Building, fixing and learning how to fix things in the face of automation

Viktoria Pammer-Schindler, Graz University of Technology, Austria & Know-Center GmbH, Austria

ACM Reference Format:
Viktoria Pammer-Schindler, Graz University of Technology, Austria & Know-Center GmbH, Austria. 2023. Building, fixing and learning how to fix things in the face of automation. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Automated systems often build something. The case that I'm using in this paper to develop my overall argument is set in car series manufacturing. However, far beyond this single use case, most mass-market products today are built using at least partially automated production systems. In addition, there is a very recent and active discourse around AI (artificial intelligence) systems that produce output that is of a more intellectual or creative type, such as ChatGPT for producing text or code; CoPilot for producing code, or DALL-E for producing images.

This article continues a discussion that is ongoing both in research and in society on the automation and future of work. The outline of the typical narrative both in research and society is that many sectors of work are experiencing rapid digitalisation, as well as automation; that a substantial number of remaining jobs will be knowledge-intensive, and lifelong learning and organisational learning will gain in importance across all sectors (e.g., [1, 2]) and specifically also in manufacturing (e.g., [8]). In this paper I do take a learning perspective as well, but from a slightly different angle: Given that building things (production, content creation) is increasingly automated, I argue that an important activity for humans is to fix any resulting errors. Secondly, I argue that this might require creating new learning opportunities, such that we can learn to fix things even though we haven't built them.

In this paper, I firstly lay out the case study (Section 2). It is set in car series manufacturing, where we specifically analysed rework. By rework we mean all activities directed towards eliminating faults that occur in highly automated series production, either in individual cars or through analysing and eliminating systematic causes of faults. The full case study, analysed from a different angle than that discussed in this paper, is documented in [5]. Rework is a "ghostly" labour in the sense that the reality that automated production requires manual fixing if errors occur is not a substantial part of public nor academic discourse.

Secondly, I will analyse this case study from a learning perspective, in the sense of asking what workers in rework need to be able to do, and how they learn doing it (Section 3). From this perspective onward, I discuss two key arguments, namely 1) that fixing things is an important human activity in an increasingly automated world; and 2) that there is need for finding and creating new learning opportunities, which could be taken up as a design opportunity and challenge within HCI, and will hopefully spark off interesting discussions at the workshop on "Behind the Scenes of Automation: Ghostly Care-Work, Maintenance, and Interference".

From a methodological viewpoint, this paper should be considered as consisting of a secondary analysis of a case study and subsequent development of a new perspective, namely a learning perspective, for analysing automation further. The original case study focused on key aspects of knowledge work, including understanding the nature and complexity of knowledge required in order to do the job, and the necessary collaboration, cooperation and communication. Naturally, learning does play an important role in knowledge work, which makes the present discussion possible.

2 REWORK IN CAR SERIES PRODUCTION

Rework in car series production are all activities that aim to detect, analyse and eliminate errors in cars and in the preceding production systems. In [5], we have described rework in a case study set in a global car manufacturing company, which produces complete vehicles for international customers (= car brand companies). There, we have analysed rework from a knowledge work perspective, highlighting that rework is knowledge-intensive, highly collaborative and involves complex dynamics of communication between stakeholders with a wide range of competencies and roles (ibid).

Data collection in this case study happened in a contextual inquiry (cp. [4]) carried out over the course of three consecutive months in 2019 in three different assembly units.

Each assembly unit is configured to produce a single car platform (something a bit more generic than a car model; e.g., a car platform can still be configured into multiple - similar - car models) for one or more customers. Each assembly unit is responsible both for planning the production of the car platform, and for operations, i.e., actually producing the car platform in series up to the point where certified single cars leave the factory and are shipped. This means, that each assembly unit also has to account for testing and, where necessary, repairing single cars, adapting the assembly process should systematic faults be identified, or triggering changes in the car design or in other parts outside of the car manufacturing company including software.

