Tampere University of Applied Sciences

Designing a Busbar Machine Interface and UX

Tatu Kankaanpää

BACHELOR'S THESIS
May 2020

Degree Programme in Media and Arts
Interactive Media

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media and Arts
Interactive Media

KANKAANPÄÄ, TATU;
Designing a Busbar Machine Interface and UX

Bachelor's thesis 30 pages
May 2020

This thesis describes the design methods and the process of creating an interface and a usability experience for a busbar machine, which was constructed in five months during the third and the fourth quarter of 2019.

The examines the theory of how to assess user experience and then considers how this knowledge was converted into practice in a field where efficiency is the key to success, without trying to sacrifice approachability and understandability, which are easily overlooked when the party that buys the product is not the party that operates the product.

The information presented in this thesis is gathered from some of the latest books available in the field as well as some classic foundation writings of their kind. The observations and imagery are the results of trial and error in attempting to fit the fields of engineering and user experience together, to make a product that is greater than the sum of its parts.

The thesis ends with a conclusion section that describes the outcome of the process and juxtaposes the needs and motives of marketing, engineering and good user experience.

---

ux, user experience, interface design, machine, busbar,

CONTENTS

ABBREVIATIONS AND TERMS

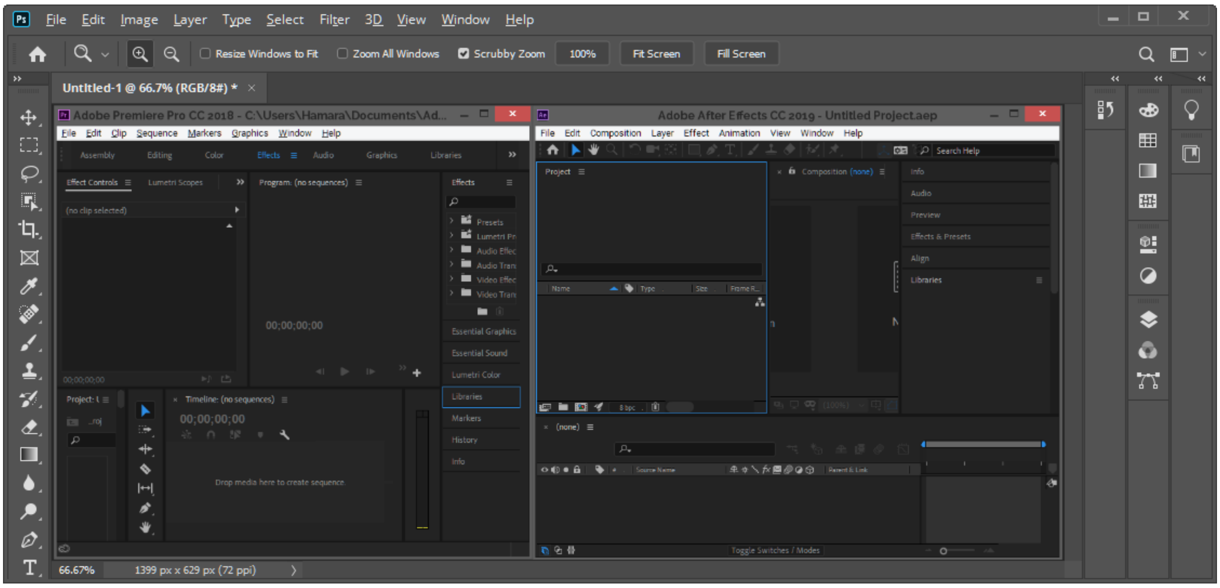
CUI	common user interface
SUS	System Usability Scale
HMI	Human Machine Interface
PLC	Programmable Logic Controller
JSON	JavaScript Object Notation, a real-time server-to-browser communication protocol that allows sites to have more extended features
CSS	Cascading Style Sheets, a file that defines how some elements, such as text or images appear on a site
HTML5	Hypertext Markup Language version 5, the newest iteration of HTML, the most common form of a website file
Busbar / Bus bar	Manufactured metal strips used in high current power distribution and battery banks
RodFINE	Rodstein's latest machine for finishing busbars
RodFLAT	Rodstein's busbar cutting and cutting machine
Modular	Consisting of modules; extensible with other machines or mechanical/digital parts
B2B	Business-to-business marketing, not directed to the general population

1 INTRODUCTION

This thesis deals with user experience of a case machine and the development of an interface for the said machine. The said machine produces busbars, which are plates roughly the size of a TV remote. Busbars are then bent into curved shapes, which then are used in large scale electronics where mere copper wiring won't be enough, such as wind turbines. Prior to the version that was done during a 6-month period from June to December of 2019, the machine lacked an interface. Our goal was to make one for the new version and as such allow customization for the parts the machine grinds without plugging a development system into it via a laptop. Our main hurdle was how to allow an untrained person to work the machine, as that was the biggest selling point of it. There was no prior set user interface standard in place, so we were at the mercy of the tools we had at that point and were to make the best of them before the machine was to be out of the shop floor. In light of what we know of what good UI and UX is, we are to take a look at what kinds of choices we made and what could've been done differently.

2 USER INTERFACE AND CONSISTENCY

Consistency in interfaces allows users to use any program, application or machine without prior knowledge or expertise on it beforehand. For the user, good consistency leads to satisfaction and increased productivity. For a company, good consistency leads to shorter training periods and reduced training costs for operators. For a vendor company, good consistency allows for a coherent architecture and its products evolve in a controlled manner and thus reduce maintenance costs when designers don’t need to design everything from scratch and developers can reuse and maintain a codebase with a clear direction and rules. This also has the added benefit of a better product definition on the market and users are more likely to purchase products of the same product family, to transition their knowhow to another product. (Nielsen, 1989, 3) Modern examples as of 2020, the Adobe product family and the Microsoft Office suite, which all have interfaces akin to each other, even to the point where the actual use case of each programs may be obfuscated to your average user due to their striking similarity in both behavior as well as looks. (PICTURE 1)



PICTURE 1. Screenshots of Adobe Premiere Pro CC 2018 (left) and Adobe After Effects CC 2019 (right) on a layer, side by side inside Adobe Photoshop 2020 of which all belong to the same product family with consistent looks and functions.

2.1 Consistency as an issue

A finished standard may save costs when it comes to development, but at the same time, an outdated standard may prevent product evolution. A standard may even enforce bad design choices, when new products are forced to follow poor rules, which haven't been evaluated for some time. If developers feel they aren't sharing ownership of the user interface under the corporation standards, they're less motivated to do their job. On the other hand, if rules are blindly followed, the developers may feel they don't need to take other factors into consideration at all. There's also a conflict of interests between people trying to make a good user interface and the need to get the product out of the door as soon as possible and it is easy to just follow the set paradigm to get there. (Nielsen, 1989, 6)

2.2 Case Korea

According to the lead programmer of the RodFINE project, there had been a technology exhibition for manufacturing machines in Korea during the summer, which Rodstein had taken part in. They had one of their coiling machines on display and many people stopped by for a user test. It soon became apparent that the interface was confusing for a layman. The reason wasn't that the interface itself was convoluted, but the machine had both physical and digital controls for the same actions. Once the physical buttons were obscured from view with some on the fly craftmanship, users started to work the interface better. Virtually nothing in the UI was changed, but the users were stripped of options that would make them second-guess their actions. Prior to that last-minute fix, the users had expected everything to be a necessary part of the whole. Having a secondary control scheme for supervisors and other people in charge over ordinary operators would still be a necessary part for the machine's deeper functionality, but it would appear the best approach to keep functionality intact is to have the bare minimum of ways to operate a machine.

3 DESIGNING UI

3.1 The practices of user analysis

When designing user experience, it is paramount to understand who your user base is. You cannot design something whose user base is everybody, since different users have different needs and as is with everything produced, too big of a scope will drag the final product's release further away. Designing aspects for a user who never uses your product is effectively working hours wasted on perfecting aspects that are actually relevant to the actual user base. Carla Viviana Coleman lists multiple ways of user research in her book *Visual Experience A Concise Guide to Digital Interface Design*, including:

- *Card Sorting*
    - Users write words, categories and sentences, either predetermined or free and organize them in hierarchies. Closed sorting is optimal for a machine interface that has only a limited amount of preset options. The aim is to make menus which contain the right feature in places where people expect to find them.

