Comp 341/441 - HCI

Spring Semester 2019 - Week 6

Dr Nick Hayward

Human memory - Recap

retrieval and transfer of new knowledge and skills

- not sufficient to simply add new knowledge to long-term memory
  - *new knowledge needs to be easily retrieved in context*
- retrieval of new skills essential for successful transfer of knowledge and experience
- knowledge successfully stored in long-term memory
  - *no use unless we may successfully recall later, as required*
- interface design necessarily needs to incorporate context to help retrieval
  - *examples and practice exercises*

Human memory - Recap

retrieval and mnemonics

- interface design necessarily needs to incorporate context to help retrieval
  - *examples and practice exercises*
  - *simple mnemonics as a child*
- mnemonics to help with easier recall
  - e.g. ***N.E.C.E.S.S.A.R.Y***, ***SOHCAHTOA***
- might ask users to apply their new knowledge relevant to actual scenarios
  - *job learning and training scenarios, role play, troubleshooting exercises...*
- often see this example within games or education applications
  - *a skill is demonstrated and then the user is asked to practice*
  - *before moving on to the main application or game*
- link or hook new knowledge to long-term memory

Video - Human memory

Sherlock Holmes' Mind Palace trick

Source - Critical Commons

Human memory

our brain forgets

- less frequently accessed chunks of information or skill processes
  - *more likely to be forgotten*
  - *natural aspect of our brain's memory structure*
- **recency effect** tends to protect daily routines...
- older facts more easily become hazy or unclear
- loss of long-term information is not universal
- highly developed motor & cognitive skills with sense of easy repetition
- some things are simply like **riding a bike**

Video - Human memory

Ten Second Tom

Ten Second Tom from 50 First Dates

Source - YouTube

Design for Memory

design considerations - part 1

- ensure interface is designed to reduce or eliminate need to memorise and recall
  - *interface elements etc within structure*
- Don Norman outlines this concept as the notion of
  - ***knowledge in the world*** *vs* ***knowledge in the head***
- eg: creating menus or lists of options for users is a good example of
  - ***knowledge in the world***
- user will be able to view the menu, read and recognise options, make selection
  - *no need to recall or memorise related information beyond the basics...*
- this same option on the command line requires memory of command...
  - *user would need to recall* ***knowledge in the head***
  - *increases potential for error and application issues*

Design for Memory

design considerations - part 2

- we can guide users through sequenced tasks
  - *provision of defined sequence of steps*
  - *guide user through the task flow step by step*
- present forms and controls in a logical and sequential order
- might even consider a **wizard** style interface
  - *user can navigate multiple pages with standard **next** & **previous** links*
- trying to reduce the amount of navigation details required by the user
- thereby reducing the amount the user needs to memorise and recall

Design for Memory

design considerations - part 3

- interface design enhanced with recognisable icons and names
  - *user can easily find interface elements as they scan a list, menu...*
- icons can act as clarifying elements
  - *icons should represent concrete and recognisable things*
- goal is to make it easier for users to create hooks from working to long-term memory
- user should not have to memorise or struggle to recognise unfamiliar icons
  - *defeats the point of using simpler graphical representations*
- if you use abstract, original icons then add some accompanying text to help the user

Design for Memory

design considerations - part 4

- naming schemes & patterns in UIs are also important
  - *helps users remember & recall information*
  - *arbitrary names are harder to recall than representative names*
- non-representative naming schemes may add to user's cognitive burden
- command line interfaces violate this principle on a regular basis
  - *consider Unix commands **more** & **less***

Design for Memory

design considerations - part 5

- good help system and search tool
  - *allows a user to quickly check and recall lost or forgotten information*
  - *user can quickly reference documentation, check usage pattern or concept...*
- in search and index systems
  - *allow users to use variations, synonyms*
  - *user may not remember the exact term, query, spelling...*
- try to avoid personalised terminology for standard UI elements, interaction concepts
- try to avoid using abbreviations or acronyms unless they are obvious or standard practice
  - *eg: **GUI**, **WYSIWYG** are well known examples...*
- be consistent in your UIs application of actions and methods
  - *eg: an action should perform in the same manner from one context to another*

