"Just Do It!"

Considerations on the Acquisition of Hackerspace Field Skills as an Ethnomethodological Research Technique

Sebastian Dahm

Abstract

In this paper I present an ethnographic approach to the research of hackerspaces. It draws upon an ethnomethodological background in order to address the role of members' skills and knowledge. To that end, I aim for an immersive ethnographic approach in order to achieve a first-hand understanding of members' practices. In this, I draw upon ethnomethodology as it provides a rich theoretical and methodological background for the study of skill and knowledge, namely the call for practical knowledge as an analytical instrument (Garfinkel 2006). In order to fully understand the implications of social movements like hacking and making communities, appropriate research methods are called for. Ethnomethodology, with its tradition in the analysis of epistemic practices and embodied knowledge, can provide the means for a more immersive and reflexive ethnography. By using materials of my own ethnography, I demonstrate how active engagement with members' practices can provide for a deeper ethnographic understanding. In order to overcome the challenges of the field, I chose to adopt a project of coding myself. This acquisition of field-specific knowledge proved to be not only a valuable resource for the ongoing fieldwork but could offer important analytical insights in itself. I will show that important facets of members' meanings were accessible only through personal experience. I suggest a broader adoption of ethnomethodological principles in ethnographic research of hackerspaces as it accommodates the underlying affinity towards experimentation prevalent in the field.

Introduction

The field of hacking and making is currently drawing increased attention of researchers from various disciplines. As earlier research has focused primarily on the unauthorised breaching of computer systems (Jordan/Taylor 1998: 757), the character of hacking as a way of creative engagement with technology has been overlooked for a long time. Recent publications increasingly take this aspect

DOI 10.14361/dcs-2017-0107

into account by using ethnographic approaches to varying degrees: in the most comprehensive work on hacking in the recent past, Coleman (2010, 2013) explores the ways in which creativity and "freedom" find its expression in various practices of hacking. Toombs, Bardzell and Bardzell (2014) demonstrate the importance of individual skill displayed in field members' ability to craft customised tools. While focusing on the criminal aspects of hacking, Steinmetz (2014) draws a similar conclusion and recognises hacking as a "transgressive craft" (2014: 141). Wagenknecht and Korn further this understanding of the "productive" features of hacking: by drawing upon the concept of "transgressive infrastructuring" (2016: 1104), the authors show how practices of "opening" are prevalent in the activities of the annual Chaos Communication Congress.

Although recent publications provide an extensive understanding of hacking as a field of individual skill, practical knowledge and creativity, the methodological implications of these findings are seldom addressed. In this paper I therefore want to discuss the specific methodological challenges associated with an ethnographic approach to the field of hacking and making. In order to do so, I turn to ethnomethodology as the main theoretical framework. Developed in the 1960s by Harold Garfinkel, ethnomethodology explores the ways in which members of society continuously produce social order in their actions (Garfinkel 2010: 1). Following this predicament, ethnomethodology has always incorporated a fundamental critique of established social theory and research methods which has in turn led to important epistemological implications. The focal point of analysis is the "ethno-methods" by which members of society produce social order. This leads to a radical scepticism towards any theoretical preconceptions, as any generalising description loses the defining qualities of the described phenomenon (Garfinkel 2002: 133). Thus, ethnomethodology has adopted a strong empiricist attitude that emphasises the importance of practical knowledge for an understanding of social life:

> It is Garfinkel's position that the knowledge of practices he is trying to introduce is not a conceptual or cognitive knowledge but, rather, an embodied knowledge that comes only from engaging in practices in concerted co-presence with others. The details of these practices cannot be seen from within the theoretical attitude. (Garfinkel/Rawls 2006: 5)

I argue that this conception of knowledge offers interesting perspectives for ethnographic fieldwork. As any ethnography demands active engagement with the field to a certain degree, ethnomethodology can provide an analytical framework for the practical experience attained in this process. This holds especially true when studying fields like hackerspaces where skill and embodied knowledge are of high importance. In the following sections I will illustrate this point by using materials from my ethnography in several German hackerspaces. Thereby I will focus on my own attainment of basic coding skills as a means of access to the practical field knowledge. "Classical" field descriptions will only be discussed cursory, as I

want to place emphasis on the relevance of ethnomethodological principles in my ethnography of hackerspaces by demonstrating my own achievement of practical knowledge.[1]

I will begin with outlining the general problems of field access in ethnography in order to introduce central ethnomethodological concepts that can offer a reflexive viewpoint of ethnographic practice. Subsequently, I am going to demonstrate the incorporation of said concepts into my own research: I will show that the attainment of basic coding skills was necessary in order to achieve a deeper understanding of field members' conduct. In the next section I will demonstrate how the practical experience of coding could be applied to the subsequent stages of fieldwork. Lastly, I am going to discuss my findings with special respect to the field of hacking and making.

