ROBERT GORDON UNIVERSITY ABERDEEN

# Cognitive Evaluation

Week 7

# Optimising Interfaces

- To optimise interfaces, we need to minimise the cognitive load!
  - People spend more time in software use every day

- According to latest pools from uSwitch.com, an average adult spends over 30 hours a week browsing the Internet, increasing to 45 hours for young adults i.e. 18-24 years old.

# Optimising Interfaces

- New guidelines for designing usable interfaces such as Web Content Accesibility Guidelines (WCAG) recommend reducing short term memory load.

- Cognitive load, i.e. working memory requests during problem solving, reasoning, or thinking or when moving a limb such as a hand, may affect users' general satisfaction and performance when completing tasks such as browsing an internet web application[1].

- In this paper, Schmutz et al. addresses some of points mentioned above.

[1] This includes motor functions which are only possible after some cognitive processes

# Optimising Interfaces

- Optimisation often is based on human testing, however it is:
  - Time consuming
  - Expensive
  - Slow processing
  - Many participants must be involved
  - Many tests must be completed by participants
  - Difficult to integrate into software development

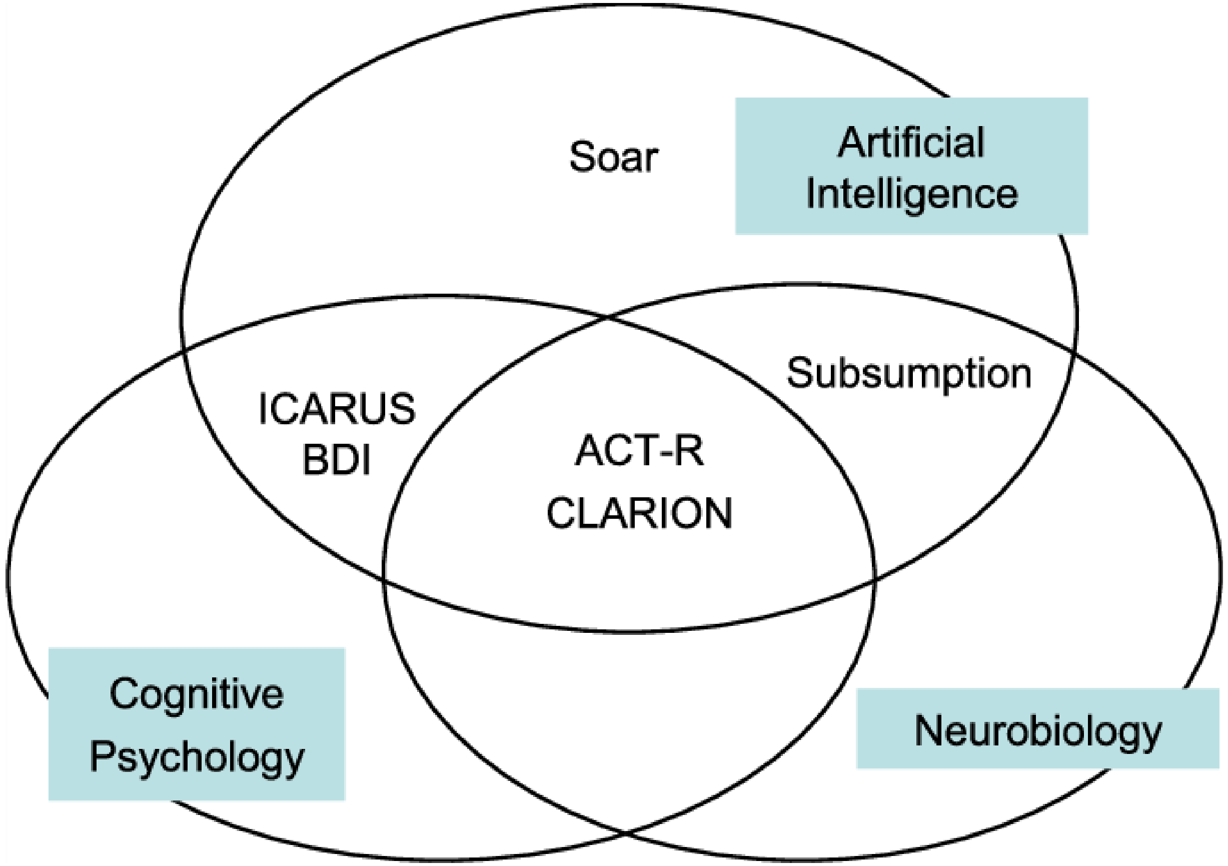
# Optimising Interfaces

- There are other methods. Using Cognitive Architectures provides:
    - Fast processing
    - No participants are involved
    - Not expensive
    - Easy to integrate into software development
    - Difficult to model human motor and cognitive functions

