COGNITIVE ERGONOMICS OF OPERATIONAL TOOLS

A. Luedeke, PSI, Villigen, Switzerland

Abstract

Control systems have become increasingly more powerful over the past decades. The availability of high data throughput and sophisticated graphical interactions has opened a variety of new possibilities. But has this helped to provide intuitive, easy to use applications to simplify the operation of modern large scale accelerator facilities? We will discuss what makes an application useful to operation and what is necessary to make a tool easy to use. We will show that even the implementation of a small number of simple application design rules can help to create ergonomic operational tools. The author is convinced that such tools do indeed help to achieve higher beam availability and better beam performance at all accelerator facilities.

INTRODUCTION

A "tool" is something that helps you to achieve a goal. It is called an "ergonomic tool" if it has been tailored to be used by humans, taking into account our physical and cognitive limitations. These days the operator in an accelerator control room pursues his goals exclusively by means of computer applications. The discipline that deals with the design of ergonomic computer applications is human-computer interaction (HCI). For the software industry this field has continuously gained in importance over the last few decades. The branch of human-computer interaction that deals specifically with the analysis of cognitive processes is known as cognitive ergonomics. It deals, for example with diagnosis, decision making and planning: all required of operators in modern industries. At the same time human-computer interaction has been widely ignored in the development of modern control systems for accelerators. The intention of this article is to advertise the basic methods of cognitive ergonomics to develop ergonomic operational tools for the control of accelerators.

MOTIVATION

One of the most important figures of merit for accelerator operation managers is beam availability. Figure 1 shows the contributions of the different systems to beam interruption time for the Swiss Light Source (SLS) in 2009. There appears to be no contribution from "bad operational tools", but "operator faults" do account for 3% of the downtime. The operational applications at the SLS are the only means of the operator to interact with the accelerator; therefore an operator fault is most likely an unintended action of the operator: he did not properly predict the consequence of his action. Or we could say that the application failed to communicate the risks of the action to the operator in a timely manner.

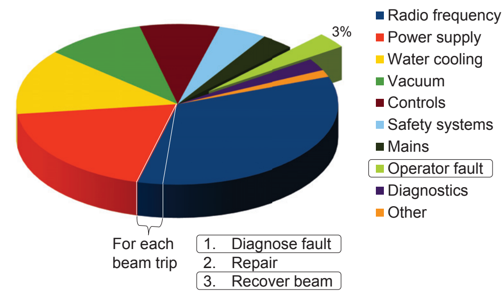

Figure 1: Beam failure statistics of the Swiss Light Source for 2009. Operational tools do affect the number of operator faults and the time to diagnose faults and recover beam.

The design of ergonomic operational tool does not just help to prevent operator faults. Every beam interruption is handled by the operator in three phases: first, he has to diagnose the fault; then he takes care that anything preventing him from making beam gets repaired and finally the operator has to recover the beam. The actual repair is often not in the responsibility of the operator, only the delegation of the repair. But as well for the proper diagnosis before as for the beam recovery after the repair he is in charge. He'll depend on his operational tools to solve these tasks quickly and efficient. If the actual repair time is on average in the same order as the time required for failure diagnosis and beam recovery, then the quality of the operational tools for these two tasks can have a significant impact on the operation statistics.

THEORY

Human-Computer Interaction

Three aspects needs to be understood for the design of good user interfaces. They are shown in Fig. 2: the goals to be achieved (the tasks), the user that has these goals (the operator) and the technology to build the user interface (the control system).

A thorough understanding of the user of operational tools, the operator, is vital to create ergonomic tools for the accelerator operation. Unfortunately many developers of operational tools are not aware that this knowledge is important. Therefore this should be the main focus of this contribution.

In some cases the operation of an accelerator facility suffers from an inappropriate specification of the tasks of the operators. The operational tools are often developed for the

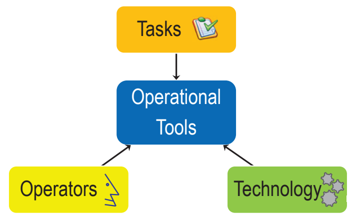

Figure 2: Tasks, operator and control system technology need to be understood for the design of operational tools.

commissioning of the facility, and therefore are optimised for the flexibility required for this phase. The tasks of the operator for the operation of an accelerator facility can differ significantly from those required for the commissioning. Problems of this type are often solved by developing new applications specifically for operation later on. A formal task analysis can help to prioritise on the development of operational tools. The task analysis section will introduce into this technique.

