Algorithms Curate Data: Four Perspectives on Data-Based Working Conditions, Using the Example of Route and Job Planning

Annelie Pentenrieder

Users of software services rarely come into contact with data in their everyday lives. There are algorithms working in the background to select, sort, classify and evaluate data for users as part of a process that turns data into information presented on a display. This makes algorithms the gatekeepers that mediate between users and relevant data. The following chapter takes up the question of how algorithms curate data for users while at the same time influencing social arrangements at work and in everyday life. For a theoretical study of data practices, one can enquire into the junctures at which data production and processing data by algorithms play a role for users and consequently must be disclosed.

There is a decades-long tradition in the use of algorithmic means to solve problems by calculating information on the basis of selected data. The omnipresence of algorithmic decision-making we can see in today's world of work and everyday life, however, is new. Algorithms can recommend books, select job applicants, sort e-mails, prioritize information in a wide variety of search engines, and help drivers navigate through city traffic. This is how algorithms have a major impact on the practices and decisions of those who use them and rely on their data analysis. Also the data on which algorithms perform their calculations are collected, processed, and prepared for algorithmic evaluation under conditions that are beyond users' view. And yet data processing does not conclude the moment a data record or a data-collection method (e.g., continuous, real-time data tracking) is set. Therefore, agorithms themselves – the calculation logics under which data are transposed into information for users – must be understood as opaque data practices in their own right. With this in mind, the focus of this chapter is on four different forms of empirical access in which drivers in the logistics sector interact with their software of route and job planning. The four situation descriptions modelled here can be used to illustrate the novelty of working conditions that are subject to algorithm- and data-driven arrangements that are multi-faceted and thus elusive:

> These algorithmic systems are not stand alone little boxes, but massive, networked ones with hundreds of hands reaching into them, tweaking and tuning, swapping out parts and experimenting with new arrangements. If we care about the logic of these systems, we need to pay attention to more than the logic and control associated with singular algorithms. We need to examine the logic that guides the hands, picking certain algorithms rather than others, choosing particular representations of data, and translating ideas into code. (Seaver 2014, 10)

Not only are the data themselves opaque for users, but also the logics that algorithms use to process these data into information. It is not just the results that algorithms recommend that intervene in users' social relationships. There are even social effects stemming from the fact that algorithms, and the underlying data are nearly impervious to scrutiny at this point in time and are used in ways that are opaque. In the following examples, the opacities involved in the process that uses algorithmic calculations to convert data records into information will be explicated, based on work practices of drivers in the logistics sector. In that setting, algorithms take different approaches with regard to the visibilities of information about road traffic and working conditions. This will shed light on the algorithmic architectures in which algorithms structure the social fabric of the people who use them. Using this spatial-theoretical metaphor, I explore the question of how "algorithmically framed perspective" can be "curated" anew, based on the perspective of the user.

Route Planners Invisibly Govern Spatial Access with their Selection of Routes

Anyone who asks a digital route planner to chart a path from Gendarmenmarkt in Berlin to the main rail station will receive a detailed description presenting step-by-step directions indicating where to turn right or left and showing how long it takes to reach the destination. As there are countless ways to get from Gendarmenmarkt to the main station, a navigation algorithm such as the one used by Google selects only a small number of the possible paths. The app developers decide which route will be pre-selected: Should a route be particularly short or quick? Or should a route instead aim to include a reliable arrival time by avoiding roads in advance that often involve unexpected delays due to traffic jams? Even the tiniest of these decisions entails different programming. Even if software providers offer their users different routes to choose from, this selection of options still conceals the technical compromises that the calculation requires. It also covers the developers' own preferences they include into programming, and the data records used to calculate the route. When displayed, a route seems to be the clearly optimal choice from among possible routes, but that is not what it is. The road intersections where a route planner is used are subject to an entire range of assumptions, priorities and conclusions previously made by developers working in a software firm.

As has been argued in Science and Technology Studies, the production of structures and facts that appear to be clear is not a "logical-rational process" or an "objective representation" (Bauer, Heinemann, and Lemke 2017, 13) of a particular subject. Instead, their production must be understood "as a social practice" (2017, 13). According to Susan Leigh Star, technical infrastructures in particular push social development processes to the background – and, with them, the question of who took part in their construction and with which interests (Star [1999] 2017). Classifications, standards, and categorizations that seem clear consolidate power relationships between those who design certain infrastructures and those who use them. This is because the production of infrastructures is inevitably tied to social considerations: "Each category valorizes some point of view and silences another" (Bowker and Star 2000, 5).

