THE "PHYSICS" OF NOTATIONS: TOWARD A SCIENTIFIC BASIS FOR CONSTRUCTING VISUAL NOTATIONS IN SOFTWARE ENGINEERING

By: Daniel L. Moody

Presented By: Kevin Shelley

Outline

- Introduction
- Background
- Communication Theory
- 9 Principles for designing effective visual notations
- Interactions among principles
- Conclusion
- Discussion

Introduction

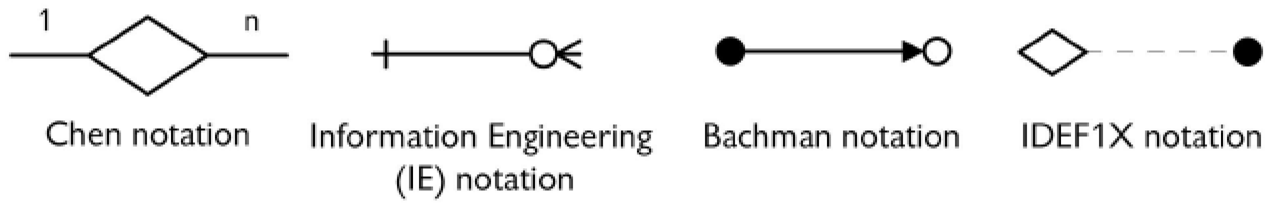
- Visual Notations are Important in SE
- Graphical conventions historically ignored or undervalued
  - frequently design rationale put into the semantics, and graphical conventions are an afterthought
- This Paper presents principals for cognitively effective notations
- Identifies some errors in common SE notations

Background

- Visual language predates written language by 25,000 years
- Diagrams play a crucial role in communicating in SE
  - Believed to convey information to non-technical people more effectively than text
- Diagrams are effective at representing information because:
  - tap into the power of the visual system (Dual Channel Theory)
  - communicated more concisely
  - picture superiority effect

Background

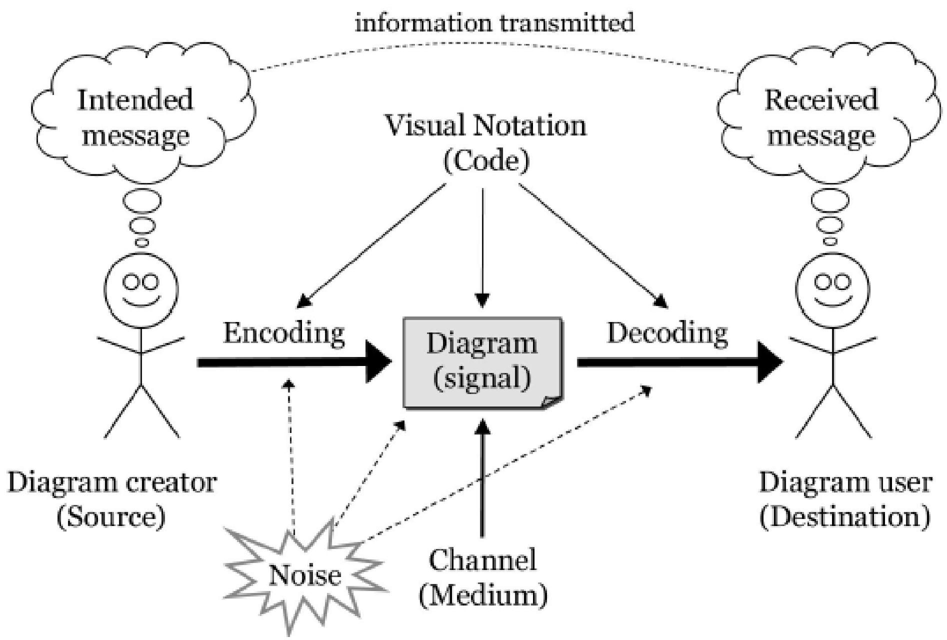
- Rationale for notation design largely absent in SE
- Graphical conventions defined without reference to theory or any justification
  - UML: class defined by a rectangle
- Lack of ability to objectively evaluate diagrams gives rise to multiple visual dialects of same language



