

Annelie Pentenrieder

## **„Nach Zuhause“. Dynamische Reflexionen zwischen verkörperten und materialisierten Navigationsroutinen im Taxi**

2018

<https://doi.org/10.25969/mediarep/13030>

Veröffentlichungsversion / published version

Sammelbandbeitrag / collection article

### **Empfohlene Zitierung / Suggested Citation:**

Pentenrieder, Annelie: „Nach Zuhause“. Dynamische Reflexionen zwischen verkörperten und materialisierten Navigationsroutinen im Taxi. In: Käthe von Bose, Hannelore Bublitz, Matthias Fuchs u.a. (Hg.): *Körper, Materialitäten, Technologien*. Paderborn: Fink 2018, S. 119–135. DOI: <https://doi.org/10.25969/mediarep/13030>.

### **Nutzungsbedingungen:**

Dieser Text wird unter einer Creative Commons - Namensnennung 4.0 Lizenz zur Verfügung gestellt. Nähere Auskünfte zu dieser Lizenz finden Sie hier:

<https://creativecommons.org/licenses/by/4.0>

### **Terms of use:**

This document is made available under a creative commons - Attribution 4.0 License. For more information see:

<https://creativecommons.org/licenses/by/4.0>

ANNELIE PENTENRIEDER

„NACH ZUHAUSE“.

## DYNAMISCHE REFLEXIONEN ZWISCHEN VERKÖRPERTEN UND MATERIALISIERTEN NAVIGATIONSROUTINEN IM TAXI

### 1. Intro

Bei der Navigation im tagtäglichen Straßenverkehr stehen sich zunehmend materialisierte und verkörperte Routinen von Fahrer\_innen und ihrer Navigationssoftware gegenüber. Menschliche Praktiken und maschinelle Prozesse interagieren miteinander und entwickeln eine „soziomaterielle Handlungsfähigkeit“<sup>1</sup>, die sich zwischen Fahrer und Software situativ und individuell ausbildet. Darüber hinaus nehmen beide zunehmend ‚lernend‘ aufeinander Bezug, indem sie durch reflexive „Selbsttätigkeiten“<sup>2</sup> gegenseitige Anpassungsroutinen aneinander ausbilden. Der Text verfolgt die These, dass durch diese dynamisch ‚lernende‘ Selbsttätigkeit beider Akteure das Navigieren von einer hybriden und kooperierenden Mensch-Maschine-Interaktion auch zu einem konkurrierenden Verhältnis werden kann. Diskurse um maschinelles Lernen und autonomes Fahren verstärken diese Verschiebung von einer Kooperation hin zur Konkurrenz. Diese These soll an der Interaktion zwischen Taxifahrern und ihren Navigationssystemen überprüft werden. Dazu thematisiere ich im Folgenden unterschiedliche Anpassungsroutinen der Navigationssoftware von manuellen Updates bis hin zur Entwicklung einer ‚learning navigation‘. Anschließend gehe ich der Frage nach, wie sich befragte Taxifahrer zu diesen ‚Selbsttätigkeiten‘ von Software verhalten und mit eigenen ‚Selbsttätigkeiten‘ zu ihr in Bezug setzen.

Zwei Vignetten<sup>3</sup> aus meiner Forschung bei Taxifahrern<sup>4</sup> eröffnen dazu empirische Spannungsverhältnisse ‚lernender‘ Mensch-Maschine-Interaktionen.

---

<sup>1</sup> Lucy Suchman, *Human-Machine Reconfigurations. Plans and Situated Actions*, 2. Aufl., Cambridge, New York, NY, 2007, S. 267. „Soziomaterielle Handlungsfähigkeit“ (im Originaltext „sociomaterial agency“).

<sup>2</sup> Hartmut Winkler, „These 9: Automatismen sind Technik und haben einen privilegierten Bezug auf Technologie“, in: Hannelore Bublitz/Roman Marek/Christina L. Steinmann/Hartmut Winkler (Hg.), *Automatismen*, Paderborn, 2010, S. 113-117: 117.

<sup>3</sup> Die Vignetten-Technik ist eine Methode der qualitativen Sozialforschung und Ethnografie, mit der Wahrnehmungen aus teilnehmenden Beobachtungen, Interviews oder anderen Quellen der Feldforschung herausgelöst werden, um relevante Szenarios und Situationen kondensiert darzustellen. Eine narrative Vignette basiert auf den lebhaften Berichten und Feldnotizen über Praktiken, die von einer Beobachterin ‚synthetisiert‘ werden. Die Aussagekraft einer Vignette ist abhängig von ihrem Grad an reichhaltiger Beschreibung konkreter Details und erlaubt dem Leser oder der Leserin als Ko-Analyst\_in der Studie tätig zu werden. Vgl. dazu Alice Juel

Aufbauend auf dem Ansatz von Lucy Suchman wird das Fazit sein, dass die synchrone Verwendung des Begriffes ‚Lernen‘ für Mensch und Maschine zwar Missverständnisse erzeugt, aber ebenso analytisch Differenzen und Problemfelder offenbaren kann.

## 2. Taxis ohne Lenkrad

Von der Rückbank eines Taxis aus werden zunehmend Szenarien sichtbar, in denen Taxifahrer mit ihren Navis die nächste Abbiegung aushandeln. Die Fahrer stimmen dabei ihre spezifischen Ortskenntnisse und bewährten Fahrgewohnheiten mit den algorithmischen Berechnungen ihrer GPS-Navigationssoftware ab, um eine optimale Route von einem Ort zum anderen zu finden. Ihre tagtägliche Entscheidung, direkt links oder erst an der nächsten Straßenkreuzung abzubiegen, wird dabei stets neu und auf Basis eines Gewebes vielschichtiger „Mensch-Maschine-Interaktionen“<sup>45</sup> getroffen.

Wenn menschliche Routinen und maschinelle Iterationen über die Nützlichkeit der nächsten Abbiegung entscheiden, greifen materialisierte und verkörperte Praktiken ineinander: Teilkompetenzen zur Wegfindung werden an „kleine Kästchen mit Saugfuß“<sup>46</sup>, an Smartphones oder eingebaute Navigationssysteme in der Mittelkonsole ausgelagert und so in Form von digitalen Straßendatenbanken und algorithmischen Protokollen materialisiert.<sup>7</sup> Gleichzeitig bringen die Fahrer herkömmliche sowie neuartige Wegfindungskompetenzen in die Navigationssituation mit ein. Ihre Praktiken verkörpern dabei Berufsinteressen, ökonomische Absichten oder individuelle Sympathien. In der Navigationsroutine sind menschliche und maschinelle Entscheidungsprozesse komplex und für gewöhnlich ununterscheidbar zu hybriden<sup>8</sup> Vollzügen verwoben.

---

Jacobsen, „Vignettes of Interviews to Enhance an Ethnographic Account“, in: *Ethnography and Education* 9, 1 (2014), S. 35-50: 41.

<sup>4</sup> Der Beitrag basiert auf einer ethnografischen Feldforschung, die ich im Herbst 2015 und 2016 an den Taxiständen dreier deutscher Städte durchgeführt habe. Trotz Bemühungen habe ich dabei nur männliche Fahrer befragen können. Bezieht sich der folgende Text auf meine Feldnotizen, so benutze ich darum ausschließlich die männliche Form, da ich auf keine Taxifahrerin in meinem Sample referieren kann.