Production is highly automated, and workers in the production lines work on a strict time budget for each step in the production process. Once a single car comes out of the production line, it is moved to systematic testing ("off-track", cp. [5]). In case any errors are detected in this testing process, the single car is moved to the rework zone. Each assembly unit has their own space and team for rework. Rework activities include error detection (identifying

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

that there is an error, understanding its type and location in the car) error treatment (fixing the error), and rework management (organising all rework activities including coordinating repairs that need multiple specialities, or that need the inclusion of stakeholders beyond the rework team such as quality management, software development, or customer representatives).

Assembly units are independent organisational units, which, despite their being part of the same car manufacturing company with the same set of overarching processes, have some differences. These differences are due to team set-up, but also specifics of the produced car platform and customer requirements and processes (customers are car brand companies).

3 ANALYSIS: REWORK FROM A LEARNING PERSPECTIVE

This section discusses rework from a learning perspective, still staying close to the above case study. The understanding of "learning" here has been developed on the wider background of workplace learning (cp. [3] for a discussion on nuanced differences in defining workplace learning, and an overview of applicable theories). More specifically, it is substantially influenced by two learning theories, both of which could be considered socio-technical learning theories. Firstly, this is activity theory (e.g., [6]) that frames understanding learning as happening through (meaningful) activities, set in a social context by being embedded in a community, and set in a historical context by being shaped by available concepts and tools (both human-made, the one immaterial, the other more material). Secondly, this is the theory of communities of practice (e.g., [9]), which has been developed on the background of substantial empirical work as a conceptualisation of learning that happens within a community that shares a set of practices ("community of practice").

Rework is necessary for automated production to work. Errors do occur in highly automated (car) series production, and someone needs to be able to fix them. These people need, as a team, to have an understanding of i) how the - by now extremely complex - cars work, ii) how to systematically analyse errors and find solutions, and iii) some understanding at least of how the - by now extremely complex - production process works.

Workers in rework are experienced workers with a diverse set of professional backgrounds. Workers in rework come from a variety of backgrounds: most are car mechanics by training, but others have a higher education degree in engineering. Of course, specialisation exists in rework, for instance, workers specialise in either mechanics or electronics. This complements the above-described mechanisms that are in place for handing over the handling of faults to departments outside of rework and hence outside of production (rework is located, as a department, in production), such as engineering or software development. Wherever workers in rework come from, however, they have developed expertise outside rework before moving into rework as a promotion. This recognises that fixing things requires substantial expertise.
Within rework then, workers again go through steps of informal apprenticeship. An expression of this is the fact that leaders in rework have substantial expertise in rework, such that their leadership is legitimated through their expertise in the domain of work that they are leading. Additionally, the whole workforce within rework is regularly re-trained to be competent for new models, and new technologies built into newer car models. This re-training includes making cars of new models available to rework to take apart and to "play around with" as a basis for being able to fix any errors that occur in series production.

There isn't a joint community of practice for building and fixing cars. Seeing this case from the perspective of communities of practice, we notice two things. Firstly, there is no joint community of practice around building and fixing cars. Workers aren't learning to fix things by first participating in building things and then moving on to fixing them in the same community of practices. Rather, workers who "fix things" come into rework with already substantial professional expertise that has been developed throughout different formal education settings and within different communities of practice. In short, workers in rework have a diverse set of professional expertise. Further, as car series production of cars is highly automated and needs little of the skilled workforce that is needed in rework, prior professional expertise typically doesn't include expertise in manually building cars. Rather, expertise is in car mechanics, automotive engineering including electrical and electronics engineering, or software development. Overall, there is a substantial separation between production and error handling (error detection, treatment, and management of these activities).