Cognitive Effectiveness

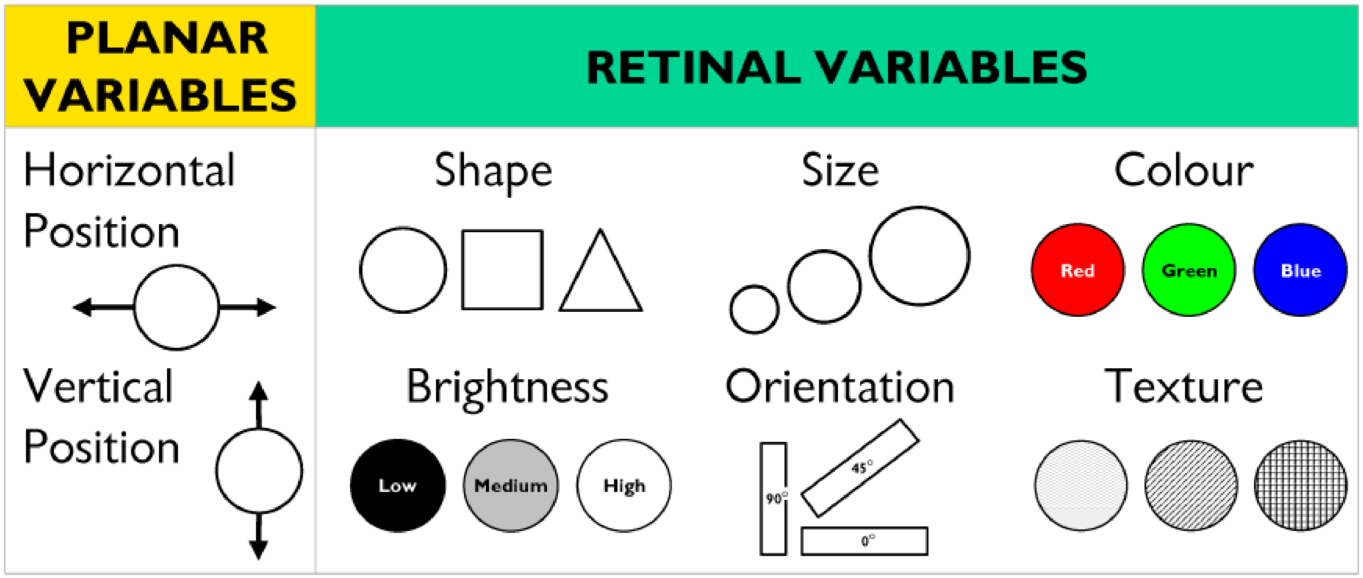
- Currently no clear design goal for visual notations
- Common goals include:
  - Simplicity
  - Aesthetics
  - Expressiveness
- Often vague and subjective
- Cognitive Effectiveness: the speed, ease and accuracy with which a representation can be processed by the human mind
  - Information processing highly sensitive to form

Communication Theory



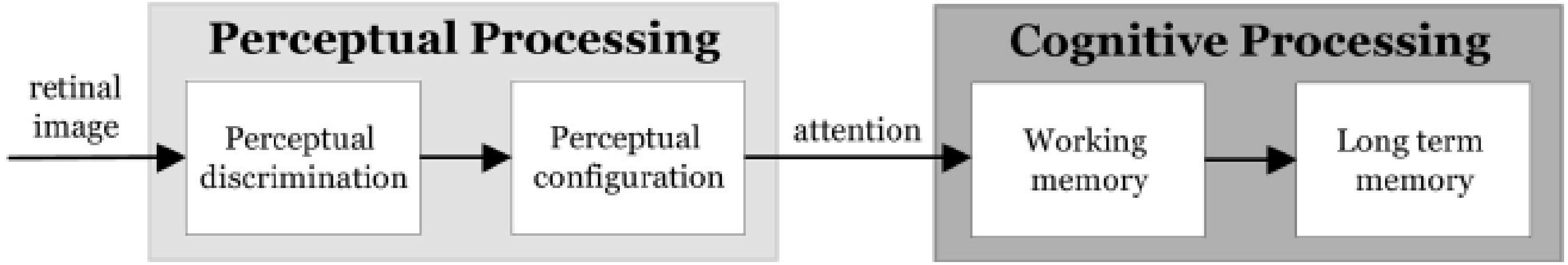
- Communication Consists of Two Processes:
  - Encoding (Design Space)
  - Decoding (Solution Space)