For a path between two locations to be recommendable at all on the basis of algorithms –to be calculable, in other words – a wide variety of expectations of a suitable route are simplified on the basis of economic and mathematical criteria. Everyday route planners are based on models in which the cost calculation forms the basis for every concept of space. Space is based on a "mathematizable calculus," and the design of spatiotemporal dimensions becomes a management task and an optimization potential for which mathematical methods and economic ordering structures are used (Neubert and Schabacher 2012, 164–165). When a route is selected, economic factors such as savings of time, money and fuel are prioritized over such harder-to-quantify aspects as convenience, aesthetics and the feelings of the traveller.[1] But even in the case of route properties for which purportedly unique data exists, as in the case of quick or short routes, route decisions cannot be made unequivocally: This has been reported to me by taxi drivers, long-distance lorry drivers and other deliverers whom I interviewed or accompanied in the course of my dissertation project between 2015 and 2019 (Pentenrieder 2020). They report complicated considerations as to which route is fast, short, or economical. Nearly all drivers surveyed used route planners for their journeys, but they always checked the calculations against their own local knowledge and additional sources of information. When choosing their route, they also took into account their own experience – how susceptible a route may be to congestion, for example – or they avoided routes suggested by the route planner that involved bridge underpasses that offered insufficient clearance for an articulated lorry. Or if the "sat nav" device recommended a short distance crossing an avenue, where tree roots

1 On the construction of road data, see: Pentenrieder (2020, 196–205), chapter 6.3: Zum Wert einer Straße: Produktionsketten unveränderlicher Elemente.

typically lift the asphalt to create a "bumpy road," they had to weigh the time savings against wear and tear to the vehicle (field notes, winter 2015). Calculations by a route planner are of little use to drivers in situations like these. Or take another example: During drives lasting several days, a relevant factor for selecting routes for long-distance lorry drivers involved determining which motorways offered places in which to sleep free of charge. This depends on whether roads and rest areas are state-owned or in private hands. Again, none of the information provided by the navigation equipment was of any use, as developers working in software firms know nothing about needs of this type.

In his essay entitled "Information Mythology and Infrastructure," Geoffrey Bowker argues that we should eliminate the "cybernetic narrative" that holds that everything is information, and that realities can be fully represented in models (Bowker 1994, 245). Nor do map data, algorithms and parameters in route planners reflect complete information and universally fitting combinations for suitable routes. They always follow the assumptions about feasibility and interests that can be found in software development firms. Considering long-distance lorry drivers' preferences in rest stops, we can see that the question of which data are available to the algorithm is a function of social and political context. Those who collect data and make them available for algorithmic evaluations take decisions as to which properties *count* and which do not. The inclusion of users' situational knowledge is thus relevant to the effort to adapt algorithmic recommendations to suit the individual situation. Up until now and beyond then, an informed route decision is based on calculable, technically inducible information as well as on the informal knowledge of the users.

And yet informal knowledge bases that users contribute to in the successful application of algorithmic recommendation systems – in this case, the considerations of the drivers themselves – often go overlooked. As "invisible work" (Star and Strauss 1999), they remain hidden when users negotiate the situations of everyday life. Success is instead attributed retrospectively to the provision of algorithmic assistance. The fact that the contribution of an informed user goes unseen owes not least to the steadily growing interpretative power of algorithmic recommendations in comparison with the knowledge of a driver: A taxi driver tells me that passengers have already asked him not to follow his own knowledge of the surroundings but to drive according to the directions shown on the route planner instead (field notes, autumn 2015). Even when I accompany a food supplier in the delivery of his wares, I witness the superior interpretative power of algorithmically calculated data in his everyday work: The software transmits his arrival time to the customer even before the driver, still at the restaurant, has managed to determine the route to take to deliver the food to the customer (accompaniment of a food delivery driver in autumn 2016). Algorithmic calculations forecast and control without offering any way for the stakeholders involved to know the underlying calculation and base data. It is the opacity that makes it difficult for users to review or contradict algorithmic benchmarks in light of other criteria and knowledge levels: "The claim that algorithms will classify more "objectively" (thus solving previous inadequacies or injustices in classification) cannot simply be taken at face value given the degree of human judgment still involved in designing the algorithms, choices which become built-in. This human work includes defining features, pre-classifying training data, and adjusting thresholds and parameters" (Burrell 2016, 3).

Economic and Political Contexts Remain Invisible

A second case in which an algorithmic pre-selection occurs without users' knowledge of the information withheld from them when selecting certain data situations, is provided by the critical urban researcher Ulf Treger, taking Uber, the ride-hailing service, as his example. His work demonstrates the significance of the private development contexts today's software is chiefly programmed. For business purposes, Uber addresses different target groups with specifically tailored map views to influence who receives which spatial access. With internal map views bearing labels such as "Heaven" or "God's view" Uber managers are able to monitor their own business processes and retrieve different layers of information about the urban space (Treger 2018, 241). Because the map information presented to drivers and passengers must necessarily be selected for a limited view in the smartphone app, this selection of information can also be used for business purposes: As Treger points out, the "Hell" map view addresses a group of Uber drivers "who also work for competing companies such as Lyft, the biggest competitor, so that their behaviour can be monitored and controlled: Such drivers are more likely to be offered rides than others in an effort to keep them on duty." (2018, 241–242) Through selection, Uber tailors the "truth content" (2018, 241–242) to certain perspectives on urban space, as is made even clearer in the "Greyball" map view: This map view conceals a secret software program designed to mislead regulatory authorities (Isaac 2017). Between 2014 and 2015, this map view was displayed to people whom Uber had identified as employees of a supervisory authority. Reserving an Uber taxi was made more difficult for this group of people by hiding available Uber vehicles in their immediate vicinity and making them so-called "ghost cars" in order to render it more difficult for the authorities to wield critical control over the company. As journalist Mike Isaac describes in an interview with CNBC, this map view is subject to a special data practice:

> If individuals working for a regulatory authority are "greyballed," they are not banned by the Uber service, as otherwise the app would no longer 'learn' about these people's behaviour: Information about how often the app is opened, and

> about the devices these individuals (e.g., police officers) use to open it, helps reveal the supervisory authorities' tactics. (Isaac 2017)

Along with the first example of the selections made by a route planner, the selection of spatial information shown in an app's map view presents a second possibility of influence in which data and algorithms help determine the social fabric precisely by virtue of the things one does not see in them: by virtue of their opacities. As the map views used at Uber make clear, the focus of a map usage is no longer to capture larger relationships, as noted by Pablo Abend for smartphones in general: "Instead, the viewer is presented with an isolated excerpt that seems to be cut off from any reference." (Abend 2013 118). It is precisely the boundedness of displays that makes it necessary to curate spatial information for users – whether by selecting a route or by selecting a map view. The asymmetrical distributions of information in the "Heaven," "Hell" and "Greyball" views show that technically necessary curation can be accompanied by the opportunity to establish hierarchies in software-structured spatial access. Software already prepares spatial information in different ways for different groups of people: In the process, some groups learn more about a city than others (Treger 2016).

As human geographer Stephen Graham points out, it is particularly software that links places, provides access, and draws boundaries that inscribes new power and knowledge structures as well as inequalities in cities (Graham 2005, 575). It makes a difference which stakeholders deploy those means of digital spatial access, and for which purposes: At the moment, technologies which provide spatial access are strongly intertwined with economic interests, as transnational companies view the urban space as a market in which to sell their technologies (Bauriedl and Strüver 2018, 21). Therefore, any analysis of software concerning spatial constellations, needs a strong focus on social, economic and cultural conditions. In software studies in particular, then, not only is there investigation into what "software is," but also into what "software does" (Mackenzie 2006, 3) and into the material and discursive conditions in which software is embedded today.

Algorithms Assign Tasks Without the Need of Explanations

In the working environment of drivers in the logistics sector, route planners and digital maps are often part of software that additionally coordinates trips, distributes tasks, and controls execution. This is known as "dispatch software." This adds further dimensions on the basis of which algorithms organize visibilities and invisibilities. In a food delivery service as part of the gig economy, for instance, digital platforms or smartphone applications are used to communicate individual orders to the drivers. Essential organizational tasks are outsourced to algorithms, while the operational activities are processed as "repetitive micro tasks" by self-employed workers. (BMAS 2016, 7; Webster 2016; Irani 2015) One social problem with these employment relationships is that the contractors have no guarantees of "adequate economic, social and legal protection" (BMAS 2016, 8).

Min Kyung Lee and her colleagues describe employment relationships such as these, in which gig workers primarily interact with algorithms as superior entities, as "algorithmic management" (Lee et al. 2015, 1603). While the members at the management level remain anonymous, the driver interacts with a fabric of different algorithms. Lee et al. distinguish three dimensions of tasks that algorithms take over in this context: The assignment and issuance of instructions for orders and tasks to the gig worker, the optimization of their work processes associated with the structuring of work steps, and the evaluation of completed tasks (Lee et al. 2015).[2] According to this structure, food delivery staff are assigned their delivery orders using an "instruction algorithm"; then, the second step is to optimize their route to the customer through an interface to Google Maps: with a single click in the same app, drivers can output routes to restaurants and customers in the form of step-by-step instructions. As the third and final step, the app continuously records their speed and location data for evaluation purposes. These data are factored into next order assignment as individual key figures. In addition, target values for punctual delivery are adjusted based on continuous measurements of average speed, i.e. data constantly mediate between drivers' practices and the algorithmically modulated framework in which they operate. These feedback processes are an important element of algorithmic regulation, as they are self-optimizing (cf. Eyert 2020). The algorithm emerges here in the form of a dynamic data evaluation, as it constantly modifies itself based on active driver practices. Unlike gig workers who perform micro tasks online, drivers in the logistics gig economy are a visible part of the cityscape as they perform their algorithm-mediated assignments. In this, they draw attention to problems that affect gig workers in virtual (work) environments as well. Here is how media scientist Carina Lopes describes algorithmic employment relationships in the logistics sector, based on the example of parcel carriers in Spain:

> Algorithms coordinate and indicate then how relations unfold and evolve – what tasks are done, the workflow steps that have been followed and which interactions take place between different parts of the system. Within the intensively computational complex urban systems, everything seems to necessarily start and finish with an algorithm. The parcel about to be delivered enters a spatial field that

2 Karen Yeung offers a similar taxonomy for the algorithmic regulation of social arrangements in working conditions. For this purpose, she distinguishes among the dimensions of setting targets, collecting data and modifying behaviours as functions of algorithmic regulation. (see Eyert 2020,1 with reference to Yeung 2018)

> has been calculated and optimised numerous times, all in name of the seamless and efficient workflow towards delivery. It enters a field of action where the bureaucratic algorithm – counting, recording, ordering – meets the automation algorithm – tracking, tagging, deleting, isolating – giving rise to contexts of action that can rapidly evolve and be adapted to environmental aspects such as weather conditions or market demand and supply dynamics. (Lopes 2016, 216)

Algorithms are used to "match" drivers with suitable tasks, as if they were merely one variable among many (2016, 213). In the gig economy, work is structured around the existence of individual orders, as media theorist Felix Stalder points out for digital platforms in general: they create "access to an action space" that "offers opportunities that cannot be found anywhere else" (Stalder 2016, 161). The orders offered on a platform are not attached to a "normative must" but only an "optional can" that either party can retract at any time (2016, 161). Although drivers are free to decide between accepting and declining each incoming order, they have only a limited view of all the factors associated with the respective orders. In the case of Uber, for example, a driver has 15 seconds in which to accept a trip request without seeing where he or she must go – and whether the trip may prove unprofitable (Rosenblat and Stark 2016, 3762).

Opacities like these distinguish algorithmic instructions from instruction made by human superiors. In a bike courier center, I observed the following situation while accompanying the work of a dispatcher:

> During the afternoon of my visit, there were five drivers carrying out incoming delivery orders. As soon as a courier had delivered an order, he or she would use "the radio" to report availability back to the dispatcher. The dispatcher then assigned the driver a spot on the waiting list, with a number between 1 and 5. This assigning of trips continued throughout the afternoon, with one driver repeatedly receiving particularly short delivery orders. (The shorter the delivery route, the less lucrative the order the couriers, as self-employed individuals, can post.) Suspecting that this courier might be 'annoyed', the dispatcher asked all the other couriers if he could assign to this courier an unscheduled, longer delivery run that had just been received. The colleagues agreed, and the driver was given the lucrative order. (Field notes, autumn 2016)

This non-algorithmized instruction situation demonstrated the need for negotiation and argumentation for each standardized assignment procedure with regard to what can be viewed as a fair allocation of work, as well as any situational adjustments that are possible. Because it is presented as indisputable, an algorithmic assignments seems more objectivity – in contrast to biased, non-impartial or distorted human decision-making. But algorithms in the same way assign their tasks in accordance to logics that are subject to negotiation. Unlike human superiors, however, they cannot be interrogated about the rules that govern these assignments. Behind the display, there are technical thresholds, parameters and data inputs that decide how a task is assigned. The algorithmic instructions require not "being in the dispatcher's good graces," as a former bicycle courier described it, but rather a breakdown of mathematical arguments. But these mathematical arguments lie beneath a series of opaque layers that can have technical and organizational, but also political, cultural or economic reasons.

As Louise Amoore has contended, technologies such as "algorithmic modelling" have become the decisive, authoritarian knowledge structures of our time (Amoore 2013, 9). What is problematic about this is that these new objects of knowledge say nothing about the logics and interests inevitably inscribed in them. This results in fractures between different knowledge bases (2013, 9). Programmed by technical development teams, often algorithms and data layers cannot even be explained by direct supervisors; consequently, a delivery driver who interacts with an "instruction algorithm" during trips has no clue of the key figures under which he or she is evaluated or compared with colleagues. If these key figures are output by a "statistical algorithm" as "features" in the context of machine learning, the selection criteria are not even known to the developers, although they may be aware of the spectrum of possible criteria. However, the algorithm "decides" situationally which criteria it deems relevant to the concrete selection, operating on the basis of historical and constantly evolving data. This makes traceability, negotiation, and potential objection of algorithmic instruction very difficult.

Order instructions issued by means of algorithms thus represent a third version of the ways in which algorithms and datasets – which are often dynamically generated and updated[3] – influence the social relationships of their users by virtue of their very opacities. The software determines who receives which assignment and when. It also determines which perspective, and which visibility and invisibility the users will receive onto larger contexts. This algorithmic formation creates "information asymmetries" between drivers and platform providers (Rosenblat and Stark 2016, 3777). But conflicts of interest are systematically resolved in favor of the company that specifies the design of the technology.