Being there: Ethnomethodology and the factor of field skills in ethnography

One of the key problems of ethnographic research lies in the ability to find access to the chosen field. While such a statement may seem trivial at first glance, the endeavour of finding a "way into the field" is persistent in all stages of ethnographic fieldwork (Hammersley/Atkinson 2007: 43). In fact, "gaining access" is one of ethnography's primary concerns: not only as a practical precondition but rather as a defining research goal. When the ethnographic enquiry aims for an immersive understanding of social worlds (Emerson/Fretz/Shaw 2011: 3), the question of access becomes a matter of constant involvement: how can field practices and meanings be pursued? How can members' experiences be understood? How can their life-worlds be described? Therefore, any understanding of access as some kind of "granted privilege" would be limiting and misleading. It can only be described as an accomplishment that is the result of ethnographers' serious attempts to engage with the field and its demands. This commitment is closely related to the development of a "working identity" (Hammersley/Atkinson 2007: 69). The ethnographer needs to utilise her own experience as an instrument

1 Although this approach focuses on personal experience, I want to avoid the term "autoethnography" at this point in order to prevent misinterpretation. While there are certain similarities regarding the emphasis on personal experience, the theoretical background of ethnomethodology and autoethnographic approaches is quite different: While ethnomethodology is primarily a theoretical and methodological stance closely related to Schütz and phenomenology (Garfinkel/Rawls 2006), autoethnography is a decidedly political discipline that is influenced primarily by post-modern theorists and aims for a value-centred approach that also incorporates literary techniques (Ellis/Adams/Bochner 2010: 2). Regardless, the methodological commonalities of both approaches could be explored in further works.

of analysis as it reflects the relevancies displayed in the field. Therefore, a field role is not simply an "admission ticket" into the field but an analytical resource, which requires competent knowledge of field practices. Eisewicht and Kirschner describe a "sense of belonging" (2015: 667) that comes with the development of a credible field role as an integral part of ethnography. Through continuous engagement in the practices of online video gaming the researchers developed practical field knowledge, which enabled them to pursue members' meanings (ibid: 668). This example illustrates the importance of field-specific knowledge for access (as an ongoing process) and ethnographic analysis. Thus, solely "being there" cannot provide the means of ethnographic insight. Intimate knowledge can only be achieved through some kind of active participation.

Ethnomethodology's preoccupation with the analysis of mundane practices has given rise to a variety of radical concepts, both theoretical and methodological. Some of those concepts are directly linked to the concerns regarding field role and immersion addressed earlier. When we understand society as a "lived order" (as opposed to a static structure predefined by rules), this assumption is also relevant with respect to practices of ethnographic field work. This means that instead of a disconnected observer, the ethnographer has to be envisioned as an active practitioner in the field: regardless of her methodological stance towards observing or participating, she always forms an active part of the local setting. Thus, attaining a specific field role means to act in accordance with the relevancies and lived structures of the field. To do so, the ethnographer as a stranger[2] in uncharted territory needs to make sense of the opaque practices of the field members.

This directly corresponds to the initial view of ethnography as an active and fluid enterprise because for any kind of understanding to be possible, the ethnographer needs to achieve certain practical knowledge of the practices at hand. With this being said, another essential feature of locally produced practices needs to be taken into account: ethnomethodology places great emphasis on the haecceity of social order, which refers to the unique quality, or, as Garfinkel puts it, the "just-thisness" (2002: 99) of practice: any locally produced action possesses unique qualities that are inseparable of the immediate context. Thus, emphasis is placed on the lived detail of the situation at hand. As noted earlier, immediate experience becomes a vital part of the analysis of social phenomena. The unique adequacy requirement represents the methodological postulation derived from this understanding of social conduct. This postulation is that any social phenomenon can only be described if the researcher is at least "vulgarly competent" in the practices through which the phenomenon is achieved (Garfinkel/Wieder 1992: 182). Ethnomethodology's emphasis on immersion becomes evident here: in order to become familiar with the strange, the researcher needs to submit herself to the demands of the field. When meaning is continuously produced through practices, any kind of knowledge about the field is derived from practical knowledge.

2 Taking the meaning of Schütz (1964).

Following the initial considerations regarding the development of a field role as an essential requirement of ethnographic fieldwork, the "ethnomethodological stance" can provide a theoretically and empirically grounded expansion of ethnographic practice. In its rejection of any theoretical preconceptions, ethnomethodology directs the "ethnographic eye" towards the meaning produced by the field members themselves. This emphasises the understanding of an active field role specified earlier: as the field role is closely related to practical knowledge and skill displayed in the field practices, it can only be attained through the development of some kind of practical competence. The ethnographic approach I suggest here, emphasises the pragmatist impulses in ethnomethodology, namely "its call for a return to experience or recovery of concrete practices" (Emirbayer/Maynard 2011: 221; cf. Rawls 2011). This allows for the analysis of "silent" ethnographic data – such as researcher's experience and "tacit knowledge" (Hirschauer 2007: 431).

In the following section, I describe the difficulties of fieldwork that I had to overcome in my hackerspace ethnography. I contrast these empirical insights with the methodological propositions established in this section. The goal is to develop a differentiated perspective on the problem of field knowledge in ethnography and the benefits that an ethnomethodological approach can offer.

