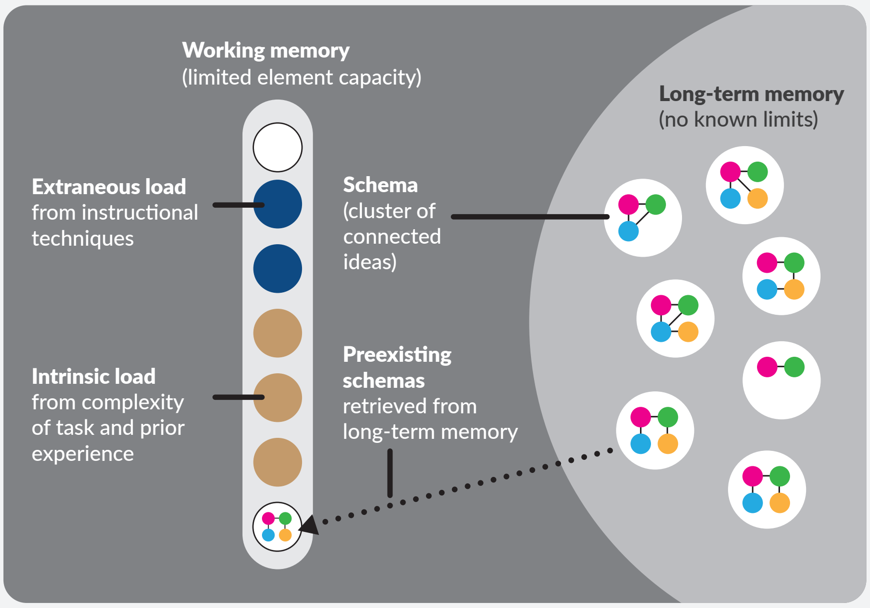
National Centre for Computing Education

Raspberry Pi

Pedagogy Quick Reads

Thoughtful instructional design reduces learners' cognitive load in computing

Cognitive load theory (CLT) is not a new idea, in that the challenges associated with designing effective teaching within the limit of working memory have been studied over the last two decades[1,2]. Whilst the theory isn't universally established, it has been applied in many settings, and offers some important insights for computing educators.


Working memory
(limited element capacity)
Long-term memory
(no known limits)
Extraneous load
from instructional techniques
Schema
(cluster of connected ideas)
Intrinsic load
from complexity of task and prior experience
Preexisting schemas
retrieved from long-term memory

A model of cognition

Cognitive load theory builds on the idea that human memory has two distinct areas, our short-term working memory and long-term memory. Whilst our long-term memory can be seen as essentially infinite, our working memory is extremely limited, with studies suggesting a processing capacity of between three and nine "information elements".[2] This capacity can easily become overloaded, impacting on our ability to process the information that we're presented with.

As we learn from our experiences, new information is stored in our long-term memory for future recall. Over time, these disparate elements of information are connected with existing understanding into collections of related knowledge or schemas.

The goal of effective learning design should therefore be to facilitate the movement of new ideas and information from working memory into conceptually sound schemas.

Summary

Human memory

- Working memory is extremely limited
- Long-term memory has no known limits
- Learning occurs when new knowledge is transferred from working to long-term memory
- Schemas are structured collections of prior learning that can be recalled from long-term memory
- Schemas only occupy a single element in working memory

Cognitive load

- Cognitive load is a stress on a learner's working memory, reducing their ability to acquire new learning
- Intrinsic load relates to the complexity of the learning task and the learner's existing understanding
- Extraneous load is the additional stress placed on the learner due to the way in which the material is presented

Managing intrinsic load

- Awareness of learners' prior experience and understanding helps predict where cognitive overload may occur
- Breaking down the learning into suitable learning episodes helps manage cognitive load

Implications for instruction

- Present only information relevant to the task in a unified way
- Present information both visually and orally as appropriate, without adding additional load
- Use worked examples to provide scaffolding for novices
- Use collaborative techniques such as pair programming, which distribute the cognitive load amongst learners

Balancing the load

Sweller's[1,2] research suggests that during a learning episode, there are two key stresses or cognitive loads acting on the learner.

The intrinsic load placed on the learner relates to the complexity (number of and interactivity of elements) of the concept(s) or skill(s) being taught, and the gap between the new learning and their existing understanding and any misconceptions they already hold. Educators should take steps to optimise intrinsic load wherever possible.

The manner in which new concepts are presented, explored, and applied can lead to an unnecessary extraneous load being placed on the learner. Having to juggle too much new information, or unnecessary information, from multiple sources, can place an increased load on the learner. Through considered instructional design, educators can minimise the extraneous load in their activities.

Managing intrinsic load

Intrinsic load stems from the gap between the learner's existing understanding and the complexity of the new concept or skill being taught. In seeking to reduce the load placed on learners, we need to reduce this gap.

- Ensure that you are aware of prerequisite knowledge and learners' existing understanding. When planning the progression between concepts, it can be helpful to use "Objective graphs", which map connections and dependencies between concepts.
- Break the learning up into smaller tasks, or even individual elements. After being taught in isolation, these elements can be revisited, making connections between them. This is useful for highly complex concepts, but a balance is needed in the classroom, where teaching time is constrained and breaking down too far could result in a lengthy and disjointed learning sequence.[3]

Optimising instructional design

Educators can reduce the extraneous load placed on learners through how they **present** their materials. Within cognitive load theory, there are a number of observable **effects** which affect the cognitive load that learners experience. These are some effects that are relevant to the teaching of computing:

Effect	Implications for computing educators
Split attention effect Learners must combine information from multiple sources, which increases cognitive load	• Combine diagrams with labels and related explanations • Annotate programs using comments, in particular identifying common sections or patterns, known as **subgoals**
Redundancy effect Learners must process and disregard repeated or unnecessary information, which increases cognitive load	• Avoid the inclusion of redundant information in diagrams and explanations • Use accessible language • Minimise the use of "boilerplate code"
Transient information effect Information that doesn't persist must be stored in working memory, which increases cognitive load	• Provide learners with programming "cheatsheets" or reference guides • Share or create concept maps[4] with learners, and highlight concepts and their relationships, to provide scaffolding and reference material
Multimodal effect Visual and oral information are processed separately, which reduces cognitive load	• Combine static images, animations, and oral presentation to spread the load • When modelling a process, narrate or prompt **self-explanation** of the thought process to make it visible to learners
Worked example effect Worked examples provide learners with scaffolding and support to develop generalised solutions, which reduces cognitive load	• Use partially or fully worked examples to provide possible solutions to problems, e.g. programming tasks, binary/denary conversions, compression algorithms, etc. • Use worked examples to model problem-solving processes, including specifying, decomposing, prototyping, and testing
Collective working memory effect Task elements are shared between a group, which reduces cognitive load	• Use techniques such as pair programming to share work between learners and thereby spread the load • Poor communication between learners can add a "cost", which could eliminate the benefits of this effect

References

[1] **Sweller, J. (1988)** Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*. 12, 257–285.

[2] **Sweller, J., van Merriënboer, J. J. G., & Paas, F. (2019)** Cognitive Architecture and Instructional Design: 20 Years Later. *Educational Psychology Review*. 31(2), 261–292.

[3] **Reif, F. (2008)** *Applying Cognitive Science To Education: Thinking And Learning In Scientific And Other Complex Domains*. Cambridge, MIT Press.

[4] **Mühling, A. (2016)** Aggregating concept map data to investigate the knowledge of beginning CS students. *Computer Science Education*. 26(3), 176–191.

Raspberry Pi