# Optimising Interfaces

- Methods to model user interaction have been researched for many decades.
- In 1983, Stuart Card developed the GOMS (Goals, Operators, Methods, and Selection) rules.
    - He also developed another form of GOMS called Keystroke-Level Model (KLM) which is simpler, easier to use and more adapted to HCI i.e. computer interfaces.
- KLM allows for time prediction for completion of tasks. These widely used modelling systems reduce a user's interaction to its basic components i.e. involving physical, cognitive and perceptual actions.
- They can be analysed and optimisation can be achieved through them.

# Cognitive Architectures


Soar
Artificial Intelligence
Subsumption
ICARUS
BDI
ACT-R
CLARION
Cognitive Psychology
Neurobiology

# SOAR

- First mentioned by Allen Newell in 1972, SOAR was the first attempt (1983) to produce an Unified Theory of human cognition.
- It is capable of simulating human learning, reasoning, skill acquisition and mental processes.
- SOAR is widely used in research involving the understanding of human cognition.

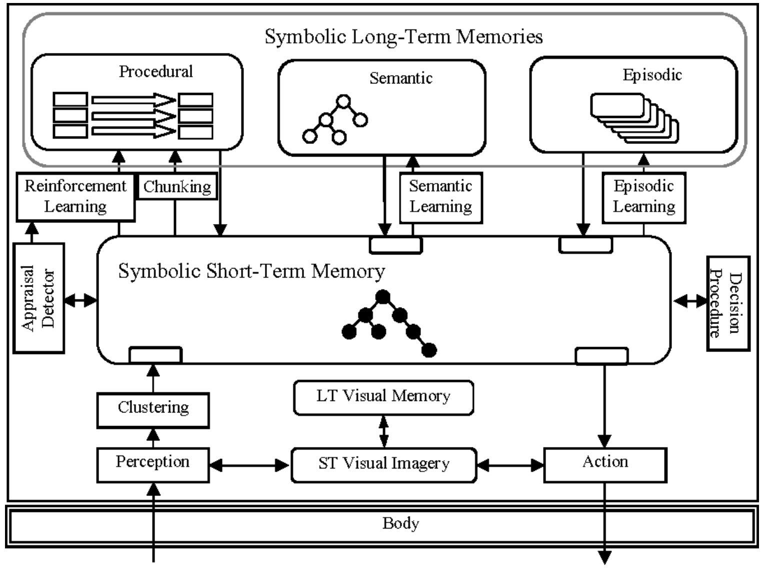


Figure 3: Soar 9

The Soar User's Manual

Version 9.6.0

https://soar.eecs.umich.edu/downloads/Documentation/SoarManual.pdf

# ACT-R: Adaptive Control of Thought—Rational

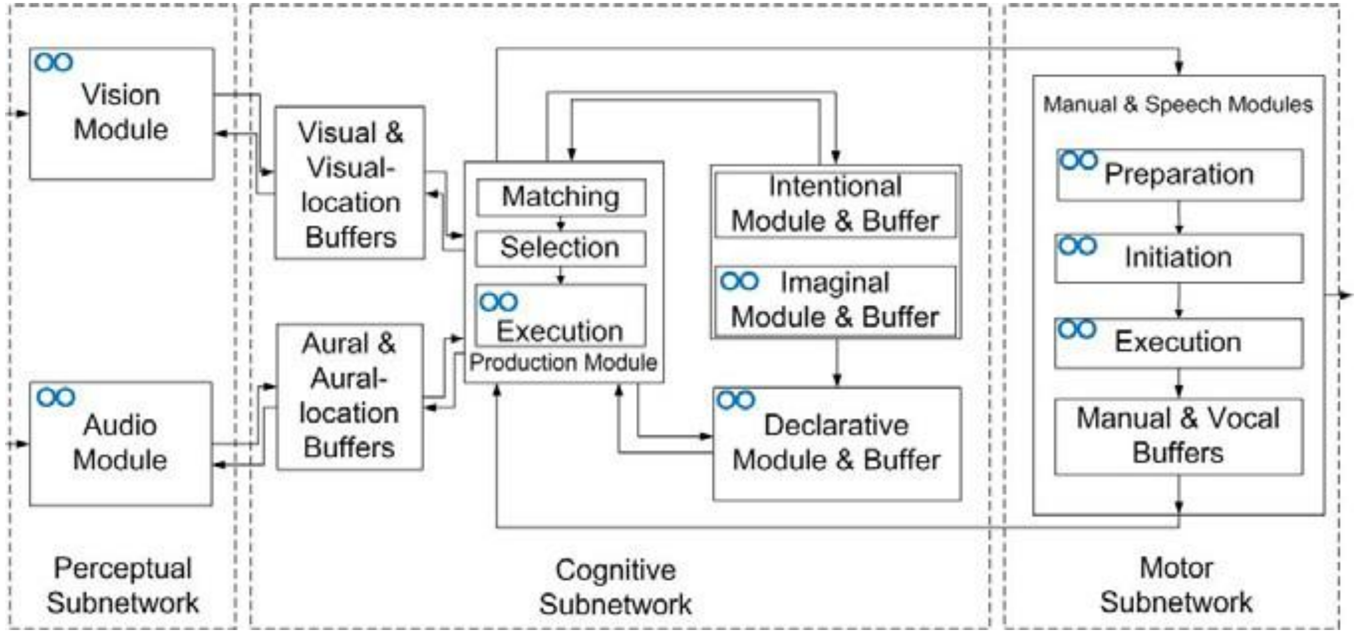
- Inspired by the work of Newell et al.[1] and Lebiere et al., who later devoted their research to create a universal theory of cognition which underpin ACT-R.
- Later, this framework was completed with the addition of human motor functions to become ACT-R/PM.
- ACT-R uses the Fitt's law as part of its latency calculations[2].
- ACT-R/PM is used in many research areas, such as military and medical applications.

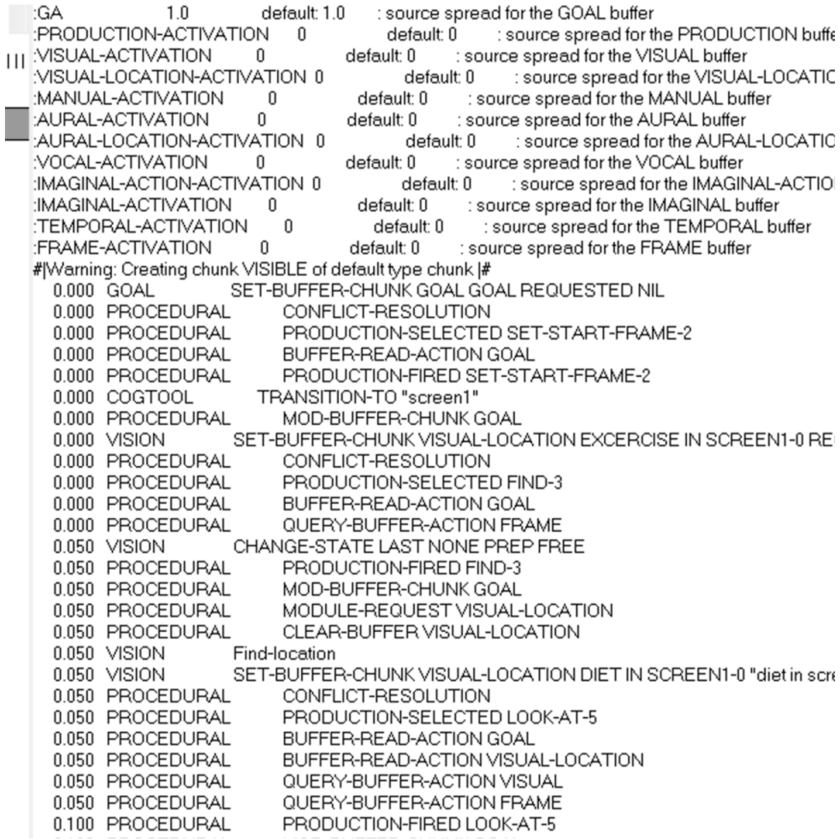
[1] Started with HAM, followed by ACT, he then went on to develop ACT-R which then was followed by ACT-R/PM
[2] In short, smaller and further is the target object to go to and higher will be the timing