Observing in order to practise: "Just do it"

The main part of the ethnography was conducted in the hackerspace of a major German city *(Weststadt)*. It has to be noted though, that at the beginning of the fieldwork phase I also visited two other hackerspaces (*Burgstadt* and *Nordstadt*) briefly. My goal was to observe – and ideally take part in – the work on one or more of the numerous projects displayed on the websites. I imagined hackerspaces to be signified by high degrees of collaborative work and hoped to be able to participate and help out in some projects. In this way, I planned to establish a field role that would be not merely observing but that could be described as a "dedicated amateur": not as skilled as the senior members, but willing and able to learn and to help. I believed that my limited skills in programming and Linux would be enough to qualify as a credible beginner, which I expected to help me gain access to common activities. However, I had to abandon that hope after a few visits: despite being featured quite prominently on the websites, most collective projects were being worked on quite infrequently, if at all. Instead of a place of collaborative pursuit of knowledge and innovation, I had found a bunch of individualists immersed in obscure projects, often seemingly consisting solely of silent keyboard hacking. In my role as observer, I experienced something Susan Leigh Star has remarked about similar fields: "At first try, using fieldwork to stand and watch people punching keys and looking at screens is terribly difficult for trying to see social order. Or in fact, to see much of anything" (2002: 108). Although regular

conversations about individual projects were taking place, I seemed to be excluded from them.

When I began to ask questions on how to best get started as a novice programmer, I got rather curt and unsatisfactory answers: no one offered any direct help and I was repeatedly told to "just do it" ("einfach machen"). This irritated me to a considerable extent, as I perceived it to be a blunt way to tell me to go and bother someone else. "Just do it," however, wasn't a request directed only towards me. During later stages of fieldwork, I noticed that it was uttered frequently in different variations. Even to the point where it resembled some sort of mantra as it seemed to be the standard response to any kind of question, be it practical or organisational. "How do I get started with Python?" – "Just do it." "The documentation of the spacebot[3] seems kind of unclear, can I fix it?" – "Just do it" – "I've got an old fish tank at home, how about we set it up in the space?" "Just do it." For the members this seemed to pose no problem at all and I began to notice that "just do it" seemed to express some kind of autodidactic stance that also pervaded members' practices: in fact, the individual pursuit of knowledge and skill seemed to be valued higher than the actual outcome. This became particularly evident on one occasion, where I told some people in Burgstadt a story of how I had managed to destroy the GPU driver of my laptop while experimenting with Linux. This tale of failure was met with great acclaim and members began to tell tales of their own failed attempts. I began to understand that "just do it" was by no means the harsh rebuke I had perceived it to be – rather it was an encouraging way to tell newbies to find their own way.

These reflections on my initial experiences in the field further develop the methodological point presented in the previous section: fieldwork requires involvement; the ethnographer assumes a position in the field and not in an aloft observation post. Without the attainment of practical knowledge even an observational role may be difficult to achieve, as the ethnographer is not able to "see anything" in the practices of the members. Although the members had provided me with a hint by telling me to "just do," this was not enough: through mere observation, I was not able to perceive the "missing what" (Garfinkel 2002: 99) of "just doing hacking." While I had achieved an understanding of the general stance associated with hackerspace practices, their specific nature still was opaque to me. I experienced situation very similar to Eisenmann and Mitchell during their study of Taijiquan and Yoga practices: just as the "internal work" (2015: 3) of meditation can only be encountered through active engagement, the pursuit of "hacking knowledge" seemed to require a practical effort. Thus, I adopted my own project

3 A spacebot is a central control device that operates the infrastructure of a hackerspace: It operates the lights, the sound system or electrics and sometimes even provides additional services, such as a timetable of the next train or bus stops. Spacebots are closely related to the culture of the respective space: They are carefully crafted and individually designed.

of learning the Python programming language. In doing so, I hoped to satisfy the autodidactic requirement posed by the principle of "just do it." In the following section I further clarify this point by presenting and analysing materials from the studies of coding that I conducted myself. Through reflection of my own immersive efforts of learning the craft of programming as an ordinary hackerspace skill, I was able to develop a basic knowledge of field practices that could be utilised in further fieldwork as well as in the subsequent analysis of observational accounts.

Practising in order to observe: Doing coding

One of the main activities in Weststadt was coding. While there were plenty of other activities and projects that were more directly associated with "making" (such as the etching of circuit boards or 3D printing), programming was still one of the most common occupations. Usually, these coding projects were quite sophisticated ones, such as custom minecraft servers, alternative firmware for computers or simply home-brew video games. I had, of course, no hope to accomplish projects like that any time soon, but that wasn't my goal anyway: for my purposes it sufficed to learn the basics of coding. The idea was to emulate not the exact practices of the hackerspace members, but the autodidactic stance associated with them. Thereby, I tried to approximate the experience of coding and to achieve a closer understanding of the practices of skill and knowledge attainment displayed by the field members. The findings of this effort are presented in this section. Although I visited the field frequently during that period of time, I conducted these studies at home. This practice was not unusual among certain senior members, who preferred to work mostly at home and to come to the hackerspace mainly in search for talk and socialising. As the work of coding is a rather absorbing and silent one, this also allowed me to focus on interaction with the field members during the valuable field time.

