund und Objektverständnis, des gesamten

2 Kathrin Passig: @gomringador, Twitteraccount, 2018 (<https://twitter.com/gomringador>, Zugriff: 20.1.2023). Hier und im Folgenden wird die Bezeichnung @gomringador für den Twitteraccount verwendet und *Gomringador* für das gesamte Objekt einschließlich des Quelltexts. Johannes Auer: free lutz!, 2005, Online-Ressource (<http://freelutz.netzliteratur.net/>, Zugriff: 20. Januar 2023), URN (Archiv): urn:nbn:de:bsz:mar1-ddo01-bsz3968932795.

3 Zur Beschreibung von Programmcode als operativer Text, den ästhetischen Dimensionen von Programmcode und zum Begriff des operativen Texts im Anschluss an Sibylle Krämer siehe Florian Cramer: *Exe.cut[up]able Statements. Poetische Kalküle und Phantasmen des selbstausführenden Texts*. München 2011.

Objekts bedingen.<sup>4</sup> Hier interessiert schließlich der Status des operativen Texts im Verhältnis zum gerenderten Objekt auf der Bildschirmoberfläche (mit den technischen Zwischenstufen) und im Verhältnis zu den Interaktionsmöglichkeiten, die mit den installativen Aspekten der untersuchten Texte zusammenhängen.

So, wie es zum Verständnis eines Briefromans ein Wissen über die historische Post braucht, ist die Interpretation von Netzliteratur angewiesen auf ein Wissen um die je spezifischen mediengeschichtlichen Bedingungen eines Werks. In welcher medientechnischen Umgebung findet ein Werk statt? Welche Voraussetzungen müssen oder mussten erfüllt sein, damit Lektüren und Interaktion passieren? Welche Funktion übernehmen die feststellbaren Teilaspekte vom Quellcode über den gerenderten Text bis zu erforderlichen Interaktionen für die Realisierung des Werks? Man muss mit Blick auf die heterogenen medialen Elemente von Netzliteratur an dieser Stelle festhalten, dass ›digital‹ oder ›Netz‹ als Distinktionsmerkmal nur eine erste Unterscheidung ist, die je fall- oder auch gattungsbezogen weiter differenziert werden muss.

### Text und Quelltext

Der Begriff »Quelltext« oder »source code« bezeichnet Programmcode, der unter bestimmten Bedingungen ausgeführt oder kompiliert und dann ausgeführt werden kann. Programmcode definiert Prozesse, die bei Ausführung des Codes ablaufen. Quelltext und Source Code sind Begriffe mit einem noch gut erkennbar metaphorischen Element, nämlich dem Bild der Quelle. So, wie eine Quelle Wasser hervorbringt, bringt der Quelltext ebenfalls etwas hervor, im Fall der hier gewählten Beispiele in erster Linie Text, in zweiter Linie Übertragungen von Text. In beiden Beispielen hat der Quelltext also eine generative Komponente und eine Übertragungskomponente. Die Metaphorik der Hervorbringung kommt hier an eine Grenze, insofern die Hervorbringung von Text durch operativen Quelltext an die Operation, das heißt die Ausführung des Quelltexts, gebunden ist. Diese Operation ist nur unter bestimmten Bedingungen möglich und führt daher weitere Komponenten in das Verhältnis von Text und Quelltext ein, etwa die Ausführungsumgebung für ein Programm und die verteilten Komponenten von Installationen, die ganz oder teilweise online stattfinden. Festzuhalten ist: Text und Quelltext stehen in einer besonderen funktionalen Beziehung, die im Fall von *Gomringador* und *free lutz!* erstens eine generative Beziehung

4 Vgl. Espenschied und Rechert (Anm. 1).

ist, das heißt, das Programm generiert automatisch Texte, und die zweitens verbunden ist mit weiteren Funktionen, die das Ausspielen der Texte über die Plattform Twitter (*Gomringador*) oder eine eigene Webseite (*free lutz!*), also die automatisierte Veröffentlichung der Texte betreffen.

Die Quelltexte der beiden hier betrachteten Gegenstände sind besonders, weil es sich um Programme und also Texte handelt, die bei der Ausführung weitere Texte erzeugen, die dann als die eigentlich literarischen Texte veröffentlicht und gelesen werden. Aus literaturanalytischer Sicht sind die Quelltexte also interessant, weil sich hier vielfältige Anschlusspunkte für Lektüren finden, die in den im Web veröffentlichten Texten nicht sichtbar sind.<sup>5</sup> Für generative Texte gehören hierzu insbesondere die formalen Strukturen des generativen Algorithmus. Denn der Quelltext definiert für den generativen Teil einen formalisierten Ausdruck der zugehörigen Poetik. Diese Poetik beschreibt nicht einzelne Texte, sondern ein Set von Texten, das in aller Regel umfangreicher ist als die Menge der veröffentlichten Texte.<sup>6</sup>

