Notes - Design - Human Memory

- Dr Nick Hayward

A brief intro to human memory relative to application and interface design.

Contents

Intro Context in user interaction is an important consideration, and helps to establish an application within a user's short-term memory.

The predominant models for human memory, and ones that we can consider for user interaction, include short-term, or working, memory, and long-term memory.

There is also general consensus that whilst the short-term and long-term aspects of human memory are separate, they are inter-related structures or systems in the brain.

Short-term, working memory Short-term, working memory is akin to a temporary memory store, which is able to hold a limited amount of information. This could include, for example, words, numbers, symbols, and so on related to a user's current thoughts, actions, or processes.

Our working memory, however, decays rapidly, and it is apparent that such memory will often simply be lost when we lose focus or switch to another task. This is where rehearsal and repetition of a required task &c. is useful. It helps to avoid losing such information on a regular basis.

The capacity of working memory is said to be about *seven, plus or minus two* items, according to Miller in 1956. It is for this reason that North American telephone numbers were chosen to be seven digits long for local dialling. It is inherently harder for most individuals to hold more than about seven digits in their mind when they hear a phone number, and then try to write it down.

Changing limited capacity in working memory We can consider working memory a bit further, and in particular that it needs to be freed up in order to replay and rehearse the new information provided in a lesson.

If we consider an analogy with computer memory, in this instant a fixed amount of RAM, we can easily see that when this limited capacity becomes full, processing becomes inefficient. This is also true for the limited capacity of our working memory. When this happens, a user's ability to learn or work is naturally reduced and slowed down.

A common example of this phenomenon is mental arithmetic. As we start to calculate larger and more complicated multiplications or divisions, we need to be able to store and access the intermediate calculations necessary to solve the problem. With working memory, it is difficult to hold even limited amounts of information and process effectively at the same time.

So, interface design and methods that overload working memory naturally makes application learning and usage more difficult. This burden on working memory, which includes information that must be held temporarily and then processed as well, is known as 'cognitive load'.

Methods that reduce cognitive load foster easier user interaction and learning by freeing working memory capacity for the task in hand.

Long-term memory Long-term memory is a more permanent, persistent store, allowing us to save and recall knowledge and memories of experiences for later recall. We can obviously store any facts, and often our brains will store less than useful information for later recall.

Long-term memory also stores procedures and skills, as needed for both cognitive and sensory-motor tasks. We can, in effect, think of long-term memory as an almost permanent memory store, although some data will deteriorate or degrade over time.

We may also experience some sense of false recall, as memory items become confused or combined irregularly.

Act of memorisation Memorisation can be considered, simply, as the act of intentionally committing something from short-term to long-term memory. This will normally be achieved through repetition. Generally, the more frequently we encounter something, the more likely we are to remember it at a later date.

For example, studying involves actively and intentionally re-reading, rehearsing, and practicing something we would like to commit to long-term memory.

However, we also need to be able to store long-term data about important, novel, surprising, and unusual information without the need for such repetition.

The exact nature of how this memory transfer occurs remains, to some degree at least, unknown. However, we believe that such memories are, effectively, stored symbolically. What this means, for us as learners for example, is that we may not remember an exact copy of an event or material. Instead, we create symbolic *hooks* to allow us easier recall of data from the short-term to the long-term memory.

Memory storage and recall Therefore, we tend to store information in logical groupings. Psychologists refer to these as *chunks*. Our memory becomes most efficient when a given chunk is associated with other existing chunks of information within our memory.

These associations are logical connections or relationships between disparate pieces of information.

When we meet people, we will tend to associate the image of their face with their name or their job title &c., which will then be augmented by knowledge about their likes and dislikes, family members, and so on.

Memory recognition and recall Our recall of information, events, and so on from long-term memory is normally triggered by a given prompt or cue. For example, recognising someone in a crowd we create a trigger to recall their name, and associated facts and details.

Naturally, successful recall is never guaranteed, and the more recently the information was memorised or accessed, the more likely we are to be able to recall its data. This is known as the *recency effect*.

It is also often noticeable that we can successfully recall a given piece of information when we've already recalled related information. It's almost as if related material is catalogued by our brain with similar tags and IDs. As we recall one piece of information, our brain is also highlighting related, contextual data for easier recall.

We also sometimes struggle to recall some piece of information, and that information may or may not come to us later on. This recall shows that long-term memory is imperfect and also inaccurate. We can often get the sense of recalling false or hazy data, a type of fuzzy recall kicks in.

Retrieval and transfer of new knowledge and skills It is not sufficient to simply add new knowledge to long-term memory. For success in usage and interaction, newly acquired knowledge needs to be transferred and stored in long-term memory in such a way that allows it to be easily retrieved when needed in a given situation. The retrieval of new skills is essential for the successful transfer of knowledge and experience. Without this retrieval, all the other psychological processes are meaningless.

In effect, it does the user little good to have knowledge successfully stored in long-term memory that cannot be applied later. It's akin to safely storing away a treasured heir-loom, and then forgetting how to retrieve it when you need some extra money, or to show your grandchild many years later.

retrieval and mnemonics To achieve this successful transfer, design and interaction must incorporate the context of the new skill in the examples and practice exercises so the new knowledge stored in long-term memory contains good points of retrieval. It's similar to using a favoured mnemonic to retrieve a formula or date. So, we can think about how to spell 'necessary' or how to calculate the sine, cosine and tangent of an angle. Each may use a common mnemonic to aid future recall of information.