As mentioned before, ethnomethodology has given rise to a great variety of studies of skill and reason. In my studies of coding I utilised an "ethnographic" approach as developed by Livingston that aims for a deeper understanding of practices by describing their characteristic "howness" (2008: 258).[4] The goal is not only to attain intimacy with a practice by actively engaging it, but to document this process in order to craft a detailed description of one's own development of knowledge and skill.[5] This is accomplished by "the stack" (ibid: 139), where notes about the activity under consideration are collected. As the notes are cultivated mainly

4 Thus, this method explicitly aims for an analysis of the haecceity of a certain practice.

5 Contrary to other ethnomethodological approaches that aim for a study of embodied knowledge through observation and the use of recording devices (Streeck 1996: 367; Nishizaka 2006: 123), the analytic emphasis here lies on the experience of the researcher (cf. Sudnow 1978; Livingston 2006).

during the activity and ordered only by its progress, the stack gives access to the temporal order of reasoning through which the particular skill was developed: it documents the thoughts, feelings, conclusions and dead ends the researcher went through during her pursuit of practical knowledge. The goal of this approach is to describe the defining core of a practice as it brings into focus the ordinary skills and knowledge that are invisible in the conduct of experienced practitioners. The following descriptions are derived from the notes that I accumulated in my stack during my studies of coding.

At the beginning of my efforts, I had only a very limited understanding of programming. Although I knew some general principles and had been learning Java once, I was barely a novice. I decided to conduct my studies with the Raspberry Pi single-board computer I had bought at some early point of the ethnography. It came with a pre-installed version of the Python programming language, and it seemed natural to use it, as it would help me not only to learn Python but to get to know the Pi itself. As a single-board computer, the Pi consists only of a circuit board the size of a credit card. In order to use it, one has to plug in a monitor and the usual input devices (i. e. mouse, keyboard). Therefore, every time I wanted to use the Pi, I had to "rig" my desk accordingly (Figure 1).

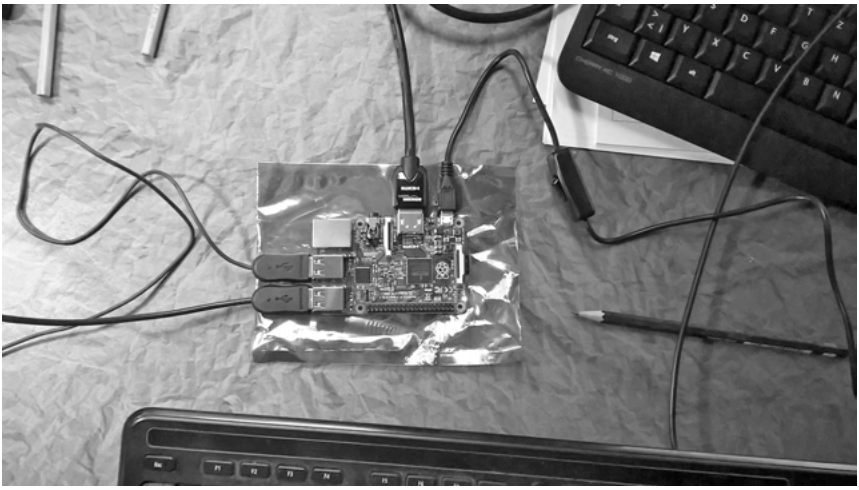

Figure 1: The Raspberry Pi on my desk

Thus the Pi was "part of the field," inasmuch as it represented a "praxeological object" (Livingston 2008: 227) that could be used to pursue the field practises at my desk. I am going to demonstrate these explorations by using a quite trivial example of my early efforts. It is derived from an exercise provided by Sweigart (2012: 21). The book presents several step-by-step instructions to teach beginners the programming of simple video games with Python. One of the basic exercises involves the creation of a desktop window that displays several geometric shapes. After completing this task I wanted to experiment with the commands I had

learned in order to gain a better understanding of Python. My first goal was to draw a yellow pentagon on blue ground. I wrote a short programme based on what I had already learned (Figure 2).