The technology to develop the operational tools compromises all aspects of the user interface of the control system: from input devices like a mouse, over GUI development software to the means of visualisation. This is generally among the core competence of the application developers and rarely a problem. Therefore we will omit it here.

Interaction Design

Figure 3 shows a simple model of the interaction of the operator with his tools. The operator will act on his tools by

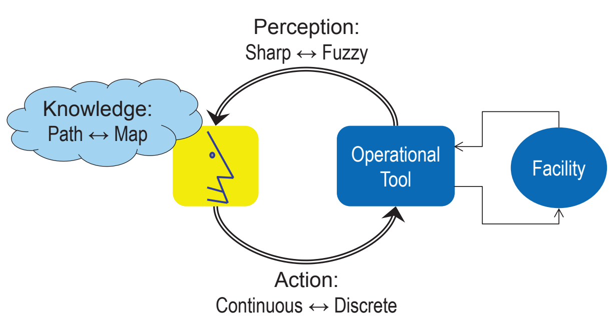

Figure 3: Simple interaction model of the operator, his operational tools and the accelerator facility.

the means of discrete control, like buttons, and continuous handles, like sliders. His actions are driven by his goal and his knowledge on how to achieve it. This knowledge can be like a path, where the operator follows step-by-step a defined procedure, or rather like a map, if the operator has a deeper understanding of his task, making him autonomous to find the way to his goal based on his knowledge. The information presented by his tool will be perceived by him either as a sharp picture, showing exactly the information he needs or rather fuzzy with room for interpretation.

A good tool should support the operator in all aspects: it should present a selection of actions appropriate to the task and easy to use by the operator, it should present all information relevant to the task in a way appropriate for the perception by the operator and it should provide all necessary knowledge required to perform the task such that they are comprehensible by the operator.

Human-computer interaction research used the theory of perception to derive guidelines for the design of effective and efficient user interfaces. The cognitive limitations of humans defining how user interfaces needs to be designed to support the user in pursuing his goals.

USER INTERFACE DESIGN GUIDELINES

While guidelines for user interfaces are formulated differently in different reference publications for user interface design, they have a great overlap in their concepts [1]. We will briefly introduce these concepts in the following, as shown in Fig. 4. Simple strategies will be provided to apply them to the design of operational tools.

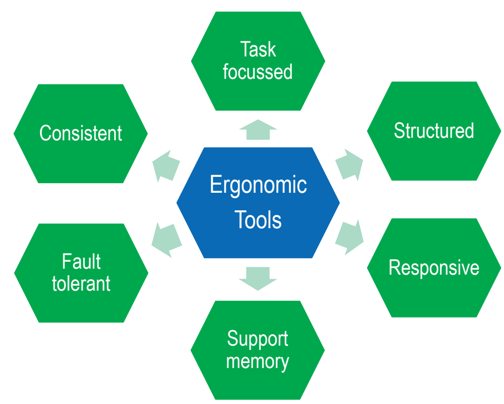

Figure 4: Design concepts for ergonomic user interfaces.

Consistency and Structure

It is obvious that consistency of the user interface has many advantages for the user. Exceptions are difficult to remember and variations in the style and layout can easily lead to confusion. Consistency is often reached without special measures, when the operational tools are created by a small number of people working closely together. If a larger number of people are involved, then a style guide helps to implement consistency in the design of the operational tools (See Fig.5). A style guide documents a desired common style for all applications designed in a given context, like all operational tools of an accelerator facility. A variety of style aspects can be documented: the general layout of screens, the types and styles of handles and buttons to be used, the standards for the display of data, the general usage and meaning of colour and symbols or the vocabulary to be used.

The benefit of standardisation derives from the limitations of our perception. People are unlikely to mix up a picture of a close friend with the picture of a similar looking stranger, but we do confuse sometimes buttons labelled "edit" and "exit", in particular if they are at the same location in two similar looking screens. Colour can be spotted very easily, for example a red word in a black on white text, but we have difficulties to distinguish different shades of red. We can spot relevant information quickly if it is presented in a clear, visual structure, but to find a specific word buried in a long text is very hard. The reasons for this are simple: we are good at things that helped our ancestors over the past hundred thousand years to survive. Reading text was no such thing within that time scale. And a red apple had a different shade of red anyway, depending on the light. This needs to be taken into account in the style guide: have a standardised layout and few defined handles and formats to provide structure [2]. Define few colours with specific meanings and select colours that are easy to distinguish [1]. Have a glossary of button and text labels following the simple rule: *"same name, same thing, different name, different thing"* [3]. In addition the vocabulary should be simple and unambiguous to the operators. Reading requires us to concentrate on the text, that will hinder us to concentrate on the task [1]. Use structure, colour and symbols to minimise the amount of text the operator has to read while performing his task. Many symbols have a known meaning, like a red triangle with an exclamation mark. Using them, e.g. for error messages, will catch the eye of the operator. Be careful to use "eye-catcher", like blinking text or pop-up windows: they do get the attention of the operator, but they can distract him from the task [1].