Die Quelltexte für *Gomringador* und *free lutz!* sind für Leser:innen nicht zugänglich. Erhaltungsansätze, die den Quelltext als konstitutives Element des Werks mit einbeziehen, sind daher auf aktive Mitteilung des Quelltexts angewiesen.<sup>7</sup> Für Webarchivierungsstrategien mit einem hohen Automatisierungsgrad bedeutet diese Unverfügbarkeit des Quelltexts von Gegenständen im Netz, dass auf die Erhaltung von nicht clientseitigen Prozessen verzichtet werden muss oder nur einzelne Gegenstände vollständig erhalten werden können, für die der manuelle Aufwand bei der Erhaltung des Quelltexts machbar ist.

5 Neben genuin ästhetischen und stilistischen Aspekten finden sich in Quelltexten eine Reihe von Anschlusspunkten für kulturwissenschaftliche Analyseansätze, siehe zusammenfassend Mark C. Marino: *Critical Code Studies*. Cambridge, Mass. 2020.

6 Ausführlicher und mit weiteren Nachweisen Claus-Michael Schlesinger: Kafka stochastisch. Rekontextualisierung und Recodierung in computergestützten Textgeneratoren, in: *Rekontextualisierung als Forschungsparadigma des Digitalen*, hg. von Simon Meier-Vieracker, Gabriel Viehhauser und Patrick Sahle. Norderstedt 2020, S. 97–105, URN: urn:nbn:de:hbz:38-294301 (<http://nbn-resolving.de/urn:nbn:de:hbz:38-294301>, Zugriff: 10. Februar 2021).

7 Der Quelltext für *Gomringador* wurde von der Autorin Kathrin Passig auf Anfrage zur Verfügung gestellt. Der Quelltext für *free lutz!* ist nicht Teil des Archivpakets, das im Zuge der Erhaltung des Werks am Deutschen Literaturarchiv Marbach erstellt wurde.

Kathrin Passig: *Gomringador*

Die Arbeit *Gomringador* von Kathrin Passig ist ein Twitterbot, das heißt es werden automatisiert Texte auf der Plattform Twitter veröffentlicht. Der erste Tweet – so werden auf Twitter veröffentlichte Texte genannt – wurde am 26. Januar 2018 um 20:59 Uhr gesendet. Der Account ist zum Zeitpunkt des Schreibens dieses Artikels im Dezember 2022 weiterhin aktiv. Bis zum 19. Dezember 2022, 19:44 Uhr wurden 2.758 Tweets gesendet. Das Programm für den Twitterbot @gomringador ist auf das Veröffentlichen von Texten ausgerichtet und beinhaltet keine automatisierten Interaktionen. Die Texte erscheinen entsprechend entweder als Einzeltexte in der Timeline von Nutzer:innen oder können auf der Accountseite als eine lange Folge von Texten gelesen werden.

Der Titel der Arbeit, *Gomringador*, ist zusammengesetzt aus dem Nachnamen des Autors Eugen Gomringer und dem letzten Wort der Konstellation *avenidas* von Eugen Gomringer, »admirador«.<sup>8</sup> Die Texte werden automatisch generiert. Grundlage für die Generierung der Texte sind ein festgelegtes Satzmuster und eine Reihe von Wortlisten, mit der die Leerstellen im Satzmuster gefüllt werden. Das Satzmuster ist, wie mit dem Titel schon angedeutet, übernommen von der Konstellation *avenidas* von Eugen Gomringer. Das Vokabular besteht aus einer Reihe von Wortlisten, die thematisch oder semantisch bestimmt sind. Die Dateinamen sind entsprechend gewählt. So finden sich beispielsweise Wortlisten zu »bauernhof«, »businessgrind«, »cryptocurrency«, »harrypotter« oder »kuehlschrank«. In Version 1 des Bots<sup>9</sup> finden sich 21 Wortlisten mit insgesamt 6.168 Wörtern. Die Texte werden gebildet, indem zunächst eine Wortliste gewählt wird. Im zweiten Schritt wird das Satzmuster aus der Wortliste gefüllt. Der fertige Tweet wird dann im dritten Schritt über die Programmierschnittstelle von Twitter zur Plattform übertragen und dort direkt veröffentlicht.

8 Die Konstellation *avenidas* wurde vielfach publiziert. Eugen Gomringer: *avenidas*, in: eugen gomringer. konkrete poesie, hg. von Museum für Konkrete Kunst Ingolstadt. Ingolstadt 1992.

9 Alle Überlegungen und Analysen zum Quelltext beziehen sich auf die erste Version des Bots. Passig hat inzwischen eine zweite, erweiterte Version von *Gomringador* erstellt, die hier nicht weiter berücksichtigt ist.