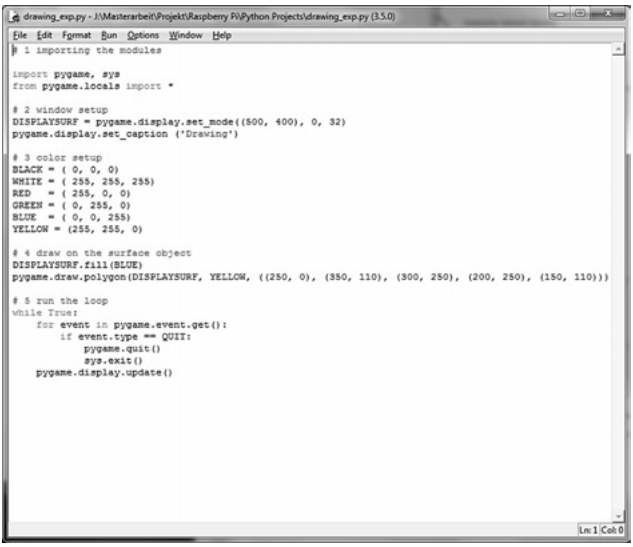

Figure 2: The code

While the task I set myself to do seemed to be simple enough, it involved several steps to be programmed. As Figure 2 shows, the code is segmented into five sections according to these steps. As I had written a similar programme before, I could use this structure and did not have to start fully anew. First, the required modules of Python had to be loaded (1). They contain the commands I was going to use in the subsequent steps. Step (2) and (3) are mainly dedicated to the creation of the objects I needed for drawing: in step (2), an empty desktop window which can be drawn upon is created. In step (3) I created different colour objects – although in this example I was using only blue and yellow. Step (4) is dedicated to the drawing of the pentagon itself, while in step (5) the programme is executed. Upon execution, the programme would create a desktop window which displayed the picture shown in Figure 3.

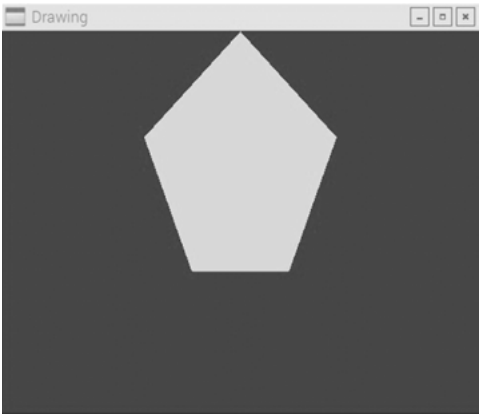

Figure 3: The pentagon

The pentagon is a direct result of the code displayed in Figure 2. In order to create it, I had not only

to perform the steps in the correct succession but also to use the correct Python syntax. For example, objects have to be created by storing them in a variable. To create the window in (2), I had to type: "DISPLAYSURF=pygame.display.set_mode((500, 400), 0, 32)." The variable I created is named "DISPLAYSURF" and has to be placed on the left side of the equation sign. On the right side the actual object is created by the command "pygame.display.set_mode((500, 400), 0, 32)." Simply put, I told Python to create a window object that is 500 pixels wide and 400 pixels high and to store it in a variable called "DISPLAYSURF." The colours in (3) are created in a similar fashion, as the numbers in the brackets refer to the red, green and blue values that are used to create the colour: "(255, 0, 0)," for example, creates red, "(0, 255, 0)" creates green and so on. The pentagon is created in (4): as the window object forms a 500×400 coordinate plane, the corners of the pentagon can be defined by their Cartesian coordinates: (250, 0), for example, refers to a point at y=0 (i. e. at the top border of the window) and x=250.

While the code seems straightforward enough on paper, its creation followed some kind of trial-and-error process, which originated in the autodidactic stance I was trying to adopt: I wasn't very skilled in the use of Python, and even though I had done similar exercises before, I was trying to create the pentagon from scratch. The main source of difficulties was part (4): although I knew the syntax for creating a polygon, my first attempts were less that satisfying: instead of creating a regular pentagon as displayed earlier, my drawing was askew (Figure 4). At first I could not make out the source of the problem and I needed several tries to fix the issue: while I had used the correct syntax and coordinates, I had inverted the x and y axis of the window, resulting in my calculations of the corner points being wrong.

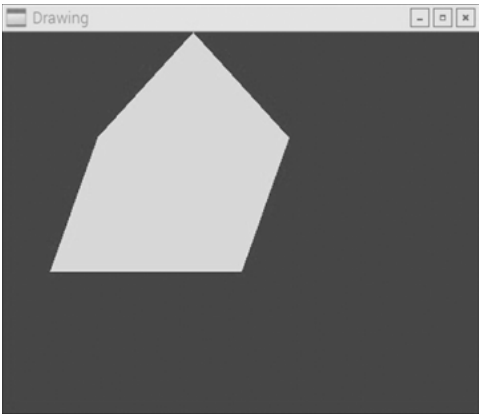

Figure 4: The distorted pentagon

Therefore I typed "pygame.draw.polygon (DISPLAYSURF, YELLOW, ((200, 0), (300, 110), (250, 250), (50, 250), (100, 110)," which resulted in the distorted pentagon earlier. I recalculated the values accordingly, but I still needed several tries, as I miscalculated some coordinates or simply performed syntax errors, which led to

the programme being aborted and an error message being displayed. The latter demanded a deciphering of the error message, as the indications given by Python in these cases tend to be rather general: I had to search the code until I came upon a missing bracket or colon, which are common mistakes in novice code writing.