Is rework a community of practice? This isn't entirely clear for the case study that was carried out: There is shared practice around fixing things for frequent errors that are handled within rework (and not moved outwards to engineering, software development, etc.) at least via shared organisational and spatial work structures; and some shared learning opportunities ("playing around" with cars of new models). This is shared practice at a rather formal level. At the individual and case-specific level, strategies for understanding errors (type, location) are informal and seem to be substantially shaped by experience, in the sense of "with experience, people become better at finding and fixing errors". Further, impressions from the available case study are that this perception of belonging exists, and that substantial pride is associated with being in rework, especially as opposed to being in series production.
At this point the secondary analysis reaches its limit. Especially, it would be interesting to know more about details of how much operative, individual work practice is really learned through participation in joint activities, and whether and what parts of practice are really shared. Due to specialisation, this isn't as clear as it could be. Knowing this would help designing learning opportunities that are potentially supported by computers. This argument is further developed in this section below, and discussion in the next Section 4.

In rework, learning is tightly embedded with working, and neither is well supported with digital technology. Using activity theory, we can frame rework as follows: In alignment with the organisational objective to produce cars that can be shipped (with all necessary quality measures in place) to customers, the collective objective of the rework team is to fix erroneous cars; and the individual objects of work are the single erroneous cars. This objective may shift if the error is more systematic. In pursuing these objectives, through doing rework, individuals and the rework team as a whole develop further, i.e., learn. Learning is embedded within work practice. There are a number of formal tools for documentation, reporting and communication. There are also standardised procedures for identifying errors (strictly speaking, such testing happens outside rework). Interestingly, such standards and tools or digital support aren't available for in-depth error analysis, nor for fixing them. Here, work becomes much more case-by-case, and knowledge intensive. Of course, there are simple errors such as a scratch in the car surface, for which this isn't an issue. Again, we are here reaching the limits of what we can discuss given the existing case study. Further investigation around the delimitation of formal, well-defined concepts and tools used within rework and informal, highly diverse, individualistic concepts that are shared in a highly informal manner within rework teams would be interesting. Such investigation would probably connect very well with investigating in what sense and how rework constitutes a community of practice.

4 DISCUSSION

Both during the case study, and during our above analysis of the case study from a learning perspective, we developed the following arguments and questions and challenge for HCI research.

4.1 Fixing Things as an Important Human Activity in the Face of Automated production

Our primary case study identified that rework is knowledge intensive work: Complex knowledge needs to be applied as intrinsic characteristic of the work [5], needs to be communicated with stakeholders of diverse backgrounds, and regularly, new knowledge needs to be acquired and created (ibid). Further, the analysis in the primary case study characterises rework not only as not automated, but in core parts of rework - the actual error detection and treatment, per individual car and error - as not really supported by digital tools (ibid). At the same time, rework is clearly necessary in order for the series production to work: it is an important human contribution to the automated production.

The question could now be asked, whether it is possible to get rid of this human contribution. Can we fully automate production including error handling (detection, treatment, and organising necessary activities around this)? In a far-out vision, we could for instance imagine that car brand companies don't need a human-run car series manufacturing company anymore. Rather, engineers from car brand companies might give full specifications or something akin to a prompt (as used now for ChatGPT) to a fully automated car series manufacturing plant. The fully automated car series The fully automated car series manufacturing plant would maybe produce several prototypes, asking the customer (=car brand company) for necessary refinements. Then, it would produce the car model (or platform) in series, deal with all occurring errors, and ship the single cars, without further need for human input. Would that work? I argue that no, this wouldn't work, because errors would still occur: Even with the current level of automation, car series manufacturing builds a complex product, namely cars. Cars contain mechanical elements, as well as electronic elements, and software. In addition, the production process itself is already complex due to automation. Hence, analysing errors and treating them has become knowledge-intensive and essentially interdisciplinary work. Error handling activities are to date not standardised at the concrete, case-by-case level, where the experience of the person handling the error is regarded as crucial. This is a weak basis for automation. Further, even assuming that error handling could be automated, the likelihood that also this process could be erroneous is high. Rework would then still be needed, but would need to consider additional layer of complexity, namely also errors in the automated error handling process. My argument is therefore that rework activities will remain human for quite some time, including error detection (understanding type and location of an error in detail) and error treatment.