<sup>5</sup> Vgl. den Interaktionsbegriff bei Suchman (2007), *Human-Machine Reconfigurations*, der genauer auf S. 123-124 erläutert wird.

<sup>6</sup> Varinia Bernau/Helmut Martin-Jung, „Wenn möglich bitte senden“, auf: *Sueddeutsche.de* vom 06.11.2014, online unter: <http://www.sueddeutsche.de/auto/zukunft-der-autonavigation-wenn-moeglich-bitte-senden-1.2204238>, zuletzt aufgerufen am 05.09.2017. „Seit 2004 [...] hat die Firma [TomTom] 75 Millionen Geräte verkauft. Die Zeit des rasanten Wachstums ist zwar vorbei, aber im Durchschnitt werden täglich immer noch 23.000 der kleinen Kästchen mit Saugfuß verkauft, das sind knapp 8,4 Millionen pro Jahr.“

<sup>7</sup> Alle diese technischen Aspekte werden im Folgenden als Bestandteile des ‚Navis‘ verhandelt.

<sup>8</sup> Das ‚Hybride‘ bricht mit dem Paradigma, Entitäten entweder als sozial oder als materiell zu bestimmen. Geräte sind damit weder nur neutraler Träger menschlichen Willens noch determinierender Faktor menschlicher Handlungen. „Each artifact has its script, its ‚affordance‘,

Diese Hybridität verstärken Fahrer und Navi in dem Moment, in dem sie ihre Praktiken und Prozesse dynamisch aneinander anpassen, Handlungs- oder Operationslogiken situativ aufeinander beziehen und dementsprechend ihre Funktionsweisen ändern. Solche reflexiven Routinen sind im Fokus des folgenden Textes. In ihren Interaktionen sammeln Fahrer\_innen Erfahrungen zu erwartbaren Funktionsweisen und programmierten Logiken der Navigationssoftware. Mit Trial-and-Error-Verfahren versuchen sie, diese zu erkennen und reagieren mit eingeübten Praktiken auf die Eigenlogiken ihrer Software. Doch auch auf der Seite der Software geschehen zur gleichen Zeit Anpassungsroutinen, die sich zunehmend auf die individuellen Fahrer\_innen einstellen. Je nach Aktualität und Programmierung der Software reichen diese Anpassungsprozesse von manuell getätigten oder automatischen Updates bis hin zur adaptiven und echtzeitfähigen „learning navigation“.<sup>9</sup>

Mit der Begriffswahl einer ‚lernenden‘ Navigation werden im öffentlichen Diskurs Parallelen zwischen Mensch und Maschine konstruiert, die der programmierten Reflexivität von Software mehr als eine bloß mechanische Reaktion auf die Aktion des Fahrers zuschreiben. Und auch Christoph Engemann und Andreas Sudmann konstatieren in ihrem kürzlich erschienenen Buch *Machine Learning*: „Im digitalen Wandel ist Lernen kein Privileg des Menschen mehr.“<sup>10</sup> Betrachtet man diese ‚lernende‘ Navigationstechnologie im Kontext des ‚autonomen Fahrens‘, so ist hier der Anspruch eingeschrieben, sich zunehmend an ihre Nutzer\_innen anzupassen, um eine vollständige Technologisierung des Autofahrens zu ermöglichen, in der die Technologie ganz „ohne Lenkrad“<sup>11</sup> und damit ohne Kooperation, Interaktion oder Assistenz des Menschen den Weg von einem Ort zum Anderen bestreiten kann: „Bis zum Ende dieses Jahrzehnt[s] sollen deutsche Automobile [...] ganz autonom sein – und von ihrem Fahrer lernen.“<sup>12</sup> Für die Navigation hieße das, den Dreiklang der Navigation von einer ‚selbsttätigen‘ Technologie ausführen zu lassen: Die „Bestimmung des augenblicklichen Standortes“, die Berechnung des Kurses „von einem Ausgangspunkt auf [einem] bestimmte[n] Weg zu einem Zielort [...] in möglichst optimaler Weise“ und das „Führen eines Fahrzeugs“ und

---

its potential to take hold of passerby and force them to play roles in its story.“ Bruno Latour, „On Technical Mediation – Philosophy, Sociology, Genealogy“, in: *Common Knowledge* 3, 2 (1994), S. 29-64: 31.

<sup>9</sup> Interview mit einem Softwareingenieur eines Softwareunternehmens für Navigationslösungen, Februar 2016.

<sup>10</sup> Christoph Engemann/Andreas Sudmann, *Machine Learning – Medien, Infrastrukturen und Technologien der Künstlichen Intelligenz*, Bielefeld, 2017.

<sup>11</sup> Daimler stellt sein Robotertaxi vor, das 2022 auf den Markt kommen soll: Joachim Becker, „Maschine mit menschlicher Gestik“, auf: *Sueddeutsche.de* vom 30.08.2017, online unter: <http://www.sueddeutsche.de/auto/smart-vision-eq-maschine-mit-menschlicher-gestik-1.3645606>, zuletzt aufgerufen am 05.09.2017.

<sup>12</sup> Joachim Becker, „Fahrplan in die Zukunft“, auf: *Sueddeutsche.de* vom 18.01.2014, online unter: <http://www.sueddeutsche.de/auto/autonomes-fahren-fahrplan-in-die-zukunft-1.1864972>, zuletzt aufgerufen am 05.09.2017.

Kurshalten auf dieses Ziel hin<sup>13</sup> wird in dem Moment vollständig an die Technik übergeben. In diesem Kontext erlernt die Navigationssoftware allmählich die Entscheidungs- und Handlungsfähigkeit der Navigationsaufgaben.

Der weite und durchaus mystifizierende Begriff des ‚Lernens‘, der auf solche aktuellen Entwicklungsmotive in der Automobilindustrie referiert, dient dem vorliegenden Text lediglich als Ausgangspunkt. Im Folgenden differenziere ich dieses ‚Lernen‘ in unterschiedlich gelagerte menschliche und maschinelle ‚Anpassungsroutinen‘, die unterschiedliche Arten dynamischer und reflexiver ‚Selbsttätigkeiten‘ aufweisen.

Aufbauend auf dem Ansatz von Lucy Suchman, soll der These eines gemeinsamen und zugleich Konkurrenz schaffenden Privilegs des auf Intelligenz beruhenden ‚Lernens‘ nachgegangen werden:

My approach has been to slow down discourses of the ‚smart‘ machine to attend closely to the practices through which purportedly intelligent and interactive artifacts are realized, including just what conceptions of intelligence and interaction are in play. The result of this is an enduring skepticism regarding the rhetorics of machine intelligence and an interest in demystifying the specific technologies and practices about which these discourses make their claims.<sup>14</sup>

Zwei Vignetten eröffnen dazu ein Spannungsverhältnis, in dem sich dynamische und reflexive Anpassungsroutinen von Fahrer und Navigationssoftware darstellen.

