DESIGN PHILOSOPHY FOR ACCELERATOR CONTROL ROOMS

High-Level Design Recommendations for Fermi National Accelerator Laboratory to Support the Accelerator Controls Operations Research Network (ACORN) Project

November 2022

Rachael Hill, Zachary Spielman, and Dr Katya Le Blanc

ACORN-doc-700
FERMILAB-TM-2787-AD
## Revision Log

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Effective Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Initial Version</td>
<td>09/12/2022</td>
</tr>
<tr>
<td>0.2</td>
<td>Final Version</td>
<td>11/09/2022</td>
</tr>
</tbody>
</table>
Acknowledgements

The authors would like to express special gratitude to the following individuals for their collaboration and support of this effort:

The Fermi National Accelerator Laboratory ACORN team, namely Erik Gottschalk, Alyssa Miller, Lila Anderson, Beau Harrison, Tia Miceli, Eileen Crowley, Anthony Tiradani, Adam Watts, and Maria Acosta.

Additionally, the authors would like to recognize and thank the many accelerator operators that participated in the human factors interviews.
## Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACNET</td>
<td>Accelerator Control Network</td>
<td>Accelerator Control Network is the primary digital control system utilized at Fermilab.</td>
</tr>
<tr>
<td>ACORN</td>
<td>Accelerator Control Operations Research Network</td>
<td>The project that will modernize the accelerator control system and replace end-of-life power supplies to enable future operations of the Fermilab Accelerator Complex with megawatt particle beams.</td>
</tr>
<tr>
<td>BBM</td>
<td>Beam Budget Monitor</td>
<td>Accelerator machinery that measures beam intensity over time.</td>
</tr>
<tr>
<td>Fermilab</td>
<td>Fermi National Accelerator Laboratory</td>
<td>US Department of Energy particle physics and accelerator laboratory.</td>
</tr>
<tr>
<td>HSI</td>
<td>Human-System Interface</td>
<td>The digital interface through which an operator interacts with the accelerator control system.</td>
</tr>
<tr>
<td>IA</td>
<td>Information Architecture</td>
<td>The overall conceptual model used to plan, structure, and assemble system information.</td>
</tr>
<tr>
<td>IDA</td>
<td>Information and Decision-Aiding</td>
<td>Logic-based algorithms of automation that assist operators in decision making.</td>
</tr>
<tr>
<td>OVD</td>
<td>Overview Display</td>
<td>Overview displays are human system interfaces that provide the highest level of information on displays that are viewable by all crew members.</td>
</tr>
</tbody>
</table>
Table of Contents

1.0 Introduction ......................................................................................................................... 11

2.0 DESIGN PHILOSOPHY ........................................................................................................ 12
   2.1 Summary of Design Philosophy ................................................................................... 13

3.0 RECOMMENDATIONS ......................................................................................................... 13
   3.1 Summary of Recommendations .................................................................................. 13

4.0 STRUCTURE AND CONTENT OF ACCELERATOR CONTROL SYSTEM INTERFACES 16
   4.1 Information Architecture ............................................................................................. 16
      4.1.1 Structure User Interfaces Based on Information Hierarchy ................................. 18
   4.2 Overview Displays ....................................................................................................... 19
      4.2.1 Summary of Design Principles for OVDs.............................................................. 20
   4.3 Role-Based User Interfaces .......................................................................................... 20
      4.3.1 Summary of Design Principles for Role-Based User Interfaces ......................... 22

5.0 REDUCE MEMORY BURDEN BY APPLYING PRINCIPLES THAT SUPPORT 22
   RECOGNITION RATHER THAN RECALL .............................................................................
   5.1 Mental Model Abstraction ............................................................................................ 22
   5.2 Support Perceptual Processing of Information ............................................................. 23
      5.2.1 Lessons Learned from Analog Displays ................................................................. 24
   5.3 Abstract and Aggregate Data When Possible ............................................................... 25

6.0 CONSISTENTLY APPLY DESIGN PRINCIPLES TO INTERFACE ELEMENTS .......... 29
   6.1 Design Navigation to Match IA and Mental Models ...................................................... 29
      6.1.1 Summary of Navigation Principles: ........................................................................ 31
      6.1.2 Design Menu to Optimize Navigation Effectiveness ............................................. 32
      6.1.3 Use Labels That Are Intrinsically Meaningful ....................................................... 33
         6.1.3.1 Summary of Menu Principles: ........................................................................ 34
   6.2 Design Pages to be Organized Contextually According to User and System Goals.... 34
      6.2.1 Summary of Pages Principles ................................................................................. 37
      6.2.2 Design Layout to be consistent with IA, Mental Model, and Navigation Scheme..... 38
         6.2.2.1 Summary of Layout Principles: ........................................................................ 42
   6.3 Display Graphics Within Context of Expected Operation ........................................... 42
      6.3.1 Reserve Use of Saturated Color to Communicate Important Information .......... 43
         6.3.1.1 Summary of Design Principles for Graphics ................................................... 45
      6.3.2 Use Plots to Illustrate Parameter Behavior Over Time ...................................... 45
         6.3.2.1 Summary of Plots Principles: .......................................................................... 47
   6.4 Display All Control Options on Screen ....................................................................... 47
      6.4.1 Designing Clear Functionality in Controls ............................................................. 49
      6.4.1.2 Controller-Component Relationship................................................................. 50
Figures

Figure 1. Process of design philosophy to user interface style guide........................................ 11
Figure 2. IA hierarchy for accelerator control rooms................................................................. 17
Figure 3. Accelerator tree structure. ......................................................................................... 18
Figure 4. Accelerator overview displays highlighted in green (“comfort displays”).................... 19
Figure 5. Accelerator role-based user interfaces, information content is customized for specific user role.................................................................................................................. 21
Figure 6. BBM: Linac design WITHOUT intuitive perceptual processing. ................................. 24
Figure 7. BBM: Linac design WITH perceptual processing. ...................................................... 24
Figure 8. ACNET injection closure (numerical values only). ...................................................... 25
Figure 9. Abstraction of data representation in a database........................................................ 26
Figure 10. Example of hex code labels in current ACNET parameter page seen under “Trims” column. ................................................................................................................................. 27
Figure 11. Left interface is unaggregated data and right interface is aggregated. ....................... 28
Figure 12. Accelerator navigation hierarchy. ............................................................................. 30
Figure 13. Manual-based navigation (left) versus aggregated navigation (right)......................... 31
Figure 14. Menu icon design (hamburger) and location (top-left corner) example..................... 32
Figure 15. Unclear labels (left) and clear labels (right). ............................................................ 34
Figure 16. ACNET index page. ................................................................................................... 36
Figure 17. General example of operator “home” page. ............................................................... 36
Figure 18. Mimic example: Fermilab Accelerator Complex. ...................................................... 39
Figure 19. Mimic example: accelerator water distribution......................................................... 40
Figure 20. “Z” scan path. .......................................................................................................... 40
Figure 21. ACNET index page versus operator role-based user interface. ................................ 41
Figure 22. Example design of tabs, pagination, and breadcrumbs............................................ 42
Figure 23. Left side illustrates an interface showing data only. The display on the right shows a simple example of that same data displayed within the three operational contexts. .................. 43
Figure 24. Illustrated on the left is the overuse of saturated colors. The example on the right shows one way to use saturated color to capture operator attention. ................................. 44
Figure 25. The left plot shows the relationship of four measures over time. The plot on the right shows the relationship of two measures over time within their operational context. .... 46
Figure 26. Bullseye plot. .......................................................................................................... 47
Figure 27. Controller example with all control options displayed............................................. 48
Figure 28. “Clicked” indications (bottom buttons have been clicked). ........................................ 52
Figure 29. Confirming original control input. ......................................................................... 53
Figure 30. The top series is the six cognitive processes human operators iteratively employ while operating, and the bottom series is a simplistic explanation of an automated process.... 55
Figure 31. Booster tunnel alarm page. .................................................................................... 63
Tables

Table 1. Tradeoffs of relating control functionality to impacted components and systems........ 49
Table 2. Guidance for selecting tasks suited for automation. ..................................................... 56
Table 3. Categorize operator notifications based on requisite urgency........................................ 61
DESIGN PHILOSOPHY FOR ACCELERATOR CONTROL ROOMS

1.0 INTRODUCTION

This report describes the foundational human factors principles needed to design user interfaces for accelerator control rooms. These principles are presented as recommendations for accelerator control system interface design and are illustrated with examples drawn from accelerator operations at Fermi National Accelerator Laboratory (Fermilab). These high-level principles are intended to serve as design objectives that apply generally across all control system interfaces developed for the accelerator complex. These principles will be mapped to more specific design guidance that will be provided in a style guide for accelerator operations.

This design philosophy document is a high-level roadmap to approach the general interface design for the accelerator complex. This philosophy applies state-of-the-art human factors principles and evidence-based design standards to an accelerator environment, which will transform the next era of accelerator interfaces as outlined by the Accelerator Controls Operations Research Network (ACORN) Project. Further work will include mapping the general
design principles included in this report to more specific guidance in the form of a user interface style guide (see Figure 1).

An extensive review of operator tasks and processes was completed prior to the development of this report to determine the information landscape and interaction design of accelerator operators. The main operator tasks are: acquire awareness of the current system state and expected operating efficiencies, monitor machines within the system for violations of operating efficiencies, identify machines at risk of or currently violating expected operating efficiencies, diagnose what is causing the potential or current violation of expected operating efficiency, and finally, act to prevent or restore the machine to operate within expected efficiency range. To summarize, accelerator operators are primarily concerned with monitoring and maintaining beam quality. Each high-level operator task can be broken down by the necessary actions or information needed for the operator to achieve operator and system goals. Therefore, system upgrades that support operator tasks will subsequently support accelerator control system upgrade objectives.

This report includes high-level design recommendations and where possible also includes information and examples of how they apply to accelerator operations. Many of the design principles are illustrated with specific examples of how to address challenges accelerator operators encounter with the current control system. The purpose of including these additional details is to explicitly highlight current interface challenges of Fermilab operators that can be easily and quickly remedied with the proposed recommendations.

The recommendations included in this report will serve as a foundation to modernize and transform accelerator interfaces by creating design consistency and standardization for all types of displays. The recommendations should be referenced and applied to all display development related to the control system, whether it’s a broad-scope modernization or a small-scale custom screen to accommodate an upcoming experiment. When additional context is needed, developers can refer to the accompanying background and explanation sections in this report, which provide illustrations and justification for design decisions.

2.0 DESIGN PHILOSOPHY

The philosophy here is general and includes high-level guidance on how to develop intuitive and more usable accelerator control system interfaces. This design philosophy includes direction for how to structure and organize an accelerator control system interface, reduce memory burden by applying principles that support recognition rather than operator recall, consistently apply design principles to interface elements, and effectively incorporate operator aids, such as
automation. After each high-level recommendation in the design philosophy are more specific design recommendations that directly map to the general philosophy.