The case study described in some detail in this paper is set in car series production. Analogies can be made to other sectors, however, of which we mention only two: 1) Software engineering: code is still manually produced to a large extent; however, with the advent of CoPilot and more recently ChatGPT, we can at least envision a future in which this isn't the case. In addition, software engineering has been developing the realisation that testing is an important activity that requires different skills than programming and may even drive programming (test-driven development) for a long time. Testing is now supported with specific tools (e.g., continuous builds, test automation, code coverage checks, concepts). 2) Writing: With ChatGPT, we can at least envision a future in which full texts are first written by a generative AI such ChatGPT, and then fixed by humans if errors have been made. Note that there seems to be a spectrum of post-processing, from fixing errors (in cars, in software, in a text that could be factual errors) to fine-tuning and adaptation (etc. tuning for cars, optimisation or context-specific adaptation in software, improvements in style in texts). For the remainder of this argumentation we will stay with the notion of "fixing" as changing something in the product that absolutely needs some change in order to fulfill standard requirements on the product.

The overall argument is therefore that *fixing errors made in automated production (series production, as well as in AI-based production, e.g., of texts via ChatGPT, source code via CoPilot, or pictures via DALL-E) is an important task of humans in an environment of automated production and content creation.*

Discourse in HCI exists around ChatGPT now, and has existed existing within software engineering around testing for quite some time (less so for bug-fixing to my knowledge). I argue that now *HCI would benefit from an broad investigation of the task of "fixing things"* - whatever word ultimately will be shared across domains: rework, repair, debugging, post-processing etc. - as a task that can be supported by suitable design. Part of this investigation will need to pertain to characteristics and specifics of this task, and part of this investigation needs to investigate *existing (socio-technical) designs* in different sectors (cp. the above description of tools that support testing in software engineering) as a basis for translating existing solutions across sectors.

4.2 Designing for Learning

The above-described shift in production from human involvement in production to being involved rather in fixing things, is paralleled with a shift in the craft professions itself: Most car mechanics today aren't involved in actually building cars. Highly automated car series production is dominant. This runs in parallel to developments in a range of craft professions where the main work that is done by skilled workers isn't building products anymore, but fixing them or making small changes. One example is tailoring, where most tailors outside series production don't cut complete dresses anymore, but rather fix small holes, make changes to the clothes, etc. Other example are car tuning, cycling and bike mechanics, or shoemakers.

The fundamental question from a learning perspective is: How do professionals become good at fixing things if they aren't learning how to build them? This question needs a few comments in order to be sufficiently nuanced. As a first comment, of course, in vocational education, apprentices still learn how to build things, and as we have described above, workers in rework also build back together a car after taking it apart. However, the professional practice of building things isn't dominant anymore. The second comment is, that of course it is possible to learn how to fix things that one hasn't learnt to build. This is as true for minor fixes around the household and for huge domains such as life sciences including medicine. There, the vast majority of the community isn't engaged in building humans or human parts (e.g., artificial teeth, limbs, organs), but is of course engaged in healing (="fixing") humans in a range of professions such as physicians or pharmaceutical chemist.
Following up on our above analysis, we see, however, that some knowledge and competence that is required cannot be built through the activity of fixing things alone. In rework, it is expected to have developed substantial knowledge in car mechanics, automotive engineering, etc. before entering rework. In medicine, substantial formal education is typically required before being certified as a physician. Further, specific learning settings are generated in which professionals can take things apart, such as dissection courses for physicians or the the provision of a car to take apart in order to learn how to fix it in rework. In the case of fixing small things around the household, this may be harder to see, but it could be speculated that some basic human skills (fine motor skills, three dimensional imagination etc.) are required that many healthy adults will have acquired throughout childhood. Remember, that many children do build things but before that take them apart (though not even intentionally of course).