### 3. Reflexive Software

Jahrelang hatten sich Fahrer O. und sein Navi ‚gut verstanden‘. Sobald der Feierabend nahte, drückte er auf den Knopf, der die Spracherkennung des Navis freigab, sagte laut und deutlich ‚Nach Zuhause‘ und das Navi nannte ihm – egal wo er war – eine optimale Route zu seiner Heimadresse. Doch eines Nachmittags, als O. wie gewöhnlich ‚Nach Zuhause‘ in sein Navi sprach, antwortete das Navi – anstatt die Berechnung zu starten – mit den Worten: ‚Zieleingabe nicht erkannt‘. Es reagierte stattdessen nun auf die korrekte Formulierung ‚Nach Hause‘ und konnte die Formulierung ‚Nach Zuhause‘ nicht mehr verarbeiten, mit der O. die im Adressbuch unter ‚Zuhause‘ gespeicherte Heimadresse bisher sprachlich angesteuert hatte. Fahrer O. und ich stellten die Vermutung auf, dass ein Update im Hintergrund die Sprachkenntnisse der Software ‚verbessert‘ hatte, indem Programmierer\_innen sprachliche Ausnahmeregelungen der Software hinzugefügt hatten.<sup>15</sup>

Die Szene fängt ein plötzliches Unverständnis zwischen einem Fahrer und seiner Navigationssoftware ein und eröffnet den Blick auf zwei zu unterscheidende Interaktionsflächen zwischen Fahrer und Navi: Die Interaktion mit der

<sup>13</sup> Brockhaus, *Eintrag zur Navigation*, Bd. 19, 21. Ausgabe, Leipzig, Mannheim, 2006, S. 423.

<sup>14</sup> Ebd., S. 242 f.

<sup>15</sup> Vignette auf Basis meiner Feldforschung an drei Taxiständen in deutschen Städten.

Funktionsweise einer Software in Abgrenzung zur Interaktion mit der Anpassungsroutine derselben Software.

Zunächst zur Funktionsweise der Software: Die Szene zeigt einen Fahrer, der seine Software mit spezifisch eingeübten Praktiken aktiviert und die Software erst damit im täglichen Umgang für sich nutzbar macht: Mit der grammatikalisch fehlerhaften Formulierung ‚Nach Zuhause‘ passt er seine sprachliche Kommunikation an die Software an und entwickelt eigene Konversationsregeln im Kontext der spezifischen Interaktion. Er bezieht seine Praktik dabei auf bestimmte Vorstellungen, die er von den grundlegenden Prinzipien einer Funktionsweise der Software hat und davon, wie die Software auf den eigenen Input reagiert: Systematisch bricht er seine Formulierung auf und trennt das ‚Nach‘ als Ansteuerungsbefehl vom ‚Zuhause‘ als Name der gespeicherten Adresse. Diese Vorgehensweise macht sein „implizites Wissen“<sup>16</sup> über die Funktionsweise von Software sichtbar. Auf Basis seines Wissens über die technische Strukturiertheit tritt er der Software mit eingeübten Praktiken gegenüber, die sich allmählich als situierte Sprachweisen „eingeschliffen“<sup>17</sup> haben. Der Erfolg seiner Praktik hängt jedoch stets indirekt davon ab, wie die Software dieselbe Praktik interpretiert.

Gleichzeitig passt sich die Navigationssoftware an die Sprachregeln eines prototypischen Fahrers an, indem Ingenieur\_innen sprachliche Spezialfälle durch ein Update in das Softwareprogramm implementieren. Die sprachliche Logik der Software wird von den Ingenieur\_innen zunehmend an die Lebenswelt der Fahrer\_innen angepasst. Sprachliche Sonderregeln werden miteinbezogen, um eine prototypische Nutzung der Software noch bequemer zu gestalten. Das zeigt, dass auch bei den Ingenieur\_innen Vorstellungen von einer bestimmten Umgangsweise der Fahrer\_innen mit der Software wirken. Ingenieur\_innen wie Nutzer\_innen beziehen sich dabei lediglich auf ihre eigenen Vermutungen der gegenüberliegenden Funktions- oder Umgangsweise.

Eine solche wechselseitige Verschränkung der verwendeten Sprachregeln ‚Nach Hause‘ und ‚Nach Zuhause‘, die die Akteure vom jeweils anderen übernehmen, verweist auf eine wesentliche Prämisse in den Technoscience Studies: Mensch und Maschine konstituieren sich immer wechselseitig, indem sie eine stetige Bezugnahme auf die Prozesse oder Praktiken des jeweils anderen herstellen.<sup>18</sup> In ihrer wechselseitigen Veränderung der Wortwahl zeigt sich die

<sup>16</sup> In Bezug auf Michael Polanyi, *Implizites Wissen*, Frankfurt/M., 1966 [1985].

<sup>17</sup> „Vorgänge und Handlungen scheinen nur dann zu Automatismen zu werden, wenn der wiederholte Gebrauch sie *eingeschliffen* hat. Wer Automatismen beobachtet, sieht sich zurückverwiesen auf deren Entstehungsprozess.“ Hartmut Winkler, „These 13: Automatismen haben einen engen Bezug zur Wiederholung, zur Gewohnheit und zur Schemabildung“, in: Hannelore Bublitz/Roman Marek/Christina L. Steinmann/Hartmut Winkler (Hg.), *Automatismen*, Paderborn, 2010, S. 234-236 [Herv. i. O.].

<sup>18</sup> Ebd., S. 268. Die wechselseitige Konstitution beruht auf dem theoretischen Verständnis der Interaktion bei Suchman: „The significance of any action and the adequacy of its interpretation are judged indirectly, by responses to actions taken and by an interpretation’s usefulness

„soziomaterielle Handlungsfähigkeit“<sup>19</sup> beider Akteure, die die Konzeption von autonomen und strikt trennbaren Entitäten auflöst: Erst durch die wechselseitige Übernahme sprachlicher Regeln wird den Akteuren ihre Handlungsfähigkeit zuteil und es kommt zu einer geteilten Verständigung, die kollaborativ und *in situ* erst in der Interaktion entsteht.

Doch das Unverständnis geschieht nicht auf der Ebene der Funktionsweise der Spracherkennungssoftware, sondern aufgrund ihrer Anpassungsfähigkeit. Mit der Anpassungsfähigkeit entstehen vielfältige Formen und Schichten von Dynamiken innerhalb derselben Interaktion. Denn kompliziert wird es für den Fahrer erst in dem Moment, in dem sich durch unangekündigte Updates die bisherigen Funktionsweisen seiner Software verändern. Nicht die Logik einer operationalen Funktionsweise der Software, sondern die Logik einer dynamischen und unvorhergesehenen Anpassung der Software an die Lebenswelt des Fahrers führt in dieser Szene zum Missverständnis.

Die verkörperten Praktiken des Fahrers geraten in Konflikt mit der dynamischen Beschaffenheit der Software, während er den Konflikt mit ihrer operationalen Funktionsweise mittels seiner Formulierung ‚Nach Zuhause‘ bereits gelöst zu haben schien. Trotz oder gerade wegen der engen Bezugnahme und wechselseitigen Konstituierung der Sprachregeln kollidiert seine Nutzungsweise zwar nicht mit der mechanischen Funktionsweise, aber mit der dynamischen Anpassungsroutine der Software.

Das unerwartete Update, durch das die Software die Eingabe des Fahrers nicht mehr prozessieren kann, macht deutlich, dass Software kein stabiles Instrument ist, das eine einmal programmierte Funktion oder Aufgabe erfüllt. Die Fahrer\_innen sind vielmehr mit dynamischen Projekten konfrontiert, bei denen sich je nach Programmierung die Kommunikationslogiken oder Entscheidungsfindungsprozesse instantan ändern können. Das Unverständnis beruht damit weniger auf der Beschaffenheit und Funktionsweise von Software als vielmehr auf ihrer Flexibilität und Prozesshaftigkeit:

Importantly, software rarely constitutes a one-off event or finished product, but rather needs to be seen as an ongoing effort, a process of becoming. As such, software can be considered a project. As projects, software requires care, maintenance, updates, revision, and work. The continuously evolving, developing, and transforming landscape of software poses a challenge not only to the producers and users of software, but also to the researcher wishing to study it.<sup>20</sup>

---

in understanding subsequent actions. It is just this highly contingent process that we call interaction.“ Ebd., S. 123-124.

