Constraint

A method of limiting the actions that can be performed on a system.

Constraints limit the possible actions that can be performed on a system. For example, dimming or hiding options that are not available at a particular time effectively constrains the options that can be selected. Proper application of constraints in this fashion makes designs easier to use and dramatically reduces the probability of error during interaction. There are two basic kinds of constraints: physical constraints and psychological constraints.¹

Physical constraints limit the range of possible actions by redirecting physical motion in specific ways. The three kinds of physical constraints are paths, axes, and barriers. Paths convert applied forces into linear or circular motion using channels or grooves (e.g., scroll bar in software user interfaces). Axes convert applied forces into rotary motion, effectively providing a control surface of infinite length in a small space (e.g., a trackball). Barriers absorb or deflect applied forces, thereby halting, slowing, or redirecting the forces around the barrier (e.g., boundaries of a computer screen). Physical constraints are useful for reducing the sensitivity of controls to wanted inputs, and denying certain kinds of inputs altogether. Paths are useful in situations where the control variable range is relatively small and bounded. Axes are useful in situations where control variable is limited, or the control variables are very large or unbounded. Barriers are useful for denying unwanted actions.

Psychological constraints limit the range of possible actions by leveraging the way people perceive and think about the world. The three kinds of psychological constraints are symbols, conventions, and mappings. Symbols influence behavior by communicating meaning through language, such as the text and icon on a warning sign. Conventions influence behavior based on learned traditions and practices, such as “red means stop, green means go.” Mappings influence behavior based on the perceived relationship between elements. For example, light switches that are close to a set of lights are perceived to be more related than switches that are far away. Symbols are useful for labeling, explaining, and warning using visual, aural, and tactile representation—all three if the message is critical. Conventions indicate common methods of understanding and interacting, and are useful for making systems consistent and easy to use. Mappings are useful for implying what actions are possible based on the visibility, location, and appearance of controls.²

Use constraints in design to simplify usability and minimize errors. Use physical constraints to reduce the sensitivity of controls, minimize unintentional inputs, and prevent or slow dangerous actions. Use psychological constraints to impose the clarity and intuitiveness of a design.

¹ The seminal work on psychological constraints is The Design of Everyday Things by Donald Norman, Doubleday, 1988.
² Note that Norman uses the terms semantic constraints, cultural constraints, and logical constraints.

See also Affordance, Archetypes, Control, Errors, Forgiveness, and Mapping.
Control

The level of control provided by a system should be related to the proficiency and experience levels of the people using the system.

People should be able to exercise control over what a system does, but the level of control should be related to their proficiency and experience using the system. Beginners do best with a reduced amount of control, while experts do best with greater control. A simple example is when children learn to ride a bicycle. Initially, training wheels are helpful in reducing the difficulty of riding by reducing the level of control (e.g., eliminating the need to balance while riding). This allows the child to safely develop basic riding skills with minimal risk of accident or injury. Once the basic skills are mastered, the training wheels get in the way, and hinder performance. As expertise increases, so too does the need for greater control.1

A system can accommodate these varying needs by offering multiple ways to perform a task. For example, novice users of word processors typically save their documents by accessing the File menu and selecting Save, whereas more proficient users typically save their documents using a keyboard shortcut. Both methods achieve the same outcome, but one favors simplicity and structure, while the other favors efficiency and flexibility. This tradeoff is standard when allocating system control. Beginners benefit from structured interactions with minimal choices, typically supported by prompts, constraints, and ready access to help. Experts benefit from less structured interactions that provide more direct access to functions, bypassing the support devices of beginners. Since accommodating multiple methods increases the complexity of the system, the number of methods for any given task should be limited to two—one for beginners, and one for experts.

The need to provide expert shortcuts is limited to systems that are used frequently enough for people to develop expertise. For example, the design of museum kiosks and ATMs should assume that all users are first-time users, and not try to accommodate varying levels of expertise. When systems are used frequently enough for people to develop expertise, it is often useful to provide simple ways to customize the system design. This represents the highest level of control a design can provide. It enables the appearance and configuration of a system to be aligned with personal preferences and level of expertise, and enables the efficiency of use to be fine-tuned according to individual needs over time.

Consider the allocation of control in the design of complex systems. When possible, use a method that is equally simple and efficient for beginners and experts. Otherwise, provide methods specialized for beginners and experts. Conceal expert methods to the extent possible to minimize complexity for beginners. When systems are complex and frequently used, consider designs that can be customized to conform to individual preferences and levels of expertise.