- *Contextual interview*
    - Interviews done in an informal setting regarding the real user experience of a product to gather qualitative data. An example could be using a bank app in a coffee shop, but regarding a machine UI, the only surrounding would be a factory or a workshop, which when building any machine, is the default surrounding anyway

- *First-click testing*
    - The contemporary user has a 30 second window to do the right thing before quitting by running out of patience. In a web interface you should aim for a 90% success rate on the first click to set yourself up for a good navigation design.

- In industrial machinery the user is whoever is working on the machine, so they're on a payroll, meaning they don't just swap out of your machine UI for another. It is still a good practice to simulate the same cause and effect with your design as tardiness is relative to the surrounding world and certainly affects sales. While it was acceptable to manufacture trains that would take over 12 hours to travel from the countryside to Helsinki a hundred years ago, it would be laughable now. Human errors and confusion in using any machine in production will also directly impact output of the buyer.

- *Focus group*
  - A focus group is a bunch of users who share their views on the UX offered by the interface. Discussions should be open-ended and flexible to allows users a state of mind that allows them to give unbiased and unguided judgement. When organizing the group data, Coleman suggests dividing by age, occupation, education, experience, gender, ethnicity, hobbies and possibly a short writing where the users describe themselves, to pinpoint possible biases.

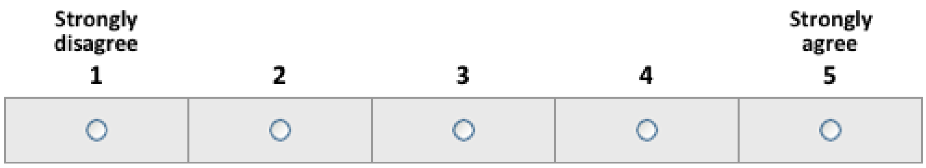
- *Creating personas*
  - The researcher creates personas for users, whose point is to tackle the user experience with set limitation and benefits. The goal of personas is to analyse each persona's background and motivation for usage and narrows down who will be using the interface or product. Coleman suggests organizing persona information by 3 groups:
    - Narrative: Personas with in-depth information
    - Table: Personas with a medium amount of information
    - The quick and dirty: Personas with little information
  - Coleman claims personas not allow for narrowing down the users, but also develop the product brand at early stages.

- *Prototyping*
  - Prototyping is the act of creating a mock-up of an interface to analyse how the user thinks it should work. Prototyping can be divided to low fidelity and high fidelity prototyping, the latter being everything made into

an actual working interface and the prior is everything you can make without electricity.

- *Online surveys*
  - In case your user research has a small set of questions that are very specific, online survey may be a good choice for reaching as many people as possible or when shared via a platform such as work or school email or possibly an enthusiast forum, it is possible to reach a relevant subset to study.

- *System Usability Scale*
  - Conceived in 1986 by John Brooke, SUS allows the evaluation of a wide range of services or products. The questions are simple declarative statements (PICTURE 2). Questionnaire is scored from 0 to 100 points. SUS score is calculated by summing the score contributions from each item. Each item’s score contribution will range from 0 to 4. For items 1, 3, 5, 7, and 9, the score contribution is the scale position minus 1. For items 2, 4, 6, 8, and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall SUS score. According to Rex Hartson and Pardha Pyla in their book The UX Book (2nd edition, 2018) SUS has been an extensively used and widely adapted public domain questionnaire. It has been a very popular questionnaire for complementing objective UX data, given it can be applied at any stage in the UX lifecycle as well as by nature it is intended for practical use in an industry context. The SUS is independent of the technology used and fast to implement as well as to analyse. It is the most recommended of all the publicly available questionnaires.



PICTURE 2. System Usability Scale has 5 stages for opinions on something.

3.1.1 Who we assume our user is

The end-user for our machine and its interface is picked by the company that buys the machine itself. Therefore it’s hard to make infallible decisions for who our user is in actuality. We can however assume that people who have a higher level of education than company leaders won’t be employed to use the machines (S. Mauno et al. 2001.) so we can assume we’re speaking of people of relatively low levels of education and who can be easily attained and replaced. We therefore have a dilemma when designing the product (PICTURE 3). We need to be able to include enough features to boost sales, as features, no matter how useless, will be seen as a positive at the time of purchase, even if simplicity would lead to a better user experience. (Norman, 2007.)

PICTURE 3. A screenshot of the video I made to demonstrate features of RodFINE. The video follows the company design guidelines, which leaves very few ways of invoking an emotional response in customers. Therefore the only way of affecting the purchase is showing features, as smaller company brands do not sell on their own.

3.1.2 The first tests

As Rodstein is a rather small company with the design department being in another city during the time the initial testing took place and some people still were on a vacation, the building had a total of five people present at the time of the first user research sessions. The scope of initial testing couldn't be based on focus groups or surveys with that few people, but I was able to get a couple of contextual interviews from the people who had used the interface of the machine that was going to be sold before RodFINE, as the machine going by the name of RodFLAT and its interface were still on the shop floor. RodFLAT is a lot bigger machine, that cuts and punches holes in flatbars and is meant for an automated production line, unlike the smaller manually driven machine we were creating. This means it doesn't need much operation but must have a wider variety of settings than RodFINE, which has no modularity. We couldn't get anything out of RodFLAT for physicality, since the amount of interaction is very low and done only via a digital display, but the people on the shop floor who had worked on such systems for years could tell what they absolutely hated about it. Everything that was unwanted, we tried to avoid implementing from the get-go.

Afterwards we did some prototyping iterations for what buttons did we need and what kinds of menus did we want. Our initial plans were mainly guided by bias and what we had seen in interfaces, like having a home button, but after some testing, it was apparent that a home button was a totally worthless one and only introduced more tabs.

For testing purposes, we always tried to see our user persona as the most simplistic one possible, given how markets drive cheap labour to operate machines that aren't already fully automated and the cheapest labour is the one with the least education and exposure to design principles to make any educated guesses, so our initial persona was a sweatshop worker in Thailand, the country where the first machine was sold to.

3.2 Usability

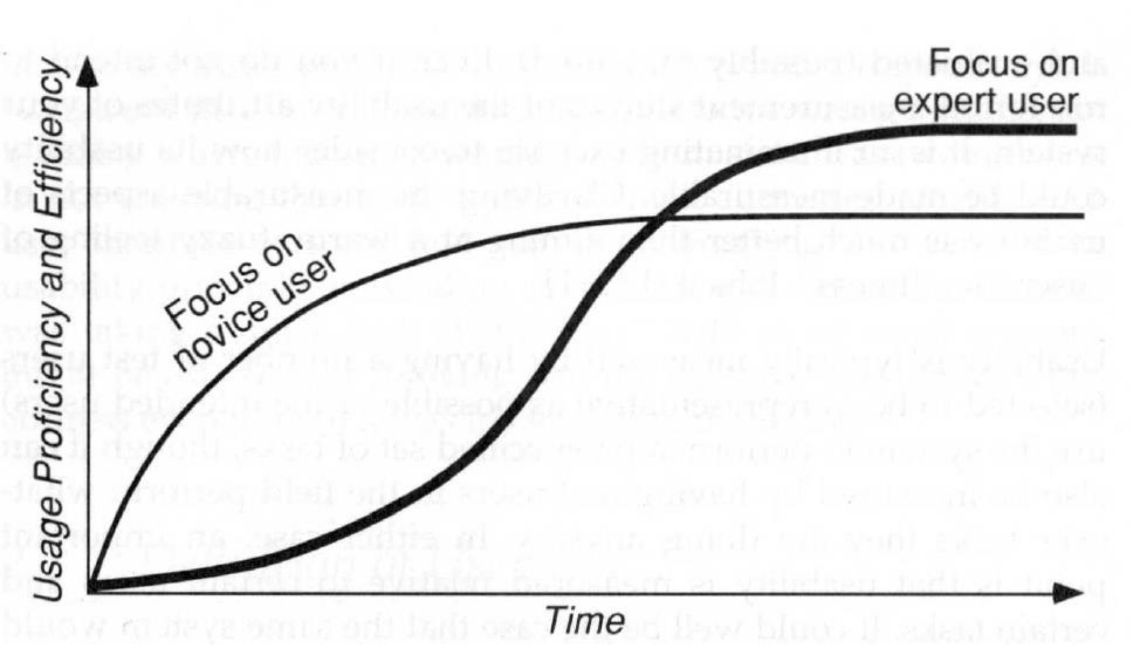
Usability is not a single one-dimensional property of a user interface. (Nielsen, 1993). Usability is traditionally divided into 5 separate groups which are as following:

- Learnability
  - The system should be learned at ease to reduce unproductive time
- Efficiency
  - The system ought to be efficient, so that once the user has learned the system, a high level of productivity is feasible
- Memorability
  - The system must be easy to remember, so that the user doesn’t need to in constant interplay with it to keep their understanding of it intact
- Errors
  - The system needs to have a low error rate, so that users make few errors during the use of it and in case they do mess up, recovery should be quick and painless. Furthermore, no catastrophic error should be possible.
- Satisfaction
  - The system should feel good to use. The user must not be dissatisfied while using the system.

Typically, usability is measured by having a group of test users use the tested system and perform predetermined sets of tasks but can also be measured by having real users in the field do whatever tasks they’d be doing anyway. What is of utmost importance is that usability is measured in relation to the user base. A program doesn’t necessarily have bad usability if the use case measures is not accurate for the intended use case. Bash terminal makes for an awful tool for writing an essay, but a good tool for establishing an SSH connection or pushing files to a git repository. A left-handed baseball mitt is an awkward object to around 90% of the world population, but an essentiality for the rest to enjoy the same game to its full extent. Overall usability is measured on the basis of the selected usability measures and their mean, but since users are known to be vastly different, it is better to consider the entire distribution of measures.

3.2.1 Learnability

Nielsen attests to learnability being the most fundamental usability attribute, as most systems must be easy to learn and the initial experience of your average user is that of learning to use the system. Essentially there is no system that doesn't start with the learning scenario and a botched learning gimps every other aspect of the system (FIGURE 1).



FIRUGE 1. Learning curves of a simple system which is easier for a novice to learn and an advanced feature-rich system that takes time and exploration to master. (Nielsen, 1993)

With our machine, it is unclear how much prior experience with operating other machines can be translated to a new system which has no prior versions in place. We can make an educated guess by assuming that systems made with the same tools are somewhat uniform in nature, as a production line can have machines from multiple manufacturers, meaning the way parts operate logically has to have enough semblance to each other. There's scarcely any documentation of standards and apparently a fair share of the practices are simply hereditary in companies and years of competition have driven them to emulate others, copying practices that they've seen to be good selling points for the opponents. In an exchange with a UX designer at Metso, she couldn't name any specific books for industrial machine interface design. They merely operated on trial and error as well as by what kinds of practices their brand had been bound to by their previous products. This is an example of a defined vendor company architecture, which is a way to enforce an interface consistency (Nielsen, 1987, 3) for each machine. The interface and user experience we were crafting for our company's machine didn't have any kind of industry-based rule set or a parent for consistency. Only one machine with a digital display had been produced before and it was vastly different. We can assume that operating RodFINE fell under the novice user learning curve, given how simplistic its job was, but one cannot say operating machines before helps using it. The user operates it with a familiarity to the general technology (Turner, 2017, 34-36).

3.2.2 Efficiency

Efficiency is defined by how stable the user's workflow or output is once the learning curve smooths out. Efficiency is only dictated by experienced users and people who haven't learned to use the systems do not reflect the actual state of the system. True efficiency may be hard to gauge, since most of user base will level off in expertise once they feel they've learned enough, even if the cost of learning a couple of advanced tricks would increase their efficiency with a hefty margin later (Rosson, 1984). To analyse efficiency, a group of experienced users is needed. Experience can be a rather arbitrary measure, given how it can be informally defined by people just saying they are experienced or by them using the system for a specified time. Formally it can be measured by the actual hours spent using the system, which is often the measure for new systems without an established user base. Finally it is possible to define the users as experienced by constantly measuring their efficiency, such as the time it takes to run a program or a macro and once it stops progressing to a higher level, they've reached the experienced status (Nielsen, 1993, 30).

3.2.3 Memorability

The third category other than novice and expert users is the casual user. In case you've ever programmed a web page or a script for some reason a while back and have returned to it tad later, you may be completely clueless as for what it does or how it even works. This is due to programming having virtually no other ways of understanding something but memorization of the syntax. Low levels of visual cues and descriptions makes recalling harder.

An interface that is easy to remember is important for users that don't actively use the system. They may have been or a break or their job has entailed something else for a while. Maybe your system is something that does not get used often whatsoever, such as emergency systems. A good learnability is analogous to good memorability, but good memorability doesn't always stand for good learnability (Nielsen 1993, 31). Idiomatic or witty descriptions and names can be hard to learn, since they don't explicitly describe something and need thinking, but once understood, they are much like a pun or a limerick stick to the user. Nielsen points out a case of a kiss and ride sign, which makes no sense without outside assistance, but is just a play on park and ride, a zone for dropping people off on a station, with an implication of a story when a spouse or a lover drives their companion to the station.

Memorability is scarcely tested in depth, but there are two ways principal of testing it regardless. The first one is testing how long does it take for a user to execute some task after being away from the system for some time. The other one, which may be less accurate, but takes less time is a memory test with the user after they're done with using the system and tell them to describe the effect of some commands or to name one, maybe even draw its icon if there is one. The amount of right answers is the score for the interface's memorability. The first test is preferred, as it reflects the contemporary graphical interfaces the best, as people may be unable to recall the content of a menu, but still have no issues using the menu once they had the visual help (Mayes et al. 1988).

3.2.4 Errors

While it is likely that a user will make errors before they can operate any system properly, they need to be safe from doing something that either causes damage or puts them in a state that they won't be able to recover from. Errors are measured by assigning a task to the user and every step they make that doesn't serve the task is counted as an error.
Just defining errors as any wrong action won't be enough, as errors may have highly varying outcomes. Small errors like pushing the wrong digital button to go to the wrong menu and then pressing the back button should be separate from larger errors, like one that destroy the user's work or ones that the user does not realise they are doing from the get-go (Nielsen, 1993, 33).

While testing RodFINE, sometimes the test user would press the wrong digital button, as they are labelled in a way that may make it hard to discern them from each other. (PICTURE 4).
A catastrophic error was also reached by first manually changing the servo positions (PICTURE 7) via the settings menu's tool diameter setup (PICTURE 14). This caused the spindle to cut too low and damage the corner of the white trough for metal chips that lays below the spindle, and against the lower locking block (PICTURE 9). The trough has no moving parts or any mechanism other than being a bin for the trash, but such move risked breaking the spindle or the servo, which have long delivery times and costs, where as steel can be welded and painted on the workshop floor in a day. The settings were behind an admin password from the start, so the worker probably would've not been able to cause any damage, but we needed to reprogram the servos to have a limited range of motion to prevent such from happening in cause of an admin error or maliciousness.

PICTURE 4. The manual jog menu labels are mysterious to someone who has never manually driven or “jogged” a servo.