2.1 Summary of Design Philosophy

- **Structure and content of accelerator control system interface**: A well-planned structure is needed to accommodate the complexity and variety of information included in an accelerator complex as well as present that information in ways that are intrinsically meaningful to operators.

- **Reduce memory burden by applying principles that support recognition rather than recall**: Humans have limited short-term memories and control system interfaces that promote recognition and reduce the user’s cognitive burden.

- **Consistently apply design principles to interface elements**: Interfaces with design consistency create predictable interactions, which in turn generate trust and efficiency.

- **Effectively incorporate operator aids**: The accelerator control system should incorporate automated tools to support operators to reduce the cognitive and physical workload and focus their attention on their most important responsibilities.

3.0 RECOMMENDATIONS

Supporting the high-level design principles are more specific design recommendations. The following recommendations were developed by consolidating findings from accelerator operator interviews, evidenced-based human factors design principles, and considerations from accelerator control system benchmarks. These recommendations include varying levels of detail: general recommendations that apply to all types of interfaces included in an accelerator complex and further detailed recommendations that cater to current accelerator operator challenges at Fermilab. The detailed recommendations include examples of how the tools (e.g., ACNET) and processes (e.g., E-log) that operators currently use do not apply human factors principles and suggestions for how to apply human factors principles effectively in all human system interfaces across the accelerator complex.

3.1 Summary of Recommendations

- **Structure and Content of Accelerator Control System Interface**
a. **Information architecture.** The accelerator control system should include an information architecture (IA) that supports a variety of information levels to simplify user comprehension and optimize operational performance. The information architecture should illustrate the information ecosystem of the accelerator control system and demonstrate how machine-based data is consolidated and presented for human consumption.

b. **Overview displays.** Overview displays should include key indications of overall accelerator system performance and apply concepts of data abstraction and aggregation to rapidly communicate degrading, abnormal, or emergency conditions through “at-a-glance” interactions.

c. **Role-based user interfaces.** The accelerator interface should support a variety of user roles to enhance the performance of users with different operational roles. Each role-based user interface should be organized contextually according to user responsibilities and the appropriate level of information and functionality to accommodate said responsibilities.

2. **Reduce Memory Burden by Applying Principles that Support Recognition Rather than Recall**

   a. **Mental model abstraction:** Prior to accelerator interface development, designers should capture and abstract user mental models. User mental model abstraction should include documenting the responsibilities, processes, resources, and functionality of identified users.

   b. **Support perceptual processing of information:** The interface should display information that supports directly perceiving the meaning of data rather than relying on operator memory or mental calculation wherever possible.

   c. **Abstract and aggregate data when possible:** The user interface should abstract and aggregate data to increase the simplicity and comprehensibility of the interface.

3. **Consistently Apply Design Principles to Interface Elements**

   a. **Design navigation to match information architecture and mental models:** The accelerator control system navigation should map to the information architecture structure. Global navigation should allow users to access their role-based interface. Local navigation allows the user to focus on specific system information relevant to their task. The accelerator control system navigation should support flexibility for search-based navigation. A navigation menu should reflect the structure of the navigation scheme.
b. **Design menu to optimize navigation effectiveness:** The accelerator interface should include a menu that is accessible and clearly labeled to optimize user comprehension and effective navigation.

c. **Use labels that are intrinsically meaningful:** The accelerator interface should include labels that are intrinsically meaningful with plain language that is concise yet descriptive to optimize user comprehension and navigation effectiveness.

d. **Design pages to be organized contextually according to user and system goals:** The accelerator interface should include pages that are contextually based and organized according to user tasks and goals and system tasks and goals. Pages should include aggregated data and perceptual processing graphics wherever possible to optimize page real estate Custom pages should be automatically deleted when they are no longer useful or evaluated for routine use. Pages should support the navigation scheme wherein global pages are accessed first and local pages are accessed next.

e. **Design layout to be consistent with IA, mental model, and navigation scheme:** The interface layout should organize information according to the mental model of the user and should maintain consistent information placement to support a reliable and predictable user experience. The page title and menu icon should be continuously presented and consistently located regardless of page context. Context-dependent information placement should support specific user responsibilities.

f. **Display graphics within context of expected operation:** Graphics should be displayed within context of their expected operation and operational limits to provide operators with diagnostic information as well as current system status.

g. **Reserve use of saturated color to communicate important information:** Create an interface environment that supports the use of prominent features that can capture and guide operator attention towards areas of interest.

h. **Use plots to illustrate parameter behavior over time:** Use plots to illustrate parameter behavior over time and the relationship between two (or more) functionally related parameters and include operational context clues as well.

i. **Display all control options on screen:** The accelerator control system and interface design should clearly distinguish the component relationship, function, and presence of all control options available on screen.

j. **Use feedback to signal system changes:** The control system interface should support a variety of feedback interactions to inform users of autonomous system changes and user-initiated changes.

4. **Effectively Incorporate Operator Aids**
a. **Automate operator tasks that can reasonably be automated:** Integrating automation should involve a review of system functions to identify opportunities where automation frees up operator cognitive resources to focus on higher-level system operation. Anything that can be reasonably automated should be automated.

b. **Design alarms to effectively capture and guide operator attention:** Reserve alarms for events that require immediate operator attention. Maintaining alarm integrity may require adjusting alarm setpoints to fit current operational expectations. Further, alarm sets representing specific operational tasks known to trigger expected alarms can alleviate unneeded alarms. Replacing less important alarms with alerts, notifications, or clear graphic and interface design will make operators aware of situations approaching an alarm event and provide clear information and navigation options with the alarm event to manage the situation.

### 4.0 STRUCTURE AND CONTENT OF ACCELERATOR CONTROL SYSTEM INTERFACES

An accelerator complex includes a variety of information from multiple sources. To accommodate the complexity and variety of information included in an accelerator complex as well as present that information in ways that are intrinsically meaningful to operators, a well-planned structure is needed. The following sections provide general recommendations and details for how to organize the information included in accelerator control system interfaces. IA discusses levels of information and how to structure information into a hierarchy. Additional sections discuss levels of human-system interfaces (HSIs) and what level of information is appropriate for those HSIs.

#### 4.1 Information Architecture

**Recommendation:**

The accelerator control system should include an IA that supports a variety of information levels to simplify user comprehension and optimize operational performance. The IA should illustrate the information ecosystem of the accelerator control system and demonstrate how machine-based data is consolidated and presented for human consumption.

**Background and Explanation:**
IA is the overall conceptual model used to plan, structure, and assemble system information (Rosenfeld, 1998). The purpose of IA is to organize and construct system information in a way that simplifies user comprehension to optimize operational performance. A successful IA helps users to understand where they are in a system, what they’re interacting with, how to navigate their interaction, and what to expect next.

IA is the foundation of consolidating and presenting machine-based information from an industrial control system for human consumption. There are multiple levels of information in an industrial control system, and the purpose of IA is to organize the information to support manageability, findability, and usefulness (Rosenfeld, 1998). The lowest level of information for an accelerator control system is detailed information regarding accelerator hardware, such as machines and devices. This is the level that includes sensors, transmitters, actuators, and switches (royal blue level in Figure 2).

The next level is the process level (light blue level in Figure 2). This level includes processes that connect and transform specific machine-based data to be more easily interpretable by humans. This level includes elements like server software, custom programs, user software, and coding scripts.

The top level of information for an accelerator control system IA is the HSI level (light gray level represented in Figure 2). This level supports an operator’s ability to comprehend and interact
with the control system. The HSI level includes two main elements: high-level accelerator information (overview displays) and operator control information (role-based user interfaces).

### 4.1.1 STRUCTURE USER INTERFACES BASED ON INFORMATION HIERARCHY

The top level of the IA hierarchy is where system data is presented to operators as rich information. Within this level, the information can be further broken down within an information hierarchy. A useful hierarchy concept for this application is called a tree structure, which organizes types of information in a top-down manner (i.e., parent-child relationship between information). The parent information is presented first or accessed most easily. The parent information includes leading indications or information that signals operators towards follow-up actions, such as issue diagnosis or discovery. The ability to drill down to more specific information (i.e., child) is included within the same screen or within a short navigational distance (e.g., a single action or click away). How information is organized depends on the goals and tasks of the user navigating the interface. For example, an interface designed to support accelerator operators should present high-level monitoring information concerning beam efficiencies as most accessible. Since the primary goal and task of accelerator operators is to monitor and manage beam quality within beam safety and quality expectations, any general information operators use to identify degrading or poor beam condition should be presented first (i.e., top of the interface). Any additional or supplemental information relating to the top-priority information should be listed next (“next level” represented in Figure 3), and so on. This not only creates a visual hierarchy of information but also a functional hierarchy of information wherein users are able to access the most important information easily and access more detailed information when needed.

![Accelerator tree structure.](image)

In a particle accelerator control system environment, two levels support an intuitive and hierarchically organized IA: overview displays (high-level, information only) and role-based user interfaces (lower-level information and operator control).
4.2 Overview Displays

Recommendation:

Overview displays (OVDs) should include key indications of overall accelerator system performance and apply concepts of data abstraction and aggregation to rapidly communicate degrading, abnormal, or emergency conditions through “at-a-glance” interactions.

Background and Explanation:

OVDs are the top level of accelerator information in the proposed information hierarchy. The OVDs support broad system awareness by abstracting and aggregating accelerator information for at-a-glance status acquisition. Operators should rely on OVDs to provide the first indications of degrading, abnormal, or emergency conditions for accelerator monitoring.

OVDs are group view displays that provide an at-a-glance status of general accelerator information. Overviews are typically located in a central location that is viewable by all crew members. OVDs help foster a shared understanding of system status. Having a shared understanding is beneficial when crew members must coordinate their actions when either claiming or delegating actions to team members.

Figure 4. Accelerator overview displays highlighted in green (“comfort displays”).

In an accelerator environment, crew chiefs (i.e., shift leaders) rely on overview displays to determine the status of accelerator equipment and delegate actions to operators.
4.2.1 SUMMARY OF DESIGN PRINCIPLES FOR OVDS:

- OVDs should be positioned in a central location that is viewable by each crew member at a reasonable distance (i.e., within reading distance with normal or corrected-to-normal vision)
- OVDs should provide an abstracted representation of overall system status by including high-priority accelerator information, such as trending equipment telemetry history
- OVDs should include embedded indications to notify crew members of impending abnormal or emergency operating conditions
- OVDs should be designed to support rapid event identification to enable an accelerator crew chief to delegate response actions to additional crew members.

4.3 Role-Based User Interfaces

Recommendation:

The accelerator interface should support a variety of user roles to enhance the performance of users with different operational roles. Each role-based user interface should be organized contextually according to user responsibilities and the appropriate level of information and functionality to accommodate said responsibilities.