<sup>19</sup> Ebd., S. 267. „Soziomaterielle Handlungsfähigkeit“. Suchman bindet diese Begrifflichkeit an den agentellen Realismus von Karen Barad und soziomaterielle „Intra-Aktionen“ (Karen Barad, „Posthumanist Performativity toward an Understanding of How Matter Comes to Matter“, in: *Signs: Journal of Women in Culture and Society*, 28 [2003], S. 801-831).

<sup>20</sup> Taina Bucher, *Programmed Sociality: A Software Studies Perspective on Social Networking Sites*, Dissertation eingereicht an der Universität Oslo, 2012, S. 65.

Dieser projekthafte Charakter von Software, bei dem die Logiken der algorithmischen Prozesse stetig im Werden („a process of becoming“) begriffen sind und Veränderungen unangekündigt auftreten, stellt Nutzer\_innen wie Forscher\_innen – und damit die außenstehende Beobachtung an sich – vor Herausforderungen. Diese technische ‚Selbsttätigkeit‘ unterscheidet die Interaktion mit Software von der Interaktion mit anderen Technologien, Werkzeugen oder Instrumenten: Die Funktionsweisen von Software sind instabil, komplex verwoben, ständig veränderlich und diffus. Eine konkrete Logik der Funktionsweise ist damit für die Nutzer\_innen schwer von anderen Dynamiken zu abstrahieren und nachzuvollziehen.

Neu ist damit, dass sowohl Fahrer\_innen als auch Software zu einer Anpassung befähigt sind und diese auch ausführen. Es sind die Anpassungsroutinen, die in dieser Szene unbekannt sind: weder dass sie stattfinden noch nach welchen Prinzipien sie ablaufen. Und das gilt für beide Seiten: Ingenieur\_innen wie Nutzer\_innen vermuten eine Stabilität und keine Dynamik der Handlungslogiken des jeweils anderen. Die Ingenieur\_innen gehen nicht von einer interaktiven Veränderung der sprachlichen Formulierungen des Fahrers entlang seines spezifischen Gerätes aus und umgekehrt geht der Fahrer nicht von Updates der Software aus, die zu Veränderungen in der sprachlichen Formulierung der Software führen. Beide Seiten vermuten, dass nur sie allein eine unidirektionale Anpassung an die jeweils andere, starre Kommunikationsstruktur vornehmen.

Den Begriff der ‚Selbsttätigkeit‘<sup>21</sup> bindet Hartmut Winkler an den latourischen Begriff der Handlungsfähigkeit mit der Idee, dass ontologisch unterschiedliche Akteur\_innen – Mensch und Maschine – mit unterschiedlichen Handlungsfähigkeiten in einer Situation wirksam werden.<sup>22</sup> Ich unterscheide für diese Abhandlung die ‚Selbsttätigkeit‘ von der ‚Handlungsfähigkeit‘ und setze mit dem Begriff der ‚Selbsttätigkeit‘ einen besonderen Akzent auf die dynamischen, reflexiven und damit ständig veränderlichen Aspekte, die in der soziomateriellen Handlungsfähigkeit zwischen Fahrer und Software entstehen. In diesem Kontext ragt die Selbsttätigkeit dann gerade durch ihre Dynamik über die funktionelle Bezugnahme, die in der soziomateriellen Handlungsfähigkeit reflektiert wird, hinaus.

Eine interaktive Nutzung von Software setzt damit nicht nur ein Wissen über die technischen Funktionsweisen, sondern ebenso ein Wissen über die projekthaften Anpassungsroutinen der Software beim Nutzer voraus:

[W]e download updates and install patches, trusting that the vendors are sure that the changes are correct and complete. We blindly hope that the latest change to each program keeps it compatible with all of the rest of the programs on our sys-

---

<sup>21</sup> Winkler, These 9, S. 117.

<sup>22</sup> Ebd.



tem. We rely on much software that we do not understand and do not know very well at all.<sup>23</sup>

Die Anpassung der Software, die ihre Sprachlogiken verbessert, geschieht hier zu einem konkreten Zeitpunkt, der vom Taxifahrer durch das plötzliche Unverständnis deutlich bemerkt wird. Doch für die Navigationssoftware im Straßenverkehr sind neben solchen punktuellen auch zunehmend kontinuierliche und sukzessive Anpassungsroutinen von Bedeutung. Diese Entwicklung kulminiert aktuell in einer adaptiven und echtzeitfähigen ‚lernenden‘ Navigation, die einen weit höheren Anspruch auf ‚Selbsttätigkeit‘ in ihrem Namen artikuliert als ihre mechanischen Prozesse es ihr tatsächlich ermöglichen: Die ‚lernende‘ Navigationssoftware erhebt Bewegungsdaten ihrer Fahrer\_innen und extrahiert daraus beliebte Orte, Heimadresse sowie gern gefahrene Strecken, um den Fahrer\_innen noch passendere Strecken, Umwege oder individuellere Erinnerungshilfen anbieten zu können. Die Software erstellt dazu ein Profil ihrer Fahrer\_innen, das aus den historischen Daten vergangener Fahrten ermittelt wird:

Beispielweise fährt ein Taxifahrer jeden Tag seine Schleichwege zum Flughafen, wo er seinen Arbeitstag mit den ersten – meist besonders rentablen – Touren beginnt. Vor allem wer seine Stadt kennt, so ein Softwareingenieur im Interview, findet meist eine schnellere Route als das Navi, das seine Route meist über die Hauptstraßen und erst im zweiten Schritt über die Nebenstraßen berechnet. Die Idee der ‚lernenden‘ Navigation ist es darum, mithilfe der Fahrer\_innen Daten zu unverzeichneten Straßen zu erheben aber auch Daten über den aktuellen Verkehrsfluss zu generieren. Darüber hinaus werden spezifische Fahrgewohnheiten aufgezeichnet, beispielsweise wie schnell ein Fahrer mit seinem Auto auf einer Strecke fährt, um individuelle ‚Fahrerprofile‘ erstellen zu können. Das Ziel dieser ‚learning navigation‘ ist es, dass der Fahrer morgens in sein Auto steigt und ohne seine manuelle Zieleingabe ‚erkennt‘ die Navigationssoftware nach der Fahrt um die ersten fünf oder sechs Straßenecken durch den Abgleich des Bewegungsprofils mit den historischen Daten in der Datenbank, dass der Fahrer vermutlich zum Flughafen fährt. Die Software ‚loggt‘ daraufhin die für den spezifischen Fahrer bewährte Route ‚ein‘ und zeigt den aktuellen Verkehrsfluss für die vor ihm liegende Strecke an. Für eine solche ‚learning navigation‘ bedarf es jedoch der Aufzeichnung aller Bewegungen des Autos.<sup>24</sup>

Gerade wegen dieser vielschichtigen – punktuellen und kontinuierlichen – Anpassungsroutinen, durch die beispielsweise aktuelle Stauangaben für individuell beliebte Strecken bereitgestellt werden, wird Navigationssoftware zunehmend auch auf bekannten Strecken und von ortskundigen Fahrer\_innen genutzt, um nicht nur möglichst schnell, sondern auch bequem und staufrei ans Ziel zu kommen. Durch dieses erweiterte Softwareangebot etabliert sich

<sup>23</sup> Eldad Eilam, *Reversing: Secrets of Reverse Engineering*, Indianapolis, IN, 2005, S. vii.

