

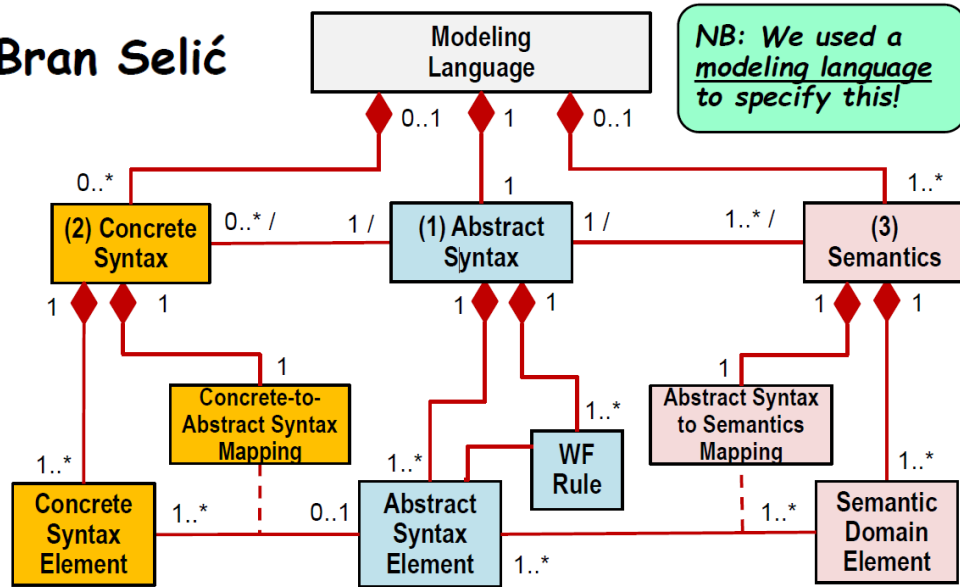
Modelling Languages: Concrete (mostly Visual) Syntax

Hans Vangheluwe

<http://msdl.uantwerpen.be/>

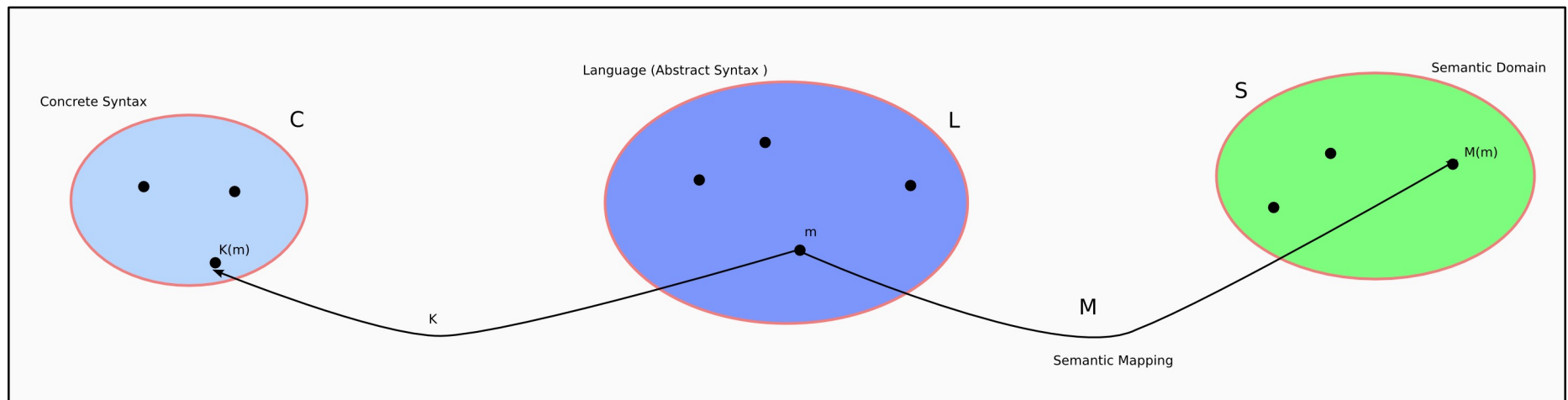
The Structure of Modeling Languages

Bran Selic

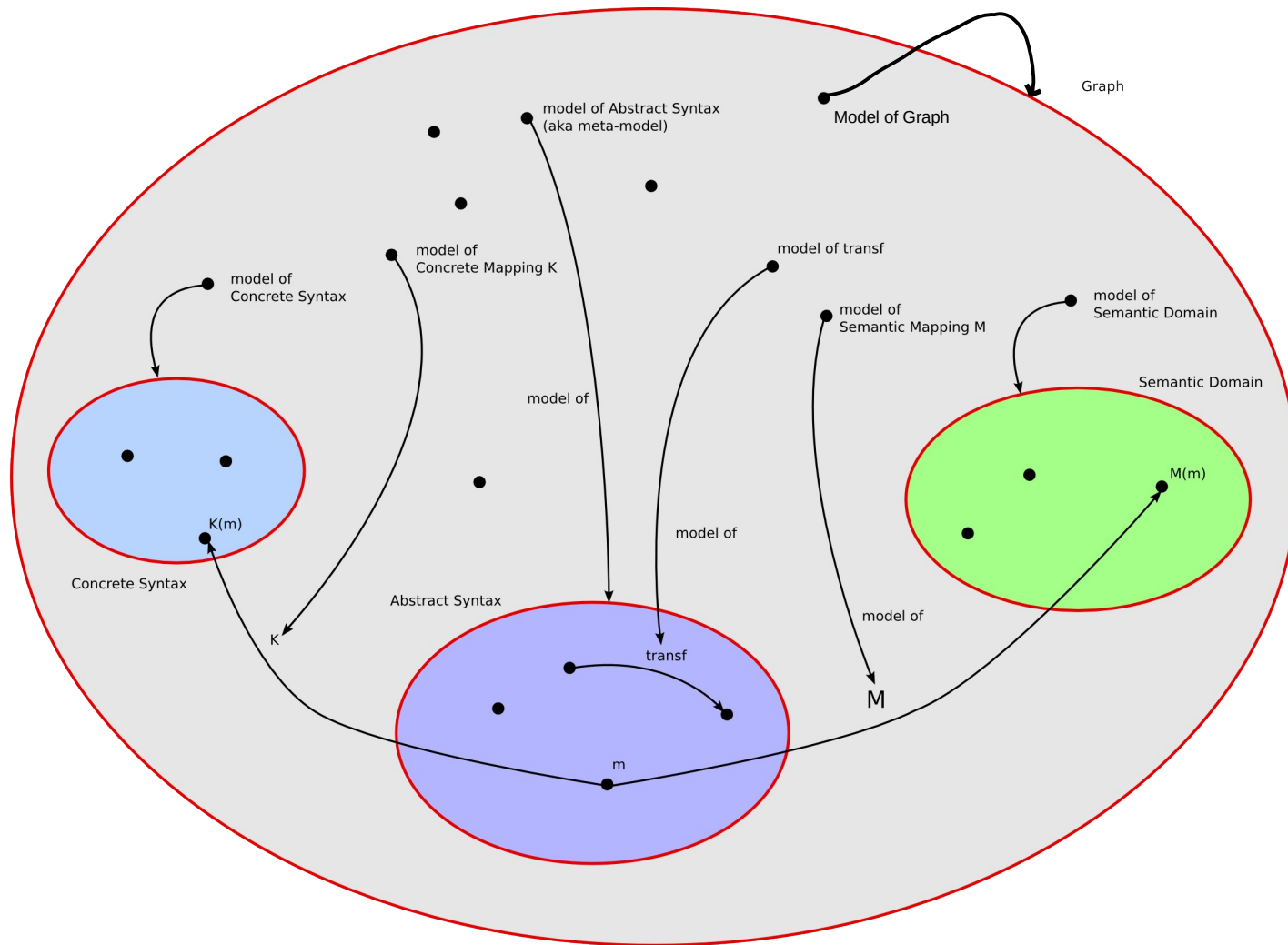


Modelling Languages/Formalisms
Syntax and Semantics

Concrete Formalism F



Modelling Languages/Formalisms Syntax and Semantics



Textual Languages

“this sentence is very short”

- Individual letters in an **alphabet**
- Combined into words
- Combined into sentences in a **language**
- Valid letter combinations in words
specified by **regular expressions**
- Valid word combinations in a language
specified by a **grammar**
- letters/words are combined by the “is to the right of”
spatial relationship (in Western languages)