## Zeitgeschichtlicher Kontext

Die Erstveröffentlichung des Bots hängt zeitgeschichtlich mit der heftigen Debatte um eine Fassadengestaltung der Alice-Salomon-Hochschule (ASH) in Berlin zusammen, die im Sommer 2017 im Zuge einer hochschulöffentlichen Debatte zur Neugestaltung der Südfassade des ASH-Campus ihren Anfang nimmt und sich dann über einen längeren Zeitraum in mehreren Wellen intensiviert. Zu den Ausgangspunkten der Debatte zählte ein offener Brief von Studierenden, der mit einer feministischen Analyse der Konstellation *in situ* auf der Hochschulfassade den männlich codierten Blick herausarbeitet und mit den sozialen Situationen im und um das Hochschulgebäude kontextualisiert.<sup>10</sup> Die Studierenden begründeten mit dieser Analyse ihre Forderung nach einer Diskussion der Fassadengestaltung im Akademischen Senat der Hochschule. In einer zunehmend von den konkreten Fassaden-Vorhaben der ASH abgekoppelten Debatte wurde diese Lesart mit Verweis auf die formale Poetik der Konstellation zurückgewiesen.<sup>11</sup> Die Umgestaltung der Fassade ist inzwischen abgeschlossen. Zu sehen ist heute ein Text der Autorin Barbara Köhler, der auf mehreren Ebenen auf die Debatte, den vorherigen Fassadentext und seine produktive Überschreibung Bezug nimmt.

Das Werk *Gomringador* kann in diesem Zusammenhang als vielschichtige Installation gelesen werden, die lyrikanalytische Positionen und literaturgeschichtliche Aspekte der Debatte aufnimmt, poetisch durcharbeitet und kommentiert. In der Serie von generierten Texten werden ganz unterschiedliche Semantiken mit dem Satzmuster von *avenidas* durchgespielt, die durch die semantischen Differenzen einerseits deutlich machen, dass der Austausch der bedeutungstragenden Nomen die Semantik des Texts betrifft. Gleichzeitig lässt sich dieses Spiel auch als konsequenter Vollzug einer poetischen

10 Studierende der Alice-Salomon-Hochschule Berlin: Offener Brief, 16. April 2016 ([https://www.ash-berlin.eu/fileadmin/Daten/Einrichtungen/Pressestelle/mehrls\\_fassade/offener\\_Brief\\_Gedicht\\_ohne\\_Unterschriften.pdf](https://www.ash-berlin.eu/fileadmin/Daten/Einrichtungen/Pressestelle/mehrls_fassade/offener_Brief_Gedicht_ohne_Unterschriften.pdf), Zugriff: 22. Februar 2023).

11 So beispielsweise Nora Gomringer: Was aber ist ein Gedicht?, in: Die Welt, 1. September 2017, S. 21; Paul Ingendaay: Diskussion um Avenidas: Kunst darf alles, nur nicht immer, in: Frankfurter Allgemeine Zeitung online, 28. März 2018 (<https://www.faz.net/aktuell/feuilleton/debatten/diskussion-um-eugen-gomrings-gedicht-avenidas-15515491.html>, Zugriff: 12. Januar 2023). Ein fortlaufend ergänzter Pressespiegel zur Fassadendebatte findet sich auf den Seiten des Allgemeinen Studierendenausschusses der Alice-Salomon-Hochschule ([https://www.ash-berlin.eu/hochschule/organisation/referat-hochschulkommunikation/presse\\_spiegel-fassadendebatte/](https://www.ash-berlin.eu/hochschule/organisation/referat-hochschulkommunikation/presse_spiegel-fassadendebatte/), Zugriff: 19. Dezember 2022, archiviert: <https://web.archive.org/web/20221219200814/https://www.ash-berlin.eu/hochschule/organisation/referat-hochschulkommunikation/pressespiegel-fassadendebatte/>).

Programmatik der Konstellation lesen, die ihre Texte eben als Spielfeld verstanden wissen will, auf dem nicht Eindeutigkeit übermittelt werden soll, sondern Sinn im Zuge der Rezeption unter aktiver Beteiligung der Leser:innen entsteht.<sup>12</sup> Und schließlich nimmt der Bot ein Phänomen auf, das schon vorher in der Debatte beobachtet wird, nämlich die Memeifizierung der *avenidas*-Syntax, das heißt die Verwendung der Syntax zur Herstellung von Texten mit anderen bedeutungstragenden Nomen.<sup>13</sup> Eine weiter gehende Analyse des Bots im Kontext der zeitgeschichtlichen Entwicklungen steht noch aus, kann hier aber aus thematischen Gründen nicht geleistet werden, was die Erhaltung von *Gomringador* als literarische Arbeit umso wichtiger erscheinen lässt.