This finally, leads to the overall argument that *in a future in which production of a wide range of products, including intellectual products such as texts, images, music, etc. is automated, it will become increasingly important to develop learning settings and learning opportunities that provide sufficient possibility for building as a basis for fixing things*. In the case study above, this was already done - without computational support - by providing concrete cars to rework teams just as an object of learning.
The rationale is twofold: The first is that this is necessary in order for production to work (unless we are envisioning a future in which no errors in complex production systems occur, which is unlikely). The second one, directly arguable on the basis both of activity theory and the theory of communities of practice: We as humans become who we are through what we are doing, and through meaningful participation in communities of practice. It can be set as an individual goal to make one's life such that learning is possible; but it could also be set as a goal in HCI to design computational tools and socio-technical practice around them such that they support lifelong situated learning (cp. e.g., [7]), as opposed to just automating without considering how that impacts users'/workers'/humans' lifelong learning and ultimately then competence and agency in the world. So *designing for learning in designing for automation* is ultimately a matter both of making production and subsequently society effective, as well as a matter of making automation serve us as humans and doesn't cripple our competence and subsequently agency in the world. From a learning perspective, we should automate processes and tasks that we as individual humans or as society don't want to be able to do (or are easily able to do anyway); and we shouldn't automate processes and tasks that we as individuals humans or as society want to be able to do. Where automation overall still makes sense, we need to be aware that just for the pleasure of doing things, we can create learning-and-doing-opportunities.

Designing for learning is a discourse within HCI that is growing, if anything, as is visible by the comparatively young CHI subcommittee on learning, education and families that has this year had two subcommittees instead of one. The question and opportunity is, I argue, now for HCI and educational technology research is to further develop ways to computationally support situated learning about mechanisms and characteristics of "things" (cars, shoes, clothes, texts, images) as well as about error handling/postprocessing/adaptation within activities around fixing things.

5 CONCLUSION

This paper puts forward using (socio-technical) learning theories, such as activity theory or the theory of communities of practice, as valuable lenses for analysing human labour in substantially automated work environments. In applying these theories to a specific argument, this paper further contributes the observation and arguments 1) that fixing things is an important human activity in an increasingly automated world; and 2) that there is need for finding and creating new learning opportunities, which could be taken up as a design opportunity and challenge within HCI.

REFERENCES

[1] Bernard Burnes, C. Cooper, and P West. 2003. Organisational Learning: the new management paradigm? *Management Decision* 41, 5 (2003), 452–464. https://doi.org/10.1108/00251740310479304
[2] Carl Benedikt Frey and Michael A. Osborne. 2017. The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change* 114 (2017), 254–280. https://doi.org/10.1016/j.techfore.2016.08.019
[3] Paul Hager. 2011. Theories of workplace learning. (01 2011), 17–31. https://doi.org/10.4135/9781446200940.n2
[4] Karen Holtzblatt and Hugh Beyer. 2017. *Contextual Design. Design for Life.* Morgan Kaufmann.
[5] Kathrin Juhart and Viktoria Pammer-Schindler. 2022. *Rework in Car Series Production through the Lens of Knowledge Work: A Case Study.* Technical Report. Know-Center GmbH, Graz, Austria. https://doi.org/10.31219/osf.io/rm24v
[6] Victor Kaptelinin and Bonnie A. Nardi. 2006. *Acting with Technology: Activity Theory and Interaction Design (Acting with Technology).* The MIT Press.
[7] Yvonne Rogers. 2006. Moving on from Weiser's Vision of Calm Computing: Engaging UbiComp Experiences. In *UbiComp 2006: Ubiquitous Computing*, Paul Dourish and Adrian Friday (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 404–421.
[8] Stefan Thalmann, Angela Fessl, and Viktoria Pammer-Schindler. 2020. How Large Manufacturing Firms Understand the Impact of Digitization: A Learning Perspective. In *Proceedings of the 53rd Hawaii International Conference on System Sciences.* 4806–4815. http://hdl.handle.net/10125/64332
[9] Etienne Wenger. 1998. *Communities of Practice. Learning, Meaning, and Identity.* Cambridge University Press, Cambridge.