Background and Explanation:

Role-based user interfaces are the next level of accelerator information in the proposed information hierarchy. The purpose of role-based user interfaces is to contextually separate accelerator information to better accommodate the variety of user roles. The more visually and functionally customized a display is to a user, the more intuitive their interaction will be.

User roles are distinguished by the responsibilities of the users that interact with the accelerator control system. The primary users include accelerator operators, machine experts, and engineers. Each of these users have different responsibilities for managing accelerator performance that require customized information and interaction capabilities.

To accommodate the variety of responsibilities, as well as optimize user performance, a role-based user interface is needed. For example, an accelerator operator's primary responsibility is to ensure beam viability by acting as a first responder to any accelerator ailments. Should an unexpected event occur, operators act swiftly to mitigate the incident by performing diagnostics and restorative controls. If the needed repair is beyond operator capabilities, machine experts are consulted. Machine experts are specialized to specific equipment and are therefore better
equipped to solve complex issues relating to their specific devices. The information a machine expert may require to diagnose problems with their equipment will differ from the general knowledge held by control room operators. These two roles have similar high-level operational goals (e.g., ensure beam quality), but they address those goals with access to different levels of information. The system should not force, but instead should optionally allow operators to interact with data and detailed information at the same level and intensity as machine experts or engineers.

To best accommodate multiple accelerator users, the control system should distinguish among three user types:

1. Accelerator Operator
   a. Crew Chief
   b. Shift Operators
   c. Operations Specialist Staff

2. System Expert
   a. Machine Specialist
   b. Physicist

3. Engineering
   a. Integrators
   b. Engineers
   c. Technicians

Each mode of the accelerator interface will optimize the interaction of the specific user by customizing visual and functional elements according to the operational goals, responsibilities, and appropriate level of information for their specific role.

Figure 5. Accelerator role-based user interfaces, information content is customized for specific user role.
4.3.1 SUMMARY OF DESIGN PRINCIPLES FOR ROLE-BASED USER INTERFACES

- The accelerator interface should support three user types: operator, system expert, and engineer.
- User interfaces should be organized contextually according to user responsibilities and the appropriate level of information and functionality to accommodate said responsibilities.
- By default, user interface displays should provide the most relevant information to the appropriate user.

5.0 REDUCE MEMORY BURDEN BY APPLYING PRINCIPLES THAT SUPPORT RECOGNITION RATHER THAN RECALL

A common goal of HSI design is to minimize the user’s memory load by making elements, actions, and options visible (Nielsen 1994a). Humans have limited short-term memories and control system interfaces that promote recognition to reduce the user’s cognitive burden. In an accelerator environment, this means an operator’s optimal performance should not be dependent on their ability to remember unique intricacies of their control system interfaces. The following sections include general principles on how to promote recognition within accelerator HSIs and reduce the dependency on operator recall.

5.1 Mental Model Abstraction

Recommendation:

Prior to accelerator interface development, designers should capture and abstract user mental models. User mental model abstraction should include documenting the responsibilities, processes, resources, and functionality of identified users.

Background and Explanation:

A comprehensive understanding of a user’s responsibilities, processes, resources, and functionality is needed to abstract a mental model. The purpose of abstracting a mental model is to intuitively replicate a user’s natural understanding of their system functions and relationships to optimize performance.

A mental model is based on what users know (or assume) about a system (Nielsen, 2010). Mental models can sometimes be restricted by the tools available to the users (e.g., the Accelerator Control Network (ACNET) for operators). The more cumbersome and confusing the
available tools are, the more difficult it is to support an intuitive interaction through mental model abstractions. Thus, the recommendations and design philosophy detailed in this report are not only meant to improve operator interface design and usability but also simultaneously improve an operator’s natural understanding of the system to optimize performance. A successful mental model abstraction captures how a user collects system information and compares it to their learned knowledge of the system to inform and select their actions. However, individual users create their own, unique mental models even when operating through the same interface. Therefore, one of the biggest challenges in interface development is to generalize a specific group of users’ mental models enough to create familiarity and predictability in the interface without hindering individual performance.

A group of users is defined as personnel with similar responsibilities, processes, resources, and functionality. While interviewing operators to understand the main control room interface processes, three groups of users were identified as distinct enough to warrant a unique role-based interface (see Section 4.3). In other words, to best support a variety of personnel interactions with the accelerator system, three types of user interfaces are necessary to replicate unique mental models: operator, system expert, and engineer. Although the content of this report is specific to accelerator operators and their accompanying mental model, further work should include abstracting the additional users’ mental models to create specific examples of design recommendations for those particular user interfaces (i.e., machine expert and engineer).

5.2 Support Perceptual Processing of Information

**Recommendation:**

The interface should display information that supports directly perceiving the meaning of the data rather than relying on operator memory or a mental calculation wherever possible.

**Background and Explanation:**

Perceptual processing refers to the capability of an operator to “look” at a display and understand its meaning with minimal interpretation and calculation. The purpose of perceptual processing is to support users with visible cues to intuitively navigate and interpret an HSI. Instead of relying on memory to operate a system, users are guided by visual cues toward what is important to them. The more users rely on memory or manual calculation to operate a complex system, the more opportunity there is for error. One example where perceptual processing philosophy can be applied in an accelerator environment is the beam budget monitor (BBM). The BBM is meant to display the overall “health” of the accelerator. Beam delivery metrics are presented for multiple areas of the accelerator complex including Booster, Linac, and Recycler. Each of these metrics are displayed with an accompanying data plot to
demonstrate the current status of beam data in relation to the listed metric (i.e., beam delivery rates). However, a high degree of tribal knowledge is required to not only interpret the data (i.e., understand if the data is within acceptable limits) but also to understand what operator actions are necessary based on that interpretation. Including additional context in the metric plots that highlight the desirable or expected outcomes would reduce the need for expert knowledge/recall to effectively operate. Additional context, such as the expected state, optimal range, and alarm limits, help operators know that the current status is acceptable without having to think about it.

Including perceptual processing wherever possible reduces users’ cognitive workload by reducing the amount of expert knowledge and recall needed for operations. This also frees up user cognitive space for focusing on more important tasks, such as decision-making (e.g., responding to alarms).

5.2.1 LESSONS LEARNED FROM ANALOG DISPLAYS

Process control industries have been transitioning their old analog (analog meters, hard controls) to digital displays. This transition has created challenges for operators when the full contextual value of an analog meter is not carried over into the new system. Many analog meters include context such as alarm limits, ranges, and expected state. This information is often replaced by digital readouts of current value(s). For example, on the Injection Closure page of ACNET (Figure 8), the current value is displayed alongside “restore” values to signify any potential previous change in the value. Operators are meant to compare these values to determine what action to take. However, a lot of tribal knowledge is required to effectively interpret these values. For example, even if a numerical change has occurred in these values, it’s not visually or immediately clear what that change means or what type of operator action is
warranted. Instead of providing numerical readouts of the data points and forcing operators to perform mental calculations to interpret the values, the interface should consolidate these values into a plot with visual perceptual processing information, such as optimal range and alarm limits.

A lack of contextual information requires operators to rely on their own knowledge of the alarm limits, measure ranges, and expected value, which requires cognitive processing. Applying contextual information, iconography, and graphical displays supports perceptual processing by allowing the operator to “look” at the value and judge its quality almost immediately.

5.3 Abstract and Aggregate Data When Possible

**Recommendation:**

The user interface should abstract and aggregate data to increase the simplicity and comprehensibility of the interface.

**Background and Explanation:**

The purpose of providing data abstraction and aggregation is to support operators by providing a simple interface free of unnecessary detail and clutter and to ensure that data is presented in an intrinsically meaningful way to operators.
Data abstraction describes transforming raw data or machine-interpretable data into information that is readily interpretable by humans. This term is borrowed from computing, and Figure 9 shows an example of data abstraction in a database.

Figure 9. Abstraction of data representation in a database.
Figure 10. Example of hex code labels in current ACNET parameter page seen under “Trims” column.

Data abstraction can also refer to directly transforming information into higher-level meaning and presenting that directly to operators. Data abstraction can be applied to monitoring data, such as processing raw telemetry and presenting it in a way that directly supports operator decision-making. It can also refer to how elements in the control system are labeled and presented to the operators.

The way parameters are labeled in the current ACNET system serves as an example for how data could be abstracted for operators. Many operators have described the challenges in remembering all the parameter names, because there isn’t a naming standard available and some are represented in the ACNET interface as alphanumeric codes, i.e., in the same way that they are stored at the logical level. Transforming the parameter names into plain language labels that are intrinsically meaningful to operators would save considerable time and effort in learning the mapping from the code to the meaning and in workload during operation.

It is important to note that the appropriate level of abstraction varies depending on the context and users. For example, machine experts may need less abstraction than operators because they need to interact with lower-level details in the control system. Users who have learned to interact with unabtracted data may have difficulty finding and interpreting information when it is abstracted. Therefore, for newly upgraded systems, the interface should support legacy users and new users by allowing information to be accessed in both the original and abstracted form wherever possible.
Data aggregation refers to combining several data inputs into a single output that is readily meaningful to operators. Data aggregation is typically used to describe summarizing data for statistical analysis, but here it describes combining data in any way that creates an output that is simplified or reduced to be more readily understood by an operator. A simple example of data aggregation is when an operator needs to check the status of multiple similar components in a system (e.g., as a prerequisite for start-up). Instead of presenting the status of each component on a summary page, the system could simply aggregate the status to say whether the system is ready or not ready. This will allow for a simpler user interface and faster recognition of system conditions. In this example, the interface should support a drill down to unaggregated data if the operator needs more detail to diagnose an issue. An example of this type of aggregation is in the channel display on the left of Figure 11 which could be simplified by showing an aggregated status with the option to drill down into more detail when needed.

Figure 11. Left interface is unaggregated data and right interface is aggregated.

Aggregation is appropriate when it simplifies the user interface without increasing interface management tasks like navigation. In the example in Figure 11, if the expected conditions read that normally the aggregated status will be good and the drill down (i.e., access additional detailed information) won’t be required unless something is abnormal, it would be appropriate to aggregate that information. If it is expected that the user will need to frequently drill down, the tradeoffs between presenting aggregated data should be weighed against the increased navigation the operator will have to perform.
6.0 CONSISTENTLY APPLY DESIGN PRINCIPLES TO INTERFACE ELEMENTS

A well-established usability heuristic law known as Jakob’s Law states that the time humans spend using digital products in their everyday lives shapes their expectations for all digital products they use (Nielsen 1994a). In other words, control system users are influenced by the digital products they use outside of the accelerator complex, and those products influence their expectations for the accelerator control system interfaces. It is impossible to translate exact design interactions meant for simple applications (e.g., using a remote controller to turn on a television) to complex applications (e.g., setting experiment parameters for an accelerator study). However, certain design conventions, such as consistency, help replicate the expected interactions of everyday digital products in an accelerator environment. To create consistency, accelerator interfaces should not include information that is irrelevant or rarely needed. Every extra element of information competes with the relevant elements of information, which can diminish their relative visibility and usability. Design consistency is about ensuring that the visual and functional elements of the interface support the user’s (e.g., accelerator operator’s) primary goals.