Thus, the creation of the pentagon in Figure 3 involved several iterations of code writing, execution of the programme and checking the results. Even this trivial exercise proved to be quite demanding for me and consumed a surprisingly large amount of time. However, it also led to unexpected insights regarding the craft of programming and the autodidactic stance, namely the experience of coding as a material craft: during the process of coding, the pentagon is not directly accessible. It can be created and manipulated through the code and only through the code, a structure of abstract objects, which is essential for the creation of the pentagon, although it remains invisible in the graphic itself. Consequently, the work of drawing the pentagon is not actually drawing at all, but rather the writing of code according to a specific syntax. The problems I encountered in writing the code reflect the nature of this connection: in my first try, I failed to shape the pentagon after my imagination, because I failed to perceive the pentagon in the code, as it is represented by a set of abstract logical relations between objects, variables and functions.

Thus, the programme consists of two different forms of materiality: the visual realisation of the image and the syntactic structure of the code. Image and code each allow for a very different perception of the same object: while the visual features of the pentagon such as colour and shape are accessible only in the image, the code contains the abstract properties that lead to this visual realisation, for example the exact coordinates of the corner points or the colour values. Thus, although the pentagon can be perceived visually only in the picture, the means of its "mechanical" manipulation lie in the code. The work of coding therefore consists not only of writing and reading the correct syntax, but more importantly, of the simultaneous perception of code and outcome – in this case the pentagon.[6]

In my attempt to learn these skills, I adopted a trial-and-error attitude: as I was able to spot my mistakes only after running the programme and reviewing the resulting image, my work flow consisted of alternating code writing and the review of mistakes. As my understanding grew, I started to view the code as a tool: it had specific functions, limitations and requirements, which I was exploring by trying them out. This was a strange experience; I've never had before using a computer: by using the code, I was able to actually create objects from scratch and thereby exploring the layers of software technology hidden behind graphical user interfaces. This insight links the work of coding to the autodidactic stance

6 This work remains inaccessible when coders are solely observed. While Rooksby highlights the importance of "sharing an understanding" (2011: 184) while programming collaboratively the embodied knowledge that is necessary to achieve this understanding individually is overlooked.

displayed in hackerspace practices. Not only had I adopted a similar attitude as the members, I had also done so by working on similar problems as they did. The experience of the material properties of code enabled me to view programming literally as a craft that enabled its practitioners to explore, manipulate and create the "hidden" layers of software. I began to understand the joy that comes from experimenting with the possibilities and limitations of technology: I came upon hackerspace knowledge which I could not only use in the fieldwork, but which posed a first analytical concept in itself. Through the engagement in a field practice, I had developed embodied knowledge about that practice, which allowed for a deeper insight into field members' conducts. In the next section, I am going to demonstrate how these insights gained from practical knowledge can open new perspectives on ethnographic material, such as members' accounts and observations.

First-hand experience as analytic resource

The active engagement in "doing coding" resulted in three major achievements, the first and most obvious one being the possibilities that opened up for me during fieldwork. Through my practical knowledge I could *engage* in field activities that I had been merely observing by that time, such as the talk about one's projects and the telling of anecdotes. Although my own experience was very limited, I could now offer my own perspective as a novice and thereby spark lengthy narrations about the practice of coding by the members themselves. At one evening, I had a conversation with Jens, one of the more active members of the Weststadt hackerspace. Originally, I had asked him about the proceedings of an open source project he was involved in, but at some point we started an excursus about programming in general which resulted in Jens offering some biographical insight his own early experience of programming as a teenager. This included memories of him experimenting with Windows 98 "running on a 133 megahertz processor, such a weeny thing made Hewlett Packart" that eventually led to the basic question which he stated as the origin of his involvement with coding: "How does this work? Somebody must have made this, you can't simply define every single pixel."

Members did not only share childhood memories; they also opened up to talk about their own unsatisfactory proceedings with their projects. On one occasion I told Michael, senior member of the Burgstadt hackerspace, about my struggles to find time to experiment with my Raspberry Pi. He responded by showing me his own long-time project of repairing an ancient game controller "I'm actually working on this for over a year now, but most of the time it's just lying around," indicating that finding time for one's project was a common problem.

The situations sketched earlier show that my practical effort enabled me to take part in members' practices, namely the telling of anecdotes and the sharing of project results. Through my practical knowledge, I could be addressed *as a practitioner* rather than an observant which resulted in a deeper involvement into

the field. Practical engagement proved to be the key to an understanding of the "ethno-methods" of the field: only one who is engaged in a project is able to participate in talk about projects.

Second, practical experience offers new perspectives on members' accounts. Some of the properties of coding that I had discovered in my own studies could also be found in members' accounts, namely the perception of coding as a "material" craft, nicely illustrated by another quote by Jens:

> I started programming very early indeed. And it was a craft where you got very quick results. [...] I've always wanted to craft stuff, I wanted to do sculpting, I wanted to do carving, I wanted to do turnery – well, I did that once, carving. But it always took so long and you didn't have that sense of achievement.