<sup>24</sup> In Anlehnung an das Interview mit einem Softwareingenieur.

ein zunehmender Austausch zwischen ortskundigen Fahrer\_innen und ‚ortskundiger‘ Software. Navigationssoftware dient damit nicht mehr nur einem ortsunkundigen Fahrer, sondern Software und Fahrer fordern jetzt wechselseitig Informationen voneinander.<sup>25</sup> Alain de Taeye, Vorstandsmitglied beim niederländischen Navigationskonzern TomTom, fordert darum im Interview mit der Süddeutschen, dass sich

[d]ie Art und Weise, wie wir Updates verteilt haben [, ändern muss, denn sie sei] veraltet. [...] [W]ir müssen schneller dabei werden, die Änderungen zum Kunden zu bringen. [Denn klar ist (Einschub des Journalisten)]: Automatisiertes Fahren funktioniert nur mit ständigen Updates. [...] Viele Kunden werden zumindest für eine Zeit lang aber lieber noch bei den gewohnten Kästchen für die Windschutzscheibe bleiben. Auch die haben sich schließlich weiterentwickelt. Die neuesten TomToms zum Beispiel kommen mit lebenslangen Updates und Internetanbindung.<sup>26</sup>

Kontinuierliche Updates dienen damit sowohl dem Fahrer wie auch der Software: Während der Fahrer für seine Fahrsituation Echtzeit-Informationen zum Verkehrsfluss erhält, erhält die Software aktuelle Informationen aus der Fahrsituation des Fahrers. Das Interesse an schnelleren Updates geht damit nicht zwangsläufig vom Fahrer aus, sondern auch vom Anspruch einer ‚lernenden‘ Navigation.

Diese Entwicklung hin zur ‚learning navigation‘ zeigt, dass die maschinellen Anpassungsroutinen zunehmend auf unterschiedlichen Schichten entstehen. Während ein Update nur punktuelle Veränderungen in der Software hervorruft, die von Ingenieur\_innen getätigt wurden, erzeugt die Programmierung dynamischer Algorithmen<sup>27</sup> einen schnelleren Datenaustausch und eine ver-

<sup>25</sup> „In acht von zehn Fällen können wir den aktuellen Verkehr auf der Grundlage historischer Daten vorhersagen“, erzählt etwa der Nokia-Entwickler Jussi Koski. „Wir haben sehr früh angefangen“, sagt auch Alain de Taeye von TomTom. 280 Milliarden Kilometer haben Nutzer von TomTom bereits zurückgelegt. Wenn ein Einzelner das schaffen wollte, müsste er 700.000 Jahre lang ununterbrochen am Steuer sitzen. Die entsprechenden Algorithmen zu ihrer Auswertung, die in jahrelanger Arbeit entstanden sind, gehören daher neben den Verkehrsdaten selbst zu den wichtigsten Geschäftsgeheimnissen der Navigationsfirmen, die streng gehütet werden.“ Bernau/Martin-Jung (2014), „Wenn möglich bitte senden“.

<sup>26</sup> Bernau/Martin-Jung (2014), Wenn möglich bitte senden.

<sup>27</sup> In Bezug auf ein anderes Schulungs- oder Lernszenario schreibt Felix Stalder zu dynamischen, adaptiven Algorithmen: „[M]an denke an die typische Ansage zu Beginn eines Anrufs bei einer Telefonhotline – ‚Dieser Anruf kann zu Schulungszwecken aufgezeichnet werden.‘ Geschult werden sollen aber immer öfter nicht die Angestellten im Call-Center, sondern die Algorithmen. Diese sollen lernen, den Persönlichkeitstypus des Anrufers zu erkennen und darauf basierend das Skript für die Interaktion mit den Kunden anzupassen. [...] Es [...] ist schon jetzt deutlich, dass an vielen Fronten die Grenzen zwischen dem, was als kreativ, und dem, was als mechanisch verstanden wird, verschoben werden. [...] Die Algorithmen, die für solche Aufgaben eingesetzt werden, sind jedoch nicht mehr nur einfache Abfolgen von statischen Instruktionen. Sie werden nicht immer wieder unverändert ausgeführt, sondern sind hochgradig dynamisch und adaptiv. Die heute zur Verfügung stehende Rechenleistung wird genutzt, um Programme zu schreiben, die sich halbautomatisch und auf der Grundlage von Feedback selbst verändern und bessern können.“ Felix Stalder, *Kultur der Digitalität*, Frankfurt/M., 2017, S. 176 f.

kehrsbasierte und profilorientierte Datenauswertung, die jenseits punktueller Softwareupdates sukzessive die Aktionsmuster der Software selbst verändern. Für den Fahrer wird es zunehmend komplizierter, in Software eingeschriebene Logiken und Funktionsweisen zu antizipieren und daraufhin entlang eigener Einschätzungen zu reagieren und gegebenenfalls mit eigenen Praktiken nachjustieren.

Die Fähigkeit zur stetigen Veränderung von Funktions- und Anpassungslogiken ist bestimmend für Software, egal ob sie auf der programmierenden Ebene durch Praktiken von Ingenieur\_innen oder auf der programmierten Ebene der Software selbst durch adaptive Algorithmenprogrammierung stattfindet: Durch punktuelle Softwareveränderungen als Resultat der stetigen Fortentwicklung in den Softwarebüros wie auch durch eine mechanische Selbsttätigkeit der Software selbst ist sie – wie es Taina Bucher im obigen Zitat formuliert – ein „stetig im Werden begriffenes Projekt“.

Die Selbsttätigkeit der lernenden Navigationssoftware geht damit über die Handlungsfähigkeit eines Automaten hinaus, selbst wenn diese Selbsttätigkeit auf mechanischen Prozessen beruht: Nicht unbedingt die mechanische Komplexität, sondern vielmehr die Vielschichtigkeit projekthafter und mechanischer Dynamiken unterscheidet die lernende Software vom Automaten. Weniger ihre Funktionsweise als vielmehr ihre Anpassungsroutinen erzeugen die dynamische und reflexive Selbsttätigkeit, die sie von anderen Techniken unterscheidet.

#### 4. Software als Aufforderung

Es stellt sich darum im Folgenden die Frage nach dem Umgang des Fahrers mit dieser vielschichtigen und anpassungsfähigen Selbsttätigkeit von Software. Denn, so Lucy Suchman, ist trotz oder gerade wegen der Anerkennung technischer Selbsttätigkeit auch die menschliche Selbsttätigkeit, die zeitgleich stattfindet, relevant:

I would propose that the price of recognizing the agency of artifacts need not be the denial of our own. Now that the agencies of things are well established, might we not bring the human out from behind the curtain, so to speak, without disenchantment? This requires, among other things, that we acknowledge the curtain's role. Agencies – and associated accountabilities – reside neither in us nor in our artifacts but in our intra-actions. The question [...] is how to configure assemblages in such a way that we can intra-act responsibly and generatively with and through them.<sup>28</sup>

Mit dem Bild des Vorhangs verdeutlicht Suchman die Undurchsichtigkeit, mit der die Fahrer\_innen in Interaktion mit der zunehmend reflexiven Navigationssoftware konfrontiert sind. Ihre Dynamik, die sowohl menschlich – von

---

<sup>28</sup> Suchman (2007), *Human-Machine Reconfigurations*, S. 285.

den Ingenieur\_innen – als auch zunehmend maschinell – von der adaptiv konstruierten ‚lernenden‘ Navigationssoftware – getätigt wird, erzeugt plötzliche oder kontinuierliche Veränderungen der technischen Logiken und Funktionsweisen.