The Spoofax Language Workbench

Report TUD-SERG-2010-014a

Rules for Declarative Specification of Languages and IDEs

Lennart C. L. Kats

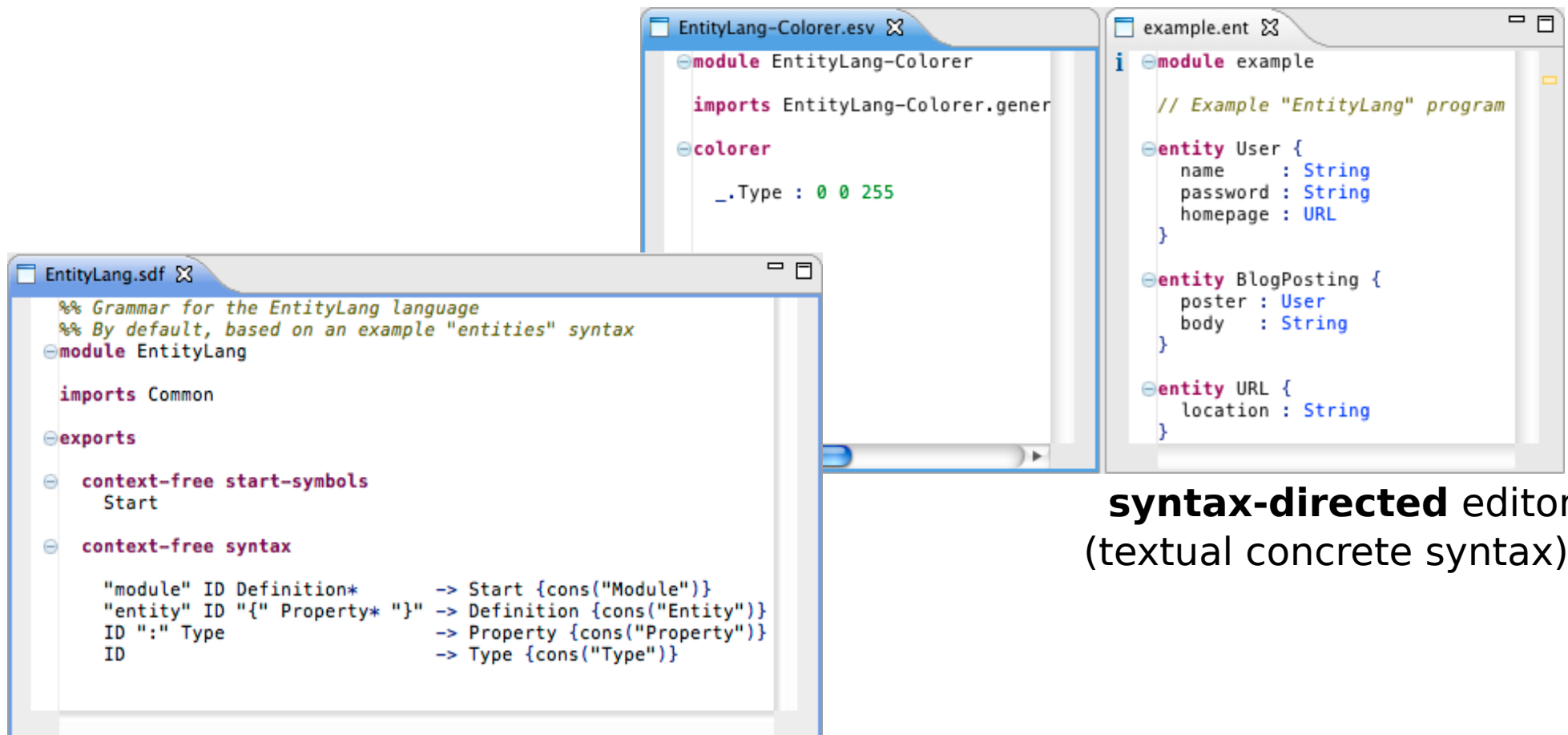
Delft University of Technology

l.c.l.kats@tudelft.nl

Eelco Visser

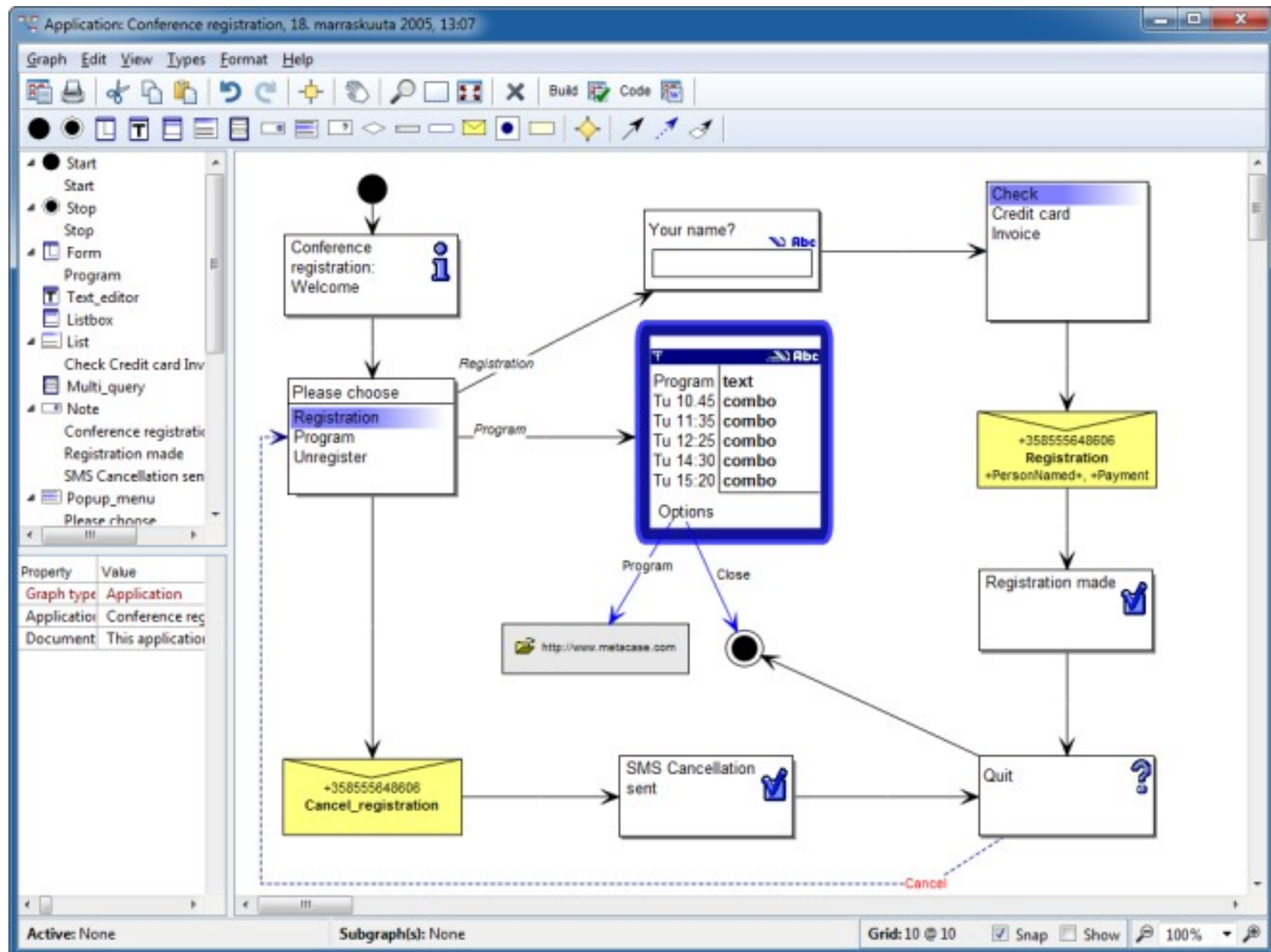
Delft University of Technology

visser@acm.org



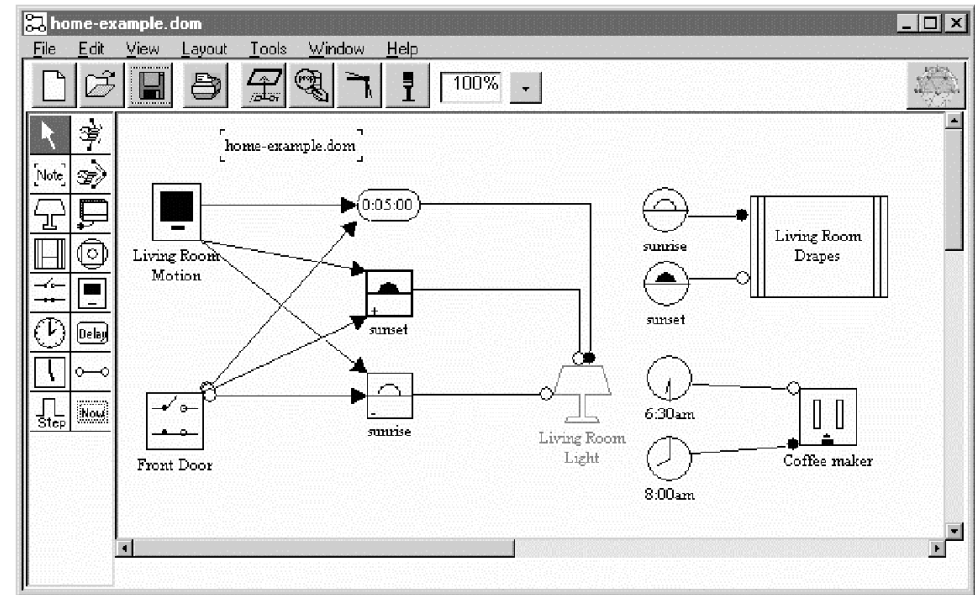
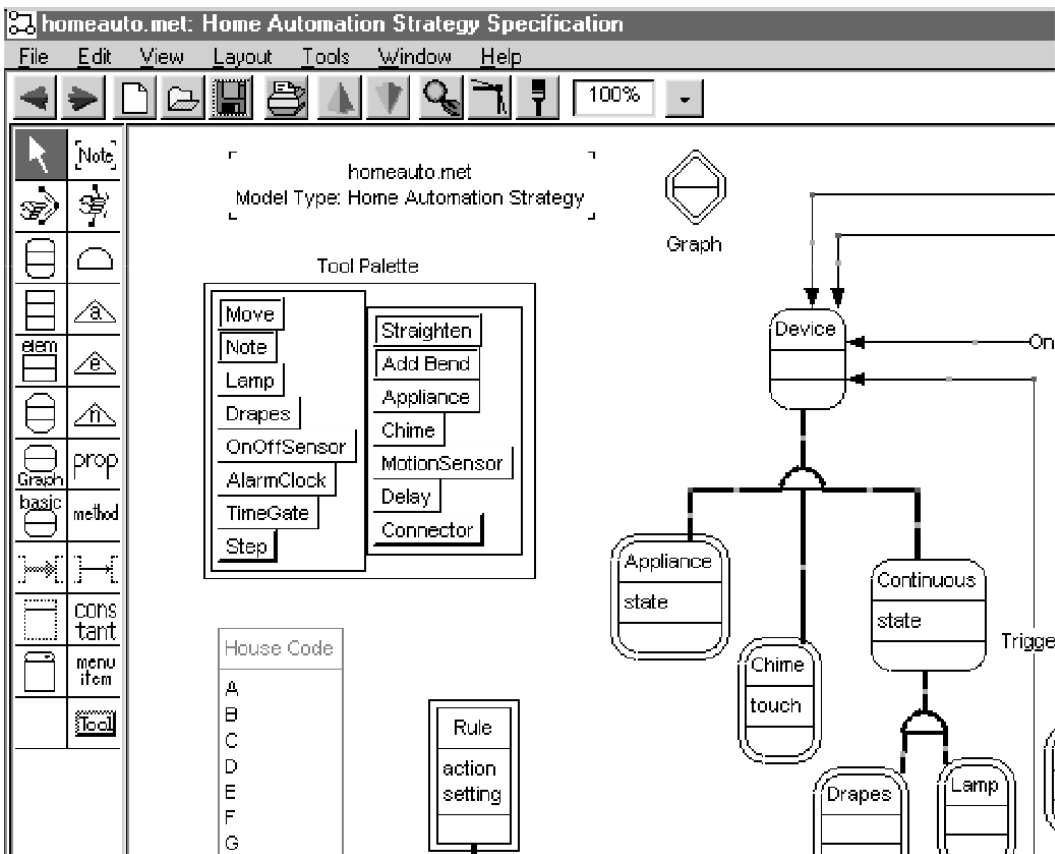
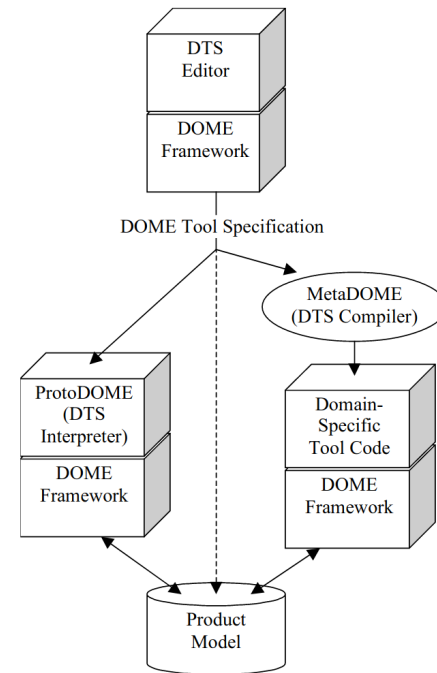
syntax-directed editor (visual concrete syntax)

Visual
Languages



1990s: Honeywell's DoME

Visual Languages

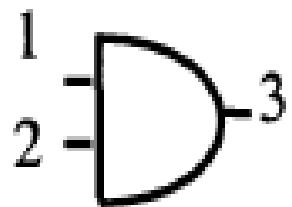




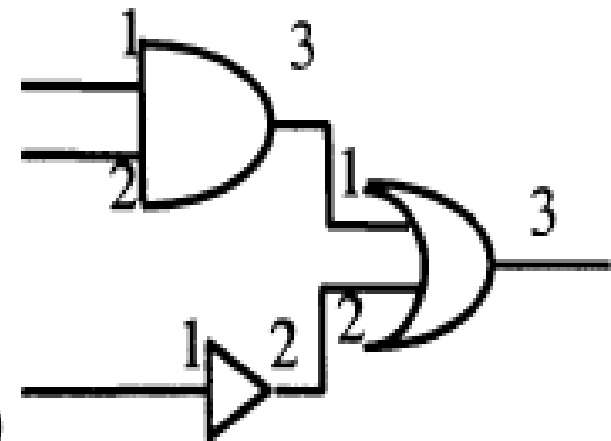
A Classification Framework to Support the Design of Visual Languages

G. COSTAGLIOLA*, A. DELUCIA[†], S. OREFICE[‡] AND G. POLESE*

Plex



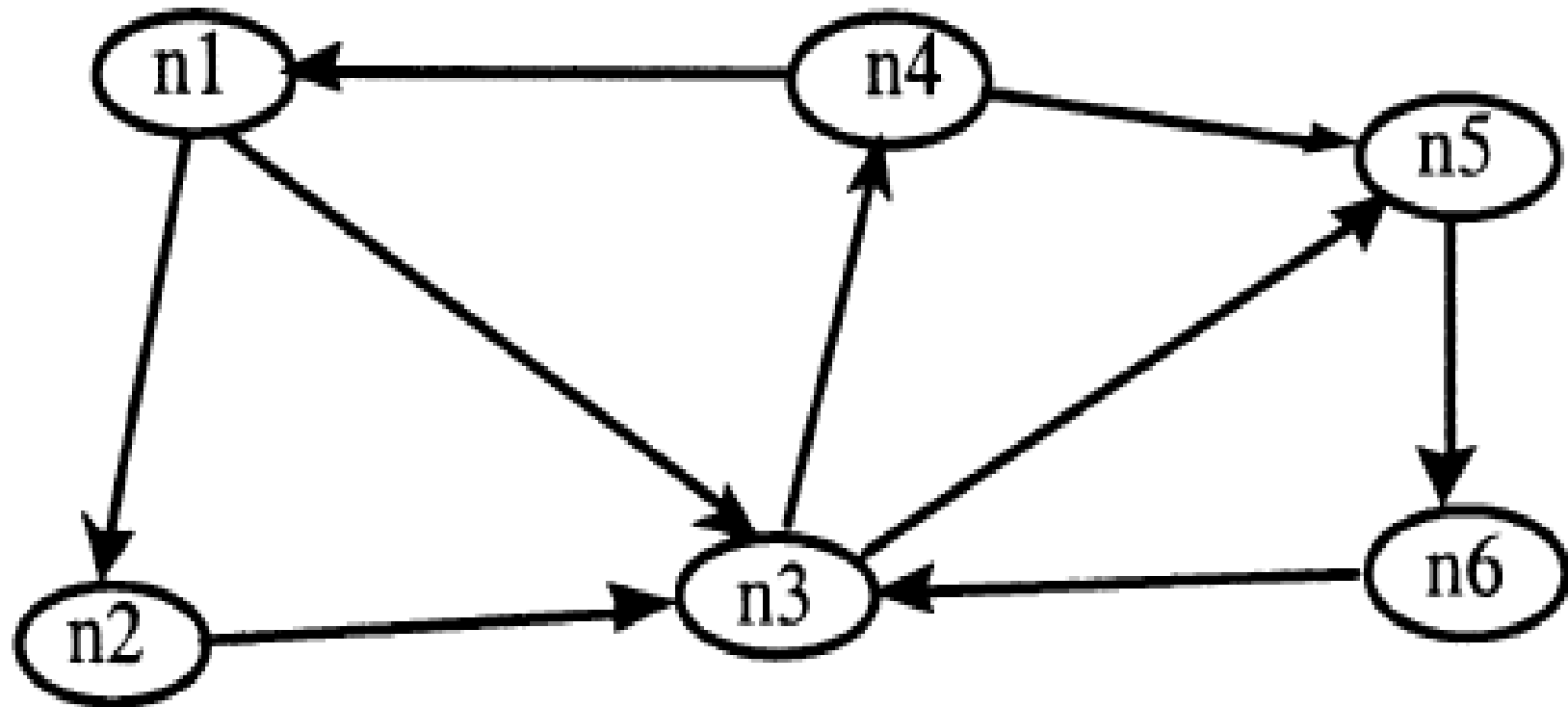
(a)



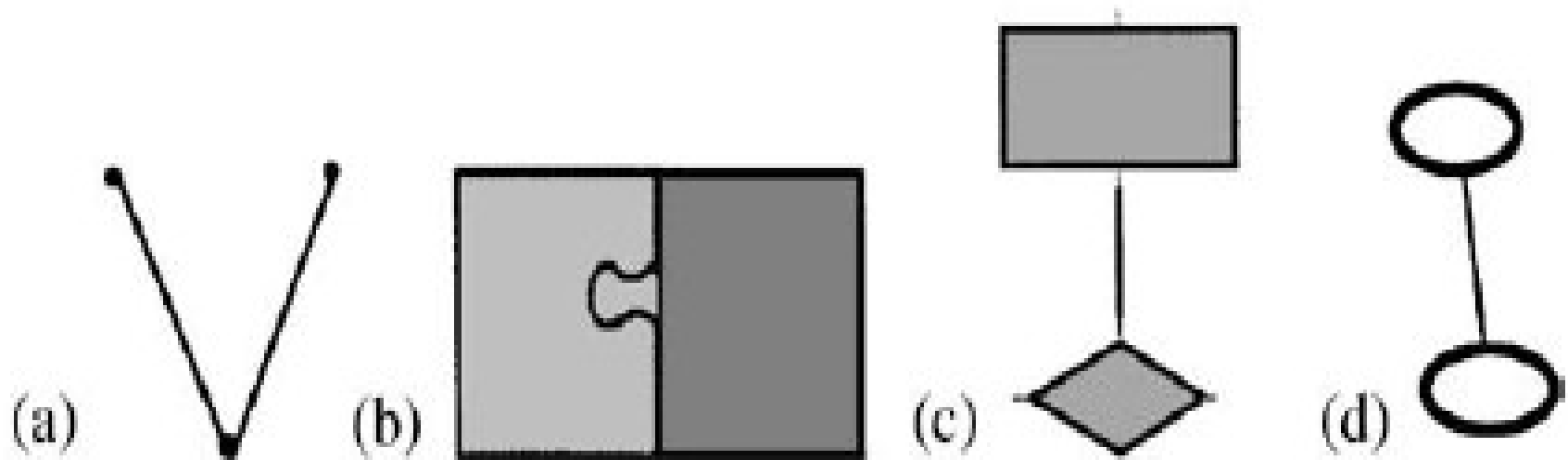
(b)

meta-model of network of blocks with “ports”?