See also Constraint, Flexibility-Usability Tradeoff, and Hierarchy of Needs.

Mapping

A relationship between controls and their movements or effects. Good mapping between controls and their effects results in greater ease of use.¹

Turn a wheel, flip a switch, or push a button, and you expect some kind of effect. When the effect corresponds to expectation, the mapping is considered to be good or natural. When the effect does not correspond to expectation, the mapping is considered to be poor. For example, an electric window control on a car door can be oriented so that raising the control switch corresponds to raising the window, and lowering the control switch lowers the window. The relationship between the control and raising or lowering the window is obvious. Compare this to an orientation of the control switch on the surface of an armrest, such that the control motion is forward and backward. The relationship between the control and the raising and lowering of the window is no longer obvious, do the pushing the control switch forward correspond to raising or lowering the window?²

Good mapping is primarily a function of similarity of layout, behavior, or meaning. When the layout of slovenly controls corresponds to the layout of burners, this is similarity of layout, when turning a steering wheel left turns the car left, this is similarity of behavior, when an emergency shut-off button is colored red, this is similarity of meaning (e.g., most people associate red with stop). In each case, similarity makes the control-effect relationship predictable, and therefore easy to use.³

Position controls so that their locations and behaviors correspond to the layout and behavior of the device. Simple control-effect relationships work best. Avoid using a single control for multiple functions whenever possible, it is difficult to achieve good mappings for a one control-multiple effect relationship. In cases where this is not possible, use visually distinct modes (e.g., different colors) to indicate active functions. Be careful when relying on conventions to attach meaning to controls, as different population groups may interpret the conventions differently (e.g., in England, flipping a light switch up turns it on and flipping it down turns it off).

See also Affordance, Interference Effects, Proximity, and Visibility.

¹ Also known as control-display relationship and output-response compatibility.
² The seminal work on mapping is The Design of Everyday Things by Donald Norman, Doubleday, 1988.
Ockham's Razor

Given a choice between functionally equivalent designs, the simplest design should be selected.1

Ockham's razor asserts that simplicity is preferred to complexity in design. Many variations of the principle exist, each adapted to address the particulars of a field or domain of knowledge. A few examples include:

- "Entities should not be multiplied without necessity." — William of Ockham
- "That is better and more valuable which requires fewer, other circumstances being equal." — Robert Grosseteste
- "Nature operates in the shortest way possible." — Aristotle
- "We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances." — Isaac Newton
- "Everything should be made as simple as possible, but not simpler." — Albert Einstein

Implicit in Ockham's razor is the idea that unnecessary elements decrease a design's efficiency, and increase the probability of unanticipated consequences. Unnecessary weight, whether physical, visual, or cognitive, degrades performance. Unnecessary design elements have the potential to fail or create problems. There is also an aesthetic appeal to the principle, which likens the "cutting" of unnecessary elements from a design to the removal of impurities from a solution—the design is a clearer, purer result.

Use Ockham's razor to evaluate and select among multiple, functionally equivalent designs. Functional equivalence here refers to comparable performance of a design on common measures. For example, gives two functionally equivalent displays—equal in information content and readability—select the display with the fewest visual elements. Evaluate each element within the selected design and remove as many as possible without compromising function. Finally, minimize the expression of the remaining elements as much as possible without compromising function.1

See also Form Follows Function, Mapping, and Signal-to-Noise Ratio.

---

1 Also known as Ockham’s razor, the law of parsimony, law of economy, and principle of simplicities. The term “Ockham’s razor” references William of Ockham, a 14th-century Franciscan friar and theologian who purportedly made abundant use of the principle. The principle does not actually appear in any of his extant writings and, in truth, little is known about either the origin of the principle or its deification. See, for example, “The Myth of Ockham’s Razor” by W. M. Thrusfield, Phil. 1996, vol. 27, p. 345-353.


---

The Yamaha Compact Silent Electric Cello is a minimalistic cell with only those portions touched by the player represented. Musicians can hear concert-quality sounds through headphones while creating little external sound, or through an amplifier and speakers for public performances. The cell can also be collapsed for easy transport and storage.

The Taburet M Steeling Stool is strong, comfortable, and stackable. It is constructed from a single piece of molded wood and has no extraneous elements.
Performance Versus Preference

The designs that help people perform optimally are often not the same as the designs that people find most desirable.