### Veröffentlichte Texte (Tweets und Tweetdaten)

Für *Gomringador* als Objekt lassen sich mehrere Elemente unterscheiden, die für die Erhaltung relevant sein können. Zu *Gomringador* als Textgenerator zählen erstens die erstellten und veröffentlichten Texte auf Twitter unter dem Account @gomringador. Auf der Seite des Accounts sind alle Tweets, die über den Account veröffentlicht wurden, zeitlich nacheinander gelistet und können, langanhaltendes Runterscrollen vorausgesetzt, vollständig gelesen werden. Die einzelnen Tweets können im Rahmen des Accounts auch als eigene Seite aufgerufen werden.<sup>14</sup>

Die Plattform Twitter bietet eine Programmierschnittstelle an (API), über die Tweets als strukturierte Daten heruntergeladen werden können. Dabei werden einzelne Tweets mit einem umfangreichen Metadatensatz bereitgestellt. Die Metadaten bieten viele Datenpunkte, die für weitere Analysen genutzt werden können und die in den browserbasierten Ansichten nicht vollständig enthalten sind oder nur schwer extrahiert werden können. Die

12 Eugen Gomringer: weshalb unsere dichtung »konkrete dichtung« nennen, in: Konkrete Poesie, hg. von dems. St. Gallen 1988 (Zur Sache der Konkreten, Bd. 1).

13 Peter Mühlbauer: Gomringer-Gedicht »Avenidas« wird stabiles Mem, in: Telepolis, 29. Januar 2018 (<https://www.heise.de/tp/features/Gomringer-Gedicht-Avenidas-wird-stabiles-Mem-3952741.html>, Zugriff: 19. Dezember 2022, archiviert: <https://web.archive.org/web/20221219194045/https://www.heise.de/tp/features/Gomringer-Gedicht-Avenidas-wird-stabiles-Mem-3952741.html>).

14 So zum Beispiel eine Konstellation, die mit dem Wort »Katzenfutter« beginnt, in: Kathrin Passig: Gomringador (<https://twitter.com/gomringador/status/1447516333000859653>, Zugriff 11. Oktober 2021, archiviert: <https://web.archive.org/web/20211011105619/https://twitter.com/gomringador/status/1447516333000859653>).

strukturierten Daten enthalten dabei keine Informationen zur Darstellung des Tweets in einem Browser, sodass die Publikation eines Texts auf der Plattform Twitter mit den strukturierten Daten nicht in der gleichen Form wiederhergestellt werden kann.

## Twitter-Plattform

Die unter dem Account @gomringador veröffentlichten Texte auf Twitter kann man nur bedingt konzeptionell von der Plattform trennen. Denn abhängig von der primären Einordnung des Objekts als Installation, die auch die möglichen Interaktionen des Publikums umfasst, oder als Textpublikation lassen sich die für die einzelnen Tweets sichtbaren Interaktionen (Likes, Retweets, Antworten/Kommentare) als Teil des Gegenstands beschreiben.

Ohne die Plattform Twitter gäbe es kein Twitterprofil, ohne Twitter-API gäbe es keinen Twitterbot, ohne Twitter-Webseite keine Nutzer:innen und ohne sie keinen Grund, die Texte dort zu veröffentlichen. Twitter-API, Twitter-Webseite und Twitter-Nutzer:innen sind die externen Komponenten des Werks *Gomringador*, anhand derer deutlich wird, dass nicht alles greifbar und erhaltbar ist. Wenn sich eine dieser Komponenten verändert, dann verändern sich die Bedingungen des Werks und damit auch das Werk, hier verstanden als Installation und als Zusammenhang von Produktion, Publikation und Rezeption. Beispielsweise könnte die Twitter-API so verändert werden, dass nur noch ein Tweet pro Tag veröffentlicht werden kann, die Webseite könnte umgestaltet werden, sodass die Beiträge nicht mehr untereinander sondern übereinander angezeigt werden und Nutzer:innen könnten die Plattform verlassen und zukünftige Posts ungelesen und unkommentiert zurücklassen.

## Quelltext

Neben all den Quelltexten, die Twitter zu Grunde liegen, ist das eigentlich operative Element von *Gomringador* ein kurzes Programm, das in der Programmiersprache PHP geschrieben ist.<sup>15</sup> Dieser operative Text implementiert

<sup>15</sup> PHP ist die Abkürzung für PHP Hypertext Preprocessor (rekursives Akronym). Die Programmiersprache wird hauptsächlich für die Entwicklung von Webanwendungen eingesetzt. Die Analyse bezieht sich auf die erste Version der Arbeit, siehe Gomringer (Anm. 8).