# ACT-R/PM

- ACT-R is a cognitive architecture: a theory for simulating and understanding human cognition as well as simple motor functions. It also contains audible and visual functions that are well developed.
- Researchers working on ACT-R strive to understand how people organise knowledge and produce intelligent and rational behaviour.
- As research continues, ACT-R evolves ever closer into a system which can perform the full range of human cognitive tasks i.e. capturing in granular detail the way we (human) perceive, think about, and act on the world the way we do.
- Theory about how human cognition works based on assumptions on numerous facts derived from psychology experiments.
- ACT-R is like a programming language; a framework where researchers create models (i.e. programs) that are written in ACT-R and that, beside incorporating the ACT-R's view of cognition embedded in this architecture, add their own assumptions about the particular task.
  - These assumptions can be tested by comparing the results of the model with the results of people doing the same tasks.
  - By "results" we mean the traditional measures of cognitive psychology i.e. time to perform the task and accuracy in the task etc.

# ACT-R/PM Example





```
:GA                    1.0          default: 1.0     : source spread for the GOAL buffer
:PRODUCTION-ACTIVATION      0              default: 0        : source spread for the PRODUCTION buffe
:VISUAL-ACTIVATION        0              default: 0        : source spread for the VISUAL buffer
:VISUAL-LOCATION-ACTIVATION  0              default: 0        : source spread for the VISUAL-LOCATIC
:MANUAL-ACTIVATION         0              default: 0        : source spread for the MANUAL buffer
:AURAL-ACTIVATION          0              default: 0        : source spread for the AURAL buffer
:AURAL-LOCATION-ACTIVATION  0              default: 0        : source spread for the AURAL-LOCATIC
:VOCAL-ACTIVATION          0              default: 0        : source spread for the VOCAL buffer
:IMAGINAL-ACTION-ACTIVATION  0              default: 0        : source spread for the IMAGINAL-ACTIO
:IMAGINAL-ACTIVATION        0              default: 0        : source spread for the IMAGINAL buffer
:TEMPORAL-ACTIVATION        0              default: 0        : source spread for the TEMPORAL buffer
:FRAME-ACTIVATION          0              default: 0        : source spread for the FRAME buffer
#|Warning: Creating chunk VISIBLE of default type chunk |#
   0.000  GOAL               SET-BUFFER-CHUNK GOAL GOAL REQUESTED NIL
   0.000  PROCEDURAL             CONFLICT-RESOLUTION
   0.000  PROCEDURAL             PRODUCTION-SELECTED SET-START-FRAME-2
   0.000  PROCEDURAL             BUFFER-READ-ACTION GOAL
   0.000  PROCEDURAL             PRODUCTION-FIRED SET-START-FRAME-2
   0.000  COGTOOL             TRANSITION-TO "screen1"
   0.000  PROCEDURAL             MOD-BUFFER-CHUNK GOAL
   0.000  VISION             SET-BUFFER-CHUNK VISUAL-LOCATION EXCERCISE IN SCREEN1-0 RE
   0.000  PROCEDURAL             CONFLICT-RESOLUTION
   0.000  PROCEDURAL             PRODUCTION-SELECTED FIND-3
   0.000  PROCEDURAL             BUFFER-READ-ACTION GOAL
   0.000  PROCEDURAL             QUERY-BUFFER-ACTION FRAME
   0.050  VISION             CHANGE-STATE LAST NONE PREP FREE
   0.050  PROCEDURAL             PRODUCTION-FIRED FIND-3
   0.050  PROCEDURAL             MOD-BUFFER-CHUNK GOAL
   0.050  PROCEDURAL             MODULE-REQUEST VISUAL-LOCATION
   0.050  PROCEDURAL             CLEAR-BUFFER VISUAL-LOCATION
   0.050  VISION             Find-location
   0.050  VISION             SET-BUFFER-CHUNK VISUAL-LOCATION DIET IN SCREEN1-0 "diet in scre
   0.050  PROCEDURAL             CONFLICT-RESOLUTION
   0.050  PROCEDURAL             PRODUCTION-SELECTED LOOK-AT-5
   0.050  PROCEDURAL             BUFFER-READ-ACTION GOAL
   0.050  PROCEDURAL             BUFFER-READ-ACTION VISUAL-LOCATION
   0.050  PROCEDURAL             QUERY-BUFFER-ACTION VISUAL
   0.050  PROCEDURAL             QUERY-BUFFER-ACTION FRAME
   0.100  PROCEDURAL             PRODUCTION-FIRED LOOK-AT-5
```