Although Jens was the only one to elaborate this, the perception of code as a material object was prevalent in members usually referring to the activity of programming as "*building* code" ("*Code bauen*"). The insights derived from my practical studies of coding are able to add another dimension of understanding to these accounts as they are able to show *how* the material properties of code can be experienced. The material connection between code and coded object resulting in the "toolic" qualities that I have shown in the previous section usually are not elaborated by the members, although they form the precondition that enables them to perceive code in such a way. In order to fully understand the casual metaphor of "building code" as the iterative, experimental combination of abstract rules, one has to engage in the activity themselves.

Furthermore, the ethnomethodological description of my coding efforts is able to show how the autodidactic stance conveyed in the dictum of "just do it" can be found in the practice of coding itself. Coding, as experienced by me, involved "exploring" possibilities and restrictions of the Python programming language by adopting a "trial-and-error" attitude. Many of the activities in the hackerspaces seemed to follow a similar pattern of exploring without a fixed goal: there was an occasion when a Weststadt hackerspace member reconstructed the encryption algorithm of the WhatsApp mobile messenger on a whiteboard "just to figure out how this thing works." By engaging with my studies of coding, I was able to adopt a similar approach as the hackerspace members and therefore experience the practice of coding in a similar way as they did. Although they were far more skilled than me, they were constantly trying new things and thereby adopting a playful "trial-and-error" attitude themselves, which could lead to unforeseen projects. The ethnomethodological description again offers insight in how this experimental approach is performed as a practice of coding through writing and rewriting after the checking of results. Thus, an ethnomethodologically structured engagement in members' practices does not only help to understand events and meanings in the field but also adds a new dimension to the ethnographic description of observations and members' accounts: it aims at the development of a practitioners'

perspective to articulate qualities of practices "taken for granted" (Garfinkel 2002) by field members themselves. Speech figures like "just do it," "building code" and so on convey meanings that can only be understood by engaging in the practices to which the expressions refer. Consequently, in order to understand the "strange" doings of the hackerspace members, I had to do what they had been telling me all the time: "just do it."

Conclusion

The goal of this article is to demonstrate an ethnomethodological approach to ethnography dedicated to the study of skills and knowledge in hackerspaces. Emphasis is placed on embodied knowledge as an important requirement for the analysis and understanding of members' practices. This is achieved by a methodological reflection of my own fieldwork process during my ethnography of hackerspace practices of knowledge production. I identify essential stages of practical insight as they occurred during the process of "doing ethnography": first, the identification of field requirements posed by members' accounts during the earlier phases of fieldwork ("observing in order to practice"); second, the active realisation of those requirements in order to achieve a practical understanding of members' practices that could be utilised for further fieldwork ("practising in order to observe"). Both stages point to an ethnomethodological understanding of ethnography that regards fieldwork as an ongoing accomplishment that demands a reflexive engagement with members' practices. The focus on learning and the acquisition of skill can provide for rich and deep ethnographic accounts that accommodate the unique nature of practices of hacking and making.

Through the development of field-specific skills and knowledge, it was possible to gain a reflexive insight into the practice of coding. In assuming such a "practical" stance towards fieldwork, I was able to utilise my embodied knowledge to come to a better understanding of field practices and members' meanings. It became clear that the practices of hacking construct technology as "hackable" by engaging it in a playful and experimental way.

Several aspects could only be addressed briefly, if at all. First, the presented findings raise questions after a broader theoretical framework in which hackerspaces can be conceived. I suggest to describe hackerspaces as "laboratories" taking the meaning of Knorr Cetina (1988). Practices of "autodidactic experimentation" could be part of a unique principle of "knowledge production" (ibid: 87–88) that emphasises an individualised acquirement of skill and technology. This connection would have to be pursued in a further article. Second, correspondences between ethnomethodology and practice theory (Reckwitz 2008: 99) need to be specified in regard to the field of hacking and making. In particular, questions of materiality and practice (Hillebrandt 2016: 82–84) pose an interesting field for further research.

List of references

Coleman, Gabriella (2010): "The Hacker Conference: A Ritual Condensation and Celebration of a Lifeworld." In: Anthropological Quarterly 83(1), pp. 47–72.

Coleman, Gabriella (2013): Coding Freedom. The Ethics and Aesthetics of Hacking. Princeton: Princeton University Press (http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10632186).

Eisenmann, Clemens/Mitchell, Robert (2015): "Ethnomethodology's Body. Making Moves in Taijiquan and Yoga Practice." Unpublished manuscript.

Eisewicht, P./Kirschner, H. (2015): "Giving In on the Field. Localizing Life-World Analytic Ethnography in Mediatized Fields." In: Journal of Contemporary Ethnography 44(5), pp. 657–673.

Ellis, Carolyn/Adams, Tony E./Bochner, Arthur P. (2010): Autoethnography: An Overview. In: Forum: Qualitative Social Research 12(1, Art. 10), pp. 1–18 (http://www.qualitative-research.net/index.php/fqs/article/view/1589), checked on 8/2/2016.

Emerson, Robert M./Fretz, Rachel I./Shaw, Linda L. (2011): Writing Ethnographic Fieldnotes. Second Edition. Chicago: University of Chicago Press (Chicago guides to writing, editing and publishing).