Although an accelerator control system is complex compared to everyday digital products, creating interfaces that provide design consistency creates predictable interactions, which in turn generate trust and efficiency. Accelerator operators should not have to wonder whether different labels, colors, or actions mean the same thing. Failing to maintain design consistency increases cognitive workload, which can lead to burnout. Although design consistency principles extend beyond what is included in the following sections, the following sections provide specific recommendations for design consistency principles that are based on observations from the operator interviews.

6.1 Design Navigation to Match IA and Mental Models

**Recommendation:**

The accelerator control system navigation should map to the IA structure. Global navigation should allow users to access their role-based interface. Local navigation allows the user to focus on specific system information relevant to their task. The accelerator control system navigation should support flexibility for search-based navigation. A navigation menu should reflect the structure of the navigation scheme.

**Background and Explanation:**
The purpose of navigation is to access and retrieve a display element or control within an interface. Navigation design is the discipline of creating, analyzing, and implementing ways for users to navigate through a system. Navigation is how a user can get from Point A to Point B in the most intuitive, expeditious, and least frustrating way possible.

Control system navigation should map to the IA structure and mental model of the target user. The closer the navigation scheme mirrors a well-designed control system IA; the more intuitive a user's interaction will be. The navigation included in these displays should be organized in a hierarchy (i.e., top-down manner) to optimize usability and accessibility. The proposed IA includes two main types of navigation organization: global navigation and local navigation. Global navigation is the top tier (i.e., general) information. It includes the highest level of organization in the navigation hierarchy. Global navigation is how users access their role-based interface. Local navigation is lower level (i.e., more specific) information that is contextual to the user's current location. This type of navigation includes child categories to the global navigation.

One way to increase navigation usability is designing for the fewest actions (e.g., clicks or keystrokes) to access necessary information. The proposed IA model wherein user roles have specific interfaces, makes applying this recommendation easier. For example, upon beginning a shift, an operator will access the operator interface, which presents the most relevant information immediately instead of requiring the operator to manually navigate to and develop
custom displays to view data. The left side of Figure 13 depicts an example of five monitoring windows that the operator must individually access. The right side depicts an operator role interface with summary monitoring, alarms, and additional information that is most relevant to the operator tasks. Combining curated interfaces that map to the IA of the system minimizes navigation requirements. Such a structure fosters ease of use and rapid access to pertinent accelerator controls and information.

Figure 13. Manual-based navigation (left) versus aggregated navigation (right).

### 6.1.1 SUMMARY OF NAVIGATION PRINCIPLES:

- Navigation should be clearly labeled so users understand where they are and where they can go
- Navigation design (icons, labels, etc.) should remain consistent throughout the entire application
- Navigation should support the ability to directly access the main display from all screens
- Navigation should support a traceable path (e.g., breadcrumbs)
- Navigation should be as simple as possible by reducing the number of required clicks and keystrokes (i.e., create the shortest path between two related points)
- Access to screens one level (above or below) from the current level should be no more than one action (i.e., click or keystroke) away
- The navigation scheme should reflect the accelerator IA to support an intuitive user interaction
- Navigation should be directed by mouse interaction (primary) and keystrokes (secondary)
• The accelerator control system navigation should support flexibility for search-based navigation.

6.1.2 DESIGN MENU TO OPTIMIZE NAVIGATION EFFECTIVENESS

Recommendation:

The accelerator interface should include an accessible and clearly labeled menu to optimize user comprehension and navigation effectiveness.

Background and Explanation:

An example of menu accessibility is ensuring that the menu (or menu icon) is visible at all times and located in a consistent spot. Since the menu is the hub of navigational information in a digital interface, it should always be accessible. Locating the menu or menu icon in a consistent spot provides reliability and predictability, both of which support situation awareness and a reduced workload. The majority of digital interfaces designate the top-left corner for menu placement (Bowman, 2020). When menus are placed outside of these areas, it can feel awkward and counterintuitive from a usability standpoint. A common and familiar icon to indicate the menu within an interface is a "hamburger" menu icon illustrated in Figure 14 (Bowman, 2020).

![Operator Overview](image)

*Figure 14. Menu icon design (hamburger) and location (top-left corner) example.*

An example of an effective menu is a menu that includes labels that are intrinsically meaningful and plainly described. Menus are only as clear as the labels and contextual terminology included in them. Labels included in a menu should follow the same guidance recommended in the labels section: plain and concise yet descriptive language that is intrinsically meaningful.
Additionally, an effective menu follows the IA. Global navigation labels should be represented at the top level of a menu, visually and functionally, followed by local navigation labels. The menu should reinforce the IA hierarchy to strengthen the comprehensibility and usability of the accelerator interface.

6.1.3 USE LABELS THAT ARE INTRINSICALLY MEANINGFUL

Recommendation:

The accelerator interface should include labels that are intrinsically meaningful with plain language that is concise yet descriptive to optimize user comprehension and interaction effectiveness.

Background and Explanation:

An additional way to improve the intuitiveness of user interaction is to clearly label titles and links with terminology that is familiar, relevant, and plainly describes the content or features included in the interface. The use of labels directly affects how well users can comprehend the information they are processing as well as predict where they are navigating to. When an interface includes clear labels, users are better equipped to understand the interface. The more intrinsically named labels are, the more powerful and optimal search-based navigation is.

Labels that are intrinsically meaningful to the user are an example of clear labels. For example, some of the main machines included in the accelerator complex are meaningfully named (e.g., the Booster machine “boosts” particles to increase their velocity). However, many accelerator subsystems are not named with intrinsic meaning, making it difficult to find devices while performing diagnostics. This leads to time delays in diagnosis and the potential loss of beam efficiencies. An opportunity for immediate HSI improvement concerning labels is renaming all devices in an intuitive way that is intrinsically meaningful to accelerator personnel.

In addition to intrinsic meaning, clear labels also contain plain language that is concise yet descriptive (see Figure 15). In the example below, the left side is a label included in ACNET. The “more info” label is used to signify access to a complete list of digital statuses since not all statuses could fit on the display. It is also used to provide access to an expanded history of certain data parameters. “More info” isn’t technically incorrect in either application; however, it also isn’t intrinsically obvious what “more info” means in either application. Alternative suggestions are represented on the right side of Figure 15. These suggestions apply the clear label guidance of containing plain language that is concise yet descriptive, and it is immediately obvious what clicking on these labels will lead to.
Including clear labels improves a user’s understanding of what information means and what an interactive element will do. This will increase navigation effectiveness and reduce confusion and unintentional action. Wherever possible, use specific labels because their meaning is immediately recognizable compared to general terms, icons, or abbreviations, which need to be inferred.

6.1.3.1 SUMMARY OF MENU PRINCIPLES:

- The interface menu or menu icon should be always visible and located in a consistent spot to provide accessibility and consistency to optimize the user experience.
- The interface menu should include clear and effective labels to reinforce the information hierarchy and support intuitive navigation.
- The interface menu should include information that is divided into contextual pages (i.e., grouping information with similar framework together).

6.2 Design Pages to be Organized Contextually According to User and System Goals

Recommendation:

The accelerator interface should include pages that are contextually based and organized according to user tasks and goals and system tasks and goals. Pages should include aggregated data and perceptual processing graphics wherever possible to optimize page real estate and avoid “blowing away plots.” Custom pages should be automatically deleted when they are no longer useful or evaluated for routine use. Pages should support the navigation scheme wherein global pages are accessed first and local pages are accessed next.

Background and Explanation:
A page in digital products can be broadly defined as a website, application, or window display. In this context, a page is a particular application screen that is a part of the accelerator interface. In control systems as complex as accelerators, multiple pages are needed to access additional information not included on the user interface home page. The pages of an interface visually group contextual information together. The purpose of having multiple pages is not only to accommodate a complex system but also to organize information according to context.

The real estate of a single page is limited (i.e., how much meaningful data can fit onto a page), but the number of pages is not. However, an endless number of pages increases the cumbersomeness of navigation. The goal of page development is to evaluate design tradeoffs when considering how to divide information between pages. For example, a current challenge of Fermilab accelerator operators is “blowing away plots,” which means that operators are opening multiple plots within a single monitor display and unintentionally exiting them due to an application threshold concerning number of pages (i.e., in this instance plots). This is especially problematic when operators are performing diagnostics and attempting to access additional data only to have their previously accessed data disappear. A seemingly simple solution would be to remove the number of pages threshold in a single display. However, the more pages (i.e., plots or graphs) operators are able to include in a single monitor, the more effort operators must dedicate to functional navigation and visual scan paths, which can lead to a higher workload and potential burnout. This is precisely why aggregating data and providing perceptual processing context is important; it optimizes the amount of information real estate included in a single page and thus reduces the cognitive workload of operators.

Therefore, accelerator control system pages should include information that is contextually related to operator goals and system goals. The content of each page is dependent on evaluation of tradeoffs identified in user interface development. A page could include multiple plots or a single plot. A page could also only include directory information. An example of a directory page included in ACNET is the index page. The index page provides visual and functional access to a list of custom programs developed by the controls department programmers. However, the index page is cluttered with programs that haven’t been used in years (see Figure 16).
Custom programs in ACNET are used to automatically populate an operator console monitor with re-selected plots (i.e., the operator doesn’t have to manually access certain data). If custom programs or pages are needed (i.e., pages with specific parameters for experiments), they should be deleted automatically when the experiment ends or be evaluated for routine use. Removing unnecessary programs and pages will reduce visual clutter and the functional cumbersomeness of the interface.

The home page (i.e., landing page) should include all the most relevant information and functionality that a user needs to perform their work safely and efficiently.
This is the page that users access first and most often and should focus the user's attention to the most important information. Although role-based user interfaces include lower-level information to support operator control, the home page of a user interface should include the highest level of information for that specific level in the IA. In other words, the home page provides general information pertinent to safe and efficient operations with embedded links to more detailed information. This way, users can access additional pages of information when needed without overcrowding the home page.

Including multiple pages in a control system interface provides the ability to group similar contextual information together. This is especially beneficial in complex systems, such as an accelerator where there is an abundance of highly specific information. Additional pages in an accelerator environment for operators will most likely include, but are not limited to:

- An alarms page with further detailed alarm information
- Expanded views and histories of monitoring parameters (e.g., Booster, Linac, Recycler)
- Device control pages
- Current experiment protocols.