3.2.5 Satisfaction

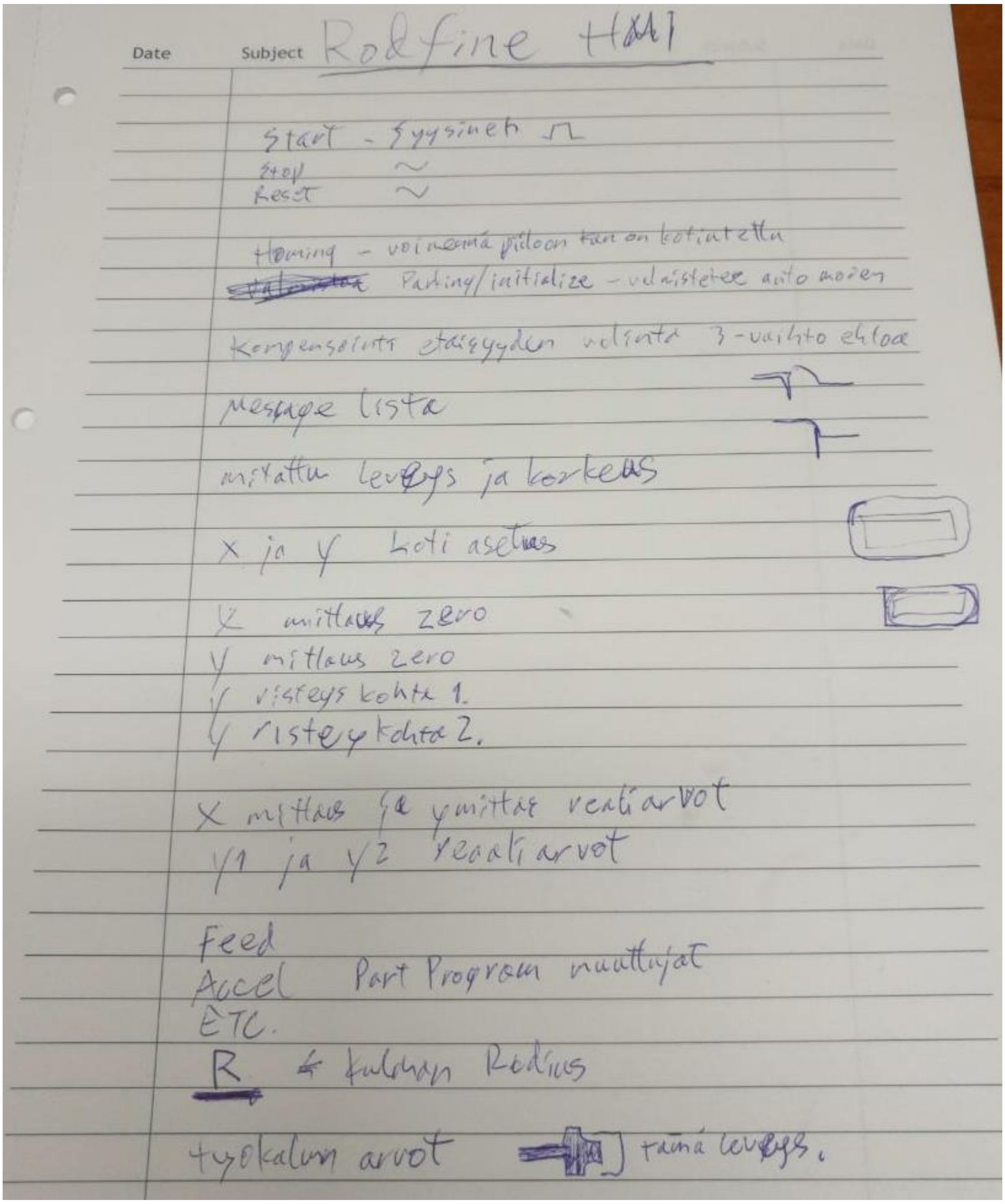
Satisfaction is plain and simple the experience the user is left with after they’re done with the task. Satisfaction is an extremely important aspect in consumer user experience, as entertainment with a bad satisfaction experience is a failing its core purpose. Nielsen argues that one should still be wary of what is the system’s fault and what is the users general comfort level with interacting with digital systems. Much of the expectations are based on how the users have previously used systems akin to the one being tested or if at all.

4 THE PROCESS

4.1 The first drafts

Starting from zero, we need to make a decision for what actions we're going to put into the interface and what are we going to install as manual switches and buttons, independent of the UI. By default, the start button ought to be a physical switch, which can be flicked on and off whenever by turning, as it allows the user to immediately shut down the whole machine's electricity if an emergency arises. According to Hartson & Pardna (2018, 668) it is imperative that a knob looks graspable or turnable. As an example, they offer a card radio, where the tuning is done by up and down buttons instead of a rotary knob. This sense of pleasure derived from action of physical objects is called physicality in UX. An industry machine is often very much pertaining to physicality as they need to be robust and able to handle years of labor and rough use, so while making physical switches for major power operations is sensible to avoid using software that could pose a security risk when malfunctioning, the human brain has a tendency to prefer physicality. As for the program in the machine itself, there is no necessity for a separate digital start when a it only runs a single program as it boots.

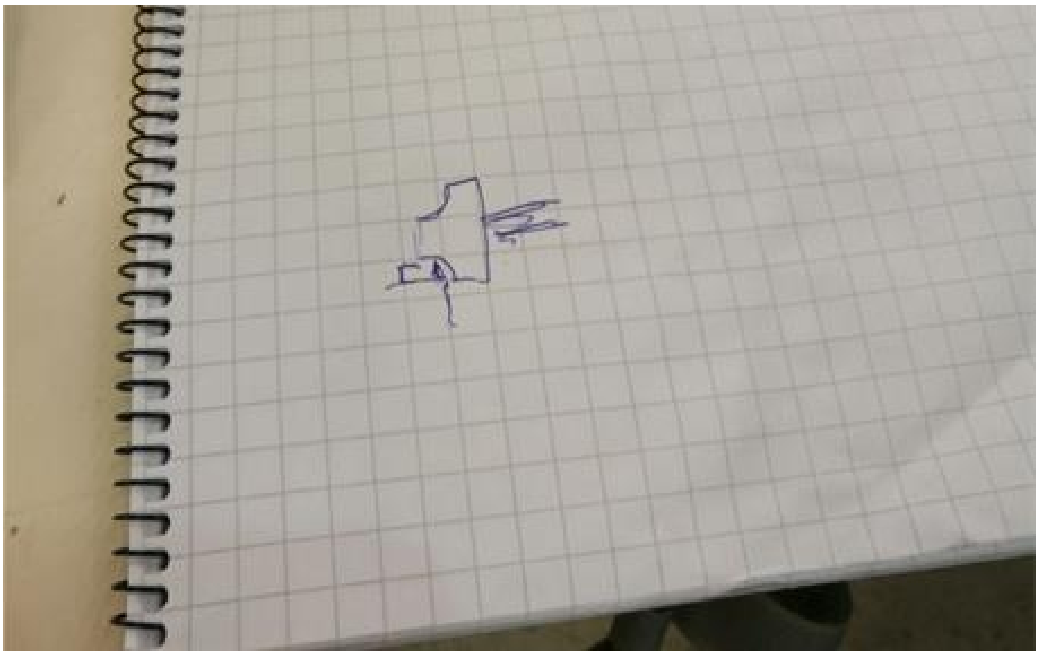
Secondly emergency stop needs to be a simple punchable switch that is easy to operate even when panicking. Operating switch for starting the servo drive was also made into physical switch. Thus, operating the machine needed no usage from the touchscreen in case the user was going to cut multiple identical pieces, which in a factory setting is the common norm, given how inefficient it would be to change settings with every run drive.