Im Alltag werden jedoch insbesondere die adaptiven Mechanismen der Software häufig pragmatisch, versehentlich oder bewusst gestoppt, behindert oder unterbunden:

Alle bis auf zwei Taxifahrer benutzen sehr alte Navigationssysteme. Die meisten machen keine [bewusst getätigten] Softwareupdates. Auch kaufen sie keine neuen Geräte – entweder aus Kostengründen oder auch weil ihnen der Funktionsumfang ihrer Geräte genügt. Ein Taxifahrer ergänzt, dass er seit Jahren kein Update mehr gemacht hat und dadurch sein Navi inzwischen besonders gut ‚kennenlernen‘ konnte.<sup>29</sup>

Der Verweis auf das ‚Kennenlernen der Navis‘ zieht sich durch meine Gespräche mit Taxifahrern. Sie wissen, an welchen Kennzahlen ihr Navi Staus misst, welche Straßen es (noch) nicht kennt, warum oder an welchen Orten man seinen Anweisungen misstrauen sollte und an welchen Straßenecken es einer manuellen Neuberechnung bedarf, um die aktuelle Verkehrslage zu erhalten.

Es ist die ‚unauffällige Kreativität‘<sup>30</sup> – wie Michel de Certeau es nennt –, mit der sich die befragten Fahrer mit eigenen Anpassungsroutinen den technischen Anpassungsroutinen widersetzen. ‚Scheinbar mühelos‘<sup>31</sup> konfrontieren sie damit effiziente Managementsysteme, die einen intensiven Datenaustausch mit den Autos auf der Straße anstreben, mit *anderen* Logiken. Die Praktiken der Taxifahrer formen die Sachverhalte nach ihren eigenen Logiken und entziehen sich damit den Logiken auf der technischen Seite. Als Gewohnheiten und damit als ‚Automatismen‘<sup>32</sup> haben sich diese Praktiken auf der Straße ‚eingeschliffen‘. Ihre Eingeschliffenheit macht sie in gewisser Weise mächtig, denn sie haben das Potenzial, technische Neuentwicklungen zu überdauern.<sup>33</sup>

<sup>29</sup> Vignette auf Basis meiner Feldforschung an drei Taxiständen in deutschen Städten.

<sup>30</sup> Michel de Certeau, ‚Praktiken im Raum‘, in: Jörg Dünne/Stephan Günzel (Hg.), *Raumtheorie. Grundlagentexte aus Philosophie und Kulturwissenschaften*, Frankfurt/M., 2006 [1980], S. 330-342: 343.

<sup>31</sup> Hannelore Bublitz, ‚These 2: Automatismen beinhalten einen qualitativen Sprung: Aus der wiederholten Einschleifung durch Übung entsteht – paradoxerweise gerade das Neue: spielerisch-mühevolle Perfektion‘, in: dies./Roman Marek/Christina L. Steinmann/Hartmut Winkler (Hg.), *Automatismen*, München, 2010, S. 23-26: 23.

<sup>32</sup> Der Begriff ‚Automatismen‘ ist in diesem Text sehr eng gefasst und bezeichnet mit seinem Fokus auf die Fahrpraktiken nur einen kleinen Ausschnitt des Gesamtkonzepts von Automatismen, wie es im Graduiertenkolleg der Universität Paderborn verhandelt worden ist.

<sup>33</sup> ‚Automatismen [...] sind – wie Technologien – Praktiken der Formung und Formierung, die in Kategorien der Funktion zu denken sind und welche sich auf eine im Voraus berechenbare Rationalität aber nicht reduzieren lassen. Automatismen sind unhintergebar und übersteigen den Horizont jeder subjektiven, willentlichen Verfügbarkeit. Sie fügen sich zu einem Regime hochwirksamer ‚Logiken‘ zusammen, ihre Wirkungen sind aber – aufgrund der unüberschaubaren Pluralität der beteiligten Kräfte – in gewisser Weise Zufallseffekte. Sie verdanken sich nicht dem Willen eines planvoll handelnden Subjekts, der sich in ihnen manifestiert, sondern sind Bestandteil eines wirkmächtigen Arrangements von Dingen, Zeichen und Subjekten.‘

Mit der Ablehnung technischer Neuerungen durch die Beibehaltung alter Navigationstechnologie erhalten sie ihre Kommunikationslogiken und verhindern, eingeübte Praktiken wieder neu formieren zu müssen. Stattdessen nutzen sie die Stabilität und Begrenztheit der technischen Funktionsweisen dazu, ihre Praktiken weiter an die Software anzupassen und auszubilden anstatt durch technische Anpassungen wie Aktualisierungen und Updates mit neuen Funktionsweisen der Software konfrontiert zu sein:

Taxifahrer J. erzählt mir bei einer Zigarette am Bahnhof, dass er sein Navi nicht nur dazu nutzt, einen Ort zu finden, sondern vielmehr auch dazu, seinen Fahrgästen ihre Ankunftszeit zu nennen. Die errechnete Ankunftszeit seines Navis stimmt jedoch meist nicht, da sein Navi ein älteres Modell ist und nur über das Radio Staudaten auswertet. Es hat keinen Stauwarner, der mit einer SIM-Card über Mobilfunk die Daten in Echtzeit abrufen. Auch das Kartenmaterial ist veraltet, und so kennt das Navi manche neuen Straßen noch nicht. Es empfiehlt somit oft andere – meist auch längere – Strecken, als er sie fahren würde. Wenn er sympathische Fahrgäste hat, sagt er ihnen: „Das Navi schickt uns jetzt eigentlich einen Umweg, ich kenne aber eine Abkürzung und dadurch spart ihr Euch 5 Euro.“<sup>34</sup>

Wie die eingangs geschilderte Szene verweist auch diese auf die zwei unterschiedlichen Interaktionsflächen zwischen Fahrer und Navi: Die Interaktion mit der Funktionsweise einer Software in Abgrenzung zur Interaktion mit der Anpassungsroutine derselben Software.

Die technischen Erläuterungen des Fahrers über die nicht-existenten Update-Möglichkeiten seines Gerätes machen deutlich, dass er sich mit den Anpassungsroutinen seiner Software auseinandersetzt und von einer *ent-dynamisierten* Software ausgeht. Er verweist darauf, dass das Navi keine SIM-Karte hat und damit keine automatischen Updates des Kartenmaterials oder aktuelle Verkehrsflussdaten jenseits der Radio-Schnittstelle in die Berechnung der Software miteinbezogen werden. Anders als der Taxifahrer in der Eingangsszene achtet dieser Fahrer darauf, dass die Software sich nicht selbsttätig aktualisiert, damit die ihm bekannten Funktionsweisen beibehalten bleiben. Er setzt ‚seine‘ Navigationssoftware damit als ein stabiles Instrument ein, dessen Funktionsweisen ihm bekannt sind und nutzt es kontrolliert für spezifische Zwecke. Im Wissen darum, welche Straßen die Software nicht kennt und welche Informationen es *nicht* in die Berechnung miteinbezieht, entwickelt der Fahrer im engen Wechselspiel mit seinem Gerät selbst situative Anpassungsroutinen. Die bekannte konservative Rechenweise der Software nutzt er als Kommunikationsstrategie für seine Fahrgäste, um diese von seiner Ortskenntnis zu überzeugen.