Video

Progress Bars

TED: How the progress bar keeps you sane
Source - YouTube

Cognitive Load

intro

- consider the physical act of interacting with a computer
  - *using a mouse, keyboard, touchscreen...touching, swiping, shaking*
- physical actions incur a cost of time and effort
  - *varying degrees of effort, both physical and mental*
- cognitive load refers to the mental taxation exerted on a user
  - *whilst performing a given task*
  - *refers to amount of sustained attention and cognitive effort required per task*
- the more complex the task, the higher the level of focused attention
  - *cognitive load will be higher as a result*
- good design strategy to try to reduce a user's cognitive load
- try reducing the amount a user has to think about
  - *general concepts, points of interaction, basic navigation, interface elements...*
- "Don't make me think , revisited: A common sense approach to web usability."
  - *Steve Krug, 2014.*

Cognitive Load

impact of interactions

Cognitive load may be impacted by the following interactions:

- scrolling, navigating, searching within an application
- choosing options such as menus, lists, forms...
- reading instructions, labels, titles...
- switching contexts (eg: switching between windows, tabs, pages...)
- switching visual attention
  - *reading text, then referring to an image, and then back to the text*
- memory recall for a specific ID, name, action, task sequence...
- simply waiting for the system or application to respond...
- recovering from a specific distraction
  - *such as an interruption not relevant to the current task at hand...*

Image - Xerox

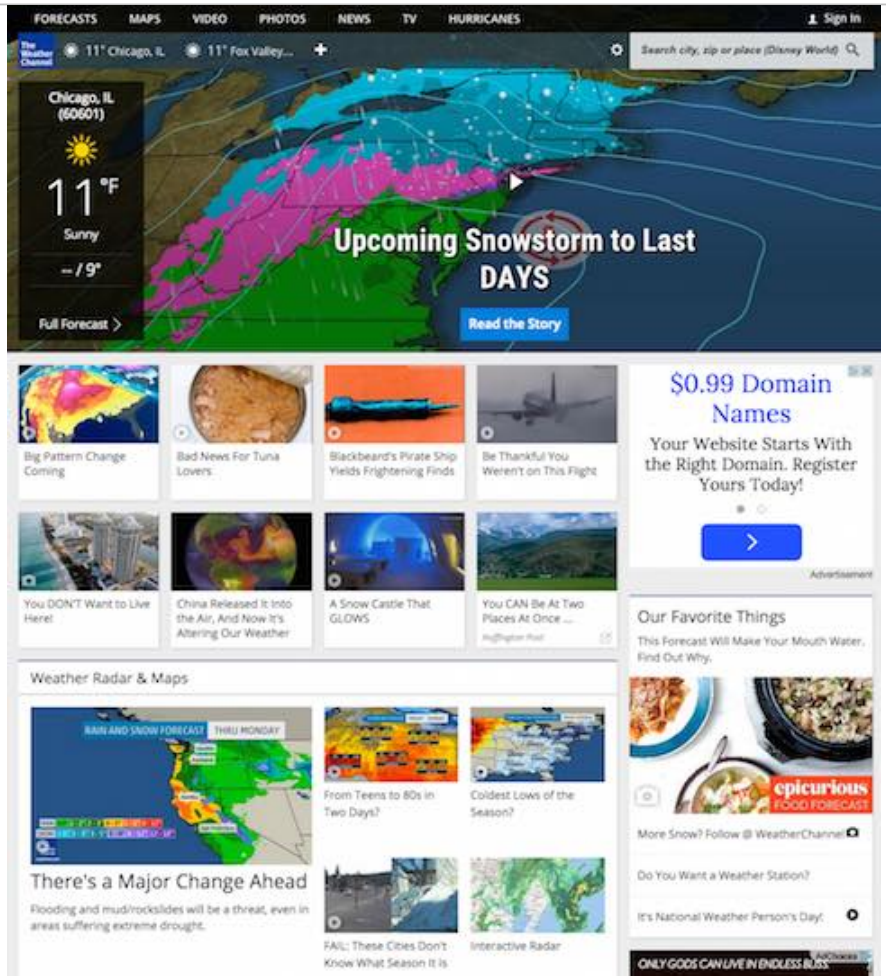
big green button...