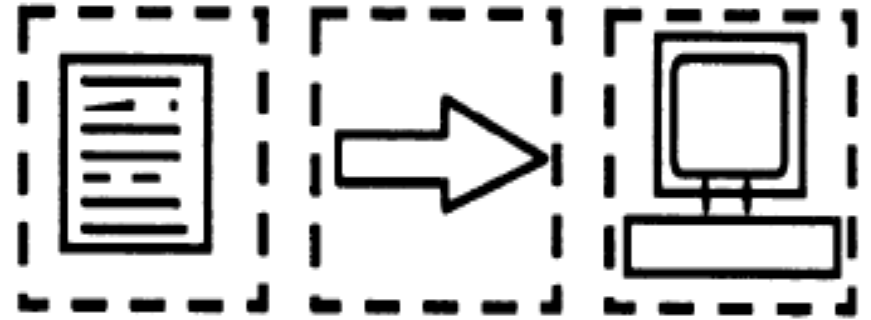
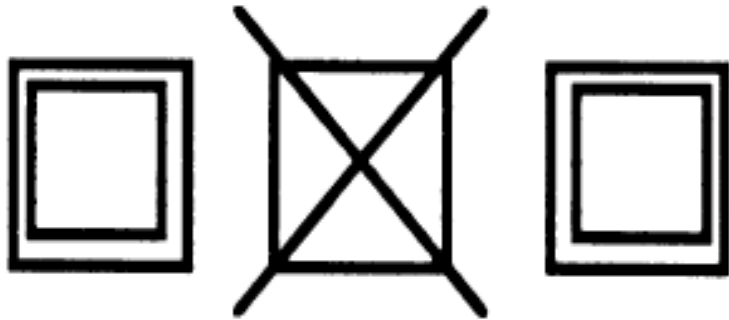
Graph



Connection Types

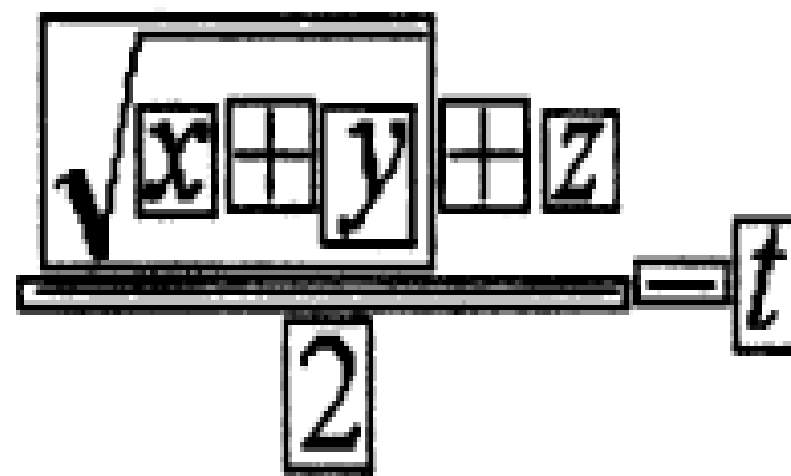


Iconic

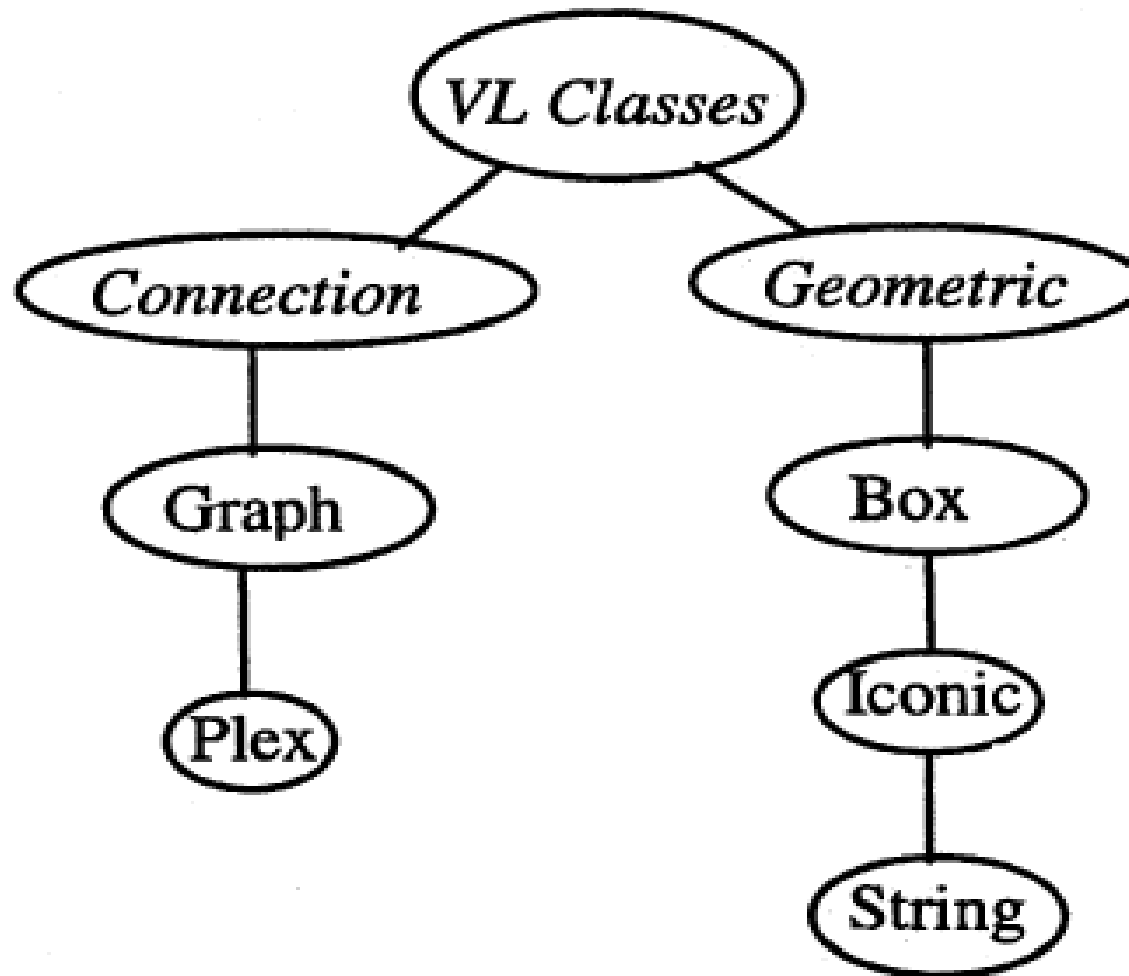


Box

$$\frac{\sqrt{x+y+z}}{2} - t$$



Visual Language Classes



Hybrid Languages

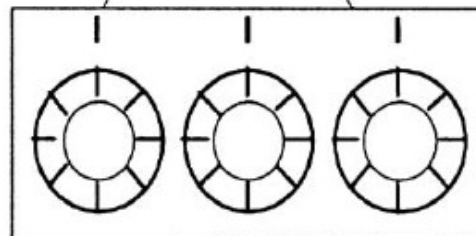
JAN 97					
SUN		5	12	19	26
MON		A1 A3 ⁶	13	20	27
TUE		7	14	A4 ²¹	28
WED	1	A2 8	15	22	29
THU	2	A3 A4 ⁹	16	23	30
FRI	A1 A2 3	10	17	24	31
SAT	4		18	25	