# ACT-Simple

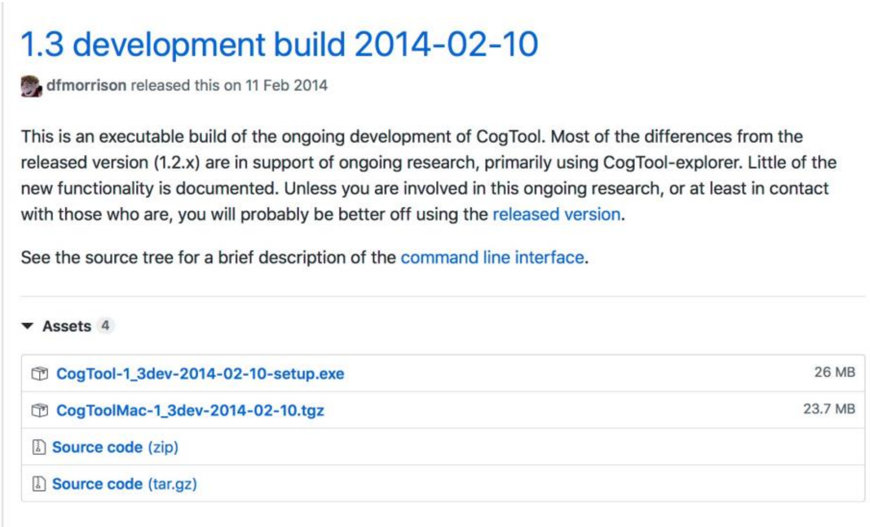
- Developed by Salvucci et al. in 2003[1] merges the concepts of KLM and ACT-R in one environment.
- User interaction is analysed and produces KLM rules which are then translated into an ACT-R model to simulate human cognitive, motor and visual functions for a scenario of tasks. It returns a latency (timing) for this series of tasks.
- It is much easier than modelling KLM and ACT-R, but is still a challenging environment to use.
- In 2004, Bonnie John created a prototyping interface called **CogTool** in which user interaction is modelled by demonstration using ACT-Simple.

[1] ACT-Simple is widely used in research for both hardware and software interfaces

# CogTool

# CogTool

- CogTool is relatively simple to use and is based on the idea of using sketches and storyboards to demonstrate the user interaction.
- It was available for free originally at the Carnegie Mellon University[1] but can now be downloaded from GitHub
  - https://github.com/cogtool/cogtool/releases



1.3 development build 2014-02-10

dfmorrison released this on 11 Feb 2014

This is an executable build of the ongoing development of CogTool. Most of the differences from the released version (1.2.x) are in support of ongoing research, primarily using CogTool-explorer. Little of the new functionality is documented. Unless you are involved in this ongoing research, or at least in contact with those who are, you will probably be better off using the released version.

See the source tree for a brief description of the command line interface.