6.2.1.1 SUMMARY OF PAGES PRINCIPLES

- The role-based user interfaces should include a home page where the most relevant information is presented for users to perform their work safely and efficiently
- Pages should include information that is contextually based (e.g., according to user tasks and goals and system tasks and goals)
- Pages should include aggregated data and perceptual processing graphics wherever possible to optimize page real estate and avoid “blowing away plots”
- When custom pages are needed, they should be automatically deleted when they are no longer useful or be evaluated for routine use
- Pages should accommodate the mental model of the specific user (e.g., operator interface should support beam monitoring capabilities as a primary task) and prioritize information placement based on user responsibilities
- The information included in a page should be visually spaced out to avoid visual and functional clutter (i.e., no occluding or overlapping information)
- User interface pages should only include data and information that is relevant to the specific user
• Pages should support the navigation scheme (i.e., global pages accessed first, then local pages).

6.2.2 DESIGN LAYOUT TO BE CONSISTENT WITH IA, MENTAL MODEL, AND NAVIGATION SCHEME

Recommendation:

The interface layout should organize information according to the mental model of the user and should maintain consistency in information placement to support a reliable and predictable user experience. The page title and menu icon should be continuously presented and consistently located regardless of page context. Context-dependent information placement should support specific user responsibilities.

Background and Explanation:

A layout supports operators by providing a well-organized interface where all visual elements included in the interface are in a consistent and predictable location.

The layout of an interface is the structure that supports all visual components included in a display. The layout organizes information contextually, which helps users make sense of what an interface consists of. A well-organized layout supports both system and user goals and helps users orient themselves in an interface.

In an accelerator environment, the interface layout is dependent on the level of information (i.e., overview versus user interface). The proposed IA inherently supports a display’s ability to focus operator attention; what is included in the layout of an overview will differ from what is included in a user interface. Similarly, what is included in an accelerator operator interface will differ from what is included in a machine expert interface. Separating the information into levels (i.e., high-level overviews and lower-level user interfaces) innately improves the visual and functional clutter of the interfaces by only including the information that is relevant to specific roles and purposes. This helps users focus on the proper content by reducing overall distractions. It also improves the display scan-ability and intuitiveness (White, 2011).

An interface layout is more than display content, it also includes content format and location, which should apply strategy based on user experience research (Farrell, 2015). For example, certain types of information, such as the page title and menu icon, should be continuously presented and consistently located regardless of page context. Context-dependent information placement is more flexible, and grouping items based on similarities (e.g., functionally, task
oriented, and system related) can reduce mental workload by helping operators easily find and interpret display information.

Functional layouts are display elements that are organized and grouped according to similarities in functional purpose and capabilities. In an accelerator environment, most display layouts should be grouped functionally. However, there might be occasional instances where mimic layouts are appropriate. Mimic layouts are visual models that display a specific view of a system (Le Blanc, et al., 2018). It is appropriate to use a mimic layout when the configuration of a system changes the status, function, or outcome of the process and when it is important to see the alignment of individual components in the overall context of the system to understand how the system will behave (Le Blanc, et al., 2018). A good example of this in an accelerator environment is for tasks where the entire beamline and related machinery (i.e., the entire Fermilab Accelerator Complex) need to be displayed, such as to plan a scheduled shutdown (see Figure 18). Another appropriate use of a mimic layout is the accelerator water distribution (see Figure 19).

![Fermilab Accelerator Complex](image)

*Figure 18. Mimic example: Fermilab Accelerator Complex.*
Another way to organize layout information placement is according to user scan paths. A common HSI scan path is a “Z” scan path (Thomas, et al., 2019). The “Z” scan path is represented in Figure 20. Content placement should be user dependent and organized according to relevancy and importance (e.g., high-level beam monitoring parameters are top of screen for operators). When user responsibilities, processes, resources, and functionality are determined, a prioritization of information content and placement can occur. Highest priority information should be placed in the number one spot on the “Z” scan path, next highest in the number two spot, and so on.

An example of how to optimize the layout of an interface is to extrapolate the specific user's
mental model of operations and accommodate it through information placement. For example, accelerator operators are primarily concerned with beam efficiencies. They ensure beam efficiencies through general monitoring, routine control, and occasional diagnostics. Therefore, their mental model likely prioritizes monitoring capabilities, necessary controllers, and investigative capabilities. To properly accommodate this mental model, the layout of the operator interface should only include visual elements (interactive and noninteractive) that support these capabilities. Any additional elements run the risk of clutter, and the more unnecessarily cluttered an interface is, the more cumbersome and difficult it is to find the needed information. In the example shown in Figure 21, the left side includes the ACNET index page. This page includes multiple embedded links to additional information, but no direct monitoring information. Furthermore, most embedded links included on the index page are old custom programs that are no longer used by current operators. Most of the information on the index page is irrelevant for operations purposes. The index page serves as a landing page of sorts (i.e., operators open it to access a variety of information). A user interface display, which is depicted on the right side of Figure 21, also serves as a landing page. However, only relevant information is included within the user interface displays, and there is a noticeable difference concerning an absence of clutter compared to the index page.

Another reason to base an interface layout on the IA and user mental model is because layout directly affects navigation. How an interface is organized impacts a user’s ability to seamlessly and intuitively navigate a display. If the layout is based on a user’s mental model, users are better equipped to predict the location of information more easily, even as novice users. This reduces time spent trying to locate information by navigating to multiple pages.
The layout should always indicate a user’s current location within the system. This is accomplished with visual cues, such as tabs, breadcrumbs, or pagination. For users to successfully and efficiently navigate an interface, they must always know where they are. Including any of the visual cues (Figure 22) or a combination of visual cues to indicate a user’s current location in an interface increases situation awareness and improves interface usability.

**Figure 22. Example design of tabs, pagination, and breadcrumbs.**

### 6.2.2.1 SUMMARY OF LAYOUT PRINCIPLES:

- The accelerator interface should support multiple user interfaces with unique content dependent on information level and user role
- User interface layouts should support a user’s ability to always know their current location in the interface (e.g., pagination)
- User interface layouts should be organized according to user role task importance (e.g., operator interface should support beam monitoring capabilities first and related controls and diagnostic information second)
- The layout of an interface should apply strategy based on user experience research to determine content placement.

### 6.3 Display Graphics Within Context of Expected Operation

**Recommendation:**

Graphics should be displayed within the context of their expected operation and operational limits to provide operators with diagnostic information and current system status.

**Background and explanation:**

A usable interface applies intention to every aspect of its design. Each graphic within an interface should translate system data into actionable information. Properly translating system
data to human operators is a nuanced practice. Displaying more data does not always mean more information is available on the interface. It is important to display data within an operational context, which typically includes individual values or sensory readings displayed within the system constraints such as safe operational parameters or alarm setpoints, and the desired operational values or operational goals of the system (Bennet, Toms, & Woods, 1993). The High-Performance HMI Handbook nicely illustrates this concept (Hollifield, et. al., 2008) with an example using blood test results for a fictional cat “Fluffy” (Figure 23).

An example of graphics with operational context is on the left (red box) of Figure 23 and are the results of different diagnostic tests. A veterinary doctor looking at these results must have the rote knowledge to interpret the results as good or bad (i.e., high, or low). However, the pet owner is unlikely to have the training to understand the results of the test. On the right (green box) of Figure 23, the values are displayed within an operational context. Results are presented on the spectrum with the value represented as a shaded box (Operational Context 1). Also indicated on the spectrum are the thresholds indicating high or low values, values that would be of concern (Operational Context 2). Naturally, the desired range exists within those boundaries (Operational Context 3). The data present in the left box was translated into something meaningful to any user who can, at-a-glance, determine their cat’s blood test results all fall within normal ranges. The green box could be improved as an interface, but this example illustrates how providing the three operational contexts within a graphic reduces the burden of interpretation from the operator.

Figure 23. Left side illustrates an interface showing data only. The display on the right shows a simple example of that same data displayed within the three operational contexts.

6.3.1 RESERVE USE OF SATURATED COLOR TO COMMUNICATE IMPORTANT INFORMATION

Recommendation:
Create an interface environment that supports the use of prominent features that can capture and guide operator attention towards areas of interest.

**Background and Explanation:**

The graphics in Figure 24 use gray-scale coloring illustrating the next point, how to leverage saliency in a graphic to guide operator attention. Saliency is the property of being noticeable or important used to draw the operator’s attention to a value or parameter that may require action. Another example from the High-Performance HMI Handbook illustrates this concept on the right of Figure 24. At first glance, it is clear which parameter has violated an operational threshold. Also, because the three operational context cues are in this graphic, it is immediately noticeable that the value is too high. However, the yellow warning only appears salient because the rest of the image uses a gray-scale coloring scheme.

---

*Figure 24. Illustrated on the left is the overuse of saturated colors. The example on the right shows one way to use saturated color to capture operator attention.*

The “Dull Screen” approach is an interface design concept based on the theory that all normal behavior should appear “dull” so that abnormal behavior detected by the system can be highlighted or made salient through the use of color (Braseth, Veland, Welch, 2004). This strategy helps operators rapidly detect events that need their detailed attention. The left side of Figure 24 is the low conductivity water flow diagram. The colors in this diagram do help differentiate functions of the low conductivity water system. However, all colors share the same approximate contrast with the background. System functionality can be differentiated via other means or “dulling” the colors here to allow a high-contrast icon and color to stand out and capture operator attention. For instance, a yellow warning icon (like that in the right-hand side of Figure 24) would likely go unnoticed unless being viewed by a highly trained eye. If a dull
screen approach was used, a new or less experienced viewer may be able to monitor the process flow diagram and determine when an alarm event was occurring with minimal training.

6.3.1.1 SUMMARY OF DESIGN PRINCIPLES FOR GRAPHICS

- Reserve saturated coloring to highlight information that may require operator intervention
- Visually differentiate between interactive graphics and information-only graphics to provide at-a-glance context of which information users can interact with
- Include operational context where possible
  - Individual values or sensor readings
  - System constraints, such as safe operational parameters or alarm setpoints
  - Desired operational values or goals of the system.

6.3.2 USE PLOTS TO ILLUSTRATE PARAMETER BEHAVIOR OVER TIME

**Recommendation:**

Use plots to illustrate parameter behavior over time. Plots can also illustrate the relationship between two (or more) functionally related parameters and include operational context clues as well.