POLICY TIER

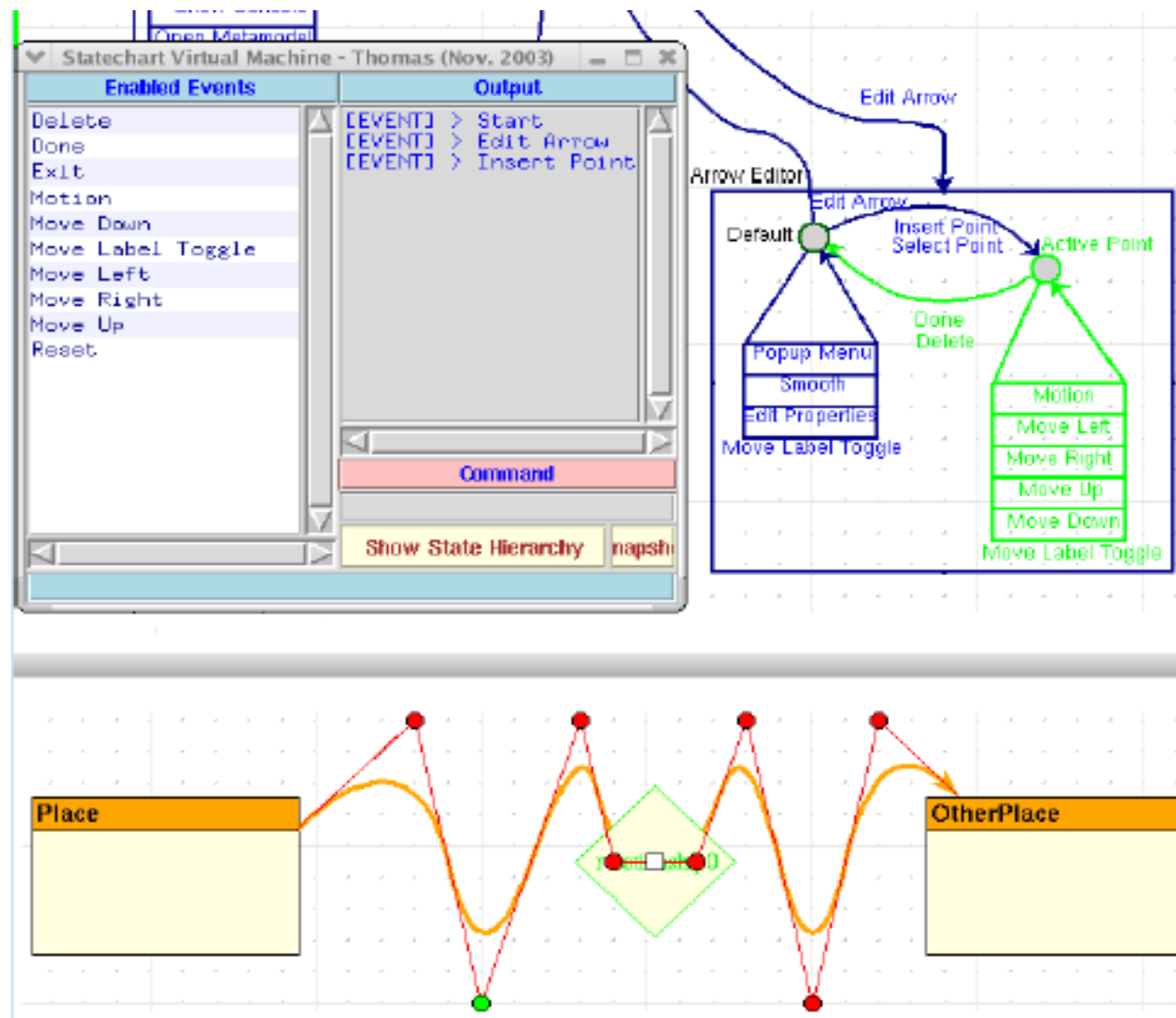
CALENDAR-FORM METAPHOR

DEFINITION TIER

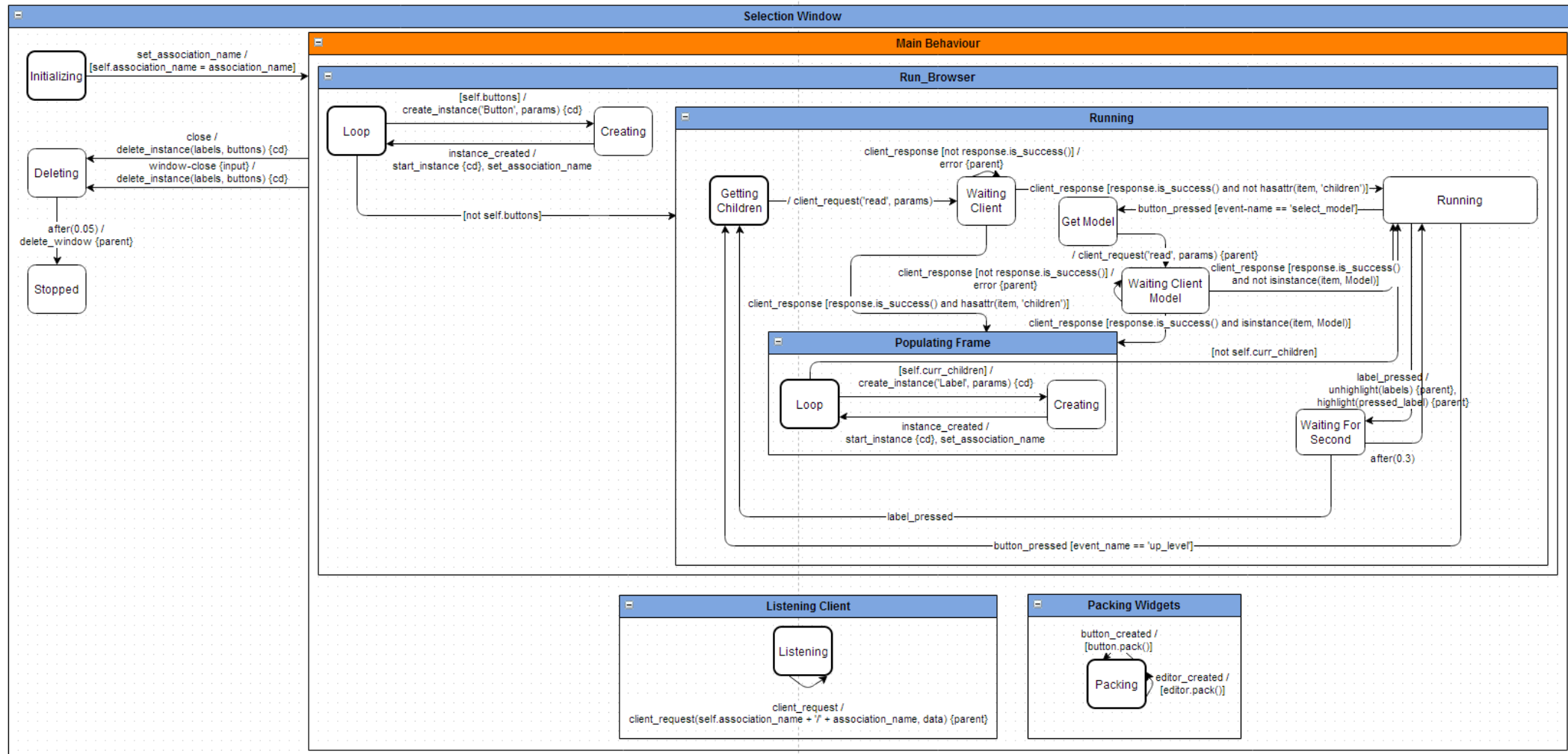
COMBINATION LOCK METAPHOR



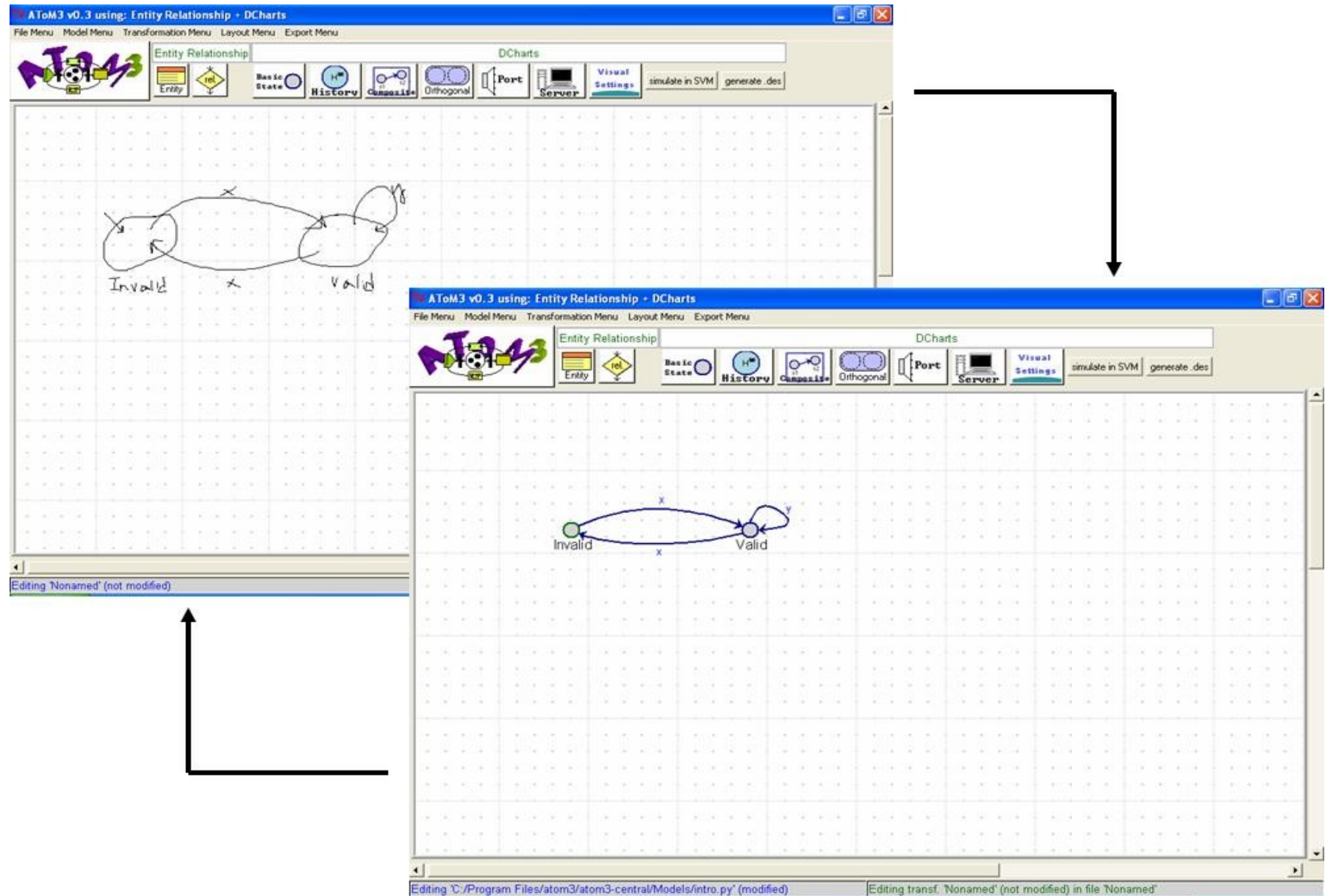
Syntax-directed Visual Editors: model and **synthesize behaviour**



Syntax-directed Visual Editors: model **behaviour**



Syntax-directed Visual Editors vs Freehand editing



Different Media: Gestural Interaction, Sound, ...



gestureworks
true multitouch for Flash and Flex

Gestures included in the open source gesture library

MULTITOUCH GESTURES

Tap Gestures

Rotate Gestures

Scale Gestures

Scroll Gestures

Hold Gestures

Swipe Gestures

Drag Gestures

Split Gestures

Flick Gestures

3D Gestures

Anchor Gestures

STROKE GESTURES

Letter Strokes

Greek Symbol Strokes

Stroke Gesture Direction: [Slider from blue to red]

Number Strokes

Shape Strokes

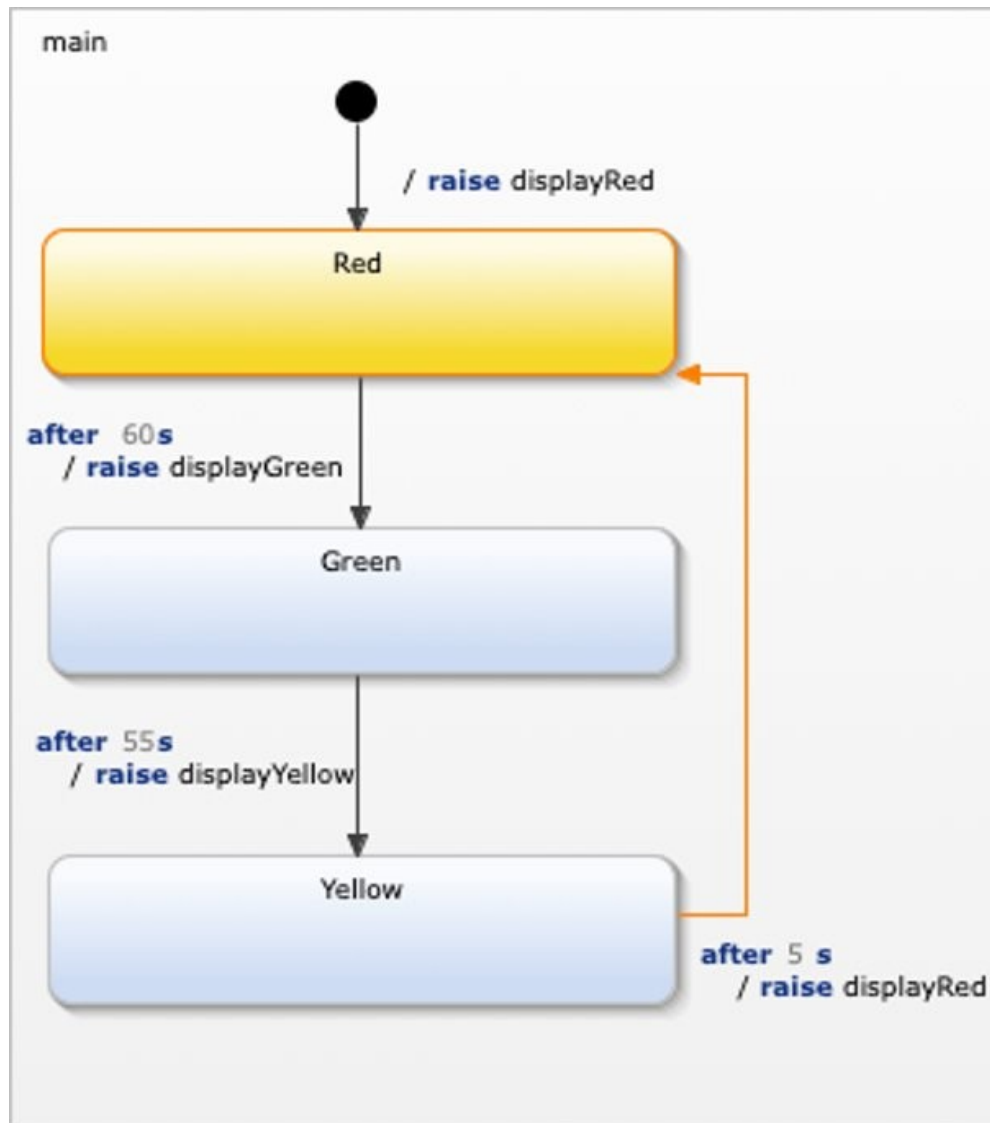
Symbol Strokes

Gestureworks is owned by Gestureworks, Inc. and other parties under license to the Creative Commons Attribution-ShareAlike 3.0 license. All content created by Gestureworks is licensed under the Creative Commons Attribution-ShareAlike 3.0 license. The Gestureworks logo is a registered trademark of Gestureworks, Inc.

CC BY SA

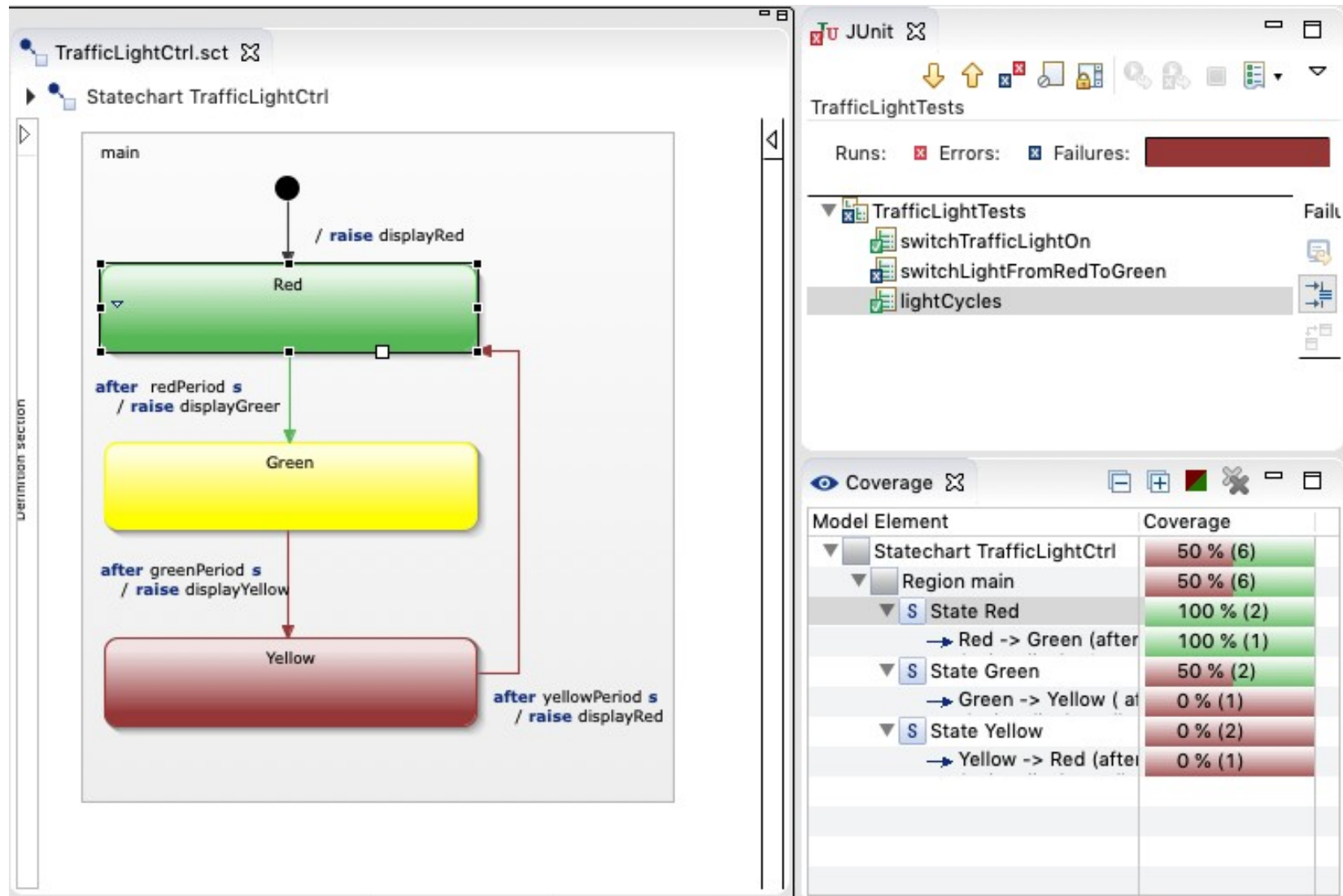
Visual Editing ~

Visual **Simulation** / Execution (**display state/snapshot**), Debugging / Testing



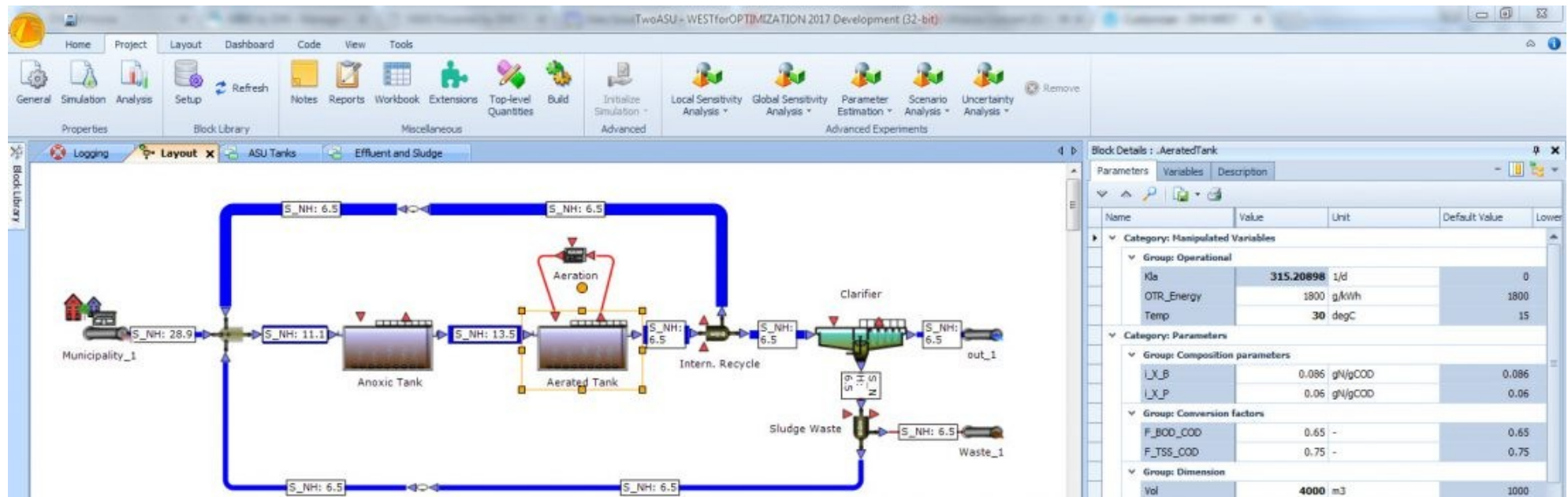
Visual Editing ~

Visual Simulation / Execution (display state/snapshot), Debugging / **Testing**



Visual Editing ~

Visual **Simulation** / Execution (**display state**/snapshot), Debugging / Testing



<http://www.mikebydhi.com/products/west>

The “Physics” of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering

Daniel L. Moody, *Member, IEEE*

Introduction

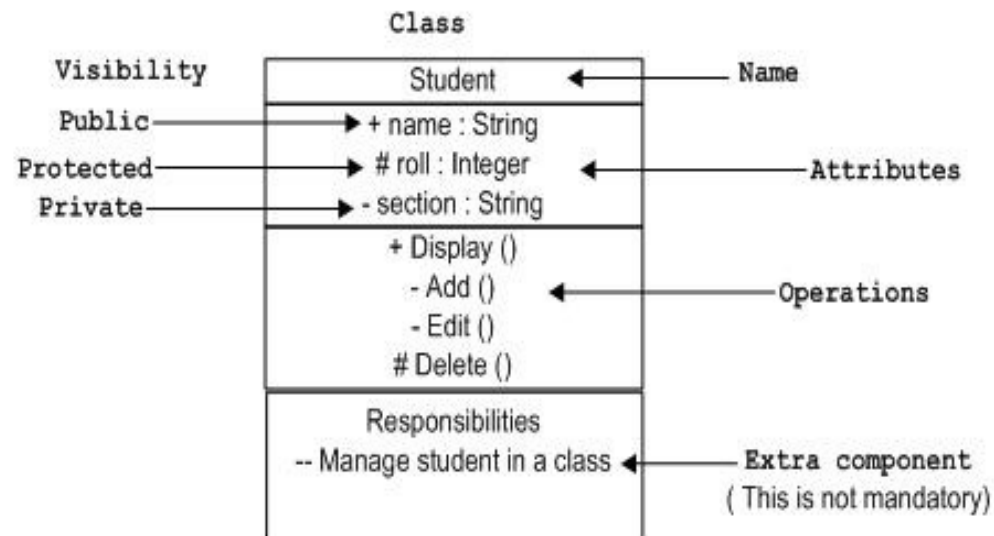
- Visual notations pre-date textual ones
- Textual is special case of Visual
- Visual notations are important for Modelling and Software Engineering
- Humans are excellent pattern recognizers
- Need cognitively efficient and effective notations.