Die Praktik des Fahrers zeigt, dass die algorithmischen Prozesse – welche Pläne, Instruktionen oder anderen Formen von vorgeschriebener Repräsentation sie auch abbilden – immer nur Angebote für die *In-situ*-Formen der sozia-

---

Hannelore Bublitz/Roman Marek/Christina L. Steinmann/Hartmut Winkler, „Einleitung“, in: dies. (Hg.), *Automatismen*, München, 2010, S. 9-16: 10.

<sup>34</sup> Feldnotiz vom Taxistand der Kleinstadt A. im Oktober 2015.

len Handlungen von Techniknutzer\_innen darstellen. Technische Prozesse können eine soziale Handlung nicht vollständig spezifizieren<sup>35</sup> und ebenso wenig haben langfristige, technologische Entwicklungsvorhaben Einfluss auf die aktuelle Techniknutzung im Vollzug. Die verkörperten Praktiken des Fahrers ziehen keine vorgebahnten bzw. „versteckten Nutzungsanweisungen“<sup>36</sup> konform nach, verfestigen keine von der Technik intendierte „Handlungsgrammatik“<sup>37</sup> konstruktiv auf der situativen und individuellen Handlungsebene oder schreiben zwangsläufig rationale Logiken der Technik fort.

Vielmehr nutzt der Taxifahrer hier nur die suggestive Wirkung, die die Vorgabe einer technischen Routenregulierung des Navis auf seine Fahrgäste hat:

[S]oftware does not merely instruct machines, but also the conduct of those who use the machines. This is not to say, however, that software instructs in a deterministic way. By means of its execution, software becomes suggestive of things, in that it enables and constrains certain kinds of action. In this sense, software can be understood in terms of the concept *affordance*.<sup>38</sup>

Der Taxifahrer impliziert, dass die Sichtbarkeit seines Navigationsgerätes eine Suggestivkraft auf seine Fahrgäste hat, weil diese vom Gerät eine kostengünstige Route erwarten. Indem er den Fahrgästen auf Basis der wesentlich begrenzten Fähigkeiten seines Gerätes eine noch kürzere oder schnellere Route empfiehlt, interagiert er mit dem Aufforderungscharakter der Software – jedoch in konträrer Weise: Er befolgt die Routenempfehlungen des Gerätes nicht, sondern erläutert entlang der wesentlich konservativen – weil entdynamisierten – Berechnung des Gerätes seine eigenen Ortskenntnisse, um den Fahrgästen seine kurze Strecke anzupreisen, die sie mit dem Navi allein nicht gefunden hätten.<sup>39</sup>

Seine Bezugnahme auf die Navigationstechnologie verdeutlicht hier erneut die wechselseitige Konstituierung soziomaterieller Handlungsfähigkeiten zwischen Fahrer und Navi: Die Praktiken des Fahrers werden von der Software zwar nicht determiniert, aber wesentlich mitkonstituiert. Sie formalisiert die Praktiken des Fahrers zwar nicht, übt aber eine interaktive Wirkung auf ihn und seine Praktiken aus, indem sie einen Vorschlag für seine Praktiken suggeriert, zu dem sich der Fahrer positioniert und ihn nach den eigenen Vorstellungen nutzt und verkehrt.

<sup>35</sup> Suchman (2007), *Human-Machine Reconfigurations*, S. 16.

<sup>36</sup> Winkler (2010), These 9, S. 7.

<sup>37</sup> Rob Kitchin/Martin Dodge, *Code/Space: Software and Everyday Life*, Cambridge, MA, 2011, S. 89.

<sup>38</sup> Bucher (2012), *Programmed Sociality*, S. 51.

<sup>39</sup> Der Fahrer muss selbstverständlich damit rechnen, dass die Fahrgäste die Kürze der Strecke auch auf ihrem eigenen Smartphone prüfen, womit sich noch eine zusätzliche Reflexionsschicht in die Aushandlungen einzieht. Dieser kann ich leider im Rahmen dieses Textes nicht explizit nachgehen.

Umgekehrt konstituiert auch der Fahrer die Handlungsfähigkeit seiner Software wesentlich mit, indem er sie durch die Nicht-Aktualisierung für seine Zwecke konfiguriert: Er reduziert sie auf abschätzbare Funktionsweisen, indem er ihre Anpassungsroutinen beschränkt.

Die eingeschliffenen Automatismen der befragten Fahrer, egal ob sie aus Kostengründen, Bequemlichkeit, Nicht-Beachtung, Unkenntnis oder bewussten Intentionen ‚veraltete‘ Navigationssoftware nutzen, wirken den technischen Entwicklungsstrategien aktiv entgegen. Mit ihren eigenen Selbsttätigkeiten überdauern sie die dynamisierenden Praktiken in den Softwarebüros, die dynamisierten Prozesse der Software sowie die Ideen einer ‚lernenden‘ Software des gesamten Industriezweiges. Die Navigationssoftware wird von jedem Fahrer immer wieder neu und je nach Situation als stabiles Instrument oder flexibles Projekt konfiguriert, genutzt und etabliert.

Suchman konstatiert, dass sich Fragen nach einer Interaktion zwischen Mensch und Maschine häufig darauf konzentrieren, ob Maschinen als ‚Objekte‘ behandelt werden oder ob sie eines Tages erfolgreich die Fähigkeiten eines autonomen menschlichen ‚Subjekts‘ ‚mimen‘ können.<sup>40</sup> Auch in der öffentlichen Debatte um das autonome Fahren sind diese Dualismen sichtbar, wie die Verweise auf eine lernende Navigation und das Fahren ‚ohne Lenkrad‘ zeigen. Doch auch neuere Software kann die technischen Leerstellen, an denen die Nutzer\_innen mit ihren Praktiken aktiv werden, nie vollständig schließen. Dennoch macht gerade die zunehmende Dynamik der Software die Funktionsweise für die Fahrer\_innen stetig weniger plausibel und nachvollziehbar. Die Umgangsweisen ortskundiger Taxifahrer\_innen, die Navigationssoftware oft bereits von Anbeginn ihrer technischen Entwicklung nutzen, verweisen auf die kontingente Stabilisierung, die die Arrangements zwischen Mensch und Maschine auf den heutigen Straßen darstellen:

The alternative perspective suggested here takes persons and machines as contingently stabilized through particular, more and less durable, arrangements whose reiteration and/or reconfiguration is the cultural and political project of design in which we are all continuously implicated.<sup>41</sup>

Das Navigieren ist damit immer als ein kontingent stabilisiertes Arrangement zwischen Fahrer und Software zu verstehen, das kulturell, ökonomisch und politisch stetig neu verhandelt wird.

## 5. Praktiken aktivieren, kogenerieren und überdauern

In der Navigationsroutine kollidieren dynamische Softwarealgorithmen mit den situativen Wegfindungspraktiken ortskundiger Taxifahrer. Hintergrundfolie zu den Aushandlungsszenarien, die hier untersucht wurden, bildet die tech-

<sup>40</sup> Suchman (2007), *Human-Machine Reconfigurations*, S. 285.