**Background and explanation:**

Plots are useful for showing the relationship between two factors of a system. The most common type of plot is a simple trend that shows how a value has changed over time. Adding the time dimension to a parameter provides the operator with two pieces of information: current value and past behavior. Accelerator operators commonly monitor parameters plotted over time as past and current behavior is indicative of system performance. Other plots show multiple parameters together over time. The capability to see multiple parameters in comparison can communicate the functional relationship shared by those parameters. A well-designed plot helps operators perceptually process information. When operating the accelerator, it is important to see values change over time, and many plots, like the plot on the left of Figure 25, display this.
The informative value of plots can be enhanced using the three operational contexts discussed earlier (Bennet, Toms, & Woods, 1993). The right side of Figure 25 is a plot that shows two functionally related values as well as the three operational contexts for what is essentially a pressurized water tank. It is immediately clear in the image that both the Y- and X-axis values are trending negatively towards an alarm threshold. Contrast this level of information with the left plot. It is clear that three values are increasing in a mostly uniform stepwise pattern. It is unclear if this behavior is desired or nearing a limit. Because of this, operators rely on training and experience to make a judgement. Using the design principles from the graphic on the right, operators can much more quickly devote their knowledge and experience towards identifying a solution. Fermilab accelerator operators have described how the overview or “comfort displays” in the control room are mostly useful to “knowledgeable people…who know what they are looking for.” A plot that shows telemetry within its operational context can expand knowledge acquisition to novice users as well.

A good example of an ACNET plot with Operational Context 3 (desired operational outcomes) is the bullseye plot shown in Figure 26. This plot shows the beam location on a horizontal and vertical axis with the desired location being the center of the bullseye (a plot that takes on cultural norms can be descriptive). This plot makes it clear where and what the error is and what needs to happen (tune left and right or up and down). This plot can be improved by increasing the readability of the information on the plot. The most important information is the location of the “green dots” on the plot, yet this information takes up the least amount of room on the window.
Plots also serve as references for expected beam efficiency given the current experiment conditions at the accelerator complex. New or degraded equipment also alter the achievable beam quality. Given these variables, operators take “snapshots” of plots that represent the new standard for beam quality. New reference images are maintained in operator E-logs requiring a manual search task from operators. New reference images should be made accessible from the plot itself for ease of use with pertinent meta-data that helps operators ensure the proper reference image is in use. Even more efficient, is incorporating the reference parameters into the plot itself to provide instant operational context. Maintaining the capability to adjust and set beam quality standards for operator reference is important to supporting operator performance.

6.3.2.1 SUMMARY OF PLOTS PRINCIPLES:

- Plots should only show functionally related information
- Plots should display the operational context related to the information within the plot for fast processing and decision-making
- Use trends over time only if the past behavior of the value is important for diagnosis
- Important information should be salient in a plot
- Clearly label the axes
- If important to operator goals, clearly display exact values in such a way to not obscure plot information.

6.4 Display All Control Options on Screen

Recommendation:
The accelerator control system and interface design should clearly distinguish the component relationship, function, and presence of all control options available on screen.

**Background and Explanation:**

Accelerator control room operators currently use a mouse and keyboard to interact with the accelerator control system. Both alphanumeric inputs from the keyboard and click inputs from the mouse are used for navigation and control actions. However, the interaction display provides little visual communication to operators regarding what is clickable, editable, or changeable. Such knowledge is gained through experience and training. There are current instances of “invisible” buttons on the screen that are used during operator tasks but provide zero indication of their presence or function. The new ACORN interface should clearly distinguish the function and presence of all control options available on screen. Figure 27 is an example of a flow indication and control faceplate. It can adjust a setpoint incrementally or by inputting a single value. The setpoint is reflected directly on the control faceplate right next to the current flow reading. Further, the “manual” and “auto” capabilities are located at the bottom, with the currently active setting clearly defined by a blue outline. Having both the noun name and controller identifier situated at the top is also important. All functionality is visible and available from the control faceplate. The function of each piece of the faceplate is also clearly distinguishable.

![Figure 27. Controller example with all control options displayed.](image)
### 6.4.1.1 Designing Clear Functionality in Controls

Table 1. Tradeoffs of relating control functionality to impacted components and systems.

<table>
<thead>
<tr>
<th></th>
<th>Proximally Located</th>
<th>Consistently Located</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Always Visible</strong></td>
<td><strong>Pros:</strong> Natural component and control association</td>
<td><strong>Pros:</strong> Predictable location for quick learning</td>
</tr>
<tr>
<td></td>
<td>Quickest control access</td>
<td>Quick control access</td>
</tr>
<tr>
<td></td>
<td><strong>Cons:</strong> Requires dedicated interface space</td>
<td><strong>Cons:</strong> Requires fewer natural methods to link controller to component</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More dependent on operator capability</td>
</tr>
<tr>
<td><strong>Always Available but Not Always Visible</strong></td>
<td><strong>Pros:</strong> Natural component and control association</td>
<td><strong>Pros:</strong> Predictable location for quick learning</td>
</tr>
<tr>
<td></td>
<td>Allows more room for informative graphics on interface</td>
<td>Allows more room for informative graphics on interface</td>
</tr>
<tr>
<td></td>
<td><strong>Cons:</strong> Risk of hiding needed information behind pop-up control faceplates</td>
<td><strong>Cons:</strong> Risk of hiding needed information behind pop-up control faceplates</td>
</tr>
<tr>
<td></td>
<td>More navigation required to access control actions</td>
<td>More navigation required to access control actions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires fewer natural methods to link controller to component</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More dependent on operator capability</td>
</tr>
</tbody>
</table>

Control buttons (generally clicks) or inputs (generally alphanumeric entries) are those that operators use to adjust a component or system status. This includes but is not limited to adjusting setpoints, clearing alarms, navigating menus, and tuning machines. All control design should have the operational context for controlling, such as current setpoints, values, and predetermined thresholds, viewable while preparing to act. How control buttons are designed and placed on-screen is dependent on the structure of the interface in which it is located.
Therefore, there are tradeoffs to consider when designing controls. Some of the design principles are mutually exclusive. Evaluating tradeoffs should be weighed against what will best support the operator for the task at hand. Controller design can be conceptualized in four categories, as described in Table 1.

### 6.4.1.2 CONTROLLER-COMPONENT RELATIONSHIP

There should be a clear association between the controller and component, or outcome associated with it. The most intuitive method to associate the control function with outcome is by collocating the control with the functionally associated parameters. Placing controls next to the graphics that represent what it controls is a natural way to create an intuitive understanding of what the control function is related to. This approach would follow a proximally located and always visible design approach.

However, some interfaces do not have the space available to fit both information and all controls within the same screen. Therefore, when collocating controls is not feasible, controls can be made always available but not always visible. This method uses a pop-up control faceplate design approach to controls. When a controllable parameter is selected, the control faceplate becomes visible. In this approach, the control faceplate that appears should follow the same guidelines when associating the control function to associated outcomes. Wherever the control faceplate is placed, it must not block visibility to the information used to monitor the resulting outcome information. If possible, also consider if important downstream information is blocked by the control faceplate pop-up.

If both collocating faceplates and not blocking pertinent information cannot be done consistently across system interfaces, another option is available. Consistently locating control faceplates on a designated section of the interface introduces predictability in the display that supports interface usability and faster learning experiences. However, consistently located control faceplates could result in unclear relationships between the component or outcome values of the control actions. This approach relies on clear, explicit links between control and component. One method is to increase the salience of the control-related information while the control is in use. Highlighting the outcome variables associated with the controller creates a visible link to shorten search time and help operators visually switch attention from controller to outcome variables. Additionally, clear and consistent labeling of the control and controller outcome information using natural language naming conventions will help operators confirm the controller faceplate that is open and available is the intended controller.

### 6.4.1.3 DESIGNING CLEAR PRESENCE AND AVAILABILITY OF CONTROLS

Accessing controls should be apparent from viewing the interface. Whether the controls are always visible or always accessible but not always visible, the interface should contain clear
indications of what is a control or how to access controls in no more than a single click. First, the control indication should contain design principles that indicate the potential for operator interaction (e.g., looks like a button). Then, the design should include labeling or iconography that signals that the interaction is (or leads to) the capability to perform control actions.

6.4.1.4 CONTROL DESIGN REFLECTS ACTION

The iconography or control configuration should conform to user population or regional stereotypes. For instance, increasing a parameter on an incremental or sliding scale should involve a left to right or down to up action. For example, if arrow iconography is used on buttons, a situation likely appropriate for instrument tuning, a right or up facing arrow should be used for increasing values, and a left or down arrow used for decreasing values.

6.4.1.5 SUMMARY OF CONTROLS PRINCIPLES

These are adapted from work by Le Blanc et al. (2018):

- Controls should be presented within the operational context (e.g., highlight equipment on overview or mimic display)
- All available or routinely used functions for control of equipment or components should be continuously visible or a maximum of one click away
- Control faceplates should be designed such that the relevant information related to operating equipment is not obscured by the control faceplate
- Where ideal locations for control faceplates cannot be consistently identified, control faceplates should appear on the screen in a manner that minimizes the distance an operator needs to move the control to ensure it does not obscure information
- Controls should provide unambiguous feedback on the status of the equipment and related information, such as current setpoints for control parameters.

6.4.2 USE FEEDBACK TO SIGNAL SYSTEM CHANGES

Recommendation:

The control system interface should support a variety of feedback interactions to inform users of autonomous system changes and user-initiated changes. Feedback messages should be consistent in style and format across the interface.

Background and Explanation:

The purpose of feedback is to inform users of progress concerning accomplishing a task or achieving a system goal. Additionally, efficient feedback informs users if errors have occurred and provides guidance on how to solve them. In control system environments, users guide
many system interactions and initiate control inputs. Feedback is important to communicate whether a control input has been received by a user. Feedback is also important to communicate the status of autonomous actions in relation to user and system goals. There are multiple ways to display feedback in a control system, most of which are visual indications. However audible indications are also used in certain contexts, usually relating to alarm events and abnormal operations.

Whenever an autonomous action has occurred, it’s important for the user to know how that action is affecting the system. Likewise, whenever a control action has been initiated by a user, it’s important for the user to know their control action was received by the system. For example, if an accelerator operator is trying to improve beam output by “knobbing” a control, it’s important for the operator to immediately know that their increase or decrease input was received to avoid exceeding a running condition limit.

Feedback indications range from innocuous indications to explicit notifications that can interrupt a normal workflow. The more risk averse a control action is, the more overt the feedback should be. For example, if beam output is satisfactory but could be improved, an operator will likely tune devices to optimize beam output. As an operator is tuning a device, they are checking the system for feedback regarding their tuning (e.g., increase or decrease). The slight increase or decrease in a device output is a subtle way to provide feedback to the operator that their control input has been received. However, this type of feedback relies on the assumption that the control input will have the desired impact (i.e., visible increase or decrease), and there might be instances where the control input is accepted but fails to result in a visible difference in device data. Fermilab accelerator operators currently experience this, so it’s important to evaluate additional opportunities to indicate operator control inputs have been received.

A simple but effective way to indicate that a control action has been initiated is to display a visible difference of the control before the action and after the action. This is a type of feedback that is very subtle and nondisruptive to normal operations but supports a user’s ability to effectively operate a control system.