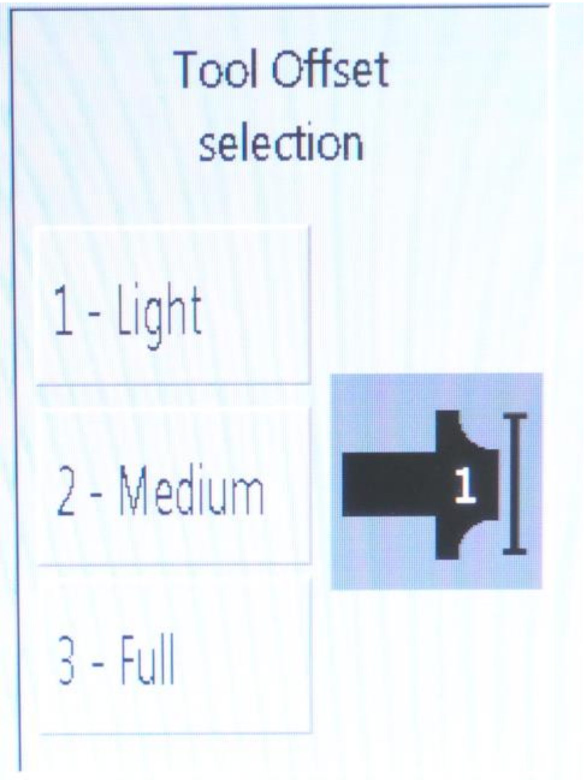
PICTURE 5. The very first scribbling of the interface plan

4.1.1 Designing buttons and icons

According to Nielsen (1993, 38-39), it has been demonstrated that people prefer icons that have both the icon and a descriptive text for what it is underneath it. We needed to create an icon to portray the spindle of the machine.

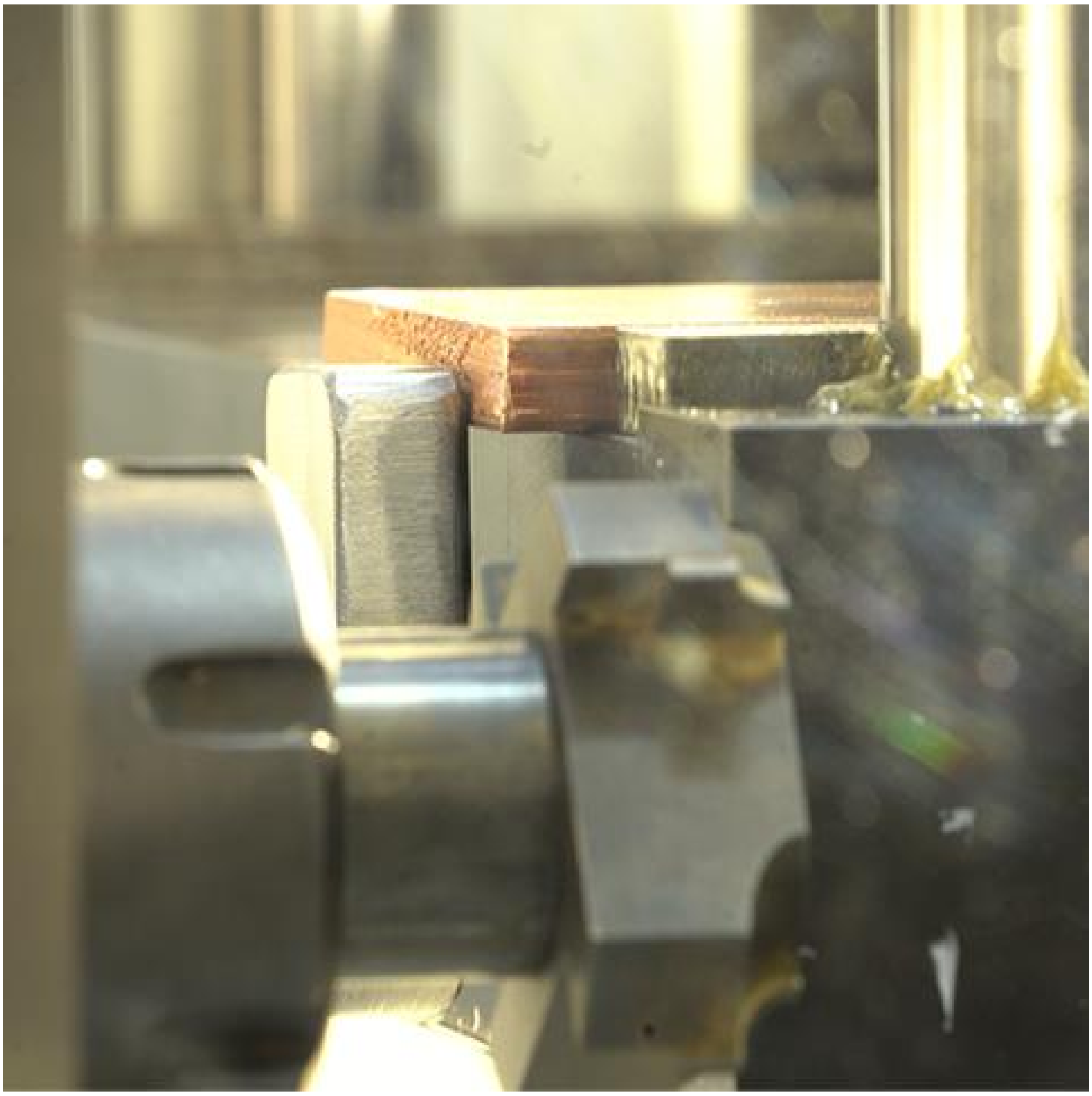


PICTURE 6. An ideation for a corner cutter adjustment

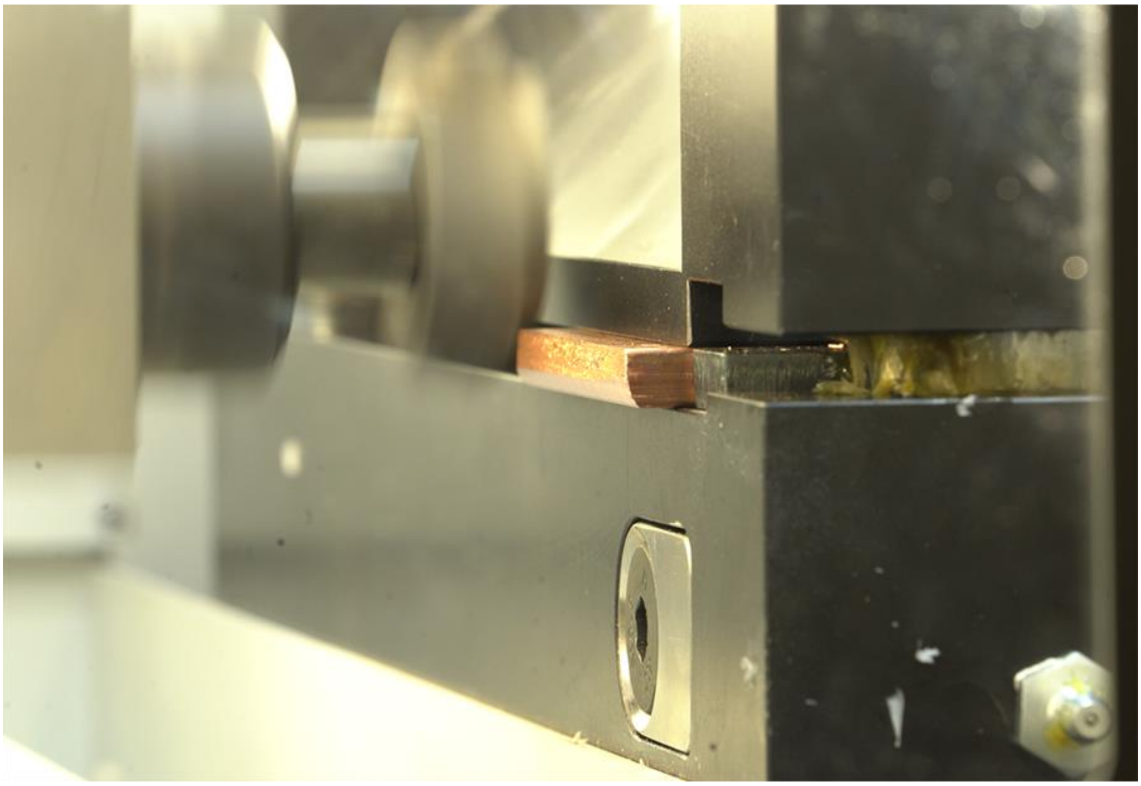


PICTURE 7. Final icon used in the machine interface. The number changes when a box-shaped button on the left side is pressed.