Designers and managers often extrapolate from the business maxim “the customer is always right” a mistaken idea that “the user is always right.” This is a dangerous misinterpretation, since in reality what helps a person perform well and what people like are not the same. For example, the Dvorak keyboard is estimated to improve typing efficiency by more than 30 percent, but has failed to rise in popularity because people prefer the more familiar QWERTY keyboard. If you asked people if they would like to be able to type 30 percent faster with fewer errors, most would answer in the affirmative. Despite this, more than 30 years have passed since the introduction of the Dvorak keyboard, and it is still more of a novelty than a practical alternative.

This underscores an important lesson for designers, the reasons people prefer one design to another is a combination of many factors, and may have nothing to do with performance. Is the design pleasing to look at? Does it compete with long-standing designs or standards of use? Does it contribute to the well being or self-esteem of the user? The balance between performance and preference must be taken into careful consideration in the development of the design requirements. If a superbly performing design is never bought or used because people (for whatever reason) do not prefer it to alternatives, the performance benefits are moot. Conversely, if a well-liked design does not help people perform at the level required, the preference benefits are moot.

The best way to balance performance and preference correctly in design is to determine accurately the relative importance of each. While surveys, interviews, and focus groups try to find out what people want or like, they are unreliable indicators of what people will actually do, especially for new or unfamiliar designs. Additionally, people are poor at discriminating between features they like, and features that actually enhance their performance; they commonly prefer designs that perform less well than available alternatives, and incorrectly believe that those designs helped them achieve the best performance.

The best method of obtaining accurate performance and preference requirements is to observe people interacting with the design (or a similar design) in real contexts. When this is not feasible, test using structured tasks that approximate key aspects of the way the design will be used. It is important to obtain preference information in context while the task is being performed, and not afterward. Do not rely on reports of what people say they have done, will do, or are planning to do in the future regarding the use of a design; such reports are unreliable.

See also Aesthetic-Usability Effect, Control, Development Cycle, Flexibility-Usability Tradeoff, and Hierarchy of Needs.

---


---

The QWERTY layout was designed to prevent the jamming of mechanical keys on early typewriters. The Dvorak layout, by contrast, was designed to maximize typing efficiency: it grouped keys based on frequency of use, and positioned keys to promote alternating keypaddles between hands, among other refinements. This resulted in a 30 percent improvement in typing efficiency, and claim to most of the world records for speed typing. Despite the clear advantages of the Dvorak design, QWERTY enjoys the following of generations of people trained on the layout, which in turn drives manufacturers to continue perpetuating the standard. Dvorak wins on performance, but QWERTY wins on preference.
Progressive Disclosure

A strategy for managing information complexity in which only necessary or requested information is displayed at any given time.

Progressive disclosure involves separating information into multiple layers and only presenting layers that are necessary or relevant. It is primarily used to prevent information overload, and is employed in computer user interfaces, instructional materials, and the design of physical spaces.  

Progressive disclosure keeps displays clean and uncluttered and helps people manage complexity without becoming confused, frustrated, or disoriented. For example, infrequently used controls in software interfaces are often concealed in dialog boxes that are invoked by clicking a More button. People who do not need to use the controls never see them. For more advanced users, the options are readily available. In either case, the design is simplified by showing only the most frequently required controls by default, and making additional controls available on request.  

Learning efficiency benefits greatly from the use of progressive disclosure. Information presented to a person who is not interested or ready to process it is effectively noise. Information that is gradually and progressively disclosed to a learner as they need or request it is better processed and perceived as more relevant. The number of errors is significantly reduced using this method, and consequently the amount of time and frustration spent recovering from errors is also reduced.  

Progressive disclosure is also used in the physical world to manage the perception of complexity and activity. For example, progressive disclosure is found in the design of entry points for modern theme park rides. Exceedingly long lines not only frustrate people in line, but also discourage new people from the ride. Theme park designers progressively disclose discrete segments of the line (sometimes supplemented with entertainment), so that no one, in or out of the line, ever sees the line in its entirety.  

Use progressive disclosure to reduce information complexity, especially when people interacting with the design are novices or infrequent users. Hide infrequently used controls or information, but make them readily available through some simple operation, such as pressing a More button. Progressive disclosure is also an effective method for leading people through complex procedures, and should be considered when such procedures are a part of a design.

See also Chunking, Errors, Layering, and Performance Load.