Figure 28. “Clicked” indications (bottom buttons have been clicked).
Additionally, feedback is important not only when inputting device controls but also for communicating logistical details within a digital system. An example where logistical feedback is helpful is if a user is trying to tune a device that someone else is currently tuning. If users aren’t permitted to access certain controls, those controls should be visually locked out and inaccessible to nonauthorized users. However, a user might have authorization to tune a device but try to tune it when someone else is already controlling it. When this occurs, it would be helpful to provide a feedback response of “You are not authorized to tune the device at this time.” This way, the user won’t assume their failed attempts to tune are a result of a system malfunction or glitch. This type of feedback is informative and minimally disruptive and informs the operator of why control inputs have been rejected.

Feedback is most crucial when a user is confirming actions that are safety related or for confirming actions that have the potential to disrupt normal operation (e.g., device shut down). A simple but effective way to provide feedback to users when they are initiating a control input that will disrupt normal operation is to prompt users with “Are you sure you want to proceed?” pop-ups. This indicates that the action they are trying to take has more serious consequences than general controls. This also protects the system from accidental control initiations (e.g., sticky keys) and requires the user to confirm their original control input.

![Figure 29. Confirming original control input.](image)

<table>
<thead>
<tr>
<th>6.4.2.1 SUMMARY OF FEEDBACK PRINCIPLES:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• A visual feedback signal should appear each time a user interacts with the control system (e.g., “clicked” indications)</td>
</tr>
</tbody>
</table>
• A visual feedback signal should appear each time an autonomous system change occurs that the user should know about (e.g., time event lapses)
• Feedback signals should be more overt when confirming actions that have the potential to disrupt normal operation and most overt when confirming action-oriented decisions related to safety critical operation (e.g., confirming original control input pop-ups)
• Audible feedback signals should be reserved for abnormal operations or emergency events (e.g., alarms).

7.0 EFFECTIVELY INCORPORATE OPERATOR AIDS

Despite the complexity of an accelerator control system, operators are highly skilled at operating accelerator equipment and maintaining beam quality. However, they are also unnecessarily burdened with an abundance of manual tasking and varying levels of information presented as alerts. Incorporating operator aids, such as automation and alarm guidance, will reduce the physical and mental workload of accelerator operators and free up their attention to focus on their most important responsibilities. The following sections include recommendations for automating manual, repetitive tasks and for reserving alarm indications for events that require immediate operator attention.

7.1 Automate Operator Tasks That Can Be Reasonably Automated

Recommendation:

Integrating automation should involve a review of system functions to identify opportunities where automation frees up operator cognitive resources to focus on higher-level system operation. Anything that can be reasonably automated and will reduce operator burden without compromising the operator's ability to perform the task, should be automated.

Background and Explanation:

Automation is a broad category of tools used to perform system functions or support operators in performing system functions. Automation has been found to improve system performance in efficiency and precision. Improved efficiency is gained by replacing tasks that require greater effort or time by the operators. Precision is gained by relying on computational power to perform complex calculations or data analysis instead of a human operator. Automation can also shift the focus of human operators to other tasks, such as judgement decisions, creative problem solving, or abstract thinking. However, if not properly implemented, automation can be a burden
to operators and instead hinder system performance. Good automation performs its tasks reliably, and its purpose and function are clear to the operator (Nof, 2009). Also, not all tasks are right for automation. Some tasks are better performed by a human operator or designed to combine both operator and automated tools. Properly implemented automation allows operators to freely focus on other tasks without the need to “babysit” the automation. Automation itself is a broad term for many different types of computerized support the system can offer the operator when operating a system.

As the automation integration process begins, it is useful to think of humans operating a system as cycling through a six-stage cognitive processes widely known as the “human information processing model” (O’Hara and Higgins 2010; EPRI 3002004310, 2015; Parasuraman, Sheridan, and Wickens 2000; Sheridan 2002). This model is partially analogous to computer processes, which helps simplify the comparison between automation and humans. Figure 30 shows the six human cognitive processes (top) compared to a simple description of automated processing (bottom) (Parasuraman, Sheridan, & Wickens, 2000). Automation may support the operator during any of the cognitive processes or, in some cases, take over a task entirely.

![Diagram of human information processing model](image)

*Figure 30. The top series is the six cognitive processes human operators iteratively employ while operating, and the bottom series is a simplistic explanation of an automated process.*

The following sections detail characteristics of good task candidates for automation. A description of types of automation that are useful for different types of tasks follows. Lastly is a description of how to design the human-automation collaboration element such that operators have a clear understanding of what automation is doing and how it supports them or how they can employ it effectively.

**7.1.1 KEY CONSIDERATIONS**

Designing for effective human-automation collaboration in a system can be categorized into three elements:

- Selecting tasks to automate
- Selecting the appropriate type of automation for each task selected
- Designing for clear communication between automation and human operators.
7.1.2 SELECTING TASKS TO AUTOMATE IN A SYSTEM

How tasks are allocated to automation has evolved from mostly binary (i.e., wholly automated or wholly manual) (Fitts, et. al., 1951) to graded versions of automation involving collaboration between humans and automation at the intermediary levels (i.e., ten levels of automation (Parasuraman, Sheridan, & Wickens, 2000). Lessons from this evolution are that automation is better at some tasks than humans, and humans are better at some tasks than automation. Also, there will most often be a level of cooperation required between the human and the automated task. As mentioned, when cooperating, it is important that the automation performs reliably so the operator can confidently focus on other tasks. The guidance presented here describes tasks that automation is typically known for performing reliably.

Table 2. Guidance for selecting tasks suited for automation.

<table>
<thead>
<tr>
<th>Type of Task</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repetitive Tasks</td>
<td>Tasks that involve many discrete inputs in a short time</td>
</tr>
<tr>
<td>Routine Tasks</td>
<td>Tasks that are performed on a schedule or following an event, request, or action every time</td>
</tr>
<tr>
<td>Rule Based (i.e., parameter driven) Tasks</td>
<td>Tasks with clear success and fail criterion that lend themselves to if-then programming and involve little to no subjective judgement.</td>
</tr>
<tr>
<td>Long-Term Monitoring Tasks</td>
<td>Alarms are the most common automated application for tasks such as monitoring parameters, equipment status, and timelines</td>
</tr>
<tr>
<td>Large Data Set or Information Abstraction and Aggregation</td>
<td>Tasks that require the gathering of resources across disparate sources or interpret and distill large sets of data for easy and accurate human consumption</td>
</tr>
</tbody>
</table>

Tasks suited for automation may contain one or more of these characteristics. Tasks allocated to some level of automation should first be filtered through these characteristics to determine suitability. Tasks outside these characteristics may require special attention and justification before automating.
Automation should be able to perform its task well and reliably. The aforementioned characteristics help select tasks that automation can typically perform reliably. Once automated task candidates are identified, the most appropriate type of automation should be defined. The three categories of automation that can serve accelerator control room operators are (Electric Power Research Institute, 2015):

- Control automation
- Information and decision-aiding automation
- Administrative task automation.

### 7.1.3 CONTROL AUTOMATION

Control automation acts on the system, adjusting parameters, monitoring for a response, then making further adjustments if necessary. The tasks can be simple (e.g., maintaining a single parameter) or complex (e.g., maintaining a system state that requires multiple parameter maintenance and a model of the relationship between them). Further, control automation can manage short tasks or tasks that may require continual adjustment over extended periods of time. Control automation can take two forms: background automation or operator-initiated automation.

Background automation is always monitoring and acting on the task it is given. This type of automation has full autonomy on a task, performing all four stages of automation continuously and with little to no supervision. Operators rely on the background automation to perform its task without the need to monitor its performance directly. Tasks suited for background automation are typically parameter driven and require frequent adjustments. Operator intervention of background automation should only occur to change task parameters or if the automation is failing.

Operator-initiated automation is a tool for operators to use on tasks that require precision, repetition, or speed. Automation of this type only performs the last processing stage of automation, only supporting operators in the final cognitive process. Operators select when to initiate this automation, and therefore, it is good for one-off tasks, rare situations, or time-consuming tasks. Situations when the desired outcomes change frequently, or special direction is required fit this type of automation. Operators should have the control options necessary to input the success criterion or parameters the automation must target. However, this type of automation can be combined or work with the information and decision-aiding (IDA) automation, which supports the first five operator cognitive processes.

Accelerator control room operators have access to “autotune” features that help align the beam within a specified millimeter range. These autotune features are a good example of when to use operator control automation, for tasks that: require precision, have clearly defined success criteria, and are parameter driven. However, operators reported that not all autotune features
are supplied the data required to reliably perform the task given. One example is the “smoothing program,” which was reported to offer unrealistic solutions requiring operators to take manual control and perform the task themselves. Automation that cannot be made reliable should not be an option in the control room.

### 7.1.4 INFORMATION AND DECISION-AIDING AUTOMATION

IDA automation reduces operator burden in the first five processing stages all the way to decision selection. Automation can collect and analyze contextual information (Human Stage 1 and 2) given a task scenario, saving the operator from manually navigating to information resources and cognitively analyzing what the information means given the context. This diagnostic capability of automation is seen, in part, in alarms. Alarms are a simple automation, monitoring a parameter and signaling the operator if the parameter has violated a constraint. More sophisticated automation may be able to detect a destabilizing system and provide possible causes, enabling the operator to act. Using IDA automation monitoring and information acquisition (Stage 1 and 2) becomes a more passive task for operators and can help guide operators where they are needed most at the time they are needed.

IDA automation is useful for performing error-prone tasks, such as complex calculations or referencing memorized information. Providing operators with this support can lower the burden on their situation interpretation and planning and strategizing processing stages (Stage 3 and 4). Reducing error and operator workload in these stages enables operators to select and implement decisions quicker and with improved efficacy.

Though easier with a highly proceduralized concept of operations, IDA automation can support decision selection by providing different options and predicted outcomes. The operator may assess what the system needs most and select the action to achieve that outcome. As noted, in a more freeform concept of operations where subjective judgement is more common, implementing a useful decision selection aid becomes more difficult. If the selection aid is not useful, it may add extra steps or inadvertently add more cognitive burden to operators during the action selection process.

A good example of IDA automation is already used in ACNET for calculating the desired oscillation trend at the injection closure. The operator judges what the appropriate buffer value is, and the system displays a superimposed “optimum” trend over the current data points. Abstracting the complex calculations required for this prediction to a simple but informative trend line supports the operator situation assessment. This capability may also be applied to areas where control automation is less well suited (e.g., cyclotrons). Providing a forecasted outcome to a judgement decision can increase operator precision and reduce their tuning time.

### 7.1.5 ADMINISTRATIVE AUTOMATION
Administrative task automation performs certain administrative tasks, such as recording data, updating databases, logging actions, or preparing and sending messages. Many systems do this already, though often the logs are not stored in a format usable by operators or require a great amount of effort to extract useful information. Administrative automation tasks are usually simple or tedious tasks that have less impact on the system but have great value to either personnel safety, system audits, or reference sources for operators. It is therefore important that the right information is captured in such a way that it is discoverable and highly usable by those that may require the information later. Typically, the benefit of administrative automation is to save operator time, improve data collection accuracy and consistency, and develop a usable resource.