Encoding



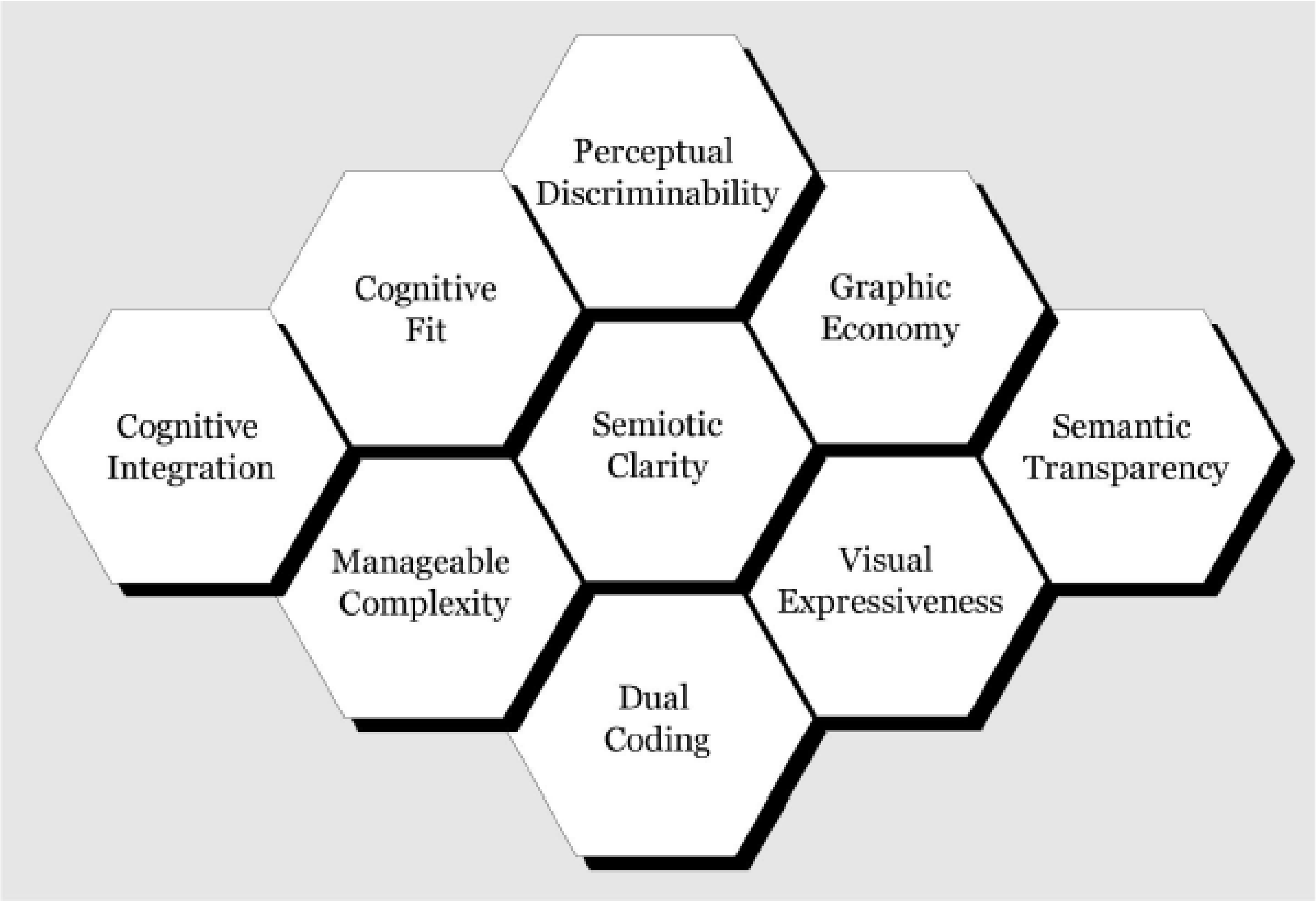
- Eight visual variables that can be used to graphically encode information

