

Sample codes:

Use of arrays and matrices

- RM1 Recognition that cell contains Matrix/Array without prompting
RM2 Recognition that cell contains Matrix/Array with prompting

Relationship between multiple cells and the matrix cell

- MC1 Recognition that multiple cells are "contained" within one cell without prompting
MC2 Recognition that multiple cells are "contained" within one cell with prompting
MC0 Failure to recognize this even when prompted

Design principles

- VS Visibility: remainder of matrix might be made visible by scrolling
EP Error-proneness: Matrix formulae reduce scope for errors
AT Abstraction tolerance: Matrix formulae allow multiple operations to be defined in one place
RV Repetition viscosity: Change without matrices is expensive because repeated many times
DV Domino viscosity: Matrices reduce the consequent effect of changes
HD Hidden dependencies: Use of matrices might hide valuable relationships between data
AI Attention investment: I think these maybe should be more specific, ie about costs, risks, (benefits are below)

Intermediate values

- IM Recognise that intermediate matrix values must be created within complex formulae.

Benefits

- LS1 Labour saving advantage from matrices without prompting
LS2 Labour saving advantage from matrices with prompting

Sample table:

Speaker	Transcript	Analysis	Coding
Alan:	This is proposed modification. Is this device self ...	"... self-explanatory": used to focus attention on design of product, rather than user's comprehension performance	
XX: 1	It's like a matrix; in some way all of these number in one	Recognises that cell contains a matrix "Number" appears to refer to cell, in that there are several "all of" cells combined into one	RM1 MC1

Blackwell/Burnett/Benton Jones
"Champagne Prototyping"