▾ Assets 4

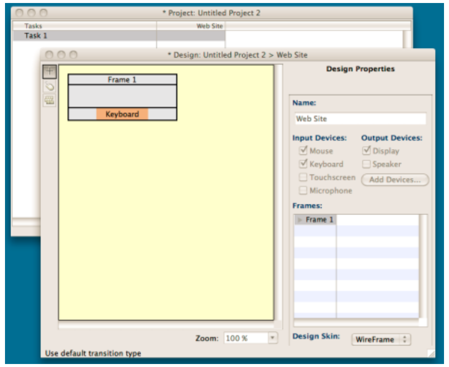
| | |
|---|---|
| CogTool-1_3dev-2014-02-10-setup.exe | 26 MB |
| CogToolMac-1_3dev-2014-02-10.tgz | 23.7 MB |
| Source code (zip) | |
| Source code (tar.gz) | |

[1] Carnegie University is well-known for its research in human cognition as well as developing ACT-R.

# CogTool Project Flow

First we create a project in CogTool. The scenario we wish to have is:

- Look at flower picture
- Look at river picture
- Click on the “Enter Gallery” link



* Project: Untitled Project 2

| Tasks | Web Site |
|---|---|
| Task 1 | |

* Design: Untitled Project 2 > Web Site

Design Properties

Name: Web Site

Input Devices: ☑ Mouse ☑ Keyboard □ Touchscreen □ Microphone

Output Devices: ☑ Display □ Speaker

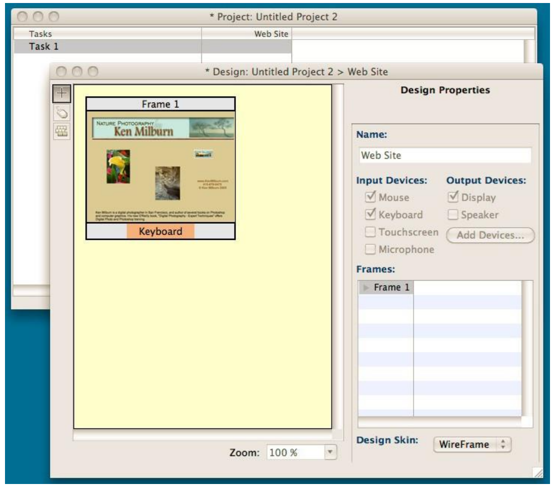
Frames: Frame 1

Zoom: 100 %

Design Skin: WireFrame

Use default transition type

The background image has been set to a screenshot of the web page we want to test



* Project: Untitled Project 2

| Tasks | Web Site |
|---|---|
| Task 1 | |

* Design: Untitled Project 2 > Web Site

Design Properties

Name: Web Site

Input Devices: ☑ Mouse ☑ Keyboard □ Touchscreen □ Microphone

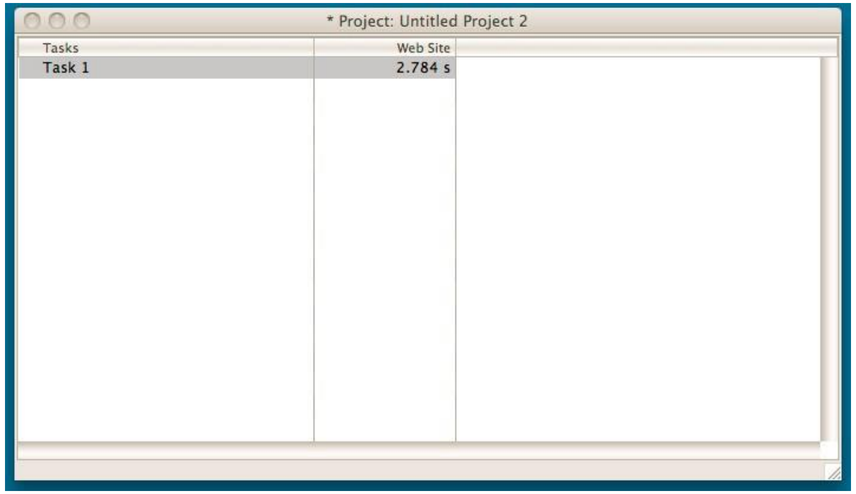
Output Devices: ☑ Display □ Speaker

Frames: Frame 1

Zoom: 100 %

Design Skin: WireFrame

Once the user interaction has been setup in CogTool, we can compute the simulated time to achieve the tasks. For this experiment, it is 2.784 second



* Project: Untitled Project 2

| Tasks | Web Site |
|---|---|
| Task 1 | 2.784 s |

# CogTool Demo