die Regeln zur Herstellung der Texte und beinhaltet außerdem die nötigen Anweisungen für die Übertragung der Texte auf die Plattform. In der PHP-Datei sind das Satzmuster von *avenidas* sowie die Anweisungen zum Füllen des Satzmusters aus dem zugehörigen Vokabular festgelegt. PHP-Datei und Vokabular sind also eine vollständige Implementierung der Regeln zur Herstellung aller möglichen *Gomringador*-Texte. Bisher hat *Gomringador* fast 3.000 Texte getwittert. Bevor allerdings der Bot alle möglichen Texte erzeugt hat, ist entweder die Sonne erloschen oder – inzwischen etwas wahrscheinlicher – Twitter existiert nicht mehr. Das vollständige Set aller möglichen Texte umfasst in etwa 1,5 Billionen Texte.<sup>16</sup>

Der Quelltext befindet sich in einem Verzeichnis auf einer Maschine mit Internetzugang, die von der Autorin kontrolliert wird. Er besteht aus dem PHP-Programm und 21 Textdateien mit Wortlisten. Die Wortlisten beinhalten eine je unterschiedliche Anzahl von Wörtern und sind zwischen 16 (»lego«) und 1.787 (»cryptocurrency«) Wörtern lang, mit einer Gesamtzahl von 6.168 Wörtern. Aus dem Quelltext ist zu lesen, dass für jede Konstellation eine Wortliste als wortgebend ausgewählt wird. Die Wortlisten mit den sprechenden Dateinamen bieten deshalb einen ersten Überblick über die thematische Ausrichtung und eröffnen weitere Lektüre- und Analysemöglichkeiten des *Gomringador*-Quelltexts.

Quelltexte von Twitterbots wie *Gomringador* werden oft nicht veröffentlicht, weil der Schwerpunkt der Veröffentlichung auf den generierten Texten oder der Installation liegt. Die Twitterstreams und -daten sind als Textrealisierungen die dem Publikum zugewandte Seite des Gegenstands, der operative oder generative Text ist aus den veröffentlichten realisierten Texten aber nur annäherungsweise rekonstruierbar. Ohne Quelltext also keine Quelltextlektüren und -analysen. Besonders ins Gewicht fällt dies bei Social-Media-Installationen mit einem generativen Anteil. Aber auch für Objekte ohne generativen Anteil lässt sich festhalten, dass eine Analyse der Funktionen, das Herausarbeiten von spezifischen Abläufen und damit auch eine möglichst weitgehende Erhaltung des Objekts mit den entsprechenden Funktionalitäten ohne den Quelltext kaum möglich ist.

16 Die Rechnung lautet, jeweils der Anzahl von Wörtern in einer Liste folgend, unter Berücksichtigung der Besonderheit, dass die letzte Zeile stets aus der Datei »gomringador\_ende.txt« mit 153 Wörter entnommen wird:  $153 \times ((177 \times 176 \times 175) + (50 \times 49 \times 48) + (171 \times 170 \times 169) + (297 \times 296 \times 295) + (1787 \times 1786 \times 1785) + (1577 \times 1576 \times 1575) + (114 \times 113 \times 112) + (106 \times 105 \times 104) + (85 \times 84 \times 83) + (225 \times 224 \times 223) + (64 \times 63 \times 62) + (16 \times 15 \times 14) + (133 \times 132 \times 131) + (119 \times 118 \times 117) + (248 \times 247 \times 246) + (142 \times 141 \times 140) + (124 \times 123 \times 122) + (347 \times 346 \times 345) + (157 \times 156 \times 155) + (76 \times 75 \times 74)) = 1.488.968.430.624.$

## Erhaltung

Aus konservatorischer Sicht ist zunächst eine Beschreibung der einzelnen Elemente des Gegenstands sowie des funktionalen Zusammenspiels nötig. Danach wird untersucht, welche ästhetischen oder konzeptuellen Bedeutungen Elemente und Funktionen für das Werk haben. Schließlich werden anhand der einzelnen Elemente des Werks die Objektgrenzen analysiert und identifiziert. Die Ergebnisse dieser Anamnese ermöglichen dann die begründete Entwicklung und Bewertung der möglichen Erhaltungsmaßnahmen.

Für *Gomringador* kann zunächst zwischen dem nichtöffentlichen Quelltext und dem veröffentlichten Teil des Werks unterschieden werden. Der veröffentlichte Teil des Werks wird von der Plattform Twitter gespeichert und bereitgestellt: Das Profil @gomringador, die Tweets und die Interaktionen anderer Nutzer:innen (Kommentare, Likes, Retweets, Follows). Für die Erhaltung von *Gomringador* insgesamt bedeutet dieser Zusammenhang: Der Quelltext muss in einer kompatiblen Softwareumgebung ausgeführt werden, die Twitter-API, auf die das Programm zugreift, muss vorhanden sein, und Twitter muss als Plattform noch funktionieren.