Decoding

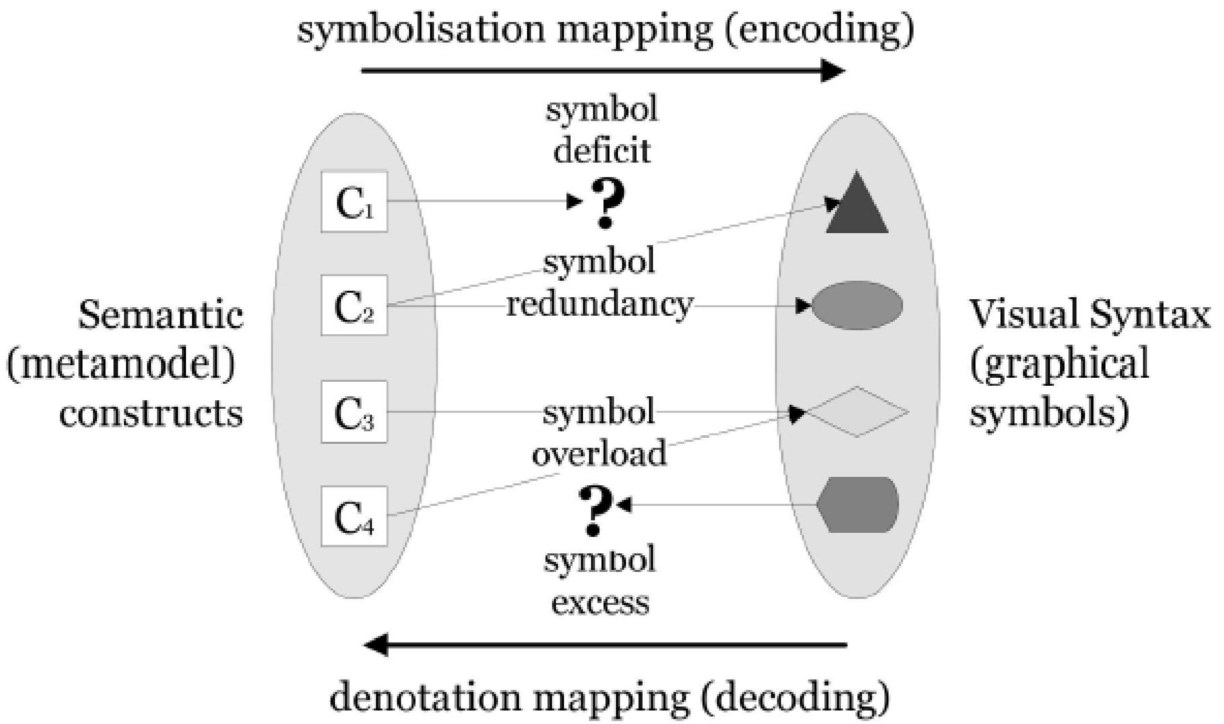


- Perceptual processing (seeing)
  - Automatic, fast, parallel
- Cognitive Processing (understanding)
  - Slow, effortful, sequential
- Computational Offloading: Shift some of the processing burden from the cognitive system to the perceptual system

Principles for Designing Effective Visual Notations



Principle of Semiotic Clarity



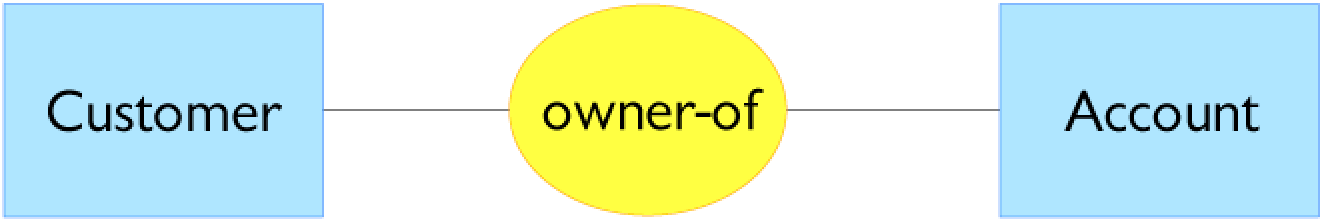
- There must be a one-to-one correspondence between symbols and their referent concepts

Principle of Semiotic Clarity

- Symbol Redundancy
  - Multiple symbols exist for the same construct
- Symbol Overload
  - Multiple constructs exist for the same symbol
- Symbol Excess
  - Graphical constructs do not have any semantic meaning
- Symbol Deficit
  - Semantic Constructs not represented by any symbol