In a real, practical situation, we might ask users to apply their new knowledge relevant to actual scenarios within help or training scenarios. So, for example, a computer technician has learnt how to fix certain hardware and software problems. We then make them test this knowledge by problem solving within role-playing or troubleshooting exercises. We often see this example within games or education applications, where a skill is demonstrated and then the user is asked to practice before moving on to the main application or game.

The point is simply to link or hook this new knowledge to parts of the long-term memory that can later be easily retrieved.

Our brain forgets Often, the less frequently a chunk of information or skill process is accessed from long-term memory, the more likely it is to be forgotten. This is a natural aspect of our brain's memory structure.

Pieces of information relevant to daily routines will be remembered more easily, due to the *recency effect*, whereas facts often studied many years ago will become more hazy or unclear.

However, this sense of forgetting long term information is, of course, not universal. Often, highly developed motor and cognitive skills that can be done with a sense of easy repetition, effectively by rote after much practice, can often still be performed with a surprising degree of competence and familiarity even after years of neglect.

Design considerations So, how do we apply this understanding of human memory to the design of our user interfaces.

Firstly, we need to ensure that our interface is structured to reduce or eliminate the need to memorise and recall things within its structure. Don Norman outlines this concept as the notion of *knowledge in the world* versus *knowledge in the head*.

For example, creating a menu or list of options for our users is a good example of *knowledge in the world*. Our user will be able to view the menu, read and recognise options, and make their selection without the need to memorise or recall related information beyond the basic motor skills to click and select the chosen item. However, if this same option was accessed from a terminal or command line, our user would need to remember the command to perform the given application option. Therefore, our user would now need to recall *knowledge in the head*, thereby increasing the potential for error and application problems.

knowledge in the world vs knowledge in the head So, compare and contrast remembered layout vs common layouts…

The idea between remembering general usage and distinct, formal patterns.

A user knows how to use a keypad, so the specific layout is not as relevant for memory.

- why different layouts - https://electronics.howstuffworks.com/question641.htm

guidance in design If our user requires a task that has a defined sequence of steps, we should guide the user through the task flow step by step. We can achieve this by presenting forms and controls in a logical and sequential order. We might even consider a *wizard* style interface, thereby presenting multiple pages our user can traverse with standard *previous* and *next* buttons. We are, therefore, trying to reduce the amount of navigation details the user needs to remember and recall in order to perform regimented tasks within our application.

icons and names Interface design can also be enhanced with the use of icons and names that are easily recognisable. For example, so our user can easily find these interface elements as they scan a list or menu. Icons can act as clarifying elements if the images themselves represent concrete and recognisable things. We are making it easier for our users to create hooks from the working to the long-term memory.

Again, if the user has to continually memorise and recall a peculiar or abstract icon, or the user must squint and pause unreasonably to recognise an icon, then it defeats the purpose of initially using such graphical representations. If abstract icons or images are still used, then they should be shown together with an accompanying text label.

Naming schemes and patterns in user interfaces are also important, and help users remember and recall information. Arbitrary names are harder to remember and recall than names that accurately describe their representation.

Non-representative naming schemes are not only problematic to learn and recall, but this additional memorisation and recall adds to the user's cognitive burden.

As you might imagine, traditional command line interfaces and applications are particularly bad in this respect, and violate this principle on a seemingly interminable basis. Consider the Unix terminal commands *more* and *less*, used with text files, and this concept becomes more apparent.

n.b. most Unix systems offer a command called `less` for showing the contents of a text file. The name `less` is a play on words; `less` is an enhanced version of another command called `more` . (`more` is a filter command that lets you view a file or other stream of data in a page-by-page fashion; its name stems from the fact that it makes the console pause until you press the space bar to show "more" of the file or stream contents.)

a few remaining considerations… A good help system and search tool allows a user to quickly check and recall lost information. They can quickly reference documentation, check a usage pattern or concept, and then return to the task at hand.

In search and index systems, it's also a good idea to allow users to use variations and synonyms just in case they are unable to recall the exact word or phrase, or even the correct spelling, needed to identify something. For example, *shut* can easily be used as a synonym for *close*.

Try to avoid creating your own terminology for standard UI elements and interaction concepts. Also, try to avoid using abbreviations or acronyms if they are, again, not obvious or standard practice. *WYSIWYG* or *GUI* are good examples of well-known acronyms.

Be consistent in your interface's application of actions and methods. For example, an action should perform in the same manner from one context to another.

Resources

- Card, S.K., Moran, T.P. and Newell, A. *The psychology of human-computer interaction.* Lawrence Erlbaum Associates. 1983.
- Krug, S. *Don't make me think, revisited: A common sense approach to web usability.* 3rd Edition. New Riders. 2014.
- Miller, G. A. *The magical number seven, plus or minus two: Some limits on our capacity for processing information.*. Psychological Review, Vol. 63, Issue 2. PP. 81-97. 1956.
- Norman, D. *The Design of Everyday Things.* Basic Books. 2013.
- Why different layouts - https://electronics.howstuffworks.com/question641.htm