3 The ways in which one's own data impact the next delivery order are illustrated by one driver's tactic of playing with his speed to generate more lucrative delivery runs. See Pentenrieder 2020, 131–144, Chapter 5.2.: Wie munitioniert man eine "Weapon of Choice?" [How does one arm a "weapon of choice"?].

Opacities Due to Algorithmic Reshaping

It is not just when job instructions are issued that algorithmic opacities have an effect. Algorithms and the opacities associated with them also influence how workflows are regulated and structured,[4] as I outline in a fourth and final empirical field observation. In autumn 2016, I accompanied a food delivery man on his lunch shift. When picking up a pizza in the restaurant, he only learned the customer's delivery address when the pizza was already hot and in front of him on the counter. Only when he actually set out on his actual route was he able to "unlock the route" and begin planning his path to the customer. The current version of the app does not provide a way to use the time he spends waiting to plan the next route. (Field notes, autumn 2016)

Regardless of whether the app has since been programmed differently, this situation illustrates how the fractures between the programmed arrangements and the knowledge bases of the user can actually hamper the efficiency maxims of logistical arrangements. While algorithmically mediated data can empower new work routines in some cases, they disempower employees in instances in which the algorithms technically reshape employment relationships to such an extent that they structurally reduce and outsource entire areas of responsibility. This prevents not only critical employee input but also constructive employee influences, offered for the sake of the work routine.[5] Here, drivers are permitted to do nothing more than 'fill in the gaps in the automation' (Wischmann and Hartmann 2018, 2). The calling-up of individual knowledge that transcends automation, as Lopes also points out, is growing increasingly difficult, in the interest of "operational efficiency and optimization" (Lopes 2016, 214–215). Work processes allow less and less room for the local knowledge of parcel couriers, for example, who know when certain people are likely to be at home (2016, 214–215). Ethnographers at the University of Hamburg demonstrate that software programming is often not sufficiently geared to the things that would be helpful in a work routine. They refer to the "requirement problem" in computer science, which holds that it is easier to make a software program operational than it is to identify the software solution that a particular situation actually requires. (Brugger et al. 2011, 182) On behalf of "good enough software" (Bialski 2018), digital working architectures simplify socially complex structures in favor of technically feasible programming.

4 In this connection, see the concept of "algorithmic regulation" by Yeung (2018), which was further developed by Eyert et al. (2020) into a framework model.

5 Eva-Maria Nyckel (formerly Raffetseder) and her colleagues reach a similar conclusion with regard to other software interactions (Raffetseder, Schaupp and Staab, "Kybernetik und Kontrolle," 2017, 232).

For decades, feminist technology researchers such as Susan Leigh Star and Lucy Suchman have advocated a "pragmatic sociology of technology" that introduces technology not simply in terms of technical feasibility but also goes on to examine the extent to which it benefits social structures (Star 2017, 35). In lieu of a Turing test that determines whether a machine is perceived as intelligent, Star proposes a Durkheim test that determines whether a machine is perceived as social (2017, 35). Taking into account the rigidities and limitations that technologies and processes of automation inevitably involve, an effort could be made to promote programming that is more oriented to everyday practice and must therefore "reckon with the untidiness of sociotechnical work": "Systems should be tested for their ability to respond to community objectives. [...] A process is deemed to be commensurate if divergent views are factored into the decision-making process in a fair and flexible way" (2017, 131).

Addressing Algorithmic Opacities with User-Centric Perspectives

In different ways, the four examples presented, ranging from the handling of software to route and job planning, demonstrate how software governs which information its users can and cannot see about road traffic and work processes. First of all, route planners grant certain streets the status of making a journey faster than others, yet they do so without disclosing their underlying data and calculation logics. Secondly, map views such as those of Uber restrict the view of the city, prescribing selective spatial access without having to disclose the underlying political and economic considerations involved. There are still more opacities in the context of logistics work: Thirdly, instruction algorithms make it difficult for their employees to critically question standardized job assignments, as even direct supervisors are often uninformed of the technical workings of an algorithm. Fourthly, even constructive contributions from employees are made more difficult in work processes if the task structure is organized in such small steps that one's own knowledge – such as an independent sense of orientation or knowledge of delivery times suitable to the customer – cannot be taken into consideration.

To reduce the instructions shown on the display, information must always be selected for the users – and algorithms evaluate certain datasets for this purpose. By selecting certain information and omitting other information, algorithmic recommendations are always deliberately designed, not objective or neutral. When performing software programming, developers make social decisions with regard to the criteria, user profiles and data that can be used to derive the fit of an individual route or follow-up order. This gives decisions of technical nature a certain social relevance. But what the display conceals are the (social) compromises made on behalf of technically feasible programming, according to which premises, for example, data selection was shaped by means of algorithms.