Principle of Perceptual Discriminability

- Different Symbols should be easily distinguishable from one another
- Primarily determined by visual distance
  - The number of visual variables on which they differ and the size of the differences
  - Higher visual distance = faster, more accurate recognition
- Shape
  - The primary visual variable



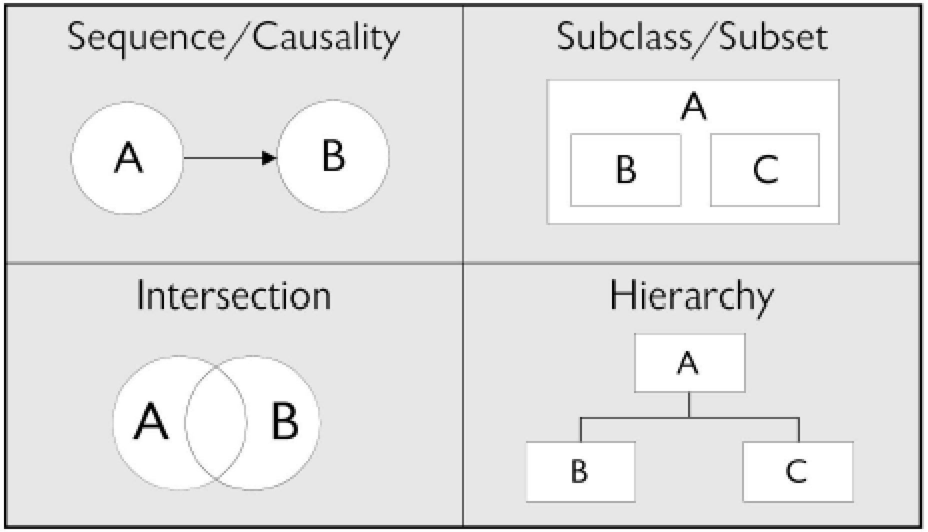
Principle of Perceptual Discriminability

- SE notations use mostly rectangle variants
- Visual distance can be increased using redundant coding
- Feature integration theory increases “perceptual popout”
- Textual Differentiation
  - Zero Visual Distance, cognitively ineffective

Principle of Semantic Transparency

- The Extent to which the meaning of a symbol can be inferred from its appearance (intuitiveness)
- Symbols can be:
  - Semantically Immediate
  - Semantically Opaque
  - Semantically Perverse
- SE notations frequently use only abstract geometric shapes
- Icons
  - Speed up recognition
  - Appear less daunting