Emirbayer, Mustafa/Maynard, Douglas W. (2011): Pragmatism and Ethnomethodology. In: Qualitative Sociology 34(1), pp. 221–261.

Garfinkel, Harold (2002): Ethnomethodology's Program. Working out Durkheim's Aphorism. Lanham: Rowman & Littlefield (Legacies of social thought).

Garfinkel, Harold (2010): Studies in Ethnomethodology. Reprint. Cambridge: Polity Press.

Garfinkel, Harold/Warfield Rawls, Anne (2006): Seeing Sociologically. The Routine Grounds of Social Action. Boulder, CO: Paradigm Publishers (Great Barrington books).

Garfinkel, Harold/Wieder, D. Lawrence (1992): Two Incommensurable, Asymmetrically Alternate Technologies of Social Analysis. In: Graham Watson (ed.), Text in Context. Contributions to Ethnomethodology. 1. print. Newbury Park: Sage Publishers (Sage focus editions, 132), pp. 175–206.

Hammersley, Martyn/Atkinson, Paul (2007): Ethnography. Principles in Practice. Third Edition. London: Routledge.

Hillebrandt, Frank (2016): "Die Soziologie der Praxis als post-strukturalistischer Materialismus." In: Hilmar Schäfer (ed.), Praxistheorie. Ein soziologisches Forschungsprogramm. Bielefeld: transcript (Sozialtheorie), pp. 71–93.

Hirschauer, Stefan (2007): Putting Things into Words. Ethnographic Description and the Silence of the Social. In Human Studies 29(4), pp. 413–441.

Jordan, Tim/Taylor, Paul (1998): A Sociology of Hackers. In: Sociological Review 46(4), pp. 757–780.

Knorr Cetina, Karin (1988): "Das naturwissenschaftliche Labor als Ort der 'Verdichtung' von Gesellschaft." In: Zeitschrift für Soziologie 17(2), pp. 85–101.

Livingston, Eric (2006): "The Context of Proving." In: Social Studies of Science 36(1), pp. 39–68.

Livingston, Eric (2008): Ethnographies of Reason. Aldershot: Ashgate (Directions in ethnomethodology and conversation analysis).

Nishizaka, Aug (2006): What to Learn. The Embodied Structure of the Environment. In: Research on Language & Social Interaction 39(2), pp. 119–154.

Pollner, Melvin/Emerson, Robert M. (2001): Ethnomethodology and Ethnography. In: Paul Atkinson/Amanda Coffey/Sara Delamont/John Lofland/Lyn Lofland (eds.), Handbook of Ethnography. London: Sage Publications, pp. 118–135.

Rawls, Anne Warfield (2011): "Garfinkel, Ethnomethodology and the Defining Questions of Pragmatism." In Qualitative Sociology 34(1), pp. 277–282.

Reckwitz, Andreas (2008): "Grundelemente einer Theorie sozialer Praktiken." In: Reckwitz Andreas (ed.), Unscharfe Grenzen. Perspektiven der Kultursoziologie. Bielefeld: transcript, pp. 97–130.

Rooksby, John (2011): Text at Work: Mundane Practices of Reading in Workplaces. In: Mark Rouncefield, Peter Tolmie (eds.), Ethnomethodology at Work. Farnham: Ashgate (Directions in ethnomethodology and conversation analysis), pp. 173–189.

Schütz, Alfred (1964): Studies in Social Theory. Photomechanical reprint, 3. print (Phaenomenologica, 15).

Star, Susan Leigh (2002): "Infrastructure and Ethnographic Practice. Working on the Fringes." In: Scandinavian Journal of Information Systems 14(2), pp. 107–122.

Steinmetz, Kevin F. (2014): "Craft(y)ness." In: CRIMIN 55(1), pp. 125–145.

Streeck, Jürgen (1996): How to Do Things with Things. In: Human Studies 19(4), pp. 365–384.

Sudnow, David (1978): Ways of the Hand. The Organization of Improvised Conduct. 3. print. Cambridge, MA: Harvard University Press.

Sweigart, Al (2012): Making Games with Python & Pygame. A guide to Programming with Graphics, Animation, and Sound. First Edition. Charleston, SC: Creative Commons.

Toombs, Austin/Bardzell, Shaowen/Bardzell, Jeffrey (2014): Becoming Makers: Hackerspace Member Habits, Values and Identities. In: The Journal of Peer Production 4(5) (http://peerproduction.net/issues/issue-5-shared-machine-shops/).

Wagenknecht, Susann/Korn, Matthias (2016): "Hacking as Transgressive Infrastructuring: Mobile Phone Networks and the German Chaos Computer Club." In: Darren Gergle/Meredith Ringel Morris/Pernille Bjørn/Joseph Konstan (eds.), CSCW'16. Proceedings & Companion of the ACM Conference on Computer-Supported Cooperative Work and Social Computing. New York, NY: Association for Computing Machinery, pp. 1102–1115.