Precisely because they are unquestionable, algorithms have become new authorities, especially when operating data. They determine which broader relationships users can grasp, understand, and question, and which knowledge of their own they can contribute towards a practice determined by algorithms. Together with all the additional technical conditions that surround algorithms – such as data formats, parameters, thresholds or the limited display space on smartphones – algorithms arrange the visual fields of their users (Pentenrieder 2020, 121–221). Users see the same technical functional logics and organizational backgrounds with which they are interacting only through a "tiny keyhole" (Suchman 2007, 11).

These viewing relationships need to be questioned, because algorithms 'curate' whether, what, when and how their users "catch sight" of a particular piece of information. Algorithmic opacities arise not from a lack of technical knowledge or competences on the part of the users. They arise through the *structural* condition of today's software, which offers too few perspectives onto algorithmic functionalities and data bases for their users.

Spatial concepts can be enlisted to vividly "visualize" these algorithmically defined perspectives in everyday practices. Taina Bucher made this impressively clear in the field of software studies, based on the example of the Facebook News Feed. She shows that, through architectural formations and arrangements, software influences social practices and brings regulatory forces to bear on them (Bucher 2012). In this connection, she projects Michel Foucault's panoptical architectures onto algorithmic arrangements: "An architectural perspective usefully highlights the ways in which spaces are 'designed to make things seeable, and seeable in a specific' way" (2012, 27). Specifically, Bucher enlists John Rajchman's interpretation of Foucault's architecture in the regulation of visibilities: "Architecture helps 'visualize' power in other ways than simply manifesting it. It is not simply a matter of what a building shows 'symbolically' or 'semiotically', but also of what it makes visible about us and within us." (Rajchman 1988, 103). Software installs visual barriers and fields of view in the same manner as spatial architecture installs windows and frames. This is particularly noticeable in interactions with route planners and tasks in the logistics sector, as these practices are determined algorithmically and unfold in spatial structures at the same time. From the point of view of a user at a street corner, one wonders how the software arrives at a route or a task recommendation. This seemingly simple question makes it possible to grow arbitrarily complex where software components and the work steps of the developers involved in the result are concerned. But this initial question makes a major contribution to the discussion of technical conditions: It focuses on the previously limited view of the software user – the view through the "keyhole" and onto the algorithmic result and the datasets – and, from the user's point of view, considers what he or she can "see," and thus know or not know, of the underlying logics of the software. The empirical representation of user interactions prioritizes users' visibilities (in the form of possibilities of knowledge) vis-à-vis the software. In a second step, once users' questions and problems have been elaborated, developers, data scientists and other designers of software can be queried about the technical principles (data, algorithms, parameters, cost functions) underlying the identified opacities that confront users (see methodology in Pentenrieder 2020). Under this approach – of first developing the user perspective and secondly analyzing the technical logics – users can determine what technical information they need to assess algorithmic results and can set their sights on the technical logics essential to this information. What makes this approach decisive is that it *curates* anew what is known about algorithms, this time working from a user-centric perspective.

Analyzing software – not only on the basis of a user-centric construct but also with creative questions in mind about visual relationships – makes a significant difference for a praxeological approach to software analysis. In contemporary urban research, an example of the value of such a shift in perspective towards the object created can be seen in the concept of "human-scale architecture" advanced by the Danish city planner Jan Gehl. In a concept that runs counter to the car-friendly city, Gehl takes the eye level of an individual, completes it with his or her interests, feelings and desires for their own city as habitat and uses this pedestrians' point of view for the planning process. As a result, the functionality and perfection of architecture and solitary buildings are no longer at the heart of urban planning (Gehl 2011). Instead, the planning strategy is defined from the perspective and experience of a pedestrian whose fields of view of his or her city are defined by an eye level at 1.60 m and a walking speed of 5 km/h. As urban research shows, incorporating visual relationships such as these, which pedestrians introduce to their use of urban spaces, significantly alters the planning process and results in cities different to those that have gone before them – "at human scale" (Wang, Sadik-Khan, and Gehl 2012).

To curate informative fields of view for users of a software program, the approach to designing the user experience (UX) calls for a similar shift in perspective – and thus a shift in paradigm. In the 1970s, the development of graphical user interfaces (GUI) marked a radical turning point in the concealment of technical processes.[6] Today, however, one can ask the question in reverse: Which perspectives on technical processes does an emancipatory use of technology now require

6 The development of the "graphical user interface" (GUI) at the research institute Xerox PARC contributed significantly towards establishing personal computers as a mass product on the market in the 1970s. (Chun 2011, 59) It changed the 'dialogue' between user and computer: Users no longer navigated between programs using text-based lines of commands, but now used windows, icons, menus and cursors instead. (Bunz 2019, 76)

for users to be able to develop expertise with regard to algorithmic recommendations – an expertise they can use to check algorithmic calculations, manage and protect personal data and profiles, and control third-party access?[7] A comparison between the user-friendly software design of the 1970s and today's requirements is paradoxical in this respect: While the effort to popularize technology in the 1970s meant that GUI design had to *protect* the user from the technical complexity of software, today's interface design should precisely re-open this protective black box to make informed, responsible or democratic software use possible.[8]