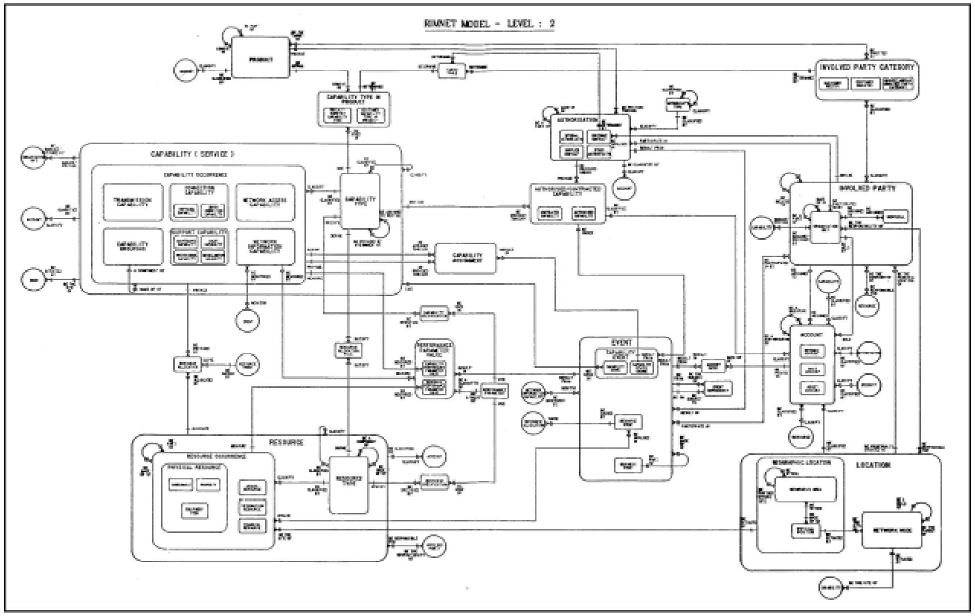
Principle of Semantic Transparency



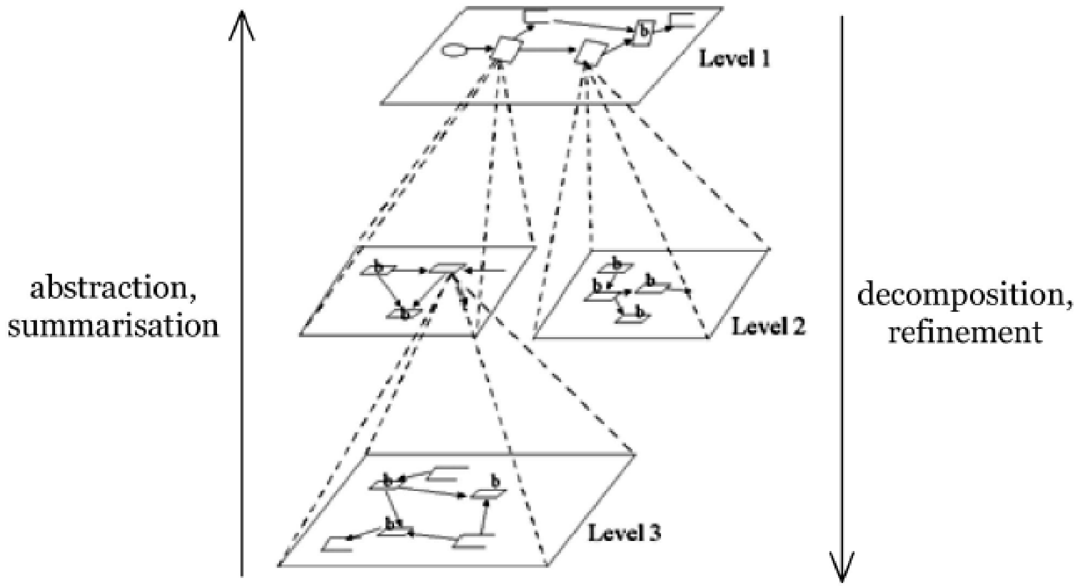
- Semantically Transparent relationships
  - Interpreted in a spontaneous or natural way
- Limited use in SE

Principle of Complexity Management

- Visual Representations do not scale well
- Include mechanisms to represent information without overloading the human mind
- Avoid Cognitive overload
- The number diagram elements to be comprehended exceeds working memory capacity



Principle of Complexity Management

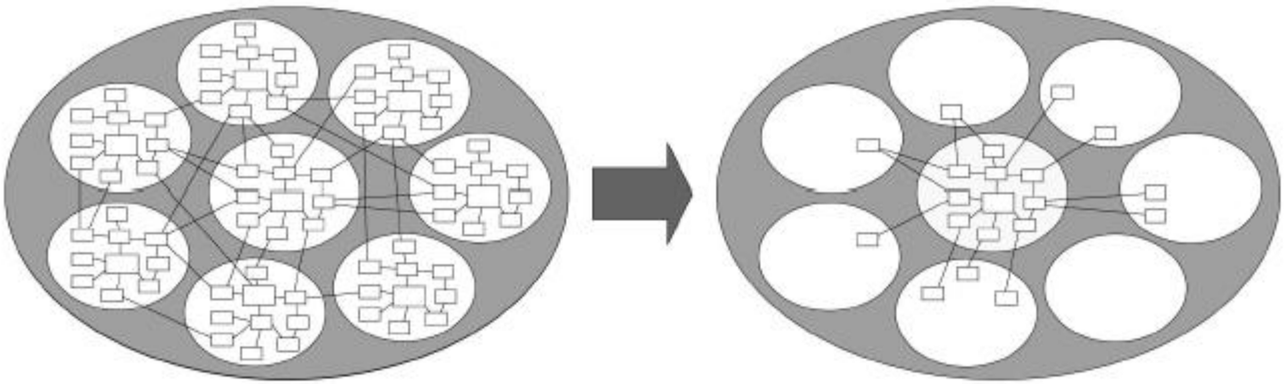


- Modularization
  - Reduces amount of information presented at a time
- Hierarchy
  - Represent a system at different levels of detail
  - Complexity limit: 7 +/- 2 elements per diagram