The machine has a custom spindle that carves the metal parts inserted, but given its custom nature, it’s not easy to communicate its behaviour to somebody who does not know what the insides of the machine are like. The innards of the machine are somewhat visible from the side, but in case of bad lighting, might be hard to discern. Thus, it’s plausible, that the user can’t just rely on the interface to make sense of the machine, and the user experience of it might even be circumstantially lacking. Manufacturing companies combat issues arising from clashes like this by an introductory visit by specialists, who show how to operate the machine. Optimally a good design would explain itself, but it’s standard company procedure for someone to go with the sold machine when it is installed on site.



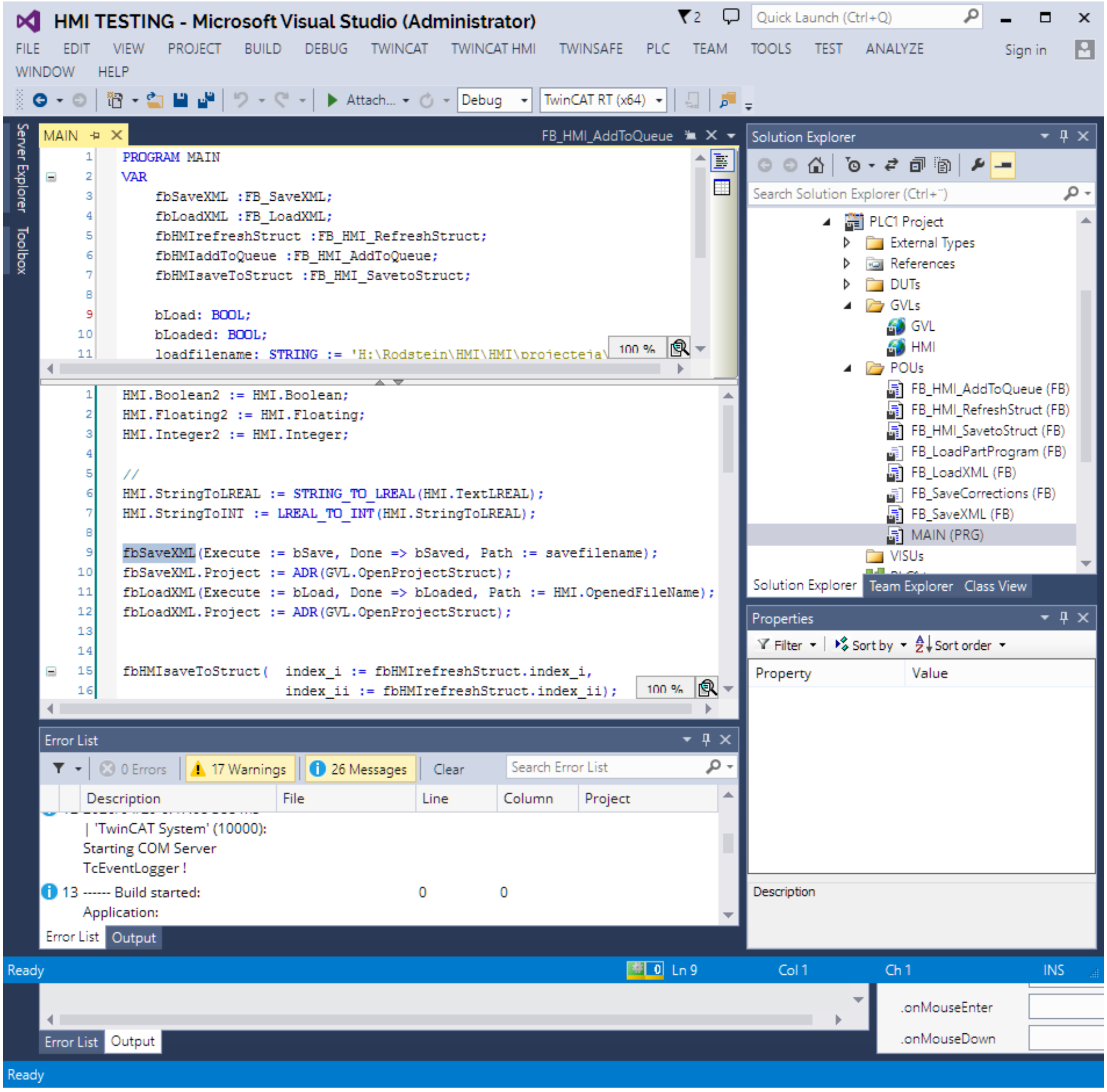
PICTURE 8. The spindle portrayed at the front of the picture. A copper block is being blocked from going too deep inside by a lock-lever, a moment before steel press lands on the block to hold it in place, starting the spindle and moving the lever aside.



PICTURE 9. Spindle in action, rounding the busbar edges

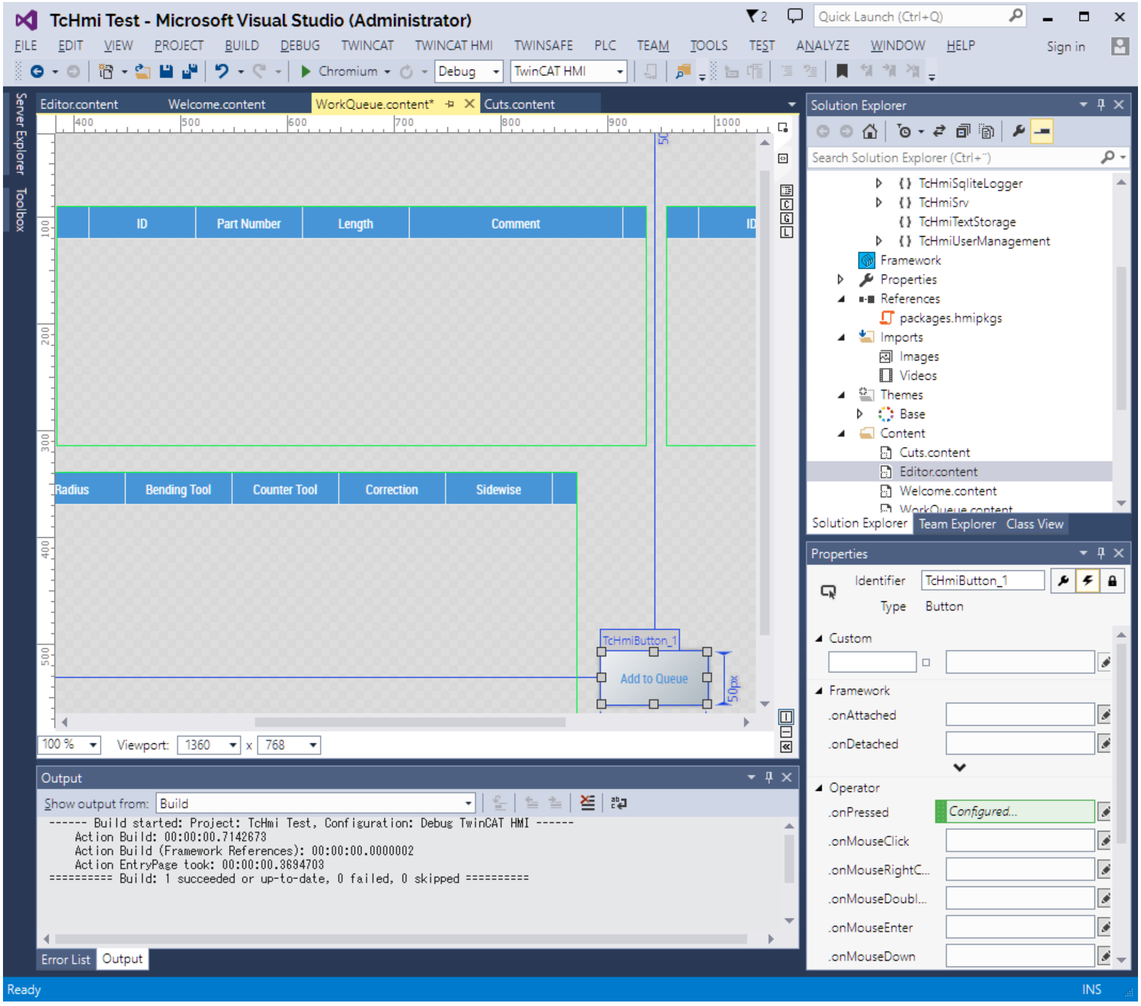
4.1.2 Working on the HMI

As every machine needs an interface, one must be first made before actual user experience testing can happen. For this, our software engineer and I constructed an interface via Beckhoff's TwinCAT PCL software (PICTURES 10 & 11), which is embedded to Visual Studio. PCL programming is much like web programming, with a separate front-end and back-end, but the languages used for the majority for the back-end are Structured Text (ST), a high level language that has a resemblance of the Pascal language and Instruction list (IL), a low level language resembles assembly language, but as of now, the latter is deprecated. Modern PLC can be programmed in BASIC and C derivatives as well.