Cognitive effectiveness =
speed, ease and accuracy with which
a representation can be processed by
the human mind

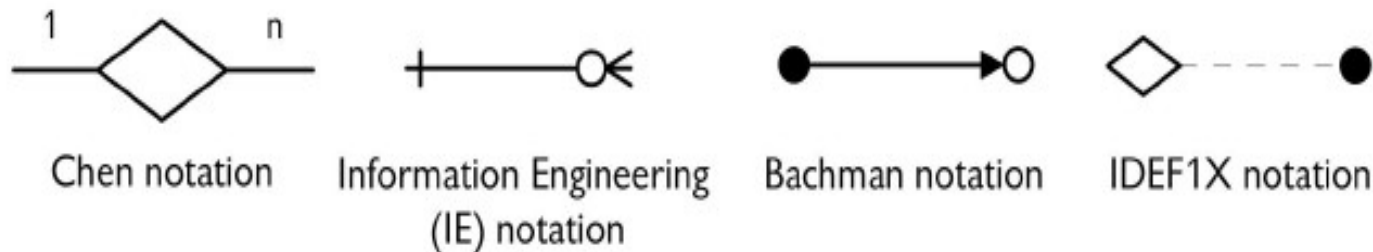


Introduction/Rationale

Visual notations are often introduced without underlying theory or rationale



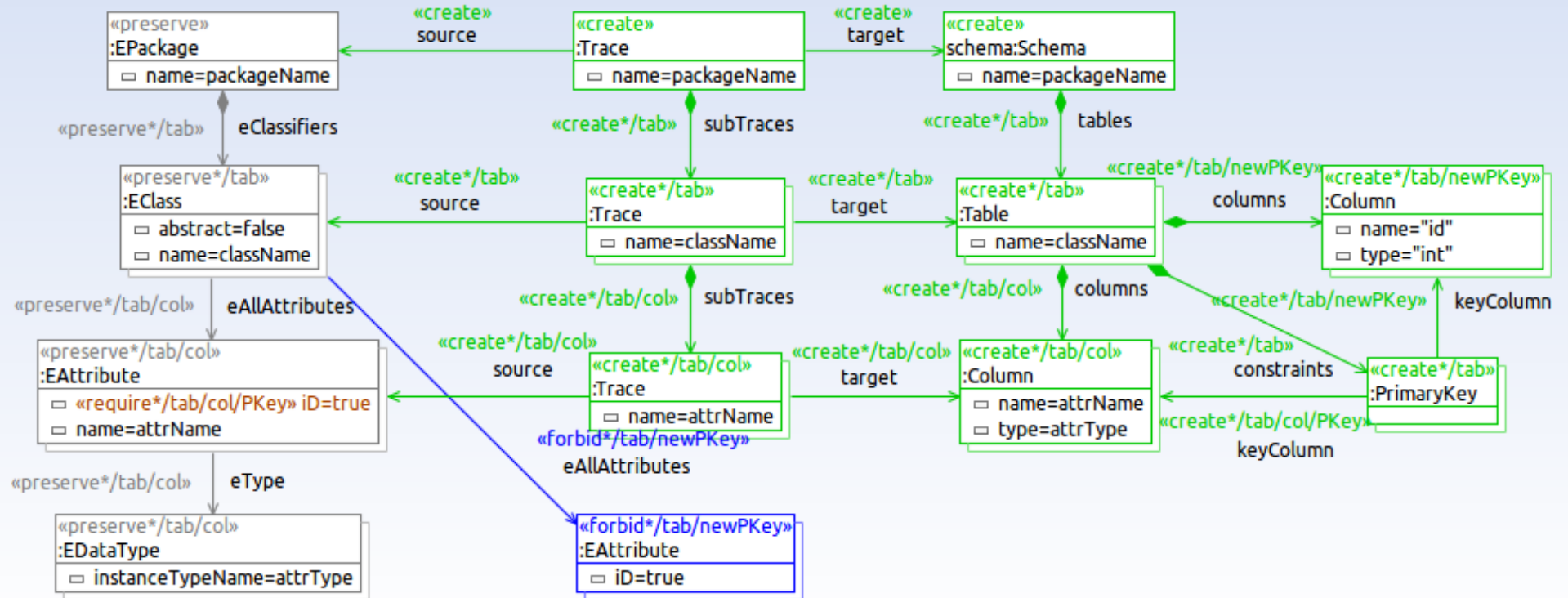
Many visual notations for the same concepts (led to **UML**):



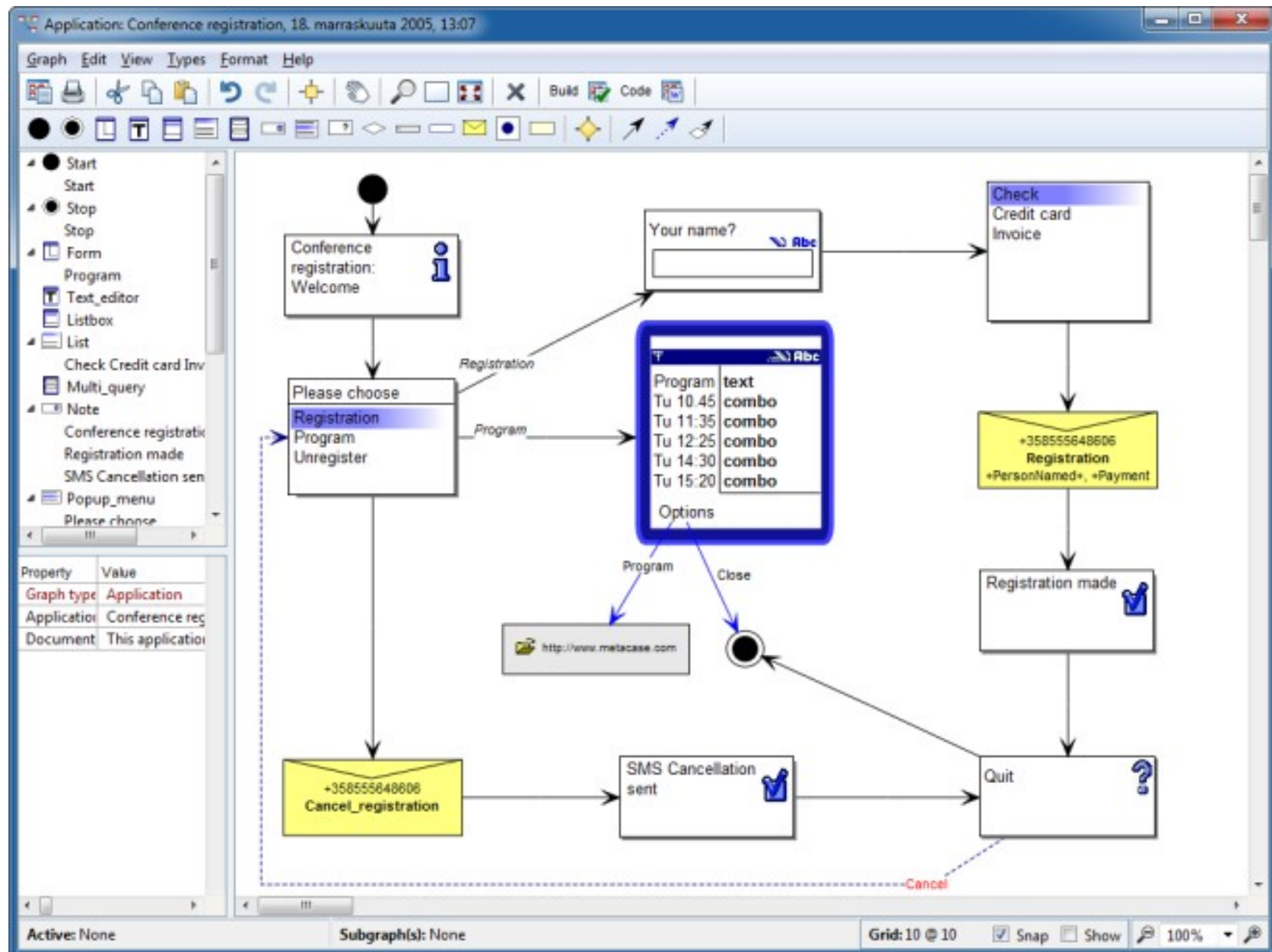
No rigorous way to **compare** effectiveness and hence no clear design goal.

Cognitive Effectiveness?

⇒ Rule CreateSchema(packageName:EString, schema:Schema)



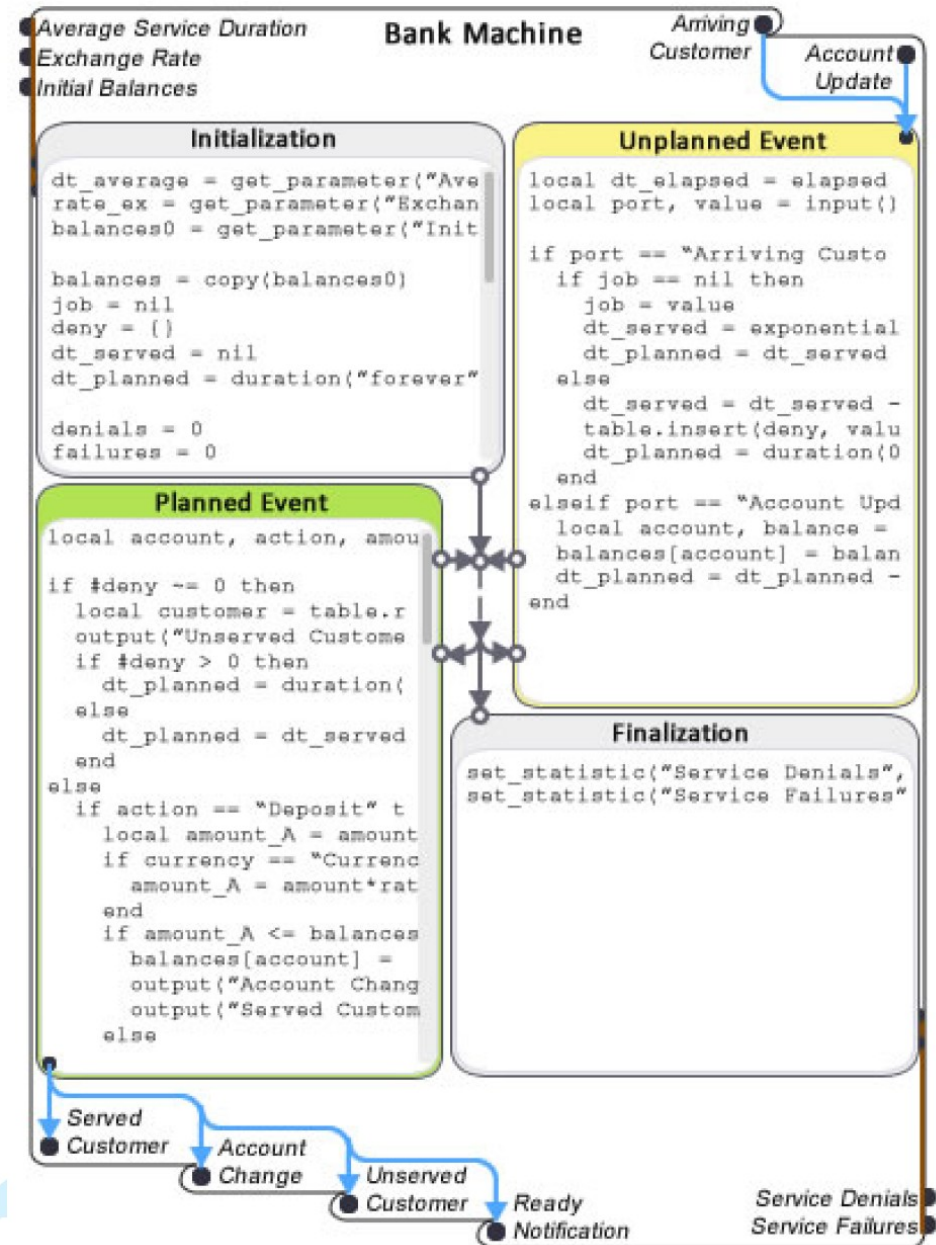
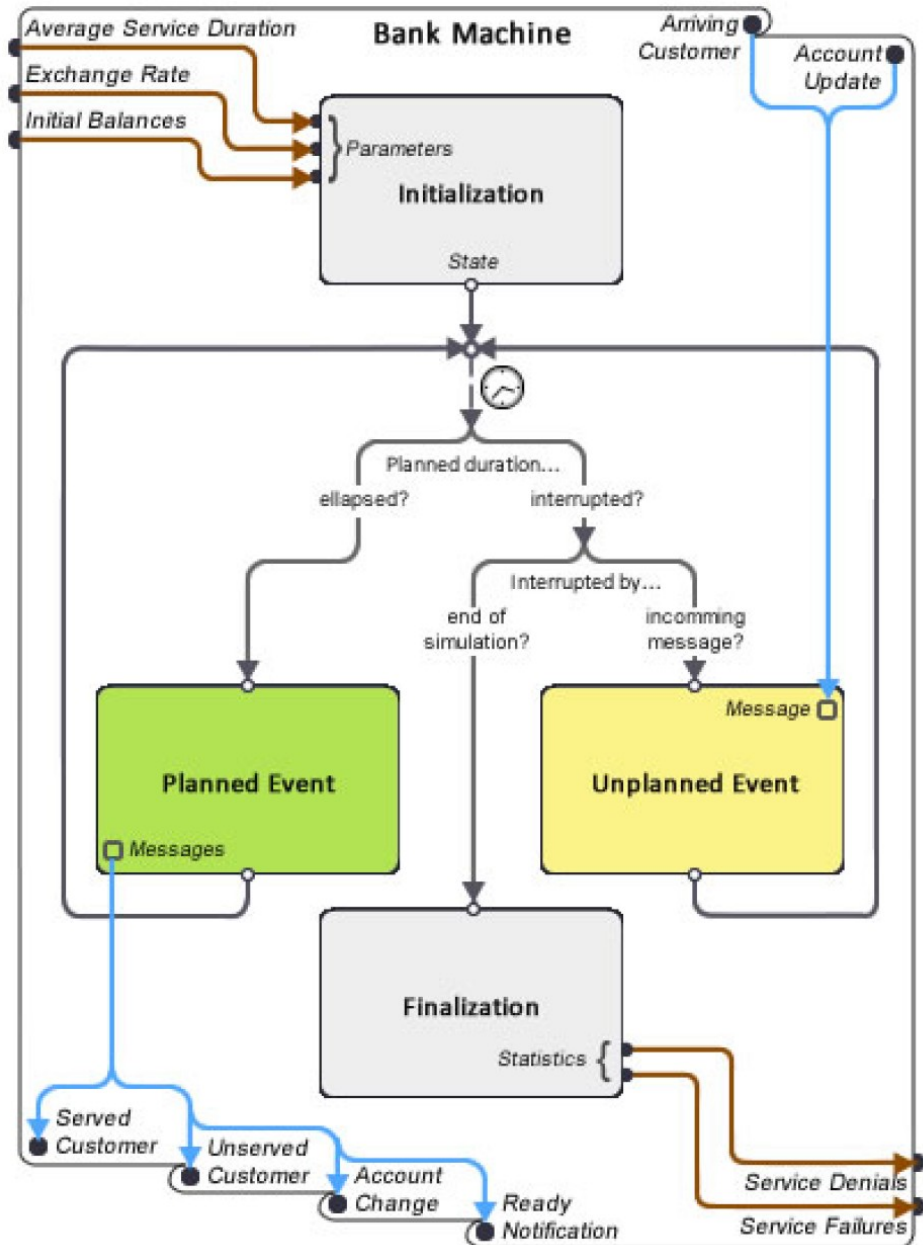
Cognitive Effectiveness?