Principle of Cognitive Integration

- Include explicit mechanism to support integration of information from different diagrams
- Applies when multiple diagrams used
  - Frequently the case in SE
- Conceptual Integration
  - Help reader assemble a coherent mental representation
- Perceptual Integration
  - Simplify navigation and transitions between diagrams

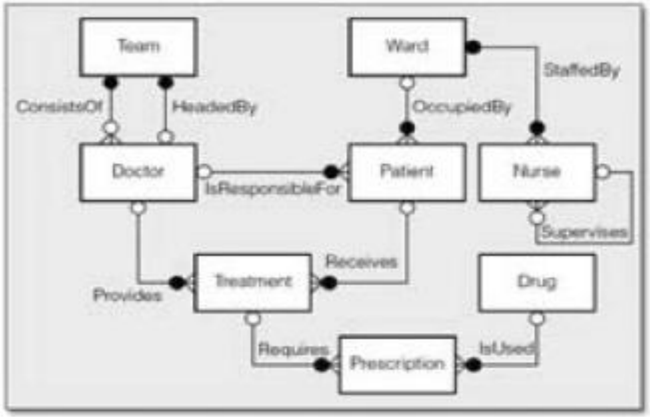
Principle of Cognitive Integration

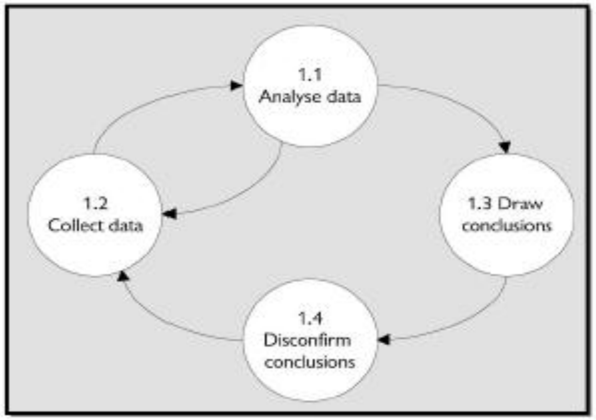


- Contextualization
  - Part of the system of interest displayed in the context of the system as a whole
- DFD
  - Long shot (summary diagram)
- UML lacks these mechanisms

Principle of Visual Expressiveness

- Number of visual variables used in a notation
- 8 Degrees of visual freedom
- Can range from
  - 0 = Nonvisual (textual)
  - 8 = Visually Saturated
- Most SE notations are visually One-dimensional (shape only)
- Colour prohibited in UML





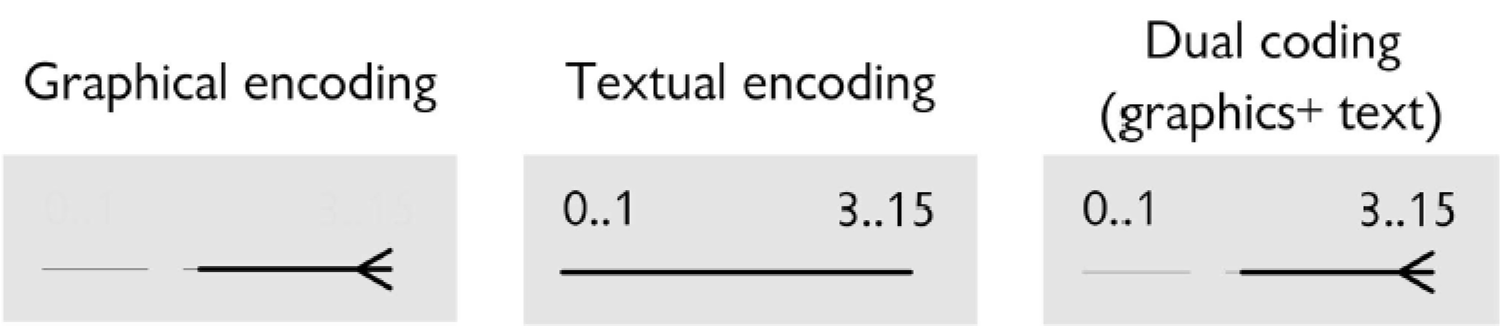
Principle of Visual Expressiveness

- Variables have different capabilities for encoding information
- Power
  - Highest level of measurement that can be encoded
- Capacity
  - Number of perceptible steps
- SE uses limited capacity of shape variable