PICTURE 10. Back-end programming with PLC. The language is Structured Text (ST or STX), which is based on the Pascal language.

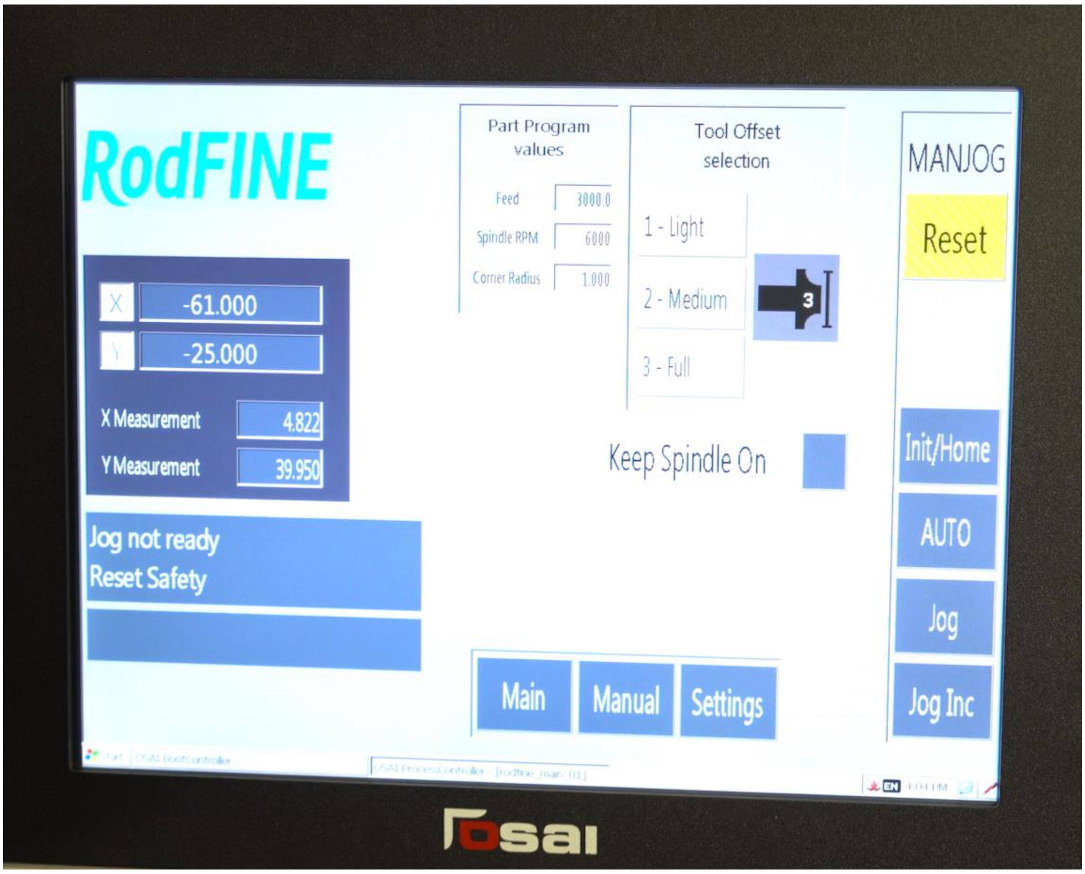
Front-end has its own set of different languages, but contemporary practices utilize the same things as web development. Our interface consists of JSON and HTML5 as well as CSS. Input files that featured different values for display given by the PLC side are in XML. Solutions are mostly built via graphical interface and just adding new items with mediocre programming skills is enough to get rudimentary HMI going. Making HMI is more of a reminiscent of some game editors with modular build-up, a game creation software, as you make do with check boxes and using variables you've hand-coded with the program's datagrids and item property tabs.