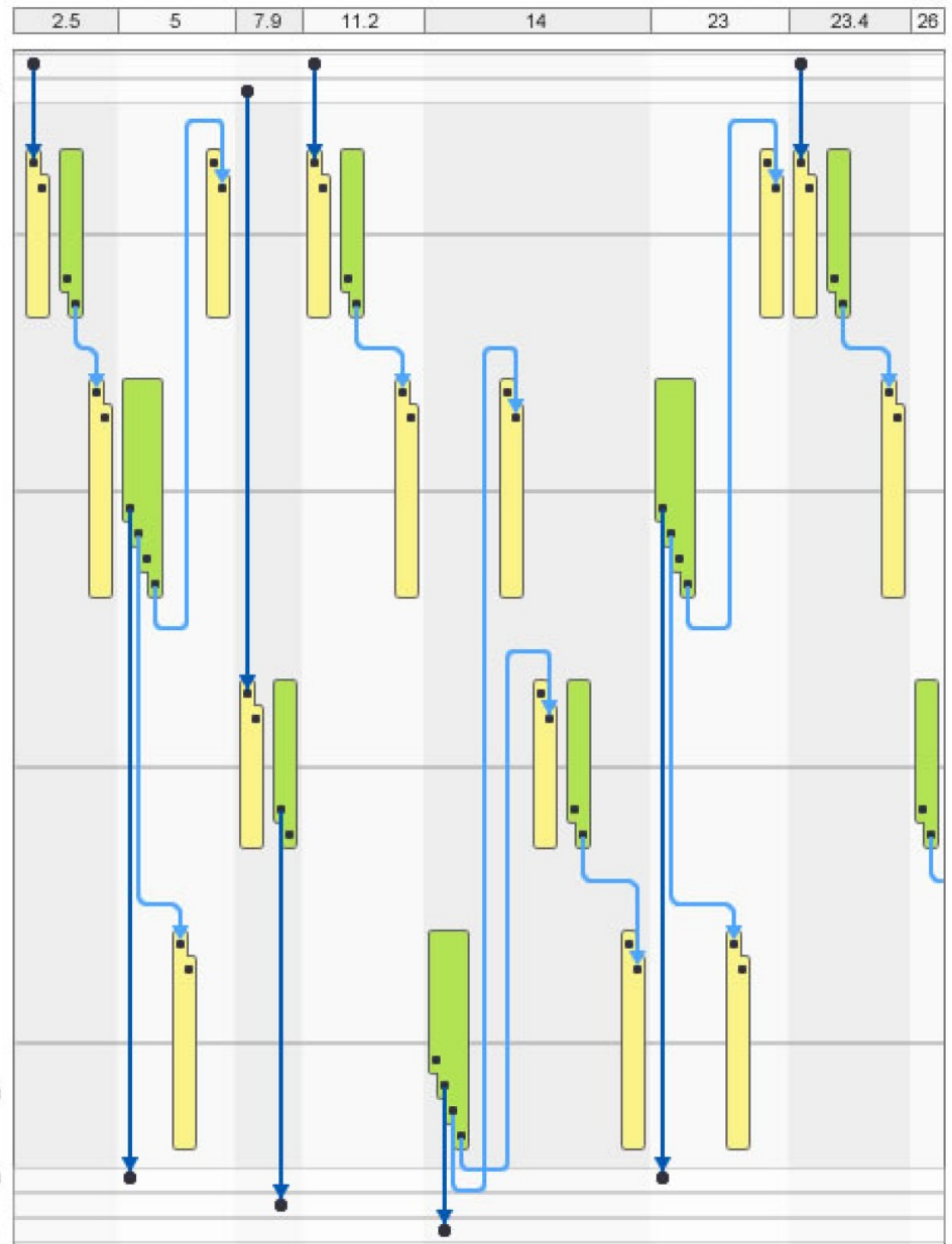
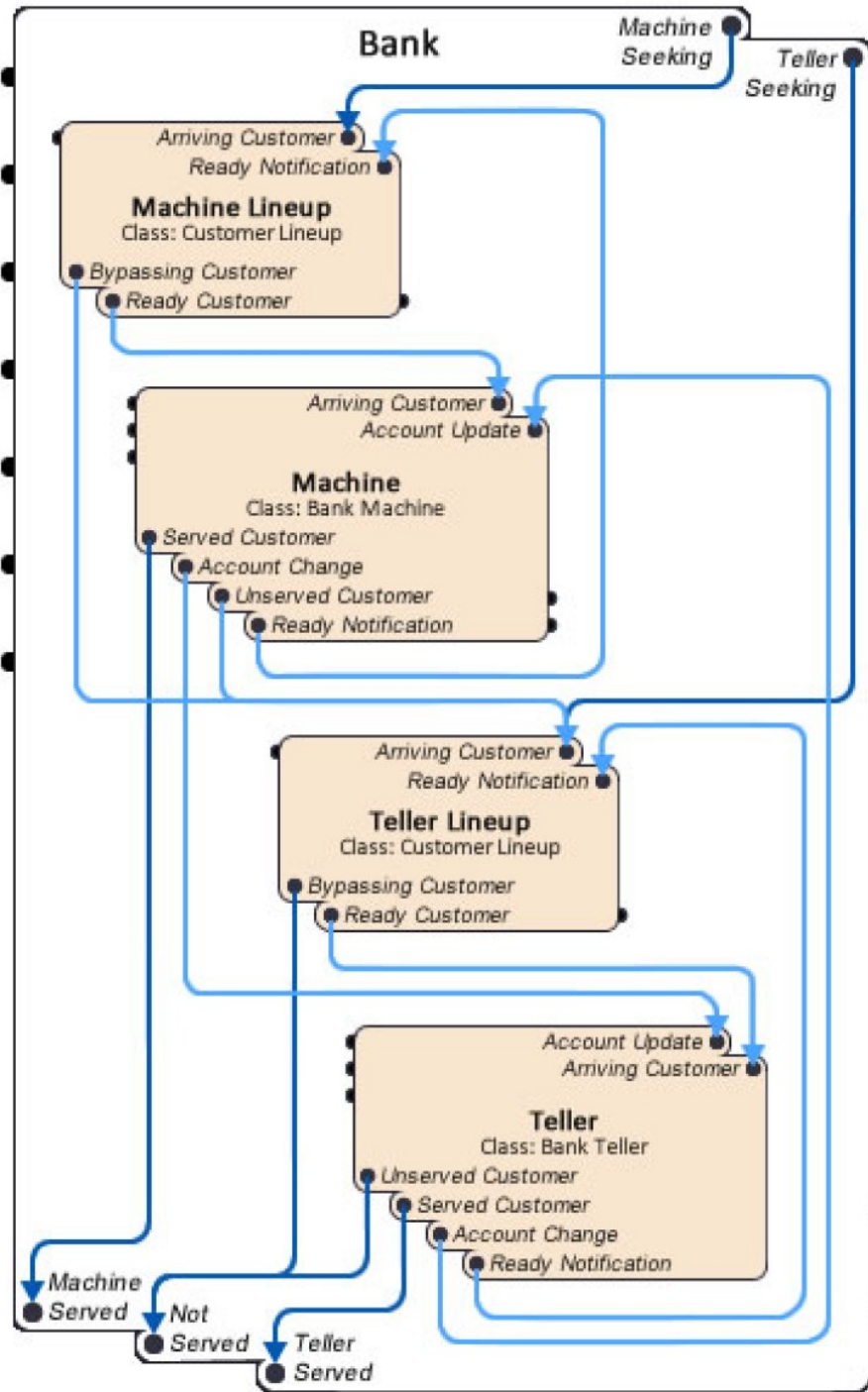
Cognitive Effectiveness?



Cognitive Effectiveness? make semantics explicit

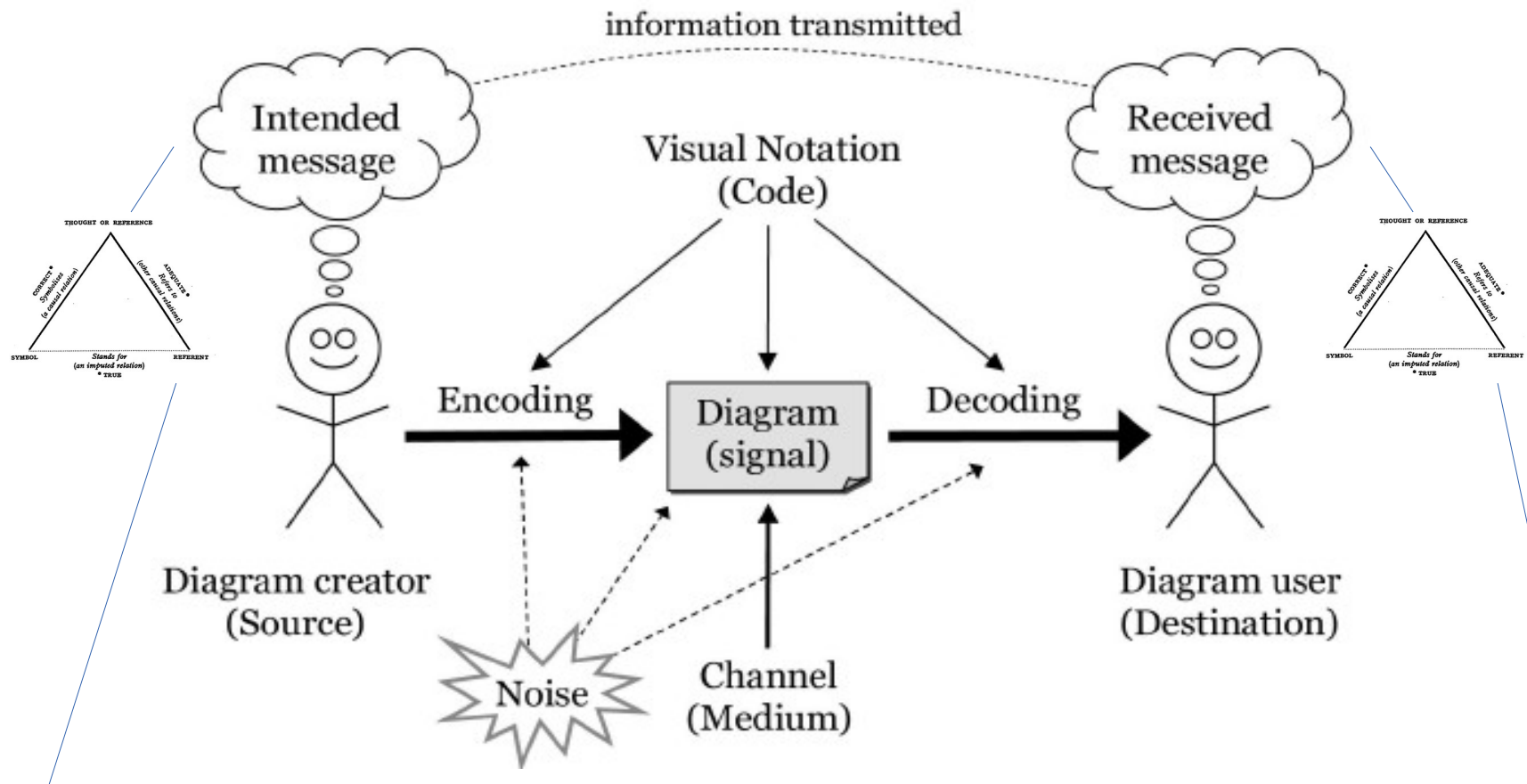
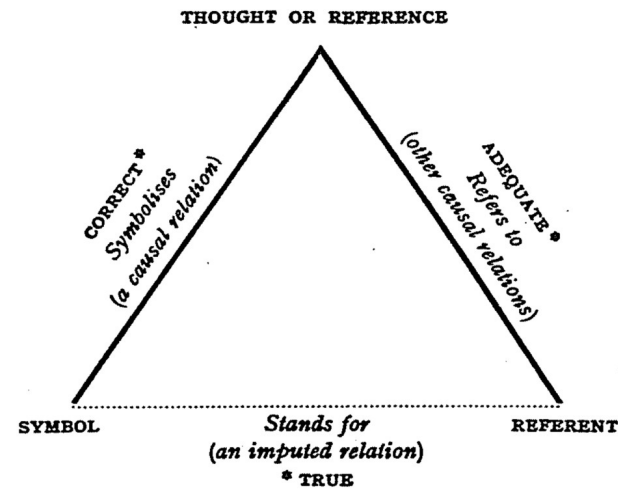


Cognitive Effectiveness?