Variable	Power	Capacity
Horizontal position (x)	Interval	10-15
Vertical position (y)	Interval	10-15
Size	Interval	20
Brightness	Ordinal	6-7
Colour	Nominal	7-10
Texture	Nominal	2-5
Shape	Nominal	Unlimited
Orientation	Nominal	4

Principle of Dual Coding

- Use Text to complement graphics
- Using text and graphics together is more effective than either on their own (dual coding theory)
- Should be used to supplement rather than substitute graphics
- Enables cardinalities to be specified
- Reinforces the meaning of the symbol

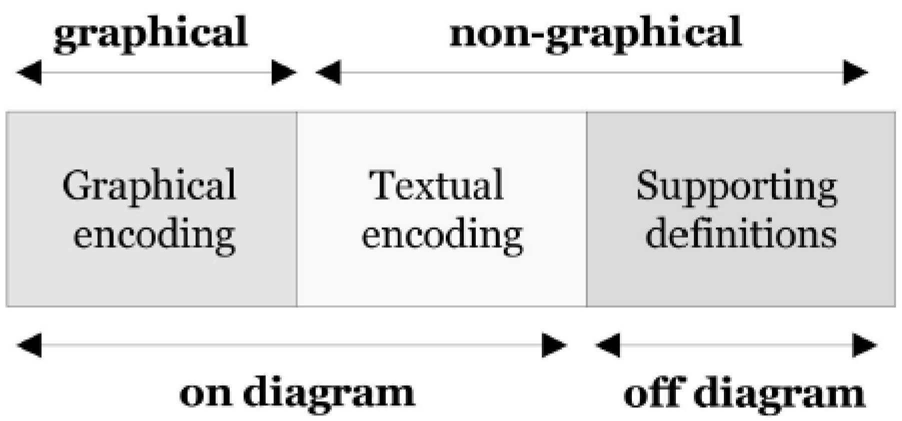


Principle of Graphic Economy

- Number of different graphical symbols should be cognitively manageable
- If too many symbols exist, a legend must be supplied
  - Increases effort for processing
- Human Span of absolute judgment
  - We can only discriminate between 6 categories per variable
  - Upper limit on graphic complexity
- DFDs and ER satisfy this principle
- UML does not

Principle of Graphic Economy

- Strategies for reducing graphic complexity
  - Reduce semantic complexity
  - Introduce symbol deficit (Graphics-text boundary)
  - Increase visual expressiveness



Principle of Cognitive Fit

- Use different visual dialects for different tasks and audiences
- Visual monolinguism
  - Use a single visual representation for all purposes (usually the case in SE)
- Expert-Novice Differences
  - More difficulty discriminating between symbols
  - Have to consciously remember what symbols mean
  - Expertise reversal effect
- Representational medium
  - Requirements for hand drawing constrain visual expressiveness

Interactions Among Principles

	Semiotic Clarity	Perceptual Discriminability	Semantic Transparency	Complexity Management	Cognitive Integration	Visual Expressiveness	Dual Coding	Graphic Economy	Cognitive Fit
Semiotic Clarity								±	
Perceptual Discriminability						+			+
Semantic Transparency		+							±
Complexity Management								−	+
Cognitive Integration	−			+				−	
Visual Expressiveness		+						+	±
Dual Coding									+
Graphic Economy		+		+		−			+
Cognitive Fit									

Summary

- Raise awareness about impact of notation design
  - Equally if not more important than semantics
- Cognitive Effectiveness
  - The primary dependent variable for evaluating visual notations
- Communication theory
  - Describes how notations communicate leveraging theories from communication, graphic design, visual perception and cognition
- Presents principles for constructing and evaluating visual notations
- Identifies serious design flaws in leading SE notations
- Profound effect on usability of notations in SE and other domains

Discussion

- Why is this paper significant?
- How does it differ from other software engineering research?
- What examples from common SE notations follow/break these principles?
- How could this research be used in conjunction with other papers in the course?
- How could these concepts be applied to other research areas?