<sup>41</sup> Ebd.

nische Entwicklung einer zunehmend ‚lernenden‘ Navigationstechnologie, die an die Entwicklung des ‚autonomen Fahrens‘ gekoppelt ist: Mit dem anvisierten Abbau des Lenkrades geht neben der Positionierung und der Auswahl der Route auch die dritte Hauptkomponente der Navigation – die Steuerung des Fahrzeugs zum Ziel – in den Kompetenzbereich von Software über.

Vor diesem Hintergrund gerät die Abbiegung an der städtischen Straßenkreuzung zwischen materialisierten und verkörperten Wegfindungspraktiken zunehmend unter Spannung: Die Updates der Navigationssoftware konkurrieren mit dem ‚Einschleifen‘ von Praktiken der Taxifahrer um das Privileg einer selbsttätigen Handlungsfähigkeit und damit um das Privileg, sich den Mechanismen der anderen Seite anpassen zu können. Software und Fahrer entwickeln zeitgleich eigene Anpassungsroutinen, mit denen sich die Software an den Fahrer und der Fahrer an die Software anpasst: Softwarefirmen etablieren mit den Algorithmen einer ‚learning navigation‘ einen immer dynamischeren Datenaustausch zwischen Fahrer\_innen und Software, der mit der Erhebung immer präziserer Fahrerdaten einhergeht und profilorientiertere Routen für den Fahrer anbietet. Im Rahmen dieser technologischen Entwicklung sind Fahrer\_innen zunehmend mit komplexeren und vielschichtigeren Anpassungsroutinen der Software konfrontiert, die den Blick auf grundlegende Funktionsweisen der Software erschweren.

Gleichzeitig begrenzen befragte Taxifahrer die ‚Selbsttätigkeit‘ ihrer Software, indem sie ihre genutzten Softwareversionen nicht aktualisieren oder keine neueren Gerätemodelle erwerben. Damit reduzieren sie die Software auf bekannte Funktionsweisen und bauen ihre Kenntnisse daran weiter aus, um die technischen Informationen situativ und kontrolliert für bestimmte Zwecke einzusetzen, beispielsweise um mit eigens ausgewählten Fahrstrecken bei ihren Fahrgästen zu werben.

Selbst wenn aktuellere Navigationssoftware durch echtzeitfähigen Datenaustausch und die adaptive Profilbildung von Fahrer\_innen immer passendere Strecken anbietet, lassen sich die Leerstellen, die die Navigationstechnologie für die Praktiken des Fahrers im Vollzug offenlässt, nie vollständig schließen. Die Fahrerpraktiken sind stets in kreativer Unauffälligkeit tätig und interpretieren technische Angebote und Aufforderungen nach ihren ganz eigenen Prinzipien. Bis zum Abbau des Lenkrades kogenerieren sie in einem unfertigen Dazwischen selbsttätig die Funktionalität ihrer Navigationssoftware im engen Wechselspiel mit derselben.



## Literatur

- Barad, Karen, „Posthumanist Performativity toward an Understanding of How Matter Comes to Matter“, in: *Signs: Journal of Women in Culture and Society*, 28 (2003), S. 801-831.
- Becker, Joachim, „Fahrplan in die Zukunft“, auf: *Sueddeutsche.de* vom 18.01.2014, online unter: <http://www.sueddeutsche.de/auto/autonomes-fahren-fahrplan-in-die-zukunft-1.1864972>, zuletzt aufgerufen am 05.09.2017.
- Becker, Joachim, „Maschine mit menschlicher Gestik“, auf: *Sueddeutsche.de* vom 30.08.2017, online unter: <http://www.sueddeutsche.de/auto/smart-vision-eq-maschine-mit-menschlicher-gestik-1.3645606>, zuletzt aufgerufen am 05.09.2017.
- Bernau, Varinia/Martin-Jung, Helmut, „Wenn möglich bitte senden“, auf: *Sueddeutsche.de* vom 06.11.2014, online unter: <http://www.sueddeutsche.de/auto/zukunft-der-autonavigation-wenn-moeglich-bitte-senden-1.2204238>, zuletzt aufgerufen am 05.09.2017.
- Brockhaus, *Eintrag zur Navigation*, Bd. 19, 21. Ausgabe, Leipzig, Mannheim, 2006, S. 423.
- Bublitz, Hannelore, „These 2: Automatismen beinhalten einen qualitativen Sprung: Aus der wiederholten Einschleifung durch Übung entsteht – paradoxerweise gerade das Neue: spielerisch-mühevolle Perfektion“, in: dies./Roman Marek/Christina L. Steinmann/Hartmut Winkler (Hg.), *Automatismen*, München, 2010, S. 23-26.
- Dies./Marek, Roman/Steinmann, Christina L./Winkler, Hartmut, „Einleitung“, in: dies. (Hg.), *Automatismen*, München, 2010, S. 9-16.
- Bucher, Taina, *Programmed Sociality: A Software Studies Perspective on Social Networking Sites*, Dissertation eingereicht an der Universität Oslo, 2012.
- de Certeau, Michel, „Praktiken im Raum“, in: Jörg Dünne/Stephan Günzel (Hg.), *Raumtheorie. Grundlagentexte aus Philosophie und Kulturwissenschaften*, Frankfurt/M., 2006 [1980], S. 330-342.
- Eilam, Eldad, *Reversing: Secrets of Reverse Engineering*, Indianapolis, IN, 2005.
- Engemann, Christoph/Sudmann, Andreas, *Machine Learning – Medien, Infrastrukturen und Technologien der Künstlichen Intelligenz*, Bielefeld, 2017.
- Ethnografische Feldnotizen aus teilnehmender Beobachtung und mündlichen, teilstandardisierten, leitfadengestützten Einzelgesprächen mit Taxifahrern an Taxiständen der Kleinstadt A., Oktober 2015, der Großstadt B., Oktober 2015, und der Großstadt C., September 2016.
- Interview mit einem Softwareingenieur eines Softwareunternehmens für Navigationslösungen, Februar 2016.
- Jacobsen, Alice Juel, „Vignettes of Interviews to Enhance an Ethnographic Account“, in: *Ethnography and Education* 9, 1 (2014), S. 35-50.
- Kitchin, Rob/Dodge, Martin, *Code/Space: Software and Everyday Life*, Cambridge, MA, 2011.
- Latour, Bruno, „On Technical Mediation – Philosophy, Sociology, Genealogy“, in: *Common Knowledge* 3, 2 (1994), S. 29-64.
- Polanyi, Michael, *Implizites Wissen*, Frankfurt/M., 1966. [1985]
- Suchman, Lucy, *Human-Machine Reconfigurations. Plans and Situated Actions*, 2. Aufl., Cambridge, New York, NY, 2007.
- Stalder, Felix, *Kultur der Digitalität*, Frankfurt/M., 2017.

- Winkler, Hartmut, „These 9: Automatismen sind Technik und haben einen privilegierten Bezug auf Technologie“, in: Hannelore Bublitz/Roman Marek/Christina L. Steinmann/Hartmut Winkler (Hg.), *Automatismen*, Paderborn, 2010, S. 113-117.
- Ders., „These 13: Automatismen haben einen engen Bezug zur Wiederholung, zur Gewohnheit und zur Schemabildung“, in: Hannelore Bublitz/Roman Marek/Christina L. Steinmann/Hartmut Winkler (Hg.), *Automatismen*, Paderborn, 2010, S. 234-236.