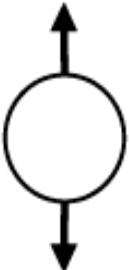
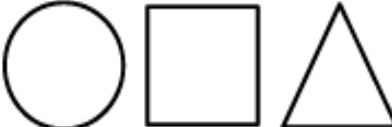
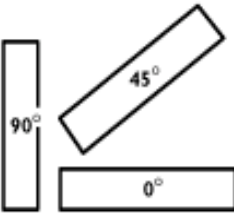
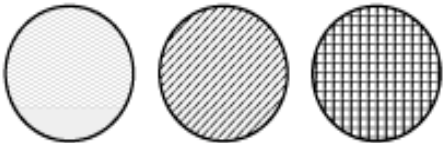


Communication Theory

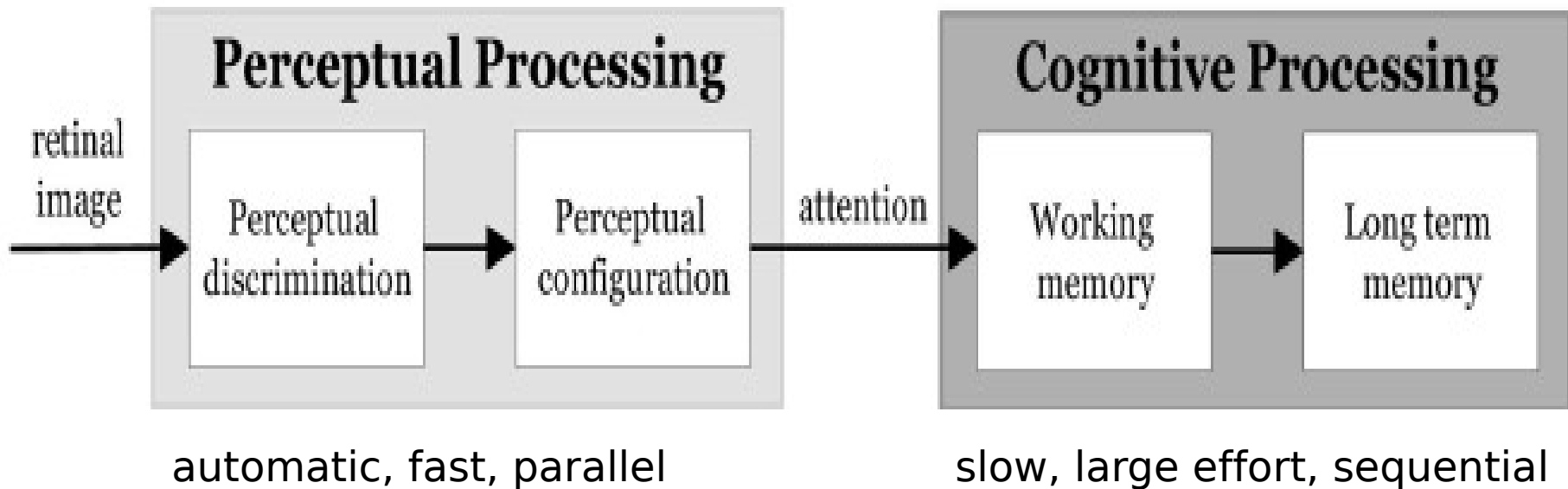
``Physics" of Notations



Encoding: 8 visual variables to (graphically) encode information

PLANAR VARIABLES	RETINAL VARIABLES		
Horizontal Position  Vertical Position 	Shape  Brightness 	Size  Orientation 	Colour  Texture 

Decoding



Appropriate notations →

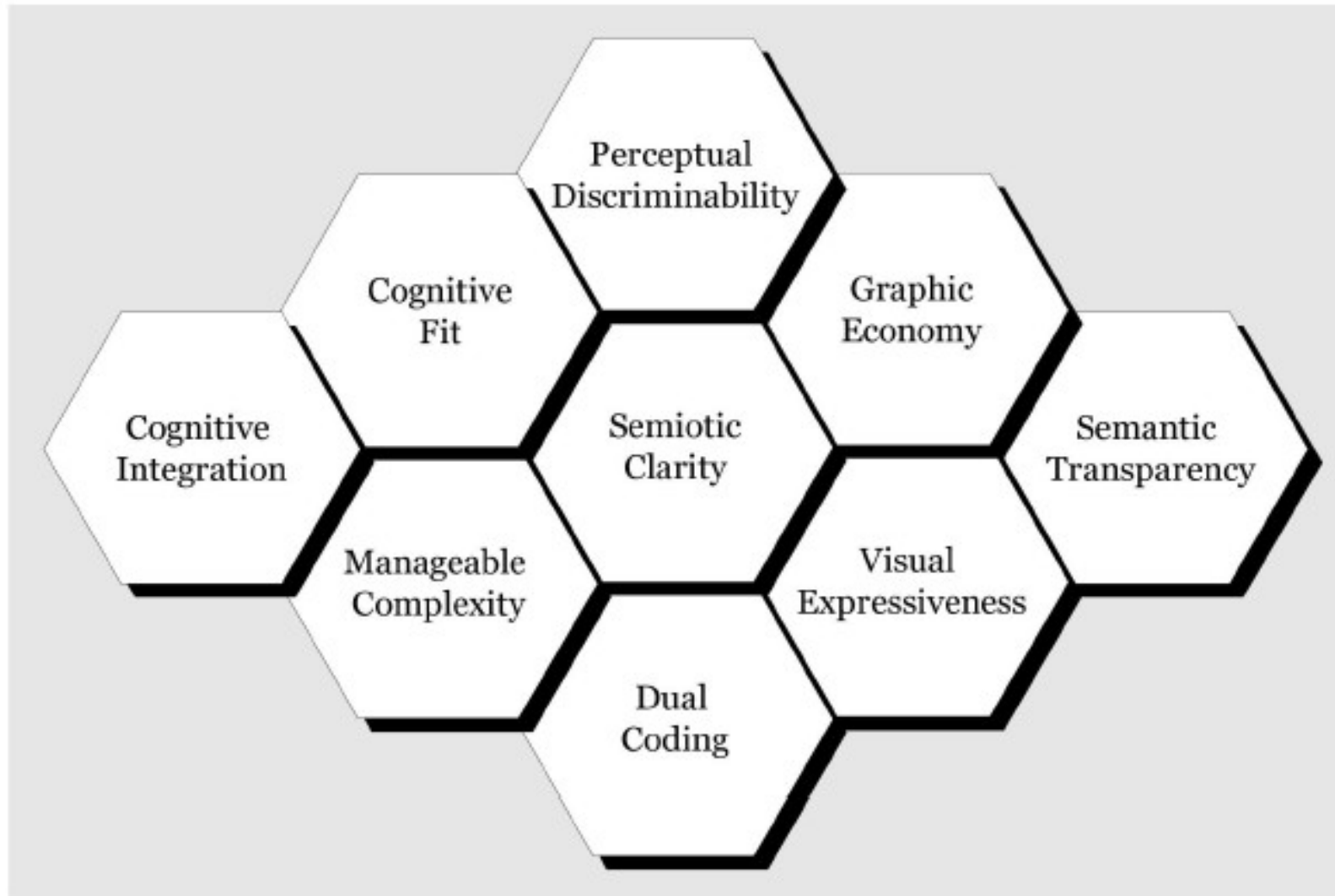
offload some of the burden from cognitive to perceptual

Note: “dual channel theory”:

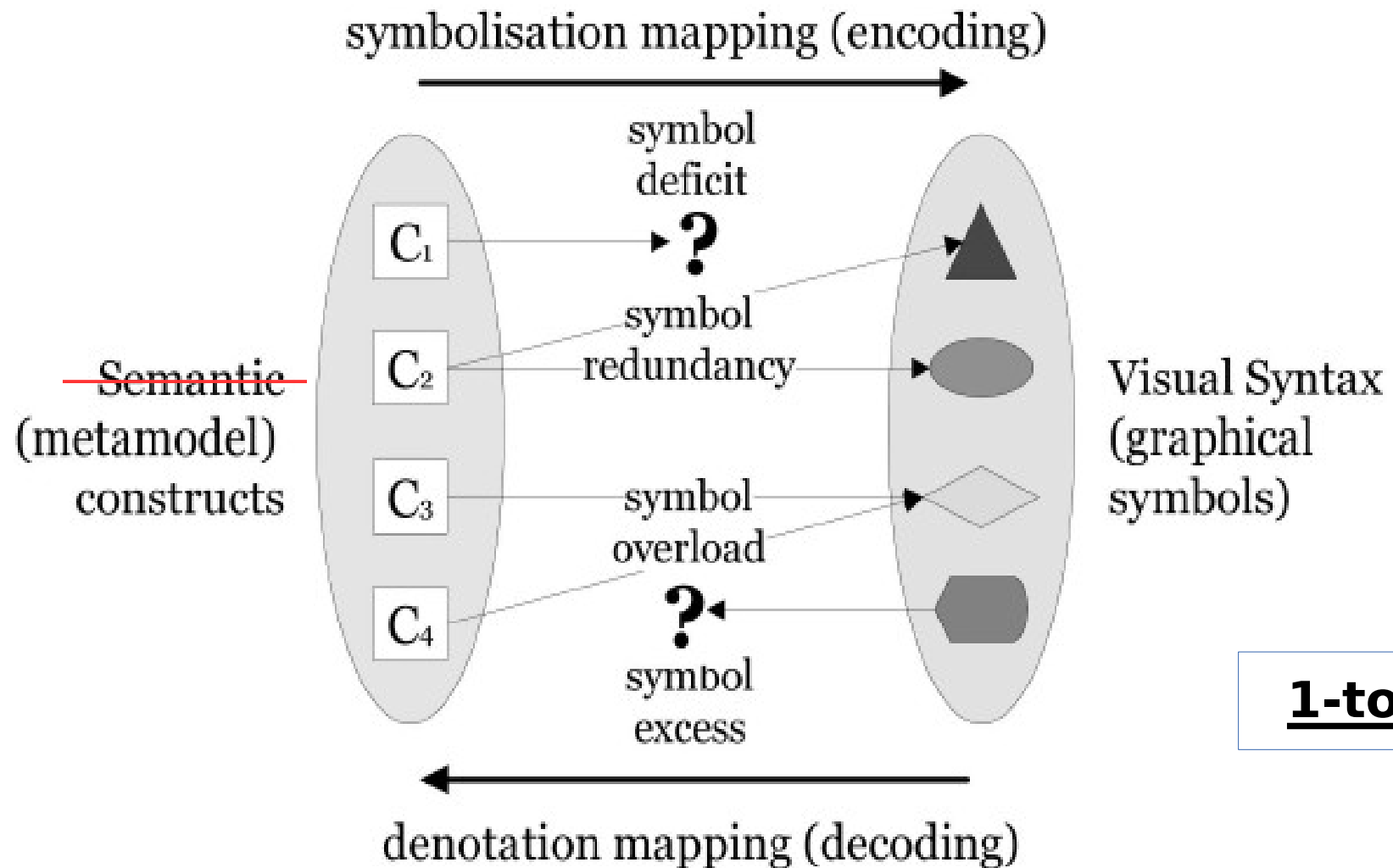
auditory/verbal channel and visual/pictorial channel are processed in parallel

Richard E. Mayer, Roxana Moreno. Nine Ways to Reduce Cognitive Load in Multimedia Learning. Educational Psychologist, 38(1), 43–5. 2003.

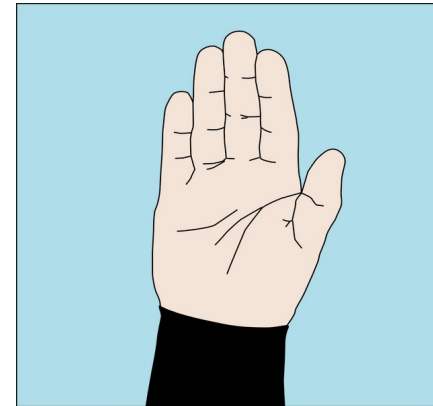
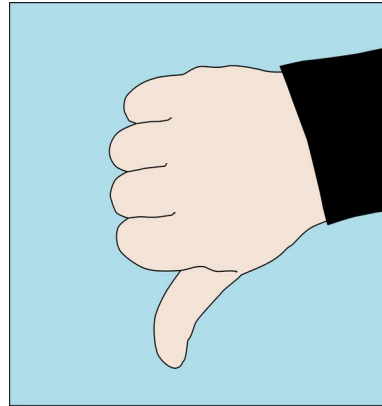
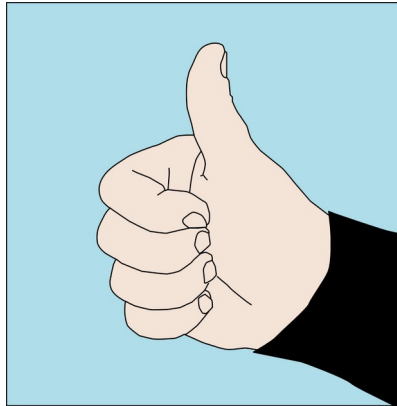
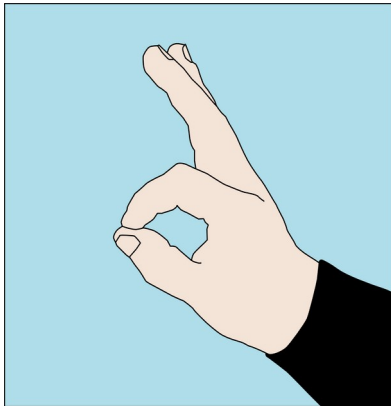
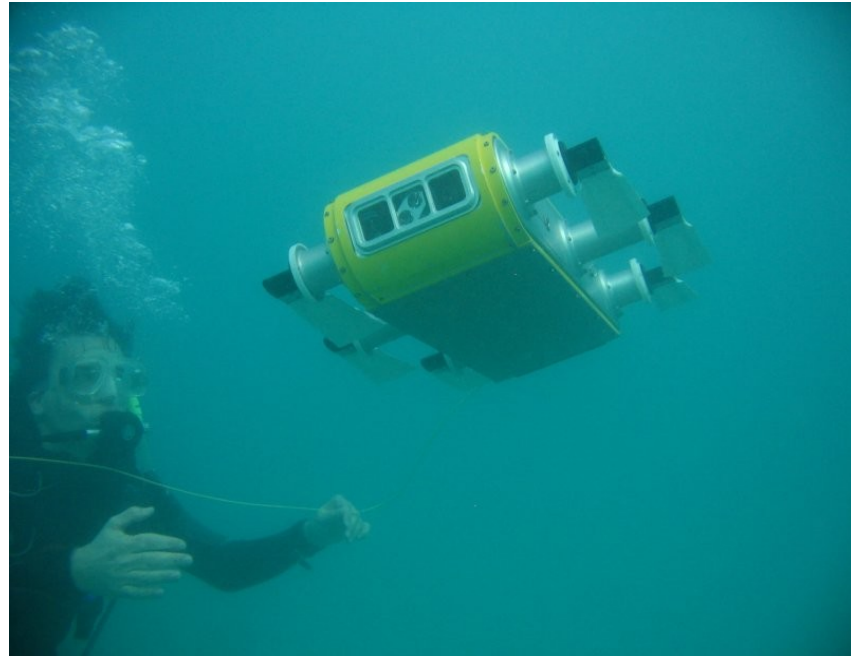
Principles for Designing Efficient and Effective Visual Notations



Semiotic Clarity (semiotics = study of signs and sign processes)



Perceptual Discriminability





(a) Divers programming Aqua2 during pool trials.



(b) A diver programming Aqua2 during an HRI trial held at a lake in central Québec.



(c) Example of command acknowledgement given on the LED screen of the Aqua2 robot during field trials.