Xerox's Big Green Button

Source - Fuji Xerox Printers

Image - Cognitive Load

Weather.com

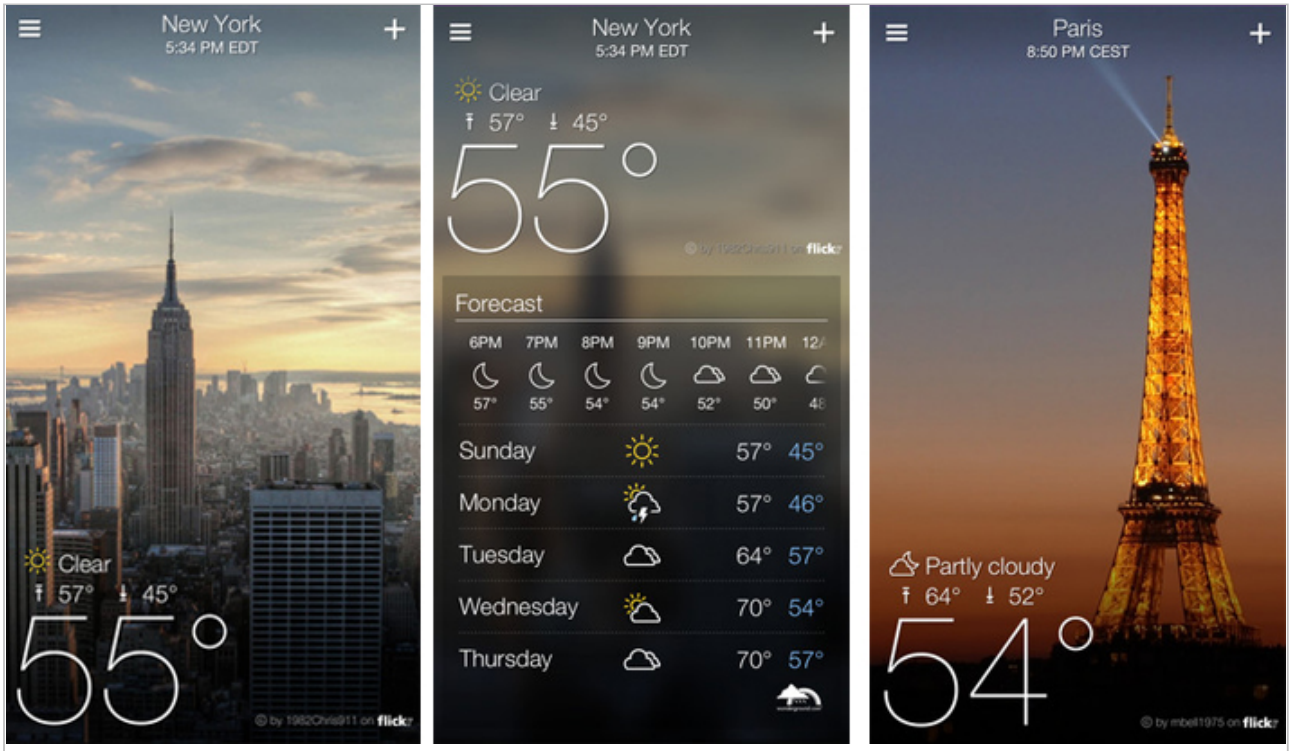


Weather.com

Source - Weather.com

Image - Cognitive Load

Yahoo Weather app



Yahoo Weather App

Source - Yahoo! Weather Mobile App

Cognitive Load

thinking

- reduce cognitive loads by awareness of types of user thinking an app requires
- for example:
  - *working out the next step in a procedure*
  - *using working memory to help complete an ongoing task*
  - *recall of commands, facts, procedures from long-term memory*
  - *memorising commands, facts, procedures etc for long-term memory*
  - *referencing information from another source*
  - *making decisions or considering judgements*
  - *mental integration of information from disparate sources*
    - including research, reference, or simply general peripheral sources...

Video - Cognitive Load

users and interaction - second try...

Filter photographs based on metadata

Source - Adobe Lightroom Tutorials

Cognitive Load

forced, unnecessary thinking

- our goal is not to reduce thinking relative to our application
  - *intellectual thinking different from forced, unnecessary thinking due to poor design...*
- our app should promote and facilitate thinking, and record results where applicable
- our app should try to limit extraneous cognitive load for activities such as
  - *active research activities*
  - *creative development and output*
  - *general problem solving and issue resolution*
  - *reading, note taking, and other general tasks...*
- cognitive load reduced by an app's focus upon
  - *the task in hand, relevancy of UI information and implementation, reduction in extraneous content...*
- reduce interface induced thinking additional to the primary task
  - *better contextual support and research*

Image - Cognitive Load

distraction free

Kindle Paperwhite

Source - Amazon Kindle Paperwhite

Image - Cognitive Load

app's *Big Green Button*

Xerox's Big Green Button

Source - Fuji Xerox Printers

Cognitive Load

quantify cognitive load

- interested in how we can quantify the cognitive load
  - *required by a user for performing a given task*
- better understanding of load issues within our application and interface
  - *helps guide us in apportioning emphasis and control in design*
- for a particular task we can compile a list of actions, steps...
  - *estimate a score (% etc) which represents our understanding of required effort*
  - *total all of the action scores to assign an overall score for the effort required*
  - *evaluate different design options by comparing overall scores...*
- KLM-GOMS model
  - ***Keystroke-Level Model for the Goals, Operators, Methods, and Selection Rules***
  - *Card et al. "The Psychology of Human-Computer Interaction." 1983.*