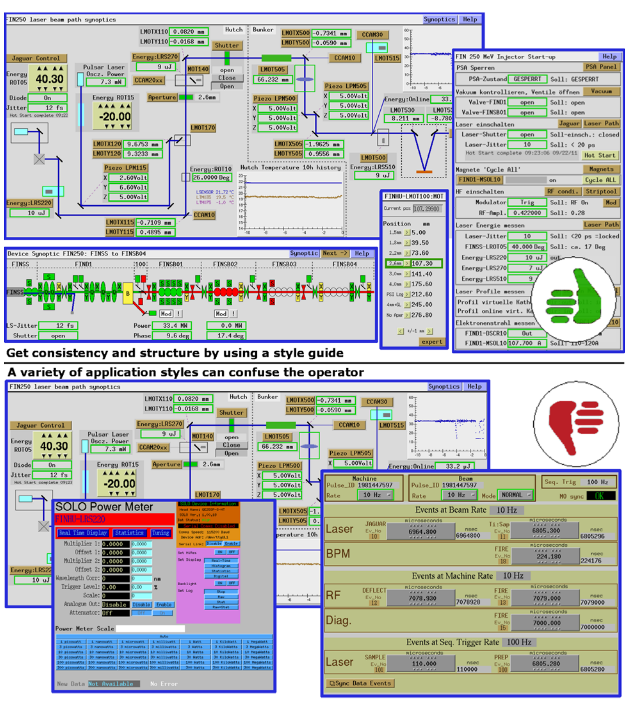

Figure 5: Use a style guide to make all operational tools similar to use. This increases the confidence of the operator in his tools and reduces the number of operator faults.

A style guide should best be ready before you start developing applications. Then it is not much extra work to adhere to it. But even after all applications have been developed, it is still a good idea to create and implement a style guide. Because many changes, like button labels or colours are very easy to change and the effort will pay off compared with the long term benefit for operations.

Task focused

A good tool should help you to achieve a goal. All your concentration should be on the task to accomplish the objective. People get easily distracted and an operator has many possible distractions: alarms to handle, telephone calls to receive, staff coming into or leaving from the control room. Therefore the tool should help the operator to stay focused on the task despite the distractions. There are several methods to help the user to focus on the task. In multi-step procedures it is useful to show which steps have been done and which still have to be done. The application should appear focused: display only actions relevant for the task and show all information useful for the task but no unrelated information.

The application itself should never distract the user from the task. Therefore it should use the task specific vocabulary, not the terminology of the application programmer. A message "division by zero, abort" will likely confuse the operator, while "the selected scan range of the measurement is zero, please select a valid scan range and retry" tells him clearly what went wrong and what he needs to do to continue his task.

Fault Tolerant

Even the best tool cannot fully prevent the operator from making errors. But ergonomic operational tools should make it hard for the operator to create significant failures and they should make it easy for him to recover from his errors. That way the operator will not just do less errors significant to operation. He'll as well feel more secure in using his tools. And his learning curve, while he's trying out new tools, will be much steeper, if he feels secure to try things out.

The confidence in his tools will even have an impact on the performance of the machine. Lets consider an SLS operator who tried to optimise the pre-accelerator with no success. He then decides to restore a saved machine setting from last week. This turns out to produce zero transmission, because - he then remembers - something had been changed yesterday on the pre-accelerator. He then learns that nobody saved the new setting. He'll be busy for an hour to restore transmission. Everyone will tell him afterwards that he should have followed the proper procedure by saving the machine settings before making many changes. This kind of stress is likely not helping him to learn the proper procedure, it is rather probable to make the operator reluctant to optimise the pre-accelerator again. It happened at the SLS, too. But we introduced an "Undo" button at the application to restore a machine setting. This allows the operator to recover to the state before the machine settings were loaded from the file. When restoring the saved settings does not produce the desired result, he can easily get back to the previous setting. In the example he would have saved the new setting after the "Undo": this would have prevented a significant fault of the accelerator, would have reminded him of the proper procedure and would have increased his confidence in continuing the optimisation.

