Automating KLM Computation: An Example

- KLM Form Analyzer (KLM-FA) automates KLM computations for web-based forms with customizable user characteristics
  - Assumptions
    - fields filled in order of occurrence
    - number of chars in "standard" named fields
    - use of widgets

http://klmformanalyzer.weebly.com
https://youtu.be/XOTE5Ad3TzQ
Model Human Processor

- $\delta_{WM} 7 \text{ [5~226] secs (no rehearsal)}$
  - Present items, keep user from rehearsing
- $\mu_{WM}$ Working memory capacity (no rehearsal)
  - Present set of letters briefly, then ask users to report ones they see. Always limited, even though they say they see all. ~ 3 chunks

Taking advantage of $\mu_{WM}$

- What3Words (https://what3words.com)
  Divide world into 3m×3m squares and assign each a three-word address
  - Eases communication of locations (emergencies, package delivery, directions,...)
Model Human Processor


- $\mu_{WM} \approx 7 \pm 2$ chunks (with rehearsal)
- George Miller, 1956
- $\mu_{VIS}$ Visual store capacity
- Present $q$ rows of $n$ letters each, followed by a pointer to one row. Then ask what was in the row. If subject gives $m$ of $n$ letters, $\mu_{vis} = (m/n) \times q$ 17 [7~17] letters

Memory Capacity and Decay

- Human Processor v. Chimpanzee Processor

Memory Capacity and Decay

- Human Processor v. Chimpanzee Processor

Try it yourself!


http://www.cambridgebrainsciences.com/play/monkey-span-ladder (Untrained human easy version!)

Seven Stages of Action

D. Norman, 1988

"The basic idea is simple. To get something done, you have to start with some notion of what is wanted—the goal that is to be achieved. Then, you have to do something to the world, that is, take action to move yourself or manipulate someone or something. Finally, you check to see that your goal was made. So there are four different things to consider: the goal, what is done to the world, the world itself, and the check of the world. The action itself has two major aspects: doing something and checking. Call these execution and evaluation."

— D. Norman,
The Psychology of Everyday Things, 1988
**Seven Stages of Action**  
*D. Norman, 1988*

1. Form goal  
2. Form intention  
3. Specify action  
4. Execute action  
5. Perceive world state  
6. Interpret world state  
7. Evaluate outcome

- **Gulf of execution**  
  - Mismatch between what you want to do and what you can do

- **Gulf of evaluation**  
  - Mismatch between what world state tells you and what you want to know
Seven Stages of Action
Norman 1988

- Principles of good design
  - Visible state and action alternatives
  - Good, consistent conceptual model
  - Good mappings between stages
  - Continuous feedback
- Points of failure
  - Inadequate goal
  - Cannot find correct user interface components
  - Cannot execute desired action
  - Cannot understand feedback

Widget-Level Models

- Apply KLM approach not just to low-level actions (key presses and mouse motion), but to interactions with high-level “widgets” (e.g., CogTool)
- Evaluate individual widgets, then make predictions about UIs composed of those widgets
- Develop standard design patterns
Transition Networks

- Show UI states and actions that cause transitions between states
- Can be used to analyze consistency/simplicity

Grammars

- Express structure of a UI
- Can be used to analyze consistency/simplicity
Evaluating UIs through Formal Grammars
P. Reisner, 81

- Evaluated “action language” of two drawing system UIs: ROBART 1 and ROBART 2
  - ROBART 2 was designed to be easier (and tests showed it was)
- Described each language as a formal grammar (modified BNF)
- Showed ROBART 2 had simpler grammar. For example,
  - ROBART 1 had two different ways to select type of object to create (text different—just type at keyboard) and differing numbers of actions for other objects
  - ROBART 2 had only one way to select type of object to create
- Used the grammars to predict how users would perform
  - Robart 2 would be easier to learn/remember
  - A user would take varying amounts of time to learn/remember how to select objects in Robart 1, but not in Robart 2
  - A user would try to treat text the same way as other objects after first learning how to select other object types to create
- Predictions confirmed by analysis of time to learn with documentation, observations of use without documentation, error rate, and questionnaires

Evaluating UIs through Formal Grammars
P. Reisner, 81

- Prior “conventional wisdom”: In order of decreasing priority:
  - Minimize # lowest level action primitives
  - Minimize length of action sequences
  - Minimize # rules
- Reisner showed this order should be reversed to ease learning for naïve users doing nonroutine tasks
Task Action Grammars (TAGs)
S. Payne & T. Greene, 86

- Context-Free Grammar maps tasks to user actions
- TAG consists of
  - Dictionary: List of simple tasks
  - Rule schemata: Grammar for language syntax
- Can be used to analyze
  - Consistency/simplicity
  - Completeness

Dictionary of simple tasks
- Move cursor one char fwd \{dir=fwd, unit=char\}
- Move cursor one char back \{dir=back, unit=char\}
- Move cursor one word fwd \{dir=fwd, unit=word\}
- Move cursor one word back \{dir=back, unit=word\}

Features and values
- \texttt{dir}\{fwd, back\}
- \texttt{unit}\{char, word\}

Rule schemata
- \texttt{task [dir, unit] \rightarrow modifier [dir] + letter [unit]}
- \texttt{modifier [dir=fwd] \rightarrow <ctrl>}
- \texttt{modifier [dir=back] \rightarrow <alt>}
- \texttt{letter [unit=char] \rightarrow "c"}
- \texttt{letter [unit=word] \rightarrow "w"}

Commands
- Move cursor one char fwd \texttt{<ctrl> c} Move cursor one char back \texttt{<alt> c}
- Move cursor one word fwd \texttt{<ctrl> w} Move cursor one word back \texttt{<alt> w}

Is user interface
- Consistent?
- Simple?
- Complete?