The downtime log is a likely function that administrative automation can perform well. The log serves to record the duration and the reason the beam was lost throughout the life of an accelerator. Neglecting to log downtime while starting up an accelerator was reported as one of the most common mistakes made by control room operators. It is important enough that operators redundantly record downtimes in the E-log. The function is important, routine, has clear rules when to initiate, and requires collecting all relevant information regarding the downtime. These characteristics make the downtime logging task a priority candidate for administrative automation.

7.1.6 MAINTAINING OPERATOR CAPABILITIES AND SKILLS AFTER IMPLEMENTING AUTOMATION

Automation should be integrated such that operators can easily understand what the automation is trying to accomplish, how it has derived its solutions, and what it is communicating. Automation serves, in part, to lower the burden on operator cognitive processes. Poorly designed automation will have the opposite effect due to replacing a difficult task with a difficult to understand tool performing the task. Interactions with automation should follow some basic criteria to support human-automation collaboration. Insights derived from Green, Ribarsky, and Fisher (2008) provide guidance for designing automation from the perspective of the human cognition model:

- IDA automation should present information and suggestions within the context of the task displaying relationships between the information shown and the task at hand
- Allow for direct interaction with supporting automation, which should not cause the operator to lose sight of the relevant information to the task or navigate away from a page or data view
- Interaction methods should be obvious, feel intuitive, and not require the user extra time or thought to understand how to interact with the automation aid.

7.1.7 AUTOMATION CONCLUSION
A key component to successful automation is using automated tools that are reliable and useful to the operator. Certain task characteristics lend themselves well to developing reliable useful automation. Once a task is identified as a candidate for automation, the proper type of automation aid is selected. A useful model to follow when selecting the type of automation aid is the human and automation information processing model (O’Hara and Higgins 2010; EPRI 3002004310, 2015; Parasuraman, Sheridan, and Wickens 2000; Sheridan 2002). This view can be used to determine what about a task is most difficult to the operator and what type of automation may be used to best reduce or eliminate that difficulty. Lastly, automation aids must be designed into the system such that they support operators and avoid causing more difficulty than the previously manual task did in the first place. Following the three principles discussed here will help avoid unusable or confusing automated aids.

7.2 Design Alarms to Effectively Capture and Guide Operator Attention

**Recommendation:**

Alarms should be reserved for events that require immediate operator attention. Maintaining alarm integrity may require adjusting alarm setpoints to fit current operational expectations. Further, alarm sets representing specific operational tasks known to trigger expected alarms can alleviate unneeded alarms. Replacing less important alarms with alerts, notifications, or clear graphic and interface design will make operators aware of situations approaching an alarm event. Providing clear information and navigation options with the alarm event will help manage the situation.

**Background and Explanation:**

Alarms, strictly defined, should only exist to make operators aware of an approaching system state with potential to harm equipment, the larger system, or put the environment or human life in danger that requires immediate operator intervention. Reserving alarms for only this purpose retains their integrity for prompting the appropriate operator response without the operator having to judge the importance or relevance of an alarm to the current situation. False alarms or less urgent alarms increase operator workload and require experienced judgement to determine the necessary response urgency.

Many process industries use alarms as notifications, confirmations of system status, or to monitor expected changes when performing work. This ad hoc approach to alarms diminishes the urgency alarms should be reserved for. It is usually a result of having poorly designed interfaces that should be providing such information already. Also, as sensor and computing technology advance, the opportunity to automatically calibrate alarms to system statuses and expectations should be used.
7.2.1 CATEGORIZING INFORMATION

System statuses that do not meet the alarm definition can be classified into other categories. Table 3 shows a potential categorizing system for other information important to the operator but should not be considered an alarm. Fermilab currently has a color scheme for the alarms that are mapped to the Table 3 categories. The alarm colors are frequently used elsewhere in ACNET for a variety of reasons. Alarm colors should be reserved for their alarm status and nothing else.

Table 3. Categorize operator notifications based on requisite urgency.

<table>
<thead>
<tr>
<th>System or Machine Status</th>
<th>Operator Action Required</th>
<th>No Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal</td>
<td>Alarm (red, yellow acknowledged)</td>
<td>Alert (blue)</td>
</tr>
<tr>
<td>Expected</td>
<td>Prompt</td>
<td>Process Variables (green)</td>
</tr>
</tbody>
</table>

The alarm category should be reserved for abnormal events that require immediate operator action. Presenting these alarms through multiple sensory cues is important and commonly done through visual and auditory indications. Fermilab already uses audible indications with English language cues to provide clear information regarding the alarm. Audible indications are intrusive and should therefore be reserved for interrupting operators from their task at hand to reorient them to an immediate need.

The audible indications should communicate the need for operators to refer to the alarm screen for information on system needs. The alarm screen should be reserved for those alarms that require attention and avoid presenting all system statuses to avoid clutter that may increase operator search time through distracting, unimportant information. A non-alarm should indicate expected system status already. Using the yellow color scheme for acknowledged alarms can be useful to indicate the problem is being addressed by an operator at which time the audible cue should cease to improve the salience of other incoming audible alarms. Once addressed, the acknowledged yellow alarm should vanish, indicating the need was addressed.

The other categories should be addressed within the operator interfaces. The operator HSI should communicate alerts (Fermi blue) if information is displayed along with relevant operational context. IDA automation should help direct the operator’s attention to process variables demonstrating behavior that may lead to an alarm event.

Prompts are a direct product of IDA automation (see section 7.1.4). IDA automation may suggest operator actions based on system goals to maintain the desired system status within normal operating conditions. Expected process variables should be easily observed on the
interface using principles of data aggregation to batch expected process variable statuses for the rapid determination of system health.

7.2.2 ALARM MANAGEMENT

7.2.2.1 BYPASSING ALARMS

The current Fermilab system allows for bypassing alarms, which takes an alarm out of an alarm state and brings it back online at the operator's discretion. A bypassed alarm could remain bypassed even after it is needed again, creating the potential for a failure to respond and the beam loss. Relying on operator memory or detailed briefings and action logging is a complex solution that burdens operator responsibility and relies on memory.

The updated control system should not require alarm bypassing. Operators report using the bypass function because the alarm is expected. Operators have described the expected alarms as a byproduct from tuning a machine known to often deviate further out of tolerance before improving beam quality. This behavior may be an artifact of poor control design, interface design, or actions better handled by an automated tool. Either way, an advanced alarm system can be informed of operator behavior and adapt to the current system goals, adjusting alarm setpoints accordingly.

7.2.2.2 FLIPPING BITS

Flipping bits is understood as the long-term alarm bypass. For instance, bringing a back-up pump online to perform maintenance on a primary pump will set off alarms for being put online and offline respectively. “Flipping the bits” manually reverses the alarm setpoint for the pumps so neither action set off an alarm for expected operations. It is reprogramming the alarm point for a machine or equipment that is expected to remain in an alarm status over a period of days or weeks. This behavior should be done automatically from a lock-out-tag-out action, or via the interface, and reversed automatically once the tag has been removed.

Operational scenarios that trigger sets of expected alarms can be anticipated. Creating a matching alarm setpoint condition that resolves multiple expected alarms before initiation can keep operations focused on monitoring for abnormal conditions that may arise during the operational scenario. Often, the argument is made that expected alarms support operators in determining that the system is behaving as planned. While this may be true in situations where operators are faced with poor interfaces and a lack of alternate indication, the information indicating proper system performance should be clear and obvious from the interface. Offloading this function from alarms reserves their integrity in clearly indicating an adverse condition and eliciting the appropriate operator response quickly and efficiently.
Bypassing and flipping bits are cited as a means to avoid crowding the alarm screen with unhelpful alarms. If too many alarms come in, alarms accumulate on a separate list that must be opened manually. A small indicator says there are more alarms coming in without displaying those alarms. This issue can be solved by automatically adjusting alarm setpoints based on system operation. It can also be managed by removing “expected status” indications from alarm screens, such as in Figure 31, showing the “Booster Tunnel Alarm Page.” There are 16 statuses shown, of which only one is in an alarm state. The remaining 15 statuses should be obvious by viewing a system overview of the Booster tunnel system, reserving the alarm page for statuses that require operator attention.

![Figure 31. Booster tunnel alarm page.](image)

### 7.2.3 EMBEDDED ALARMS

Enhanced operator awareness is possible through embedded alarms within an interface. As an operator monitors a system or is guided to a system based on an audible cue from the alarm page, the status in alarm state could appear directly on the HSI, directing the operator to the problem. This method is most effective after applying “dull screen” and saliency principles.

### 7.2.4 SUMMARY OF ALARM PRINCIPLES

- Alarms should be used to identify abnormal operating conditions and only interrupt operators when there is an immediate action that they must take
• Alarms should be presented alongside guidance for operators on appropriate actions that should be taken in response to the alarm
• Alarms should not be used to identify normal operating conditions
• Alarm setpoints should reflect the current operating condition of the control system under routine operation
• Alarm limits should be flexible to accommodate accelerator experiments (i.e., no “flipping bits”)
• Parameters or equipment that are in an alarm state should highlighted on any overviews or system mimics
• Alarms should not be used for information-only alerts as they do not require an immediate operator action or provide operationally relevant information
• Alarms presented on a list should be prioritized and should provide the operator with easy methods of searching and sorting based on priority.

8.0 FURTHER CONSIDERATIONS

This report describes the foundational human factors design principles needed to construct user interfaces for an accelerator complex. These high-level principles are intended to serve as design objectives that apply generally across all accelerator control system interfaces. These principles will be mapped to a style guide, which will include very detailed design guidance based on requirements for accelerator operation.

The main concepts introduced in this report are heavily geared towards presenting information in an intuitive manner that is easily recognizable and actionable by operators. However, operators are not the only users interacting with the accelerator control system, and future work should describe specific examples of interface design for additional users. Therefore, a next step in this effort is to perform an extensive review of additional users (i.e., machine experts and engineers) to extrapolate their mental models to develop specific examples and potentially unique philosophies.

Additionally, future work should also describe other levels of the proposed IA. Although an IA extends beyond the user interfaces, the content of this report primarily focused on the top (the HSI) level of the IA. However, human factors considerations of lower levels included in the information hierarchy, such as process-level topics and hardware-level topics, are also needed. The importance of design and process consistency exceeds the HSI level, and the philosophies introduced in this report should be applied to additional levels of the IA. Furthermore, the interdependencies between levels and the implications of human performance concerning those interdependencies should also be considered. For example, how data is captured on a
hardware level can sometimes unintentionally constrain how that data is displayed and interpreted on an interface level. A next step in this effort is to identify any interdependencies that might constrain user interface usability and develop creative solutions to ensure implementation of intuitive processes for each level of the IA.
REFERENCES