Responsive

Our brain has time requirements in order to perceive causality. If we click a button and nothing happens for ten seconds, then we will not be confident that things that happen afterwards are related to our actions. The following time requirements have been identified for human-computer interaction [1]:

- *0.1 sec*: immediate reaction to user input, e.g. a button appears "pressed" or a wheel-switch "turns".
- *1 sec*: status feedback if a process has not completed.
- *10 sec*: estimate time, if a process takes longer.
- *100 sec*: the typical time to make critical decisions.

A tool does not need to be fast in performing the task to meet those time requirements. It just needs to respond quickly to user input and if the task takes longer it needs to keep the user informed about the process. Human-computer interaction scientist found that responsiveness has a high impact on user satisfaction. More important for the control room is, that a responsive application clearly tells the operator what it is doing and if his attention is currently required. If it is not he can use his time for something else: fix problems to improve the performance, return phone calls to improve communication or have a cup of coffee to improve operator satisfaction.

Some examples to implement reactiveness in applications are shown in Fig. 6.

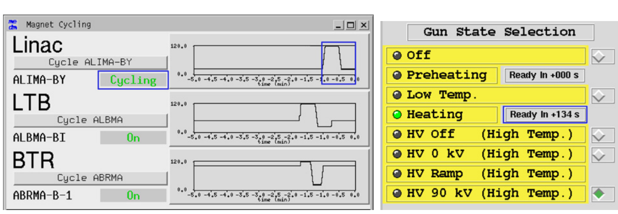

Figure 6: The magnet cycling application on the left shows the magnet current as visual feedback to the operator and the status "Cycling" while the process is still ongoing. The control panel of the electron source on the right shows a count down to forecast the end of the heating process.

Support Memory

The obvious approach to support the long term memory of an operator in performing a task is to have a thorough documentation of all the steps necessary to perform it, with a direct link from the tool to the document. Practice shows, that this works initially, most operators will read the documentation to learn the task. Some will prefer to learn the task from other operators. Once they learnt the task, only very few will read the documentation again. That should be taken into account when the documentation is updated: the operator needs to be informed about what parts have been updated and when.

There are additional methods to support the memory of the user. Each tool should clearly state it's purpose, e.g. by a descriptive title. It is often useful to provide links to related tools, to help the operator to find the tool he's looking for. Our memory works by association, therefore it will be much easier for the operator to select the proper action from a list of choices, than to recall the proper action from memory. If the number of choices is very large it is often useful to sort them by the expected frequency of usage: make those choices most visible that will be needed frequently.

A tool should provide the operator with all information required for the task. Do not expect the operator to collect the necessary information from other tools. This is particularly important during critical decision making: under stress the operator will base his decision on the available information, if the decision needs to be collected from several tools he'll may miss important information and make a bad decision.

If a tool is used to recover from a failure, then it should support documenting the failure. Do not expect the operator to type in the failure information into the electronic logbook after the failure! It is error prone and a waste of the operators time. Figure 7 shows how the magnet power supply control application at the SLS allows to save the information of the fault into the electronic logbook. This procedure is convenient for the operator but will at the same time assure that the fault is documented with complete and consistent information.

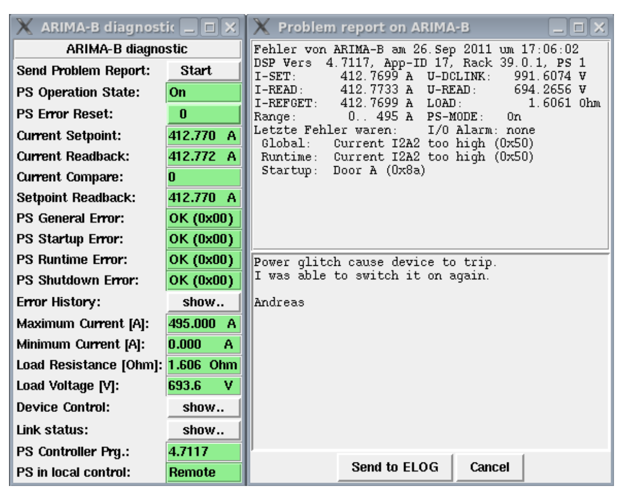

Figure 7: A problem report can be issued from the magnet power supply diagnostics application. It collects all information of the last failure and allows the operator to add a comment and save it the the electronic logbook.