Im Bereich der Softwareerhaltung schlagen Klaus Rechert und Dragan Espenschied vor, für digitale performative Objekte, also Software, die Objektgrenzen zu untersuchen und davon mögliche und nötige Erhaltungsmaßnahmen abzuleiten. Sie definieren dafür drei verschiedene Kategorien. *Bound* («gebunden») beschreibt Objekte, die eine klar definierbare Grenze haben. *Blurry* («unscharf») meint Objekte, bei denen die Grenze nicht als eindeutig oder trennscharf begründet werden kann. *Boundless* («grenzenlos») sind Objekte, die keine definierbare Grenze haben.

Der generative Gomringador-Quelltext und die daraus entstandenen Texte haben klare Grenzen, sind aber qua im Programm festgelegtem Publikationskanal mit der Plattform Twitter verbunden, die als grenzenlos zu bezeichnen ist. Daher handelt es sich bei dem gesamten Objekt um ein grenzenloses Werk, weil ein Teil davon, die Plattform Twitter, nicht in ihrer Funktionalität erfasst und erhalten werden kann. Erhalten werden können die generierten Texte und der zugrunde liegende Quelltext, der für die Archivierung bei der Autorin angefragt werden muss. Eine Analyse der Funktionsweise und der Abhängigkeiten des Quelltexts ist notwendig, um diesen wieder lauffähig zu machen. Die Ausführung ist abhängig von einer ausführenden Umgebung, zu der eine bestimmte Version der Programmiersprache PHP mit einem kompatiblen Betriebssystem gehört.

Die Tweets von @gomringador können entweder über die Twitter-API angefragt und heruntergeladen werden oder innerhalb der Twitter-Webseite

mit Verfahren der clientseitigen Webarchivierung gespeichert werden. Bei dieser Methode werden die Ressourcen, die Twitter dem Browser beim Laden der Seite schickt, im dafür entwickelten Archivformat WARC (für Web ARChive) gespeichert. Die solchermaßen erzeugten WARC-Dateien bleiben dann dauerhaft und unabhängig von der Originalseite abspielbar. Man hat also eine archivierte Version der Webseite <https://twitter.com/gomringador> zu einem bestimmten Zeitpunkt. Neben den Tweets sind die Twitteroberfläche und vermerkte Interaktionen anderer Nutzer:innen mit erfasst.

Ein Vorteil des zusätzlichen Abrufs der Tweets über die API ist, dass dann die Metadaten zu den einzelnen Texten und zum Account @gomringador in einer strukturierten Form enthalten sind, die weitergehende computergestützte Analysen ermöglichen, beispielsweise bezüglich des Zeitverlaufs der einzelnen Postings, der Textlängen usw., die auf der Grundlage von WARC-Dateien nur mit hohem Aufwand möglich sind.

Im Fall *Gomringador* muss das Ergebnis der clientseitigen Webarchivierung allerdings eher zur Dokumentation als zur Erhaltung des Werks gezählt werden. Denn diese Methode sichert zwar die veröffentlichten Ergebnisse, ermöglicht aber nicht die Erhaltung der Funktionalität des gesamten Objekts. Das schmälert nicht die Bedeutung der archivierten Seiten, sondern zeigt im Gegenteil die wichtige Rolle der Dokumentation für die bestmögliche Erhaltung von grenzenlosen Objekten. Weil die Webarchivierung der Twitterseiten von *Gomringador* stets ein Abbild des jeweils aktuellen Zustands der Seiten erzeugt, muss diese Dokumentation regelmäßig durchgeführt werden, zumindest solange @gomringador noch neue Texte postet.

*Gomringador* kann im Ergebnis nicht vollständig erhalten werden, weil die Twitter-Plattform und die Twitter-API nicht erhalten werden können. Dem vollständigen Verschwinden von *Gomringador* lässt sich aber mit der Erhaltung des Quelltexts und der Dokumentation der Inhalte auf Twitter via Webarchivierung entgegenwirken, sodass zumindest Teile der Funktionalität – die Textgenerierung – und die veröffentlichten Texte in ihrer veröffentlichten Form erhalten werden können.

## Ästhetik und Konzept

Für die Erhaltung zeitbasierter Medienkunst ist es wichtig, die ästhetischen und konzeptuellen Bedeutungen der Bestandteile eines Werks aus Sicht der Autor:innen oder der Künstler:innen zu kennen. Ein klassisches Beispiel in diesem Zusammenhang ist das Problem der mediengeschichtlichen Gebundenheit von Kunstwerken und der intentionalen Auswahl von Medien-

technik für Installationen oder Ausstellungen eines Werks: Darf ein initial verwendeter CRT-Monitor einer Videoinstallation bei einer Ausstellung mit einem LED-Monitor ersetzt werden, wenn der originale CRT-Monitor nicht verfügbar ist? Auf diese Frage gibt es keine allgemein gültige Antwort. Sie muss im Einzelfall abgewogen werden, weil ein originaler Bestandteil werkkonstitutiv sein kann, aber nicht sein muss. Erst das Wissen darum, welche Bestandteile für die Erhaltung und Rekonstruktion angepasst oder ersetzt werden dürfen, erleichtert oder ermöglicht sogar erst eine Wiederaufführung.