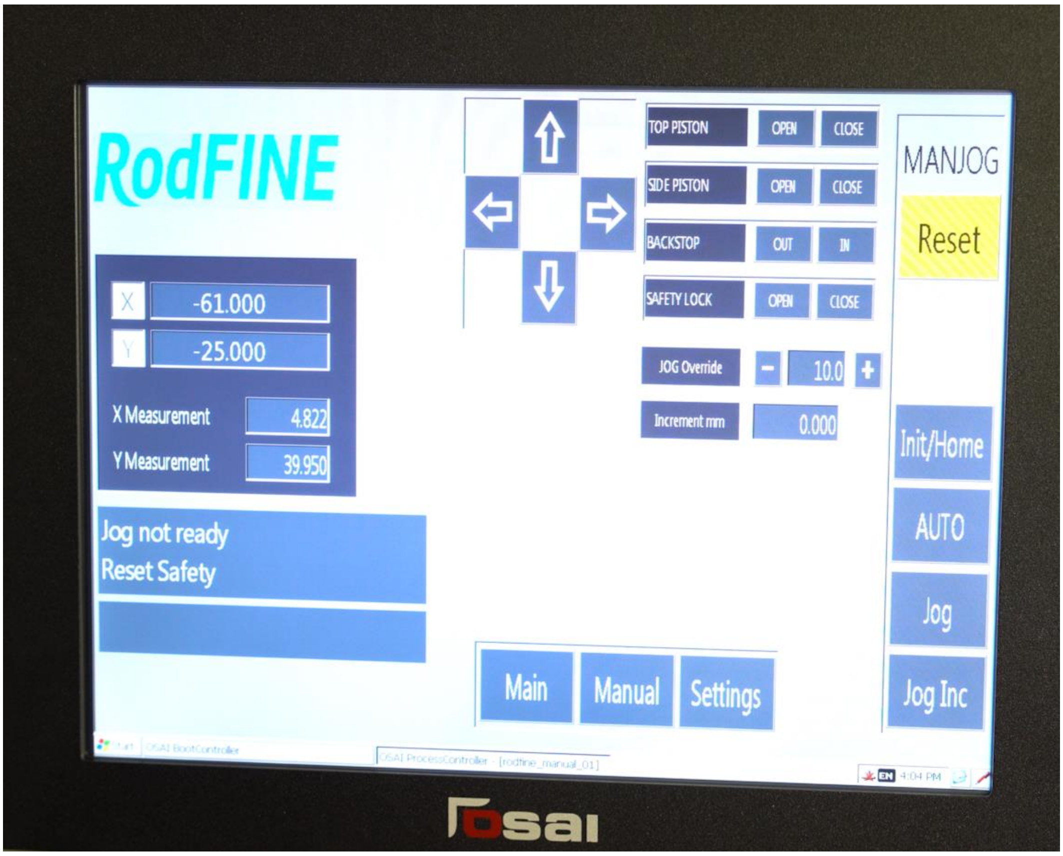
PICTURE 11. HMI front-end development

4.1.3 The final look of the UI

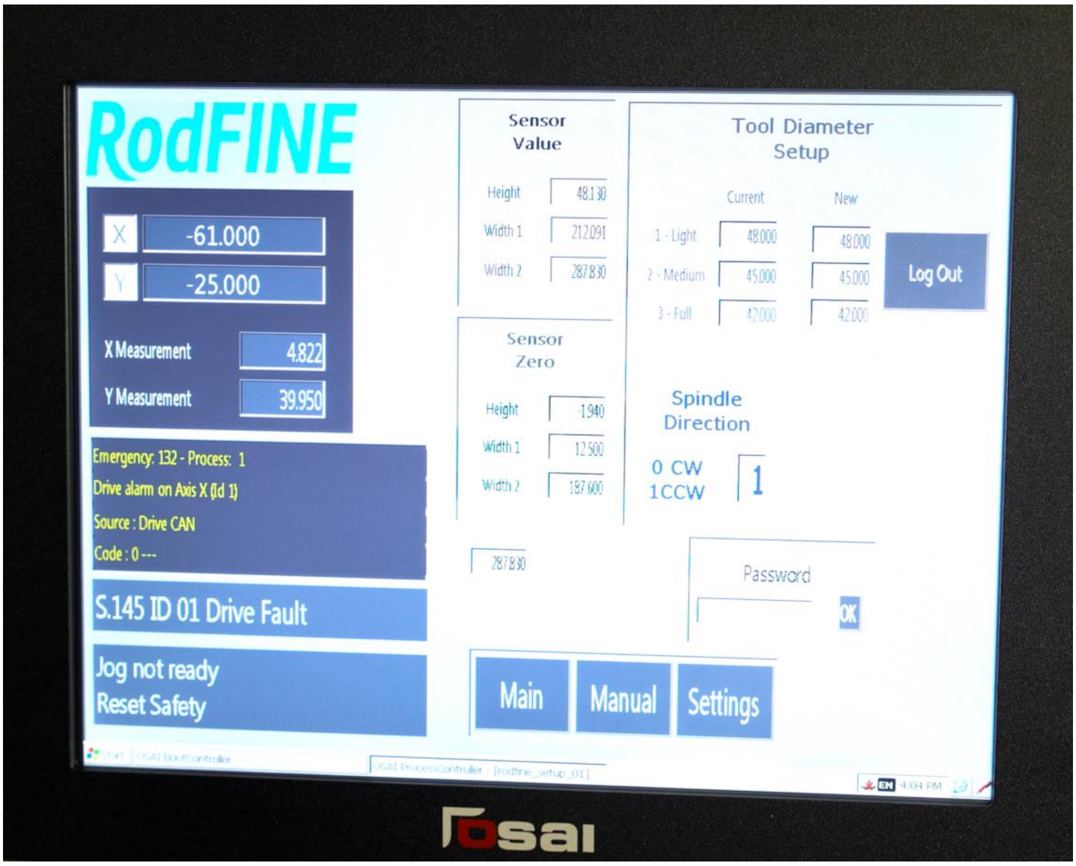
The interface has 3 different screens, of which 2 are actively used. The initial screen (PICTURE 12) is used to dictate how much rounding the user wants to get for a busbar. The second screen (PICTURE 13) is the manual drive screen, which the user may lock and unlock the safety, possibly to change the spindle or to clean or lubricate some parts. The third screen being the setting screen (PICTURE 15) mainly has measurement data for the user, but given a password, they can adjust tool preset values to what they need for the automatic drive.



PICTURE 12. RodFINE Main screen



PICTURE 13. RodFINE Manual Drive screen



PICTURE 14. RodFINE Settings screen

4.2 Building

Each machine has its own frame, which is manufactured separately from the rest of the parts. The frame is first modelled by designers, but a lot of the operator parts aren't forced to any specific location by default, in this case, security switches and touchscreens, which were put where we felt it caused the least issues, either for transporting the machine or just for operating it without awkwardness. Hartson & Pardha (2018) claim that a badly placed screen will cause fatigue and jeopardize the workflow. In Rodfine, the screen's position was limited by the small size of the machine and its sizable top hood. For an average Finnish male, the screen is at a good level, but it is unclear, whether it's in an awkward position in some countries it was sold to, where the average stature of a person is shorter. (PICTURE 15).

PICTURE 15. RodFINE user interface being used below the shoulder level.

4.3 Final testing

Before the first machine left the shop floor, a short guide for operation had to be constructed. Each person had to do the basic run and while it was extremely easy, the error messages could be taunting. The Reset Safety message for example couldn't be dealt with the Manual tabs Safety Lock button. This means that there is a separate state that was independent of the actual state of the safety lock, but was invoked by the safety lock being opened, not removed by it being closed. The solution was to just hit the Reset button, it did cause some confusion. Some popup menu elements in the Settings menu were also scalable by a finger pinch like zooming on phone, which was an unintended property for them and thus a bug, although likely not discovered by anyone who isn't trying to find it. The error messages could've been clearer and more consistent.

5 CONCLUSION

When designing interface for industry machines, one must first have some kind of an understanding of the market. While good UX is good for everyone, niche UX can be abysmal for some and good for others. The biggest hurdles seemed to be, how can we produce modern UX that we are so used to with other machines and interfaces into something so vehemently established in practices as machining. A fair share of the business is B2B marketing, so the image of the product is often made with the social skills of the marketing department. For Christmas season I had to craft a video portraying how the machine works. With companies such as these, the presence of engineers and their culture is omnipresent, which made it fairly difficult to be creative with it. During the development of the interface as well as the filming, it was fairly clear that the material that was marketed was very physicality focused, as were all the tools we worked with. Many aspects of the development tools were very rudimentary, and they had a 90s software vibe to them. The whole image of engineering that dictates acceptable design is pretty inflexible and as such the presentation can suffer from it, even if sales don’t happen just by showing technology. This may gimp the UX, since modern interfaces need development tools that allow constructing aesthetic compositions, which is not going to happen when the tools are streamlined for binary functionality. The rigidity of branding also leads to some very campy looking elements. Essentially the colour palette is limited to shades of blue, red, black, white and different metals with a tint, which when utilized in how we’d like to for example see a mobile interface or a tablet look, looks gaudy.

On a final note I personally am not completely satisfied with the ambiguous titling of some buttons, but there is no definite way of knowing how the system portrays itself to someone who has handled many systems before RodFINE. For the most part the machine doesn’t need any interaction with the screen and only works lifts the finger cover and presses the green button for the machine to do its job. Thus, the physical UX is fairly pleasant, although there could’ve been another way to design the said finger cover, to reduce the range of motion required to operate the machine.

REFERENCES

Coleman, C. V. (2018) Visual Experiences A Concise Guide to Digital Interface Design. Boca Raton: Taylor & Francis Group

Hartson, R., Pardha, P. (2019). The UX Book. Cambridge: Morgan Kaufmann

Turner, P. (2017). A Psychology of User Experience
Involvement, Affect and Aesthetics. Edinburgh: Springer

Nielsen, J. (1987). Coordinating User Interfaces for Consistency. San Diego: Morgan Kaufmann

Nielsen, J. (1993). Usability Engineering. Mountain View: Morgan Kaufmann

Mayes, J. T, Draper, S. w., McGregor, A. M., and Oatley, K. (1988). Information flow in a user interface: The effect of experience and context on the recall of MacWrite screens. Proceedings of the Fourth Conference of the British Computer Society on People and computers IV. New York: Cambridge University Press

Mauno, S., Leskinen, E., & Kinnunen, U. (2001). Multi-wave, multi-variable models of job insecurity: applying different scales in studying the stability of job insecurity. Journal of Organizational Behavior, 22(8), 919–937. Pittsburgh: John Wiley & Sons, Ltd.

Norman, D. A. (2007). Simplicity is highly overrated. Interactions, 14(2), 40. New York: Association for Computing Machinery

Rosson, M. B. (1984). Effects of Experience on Learning, Using, and Evaluating a Text Editor. Human Factors: The Journal of the Human Factors and Ergonomics Society, 26(4), 463–475. New Jersey: Sage Publishing