TASK ANALYSIS

The method of task analysis has been developed in applied behaviour analysis and is not specific to human-computer interaction. It is the process of observing and analysing the steps of the performance of a specific task. There are different methods of task analysis and many books delve into this topic [1] [4]. We will just give a motivation here, how to use task analysis for the design of ergonomic operational tools.

Task analysis starts with observation: you watch the operators performing a task and record the steps they take to achieve their goal. The recorded steps are then organised, e.g. in the hierarchical task analysis as a tree of tasks and sub-tasks. One of the primary utilisation for task analysis is the creation of good documentation: you can use the recorded, organised steps to document the task. If you have documented the most frequent tasks of the operators, you can analyse the tools he is using for the task: Which tools are used frequently for the operator tasks? Are they appropriate for these tasks? What steps are difficult for he operator? Where are risks involved, like pressing the wrong button could cause beam interruption? Is the usage of these tools consistent or do they differ significantly?

Once you've answered these questions for the set of your operation tasks you probably know already which tools would need to be modified most urgently. But instead of starting to adapt your tools you should now start to develop a conceptual model of the operation of your accelerator. That is an idealised view on how operation should work: what tools would be useful for operation, what devices would they be used on, what are the relationships between the devices and the tools, how should they be used by the operators? Make the conceptual model as simple as possible and focused to the operator tasks, and you'll get a good starting point to prioritise your work on your operational tools.

USABILITY TESTING

Regardless on the amount of planning you've spend on the design of your operational tools: they will likely fail to meet some requirements of the users. The only way to find out is to test them on your users: the operators. Usability testing is a common method in application design [5].

A usability test will start, like a task analysis, by observing the operator using the operational tools. Here the focus is on how the operator is using the tool: Does he use it the way it has been intended? What time does he need to perform the intended task? At which point is he uncertain, where does he need guidance?

Industry builds specialised usability laboratories where hundreds of users are tested for a new product and even all eye movements of the users are recorded and analysed with special software: how long does the user look where at the screen, where does he moves the mouse pointer and when does he click where. But there is a very simple but powerful alternative for accelerator facilities with limited resources: the "think aloud" technique [2]. Tell the operator to think aloud while he's using the tool to perform the desired task. Sit behind, watch the operator and take notes. A variation of this technique is to have one operator explain to another what he's doing to perform the task. Since he is forced to explain what he is doing it will produce more information about what's going on in his mind while he's using the tool.

We use this technique frequently at the SLS. We call it "operator training": at least once a month we dedicate 90 minutes of beam time to the training of the operators. A special application called "sabotage" generates randomised system failures on demand. While one operator is responsible to fix the problem, several others are watching. His duty is to fix the problem and recover beam, to explain what he does or attempts to do and to document his actions. While the primary goal of the operator training is to exercise failure analysis and beam recovery with the operators, it serves at the same time as usability testing. Even if an experienced operator uses the tools quick and efficient, the questions from the observing operators often help to find possible improvements of the tools for novice operators.

SUMMARY

In order to design effective and efficient tools for the operation of an accelerator one needs a thorough understanding of the operational procedures. The well established user interface design guidelines of human-computer interaction theory can then be utilised to create operational tools that are intuitive and easy to use.

The optimisation of the operator interface should be an ongoing process. The techniques of task analysis can help to get a better understanding of your operational procedures. Usability testing is a proven method to improve the user interface of your operational tools. Both techniques can be implemented by regular operator training exercises on the real machine: by watching the operator actually performing his operational tasks.

The creation of ergonomic operational tools can help to achieve higher beam availability and better beam performance at all accelerator facilities.

REFERENCES

[1] J. Johnson, "Designing with the mind in mind: simple guide to understanding user interface design rules", Amsterdam, Elsevier/Morgan Kaufmann Publishers, 2010.

[2] B. Shneiderman and C. Plaisant, "Designing the user interface: strategies for effective human-computer interaction", 5th ed., Boston, Mass., Addison-Wesley, 2010.

[3] C. Jarrett, "Forms That Work: Designing Effective and Efficient User Dialogs.", Proceedings of the 50th Annual Conference of the Society of Technical Communication, Dallas, Texas, USA, 2003.

[4] J. Hackos and J. Redish, "User and Task Analysis for Interface Design", 1st ed., John Wiley & Sons, Inc., 1998.

[5] S. Krug, "Don't Make Me Think! : a common sense approach to web usability", 2nd ed., Berkeley : New Riders, 2006.