Für den installativen Aspekt von *Gomringador* muss in ähnlicher Weise nach der Bedeutung der Bestandteile gefragt werden, gerade wenn sie Risikofaktoren für die Erhaltung darstellen. So ist zum Beispiel realistisch denkbar, dass die im Programm verwendete Twitter-API von den Betreiber:innen verändert wird. Aus Erhaltungssicht stellt sich für einen solchen Fall die Frage, ob der Quelltext von der erhaltenden Institution angepasst werden darf, um die Funktionalität zu erhalten. Solche Fragen können schließlich auch für weitere Teile des Werks gestellt werden, um deren Bedeutung festzustellen und erweiterte Erhaltungsmaßnahmen durchführen zu können.

### Johannes Auer: *free lutz!*

Die Arbeit *free lutz!* (2005) von Johannes Auer ist als Installation und Performance konzipiert.<sup>17</sup> Sie besteht aus einer Netzinstallation in Form einer interaktiven Webseite, die dauerhaft online ist. Der Autor hat außerdem in den Jahren 2005, 2007 und 2009 mehrere Lesepresentations durchgeführt, bei denen die Netzinstallation als interaktives Element ebenfalls zum Einsatz kam. Website und Performances wurden vom Autor selbst schriftlich und zum Teil per Video dokumentiert.<sup>18</sup> Netzinstallation und Dokumentation sind auf dem Server netzliteratur.net gehostet, der vom Autor betrieben wird. Neben den eigenen Projekten sind dort auch Webseiten mehrerer anderer Autor:innen gehostet, dazu eine Vielzahl an theoretischen und poetologischen Texten.

*free lutz!* basiert auf einem Textgenerator, der die Struktur der *Stochastische[n] Texte* von Theo Lutz mit Eingaben von Nutzer:innen verbindet.

17 Im Folgenden werden die bis hierher entwickelten Aspekte zur Rolle von Quelltexten für die Archivierung und Analyse eines Gegenstands anhand eines zweiten Beispiels durchgespielt, das durch den eingebundenen Textgenerator einige Ähnlichkeiten, als vollständige Webseite und hinsichtlich des Zugangs zum Quelltext aber auch signifikante Unterschiede zu *Gomringador* aufweist.

18 Vgl. Passig (Anm. 2).

Nutzer:innen können über eine kleine Textbox Nomen und Adjektive eingeben (beliebige Zeichenketten sind möglich), die dann in die Struktur der *Stochastischen Texte* eingebaut werden. Auf der Seite wird ein fortlaufender Strom von Sätzen ausgegeben, wobei alle Eingaben berücksichtigt werden. Das System kann mit einer Schaltfläche zurückgesetzt werden und gibt dann Sätze im Stil der *Stochastischen Texte* aus bis zur nächsten Eingabe von Nutzer:innen. Auf diese Weise interagieren Nutzer:innen sowohl mit dem System als auch miteinander. Die Webseite kann deshalb als fortlaufende Installation mit einem performativen Anteil verstanden werden.

Die Webseite basiert auf einer PHP-Anwendung, das heißt, das Programm zur Herstellung der Texte und zur Verarbeitung der Eingaben von Nutzer:innen wird auf dem Server ausgeführt. Es kann nur mit Zugriff auf den entsprechenden Bereich auf dem Server eingesehen werden und ist nicht veröffentlicht. Eine Lektüre des Programms ist damit nicht möglich. Die Archivierung mithilfe eines Webcrawls ist nicht möglich. Die Aufführung endet, wenn der Server vom Netz geht. Eine museale Wiederaufnahme der Installation wäre auf den Quelltext angewiesen, sofern eine Rekonstruktion angestrebt ist, die möglichst nah am Originalmaterial arbeitet. Ein Nachbau, der lediglich auf die dokumentierten Funktionen und das Interface Bezug nimmt, ist auch ohne Zugriff auf den Quelltext möglich.