In order to determine which perspectives informed users require, media-theoretical reappraisals of the software itself must be supplemented with the practices of interested users. This is why informed users have also prompted the vignettes formulated above: Their experience shifts the view of the software and the facts of interest that underlie the technology. When describing their everyday lives, long-distance lorry drivers and food couriers also have indirect questions about the production of algorithms and data, along with other basic principles of technology. Some users take this a step further and compensate their limited view by using various reverse-engineering methods and plausibility strategies to reconstruct algorithmic recommendations and relevant software logics (see Chapter 6 in Pentenrieder 2020). These informed and curious drivers try to anticipate the operational principles of software in use, for example by tracking whether it is an algorithm's own logic, movement data the users have generated themselves, or systematic decision-making by a manager or software developer that determines how a task is assigned. Their practices permit conclusions about the social implications that some technical principles involve – principles that thus require social-scientific analysis. This is how emancipatory user practices provide a methodological guide to locating the aspects of a software program that require transparency for users in the first place.[9]

The four scenarios presented here of drivers' interactions with opaque algorithms are based on a combination of spatio-theoretical software analysis and praxeological user research. When users follow a route (1), look at a map on their smartphone (2), accept a delivery order (3) and follow step-by-step work instructions (4), there are algorithms at work that govern the visibilities of data layers for road traffic

7 Cf. the debate around explainable AI: Wachter, Mittelstadt, and C. Russell, "Counterfactual Explanations Without Opening the Black Box," 2017, Kroll, Huey, and Barocas, "Accountable Algorithms," 2017, Spielkamp, Automating Society. Taking Stock of Automated Decision-Making in the EU, 2019.

8 Many thanks to Timo Kaerlein for the inspiring discussion on this topic.

9 In this connection, see methods from participatory design research, such as Suchman 1986/2007.

and working conditions. The empirical case studies manifest the viewing relationship as the source of algorithmic opacities. Particularly the focus on emancipated users reveals architectures in which algorithms govern social structures. Based on user-centric scenarios such as these, specific consideration can be given to how "algorithmically framed fields of view" might be "curated" to make algorithmically conditioned work environments auditable by and accountable to the people who use them.

References

Abend, Pablo. 2013. *Geobrowsing: Google Earth und Co: Nutzungspraktiken einer digitalen Erde.* Bielefeld: Transcript.

Amoore, Louise. 2013. *The Politics of Possibility: Risk and Security beyond Probability*. Durham: Duke University Press.

Bauer, Susanne, Torsten Heinemann, and Thomas Lemke, ed. 2017. *Science and Technology Studies: Klassische Positionen und aktuelle Perspektiven*. Berlin: Suhrkamp.

Bauriedl, Sybille and Anke Strüver, ed. 2018. *Smart City: Kritische Perspektiven auf die Digitalisierung in Städten.* Bielefeld: Tanscript.

Bialski, Paula. 2018. "Hiding in the Stack: Practices of Resistance in Corporate Software Development." Lecture at GfM Annual Conference, September 2018, Siegen.

Bowker, Geoffrey. 1994. "Information Mythology: The World of/as Information." In *Information Acumen: The Understanding and Use of Knowledge in Modern Business*, edited by Lisa Bud-Frierman, 231–247. London/ New York: Routledge.

Bowker, Geoffrey, and Susan Leigh Star. 2000. *Sorting Things out: Classification and Its Consequences: Inside Technology.* Cambridge, MA: MIT Press.

Brugger, Senana Lucia, Katharina Wolter and Steffi Beckhaus. 2011. "Ethnographisch, Praktisch, Gut! Perspektiven für Ethnologen in der Softwareentwicklung am Beispiel eines konkreten Projektes." *Ethnoscripts* 13 (1): 181–198.

Bucher, Taina. 2012. *Programmed Sociality: A Software Studies Perspective on Social Networking Sites*. Oslo: Universität Oslo.

BMAS (Bundesministerium für Arbeit und Soziales) und iit Berlin. 2016. "Foresight-Studie Digitale Arbeitswelt." Berlin.

Bunz, Mercedes. 2019. "The Force of Communication." In *Communication* edited by Paula Bialski, Finn Brunton, and Mercedes Bunz, 51–92. Lüneburg: Meson.

Burrell, Jenna. 2016. "How the Machine 'thinks': Understanding Opacity in Machine Learning Algorithms." *Big Data & Society* 3 (1): 1–12.

Chun, Wendy. 2011. *Programmed Visions: Software and Memory, Software Studies*. Cambridge, MA: MIT Press.

Dourish, Paul. 2016. "Algorithms and Their Others: Algorithmic Culture in Context." *Big Data & Society* 3 (2): 1–11.