KLM-GOMS Model

intro

- users divide goals into a series of tasks
  - *each task requiring some initial thought and preparation*
- preparation known as **task acquisition time**
  - *can be very short for simple, routine tasks*
  - *may be much longer, perhaps a few minutes, for more creative, original tasks*
- user will then continue with their chosen task
  - *using a sequence of actions or operations*
- total required time to complete the actions is known as **task execution time**
- total time required to complete task is the sum of
  - ***task acquisition time*** + ***task execution time***
- modified models for mobile devices, such as phones...
  - *eg: Keystroke-level model for advanced mobile phone interaction*

KLM-GOMS Model

usage

Code	Operation	Time (in seconds)
K	Key press & release (keyboard)	Best Typist (135 wpm) = 0.08
		Good Typist (90 wpm) = 0.12
		Avg. Skilled Typist (55 wpm) = 0.20
		Poor Typist (40 wpm) = 0.28
		Typing Random Letters = 0.50
		Typing Complex Codes = 0.75
		Worst Typist = 1.20
P	Point mouse to an object on screen	1.10
B	Button press or release (mouse)	0.10
H	Hand from keyboard to mouse & vice-versa	0.40
M	Mental preparation (operation)	1.20
T(n)	Type string of characters	n x K seconds

wpm = words per minute

Source: Kieras, D. 1993. Wikipedia

KLM-GOMS Model

example

Example implementation - text search including mental operators

Action	KLM-GOMS Code	Time (in seconds)
move mouse to **search** menu	H (hand to mouse)	0.40
	M + P (search menu)	1.20 + 1.10
select **search** menu...	BB (select search menu)	2 * 0.10
click on ***find text*** link...	M + P (find text menu item)	1.20 + 1.10
	BB (select menu item)	2 * 0.10
	H (hand from mouse to keyboard)	0.40
enter search term ***et***	KK (type ***et*** characters)	2 * 0.20 (avg. typist)
click the ***OK*** button	H (hand from keyboard to mouse)	0.40
	M + P (***OK*** button)	1.20 + 1.10
	BB (click button)	2 * 0.10
Total		9.10

BB = double button press to simulate mouse click and release (0.20 seconds)

Reducing Cognitive Load

a few tips and tricks...

- consistent use of icons, labels, names, and general visual presentation
  - *consistency should include design for multiple tasks as well*
- clear navigation for process steps...wizards, paged results etc
- include visual cues and clues...saves users having to remember functionality
- avoid popups except for explicit intervention reasons...warnings, errors etc
- avoid redundancy in content and rendering
- relational material should be organised in close proximity to one another
- identify and remove unnecessary steps
- automate processes, steps where possible
- reduce delays and latency as much as possible...use progress updates, bars
- option for templates, tutorials for new documents in productivity apps etc
- video and audio tutorials often easier to follow and understand than text only
- repetitive user data entry can be avoided
  - *app should not force a user to continually remember such data and information*

flow

Concept of **Flow** by **Mihaly Csikszentmihalyi**

- user's creativity and productivity are high
  - *performance of activity occurs naturally and unconsciously*
- user experiences deep concentration and immersion in their current activity
  - *user is effectively both alert and relatively relaxed*
- living in the moment
  - *sensation of being so engrossed in an activity a user is unaware of the passage of time*
- balancing interest and challenge
- user is confident and exhibits a sense of control over their current situation
- user is working progressively towards achieving a specific goal
  - *eg: in games this might be as simple as getting to the next level*

TED 2004 - Flow, the secret to happiness

Video - Concept of Flow

working memory and the concept of flow

TED 2013 - Peter Doolittle: How your "working memory" makes sense of the world

References

- Card, S.K., Moran, T.P. and Newell, A. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates. 1983.
- Holleis, P. et al. *Keystroke-level model for advanced mobile phone interaction.* CHI' 07. New York, USA. 2007.
- Kieras, D. *Using the Keystroke-Level Model to Estimate Execution Times*. 1993. http://courses.wccnet.edu/~jwithrow/docs/klm.pdf
- Krug, S. *Don't make me think, revisited: A common sense approach to web usability.* 3rd Edition. New Riders. 2014.
- Norman, D. *The Design of Everyday Things*. Basic Books. 2013.