## Erhaltung

Zugang zum Werk existiert damit zunächst also nur über die öffentliche Website. Die folgenden Überlegungen beziehen sich auf das deutschsprachige Webinterface des Werks, das auch in weiteren Sprachen verfügbar ist (Englisch, Französisch, Polnisch, Portugiesisch). Im Rahmen des Forschungsprojekts *Aufbau eines Quellencorpus für die seit den 1990er Jahren entstehende Literaturgattung »Netzliteratur«* am DLA wurde untersucht, ob das Werk mittels Crawling durch Webarchivierungstools erhalten werden kann.<sup>19</sup> Diese Methode war jedoch nicht erfolgreich. Der Seitenquelltext, der

19 Deutsches Literaturarchiv Marbach: *Aufbau eines Quellencorpus für die seit den 1990er Jahren entstehende Literaturgattung »Netzliteratur«*, o.J., (<https://www.dla-marbach.de/bibliothek/projekte/aufbau-eines-quellencorpus-fuer-die-seit-den-1990er-jahren-entstehende-literaturgattung-netzliteratur/>), Zugriff: 20. Januar 2023, archiviert: <https://web.archive.org/web/20221119202326/https://www.dla-marbach.de/bibliothek/projekte/aufbau-eines-quellencorpus-fuer-die-seit-den-1990er-jahren-entstehende-literaturgattung-netzliteratur/>).

vom Server an den Browser gesendet wurde, gibt hier weitere Hinweise.<sup>20</sup> Der textgenerierende Quelltext ist dort nicht enthalten. Es werden Dateien im HTML-Quelltext eingebunden, die diese Funktionen auf serverseitigen Quelltext auslagern: *lutz\_ausgabe.php*, *eingabe.php*, *start.php*. PHP ist eine Programmiersprache, die für die serverseitige Programmierung von Webanwendungen und -seiten verwendet wird. Bei einer PHP-basierten Seite wird das Programm also auf dem Server ausgeführt, im Ergebnis werden dann Daten, zum Beispiel eine Webseite, an den Browser gesendet. Der Inhalt der Datei *Lutz-ausgabe.php* ist also auf dem Server ein anderer als im Browser – das PHP ist verschwunden, weil es bereits serverseitig interpretiert wurde.

Anhand der Beobachtungen und der oberflächlichen Analyse können folgende Bestandteile identifiziert werden: Erstens der Textgenerator, der in PHP geschrieben ist. Teil des Textgenerators sind ein Satzmuster sowie Wortlisten, die mit den Eingaben von Nutzer:innen ergänzt werden. Zweitens die Dokumentation auf der Website *freelutz.netzliteratur.net*. Und drittens die in der Dokumentation benannten und zum Teil auch dokumentierten Performances im Rahmen von Lesungen und Vorträgen.

Aus Sicht der technischen Erhaltung scheint *free lutz!* also ein Werk mit klaren Objektgrenzen (*bound*) zu sein, da die Ressourcen des Textgenerators und der Website lokal abgespeichert werden und durch eine passende Softwareumgebung ausgeführt werden können. Die Bedingung für die Erhaltung ist daher, dass die Ressourcen auf dem Webserver zugänglich sind und von dort kopiert werden dürfen. Die Ressourcen, zu denen der Quelltext gehört, müssen durch eine passende Softwareumgebung ausführbar gehalten werden. Denn nur im ausführbaren und funktionsfähigen Zustand ist der Anteil »Webseite/Netzinstallation« des Werks erhalten. Die serverseitige Archivierung ist allerdings nicht so weit verbreitet, doch ist, wie das Beispiel *free lutz!* zeigt, für manche Objekte notwendig, damit das Werk erhalten werden kann.

## Fazit

Literaturwissenschaftliche Lektüre und archivarische Anamnese haben gezeigt, dass Quelltexte von Netzliteratur literarische und funktionale Ebenen haben, die für das Verständnis und für die Erhaltung eines Werks relevant sind. Das gilt sowohl für Quelltexte, die mit automatisierten Verfahren

<sup>20</sup> Zu unterscheiden ist hier zwischen dem Quelltext des auf dem Server ausgeführten Programms und dem Seitenquelltext, der als Ergebnis dieser Ausführung zur weiteren Darstellung an den Browser geschickt wird.

wie Webcrawling archiviert werden können, als auch für solche, die für die Lektüre oder Archivierung nicht direkt verfügbar sind, sondern manuell und in Zusammenarbeit mit den Autor:innen gesichert werden müssen. Dabei ist der Quelltext in der Regel Teil eines Gesamtensembles, dessen einzelne Komponenten gegenstandsabhängig mit Blick auf Funktionalität und Ästhetik eingeordnet und bewertet werden müssen. Der Quelltext gehört zum Werk und stellt für bestimmte Objekte, beispielsweise *Gomringador* und *free lutz!*, ein zentrales Element dar. Gleichzeitig ist der Quelltext als Teil einer Ausführungsumgebung zu betrachten, wenn funktionale Aspekte wie die Generierung von Texten oder interaktive Elemente einer Webseite erhalten werden sollen. Eine authentische Archivierung sollte daher den Quelltext bei der Prüfung und Bewertung eines Gegenstands hinsichtlich ästhetischer, funktionaler und performativer Dimensionen einbeziehen.