Eyert, Florian, Florian Irgmaier and Lena Ulbricht. 2020. "Extending the framework of algorithmic regulation. The Uber case." In *Regulation & Governance* 1–22.

Gehl, Jan. 2011. *Life between Buildings: Using Public Space*. Washington, DC: Island Press.

Graham, Stephen. 2005. "Software-Sorted Geographies." *Progress in Human Geography* 29 (5): 562–580.

Hartmann, Ernst A. 2015. "Arbeitsgestaltung für Industrie 4.0: Alte Wahrheiten, neue Herausforderungen." In *Zukunft der Arbeit in Industrie 4.0*, edited by Alfons Botthof and, Ernst A. Hartmann, 9–22. Berlin: Springer Vieweg.

Irani, Lilly. 2015. "The Cultural Work of Microwork" *New Media & Society* 17 (5): 720–739.

Isaac, Mike. 2017. "Uber Greyball Programm Evade Authorities." New York Times, March 3. Accessed 06.04.2021. https://www.nytimes.com/2017/03/03/technology/uber-greyball-program-evade-authorities.html.

Lee, Min Kyung, Daniel Kusbit, Evan Metsky, Laura A. Dabbish. 2015. "Working with Machines: The Impact of Algorithmic and Data-Driven Management on Human Workers." *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI'15: 1603–1612.

Lopes, Carina. 2016. *Understanding Relational Locations and Complex Urban Systems: Mapping The Relations Between Computation, Space and Infrastructure*. London: Goldsmiths University London.

Mackenzie, Adrian. 2006. *Cutting Code: Software and Sociality, Digital Formations*. New York: Peter Lang.

Neubert, Christoph, and Gabriele Schabacher. 2012. "Logistik." In *Handbuch der Mediologie: Signaturen des Medialen*, edited by Christina Bartz, Ludwig Jäger, Marcus Krause and Erika Linz, 164–169. München: Wilhelm Fink.

Pentenrieder, Annelie. 2020. *Algorithmen im Alltag: Eine praxistheoretische Studie zum informierten Umgang mit Routenplanern*. Frankfurt am Main: Campus.

Raffetseder, Eva-Maria, Simon Schaupp, and Philipp Staab. 2017. "Kybernetik und Kontrolle: Algorithmische Arbeitssteuerung und betriebliche Herrschaft" *PROKLA* 47 (187): 229–248.

Rajchman, John. 1988. "Foucault's Art of Seeing." *October* 44 (1): 88–117.

Rosenblat, Alex, and Luke Stark. 2016. "Algorithmic Labor and Information Asymmetries: A Case Study of Uber's Drivers." *International Journal of Communication* 10 (1): 3758–3784.

Seaver, Nick. 2014. "Knowing Algorithms." Lecture at *Media in Translation 8*, Cambridge, MA, April 2013.

Stalder, Felix. 2016. *Kultur der Digitalität*. Berlin: Suhrkamp.

Star, Susan Leigh. 2017. "Die Ethnografie von Infrastruktur." In *Grenzobjekte und Medienforschung*, edited by Sebastian Gießmann and Nadine Taha, 419–436. Bielefeld: Transcript.

Star, Susan Leigh and Anselm Strauss. 1999. "Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work." *Computer Supported Cooperative Work* 8 (1–2): 9–30.

Suchman Lucy. 2007. *Human-Machine Reconfigurations: Plans and Situated Actions*. Cambridge / New York: Cambridge University Press.

Treger, Ulf. 2016. "Space Making/Space Shaping: How Mapping Creates Space, Shapes Cities and Our View of the World." *Media.CCC*. Accessed December 9, 2018. https://media.ccc.de/v/33c3-7958-space_making_space_shaping.

Treger, Ulf. 2018. "Die Stadt als Bildschirm: Wahrnehmung und Nutzung urbaner Räume durch digitale Kartographie, urbane Dashboards und die Praxis der Navigation." In *Smart City: Kritische Perspektiven auf die Digitalisierung in Städten*, edited by Sybille Bauriedl and Anke Strüver, 237–248. Bielefeld: Transcript.

Wang, Jiangyan, Janet Sadik-Khan, and Jan Gehl. (2012). "The Human Scale.,"Eurovideo Medien.

Webster, Juliet. 2016. "Microworkers of the Gig Economy: Separate and Precarious." *New Labor Forum* 25 (3): 56–64.

Wischmann, Steffen, and Ernst Hartmann (2018). "Zukunft der Arbeit in Industrie 4.0 – Szenarien aus Forschungs- und Entwicklungsprojekten." In *Zukunft der Arbeit. Eine praxisnahe Betrachtung*, edited by Steffen Wischmann, and Ernst Hartmann, 1–7. Berlin: Springer Vieweg.

Yeung, Karen. 2018. "Algorithmic Regulation: A Critical Interrogation." *Regulation & Governance* 12 (4): 505–523.