Perceptual Discriminability

should be easy to **distinguish** visual symbols

ability to distinguish is determined by **visual distance**

larger visual distance → faster, more accurate recognition

- **number** of visual variables on which they differ and the **magnitude** of the differences
- **shape** is the main visual variable



Perceptual Discriminability

Software Engineering notations mostly use rectangle variants

Use **redundant** visual encoding to **increase distance** (e.g., textual + visual)



Semantic Transparency

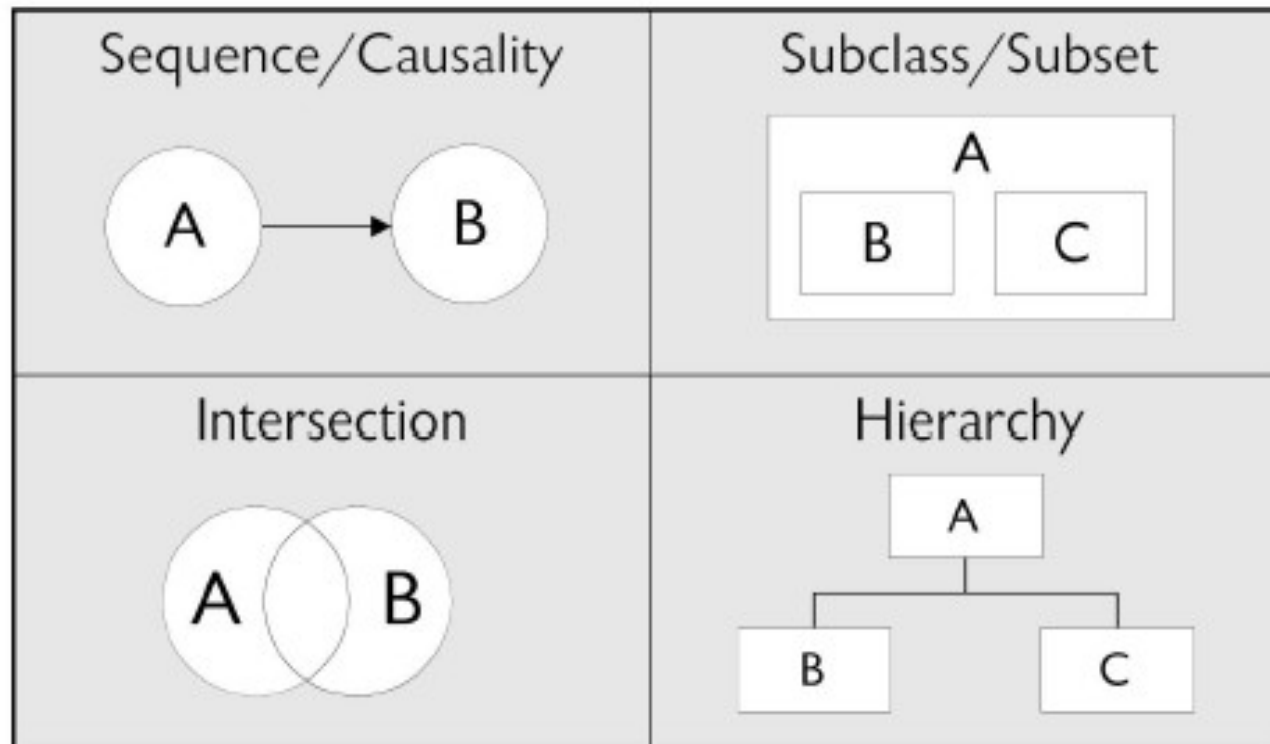
The **meaning** of a symbol can be **inferred** from its **appearance** (intuitive)

Symbols can be:

Semantic Transparency: semantically **immediate** symbols ``Physics'' of Notations



Semantic Transparency

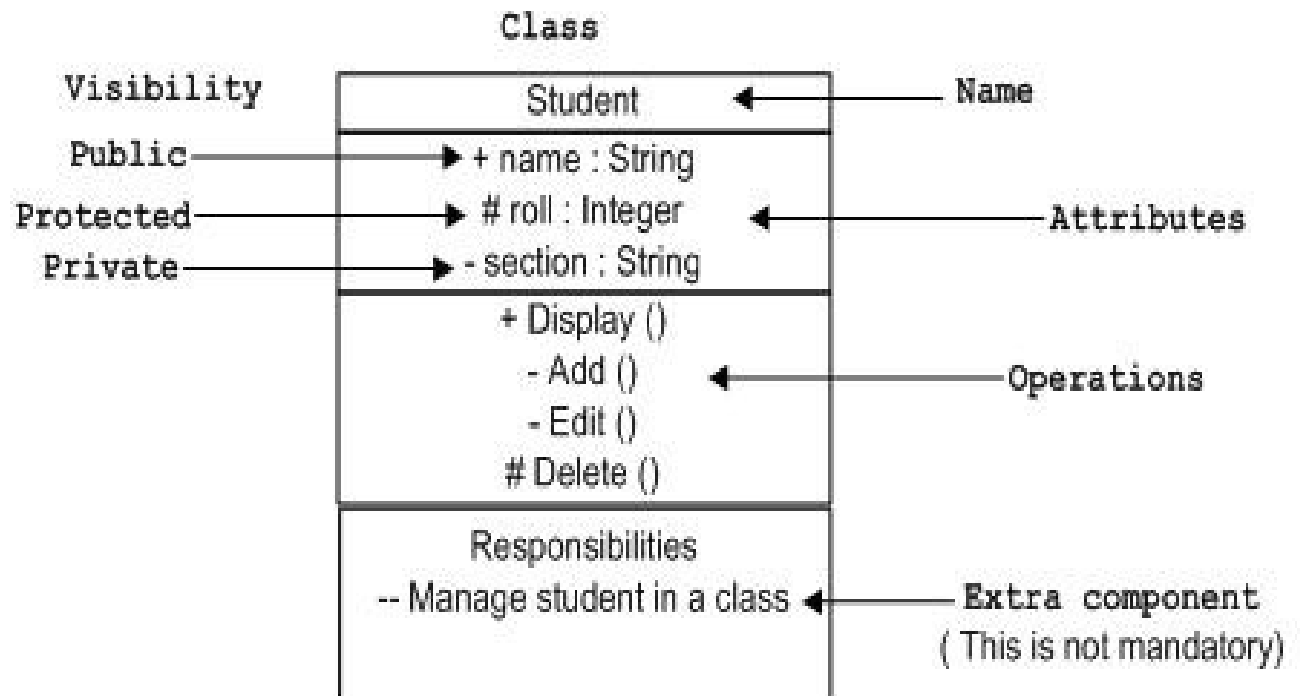


Semantic Transparency

The **meaning** of a symbol can be **inferred** from its **appearance** (intuitive)

Symbols can be:

- Semantically Immediate
- Semantically Opaque



Software Engineering
notations are usually
abstract (non-intuitive)

Semantic Transparency: semantically **perverse** symbols

``Physics" of Notations



Semantic Transparency: semantically **perverse** symbols

``Physics'' of Notations



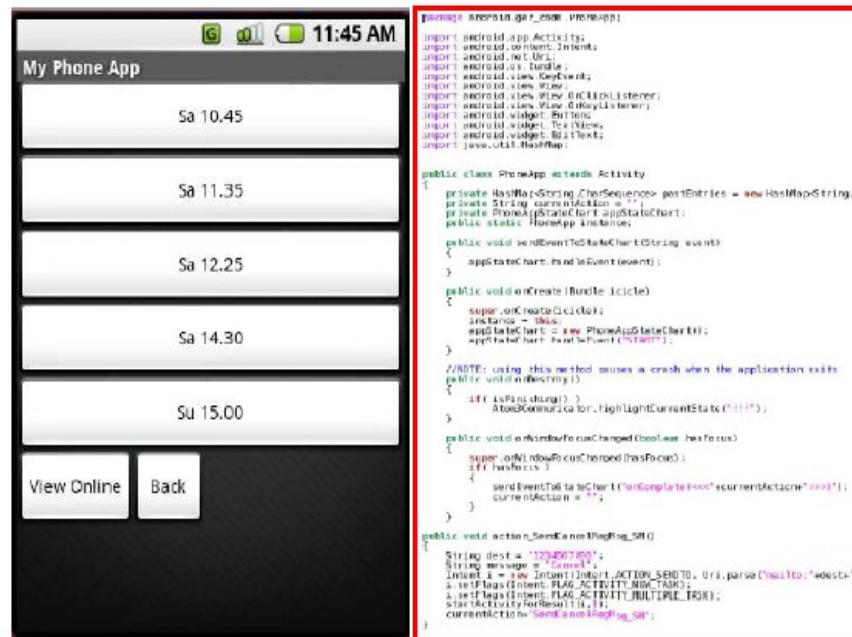
Semantic Transparency

The **meaning** of a symbol can be **inferred** from its **appearance** (intuitive)

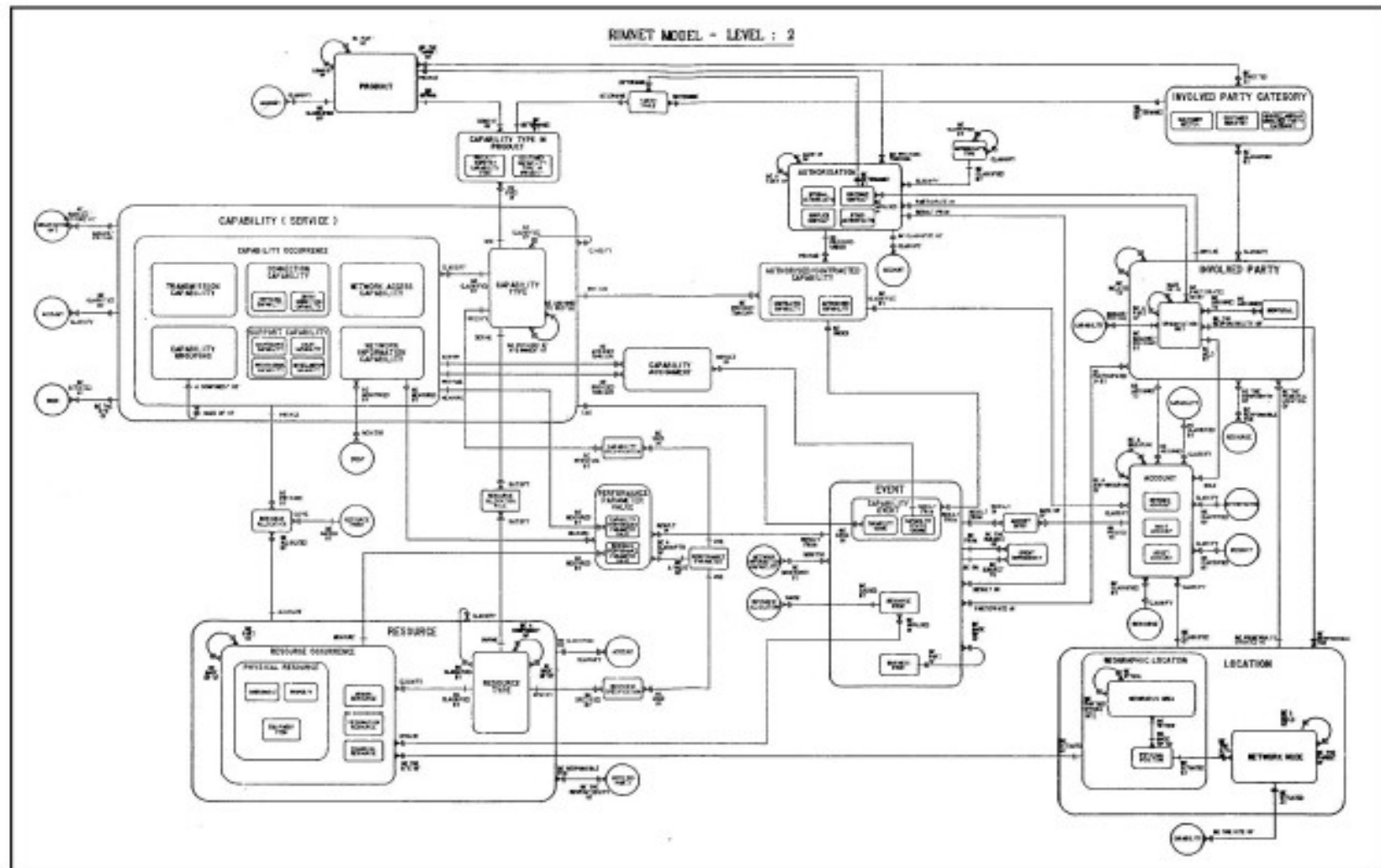
Symbols can be:

- Semantically Immediate
- Semantically Opaque
- Semantically Perverse

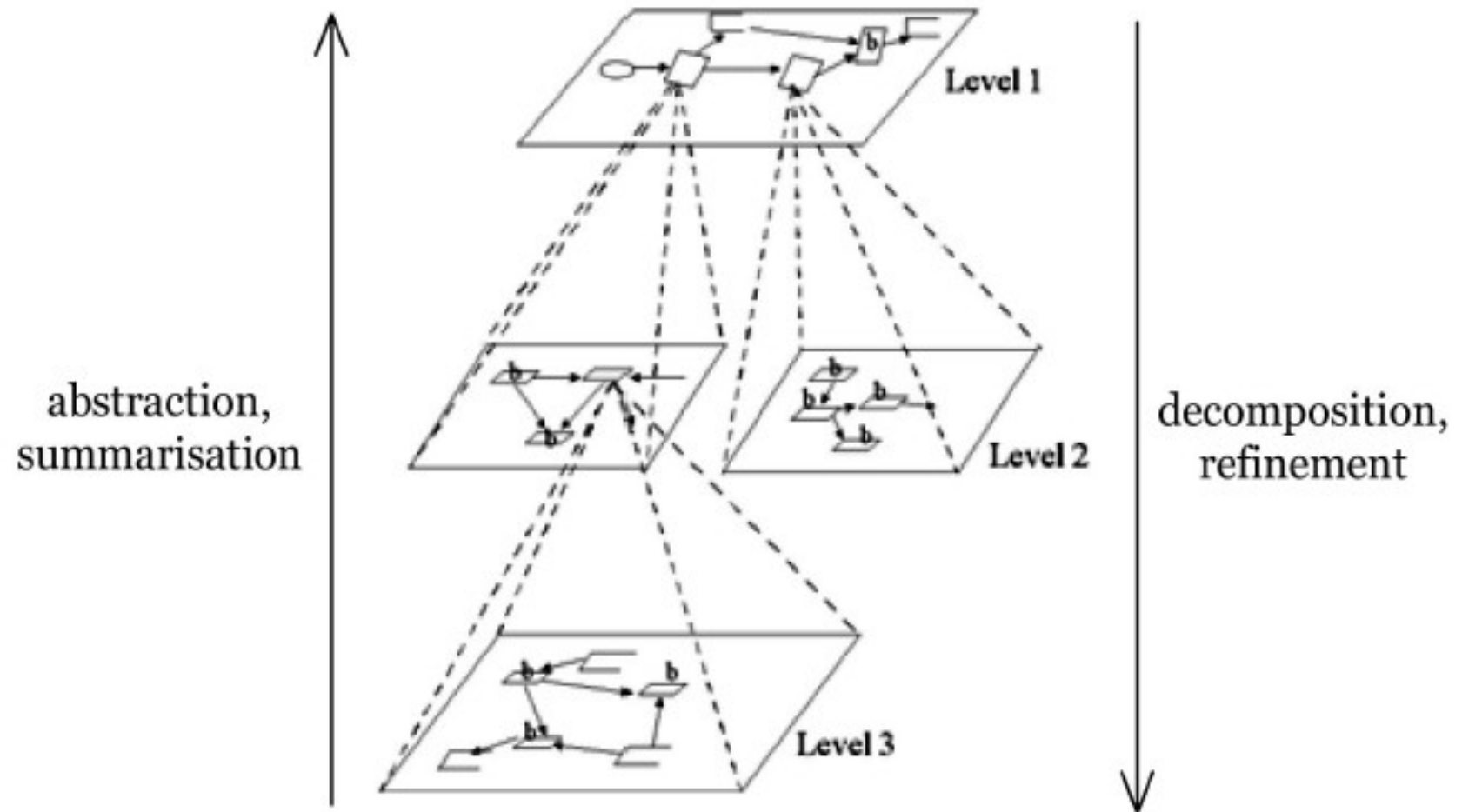
Domain-specific icons
and visual arrangement
should be intuitive



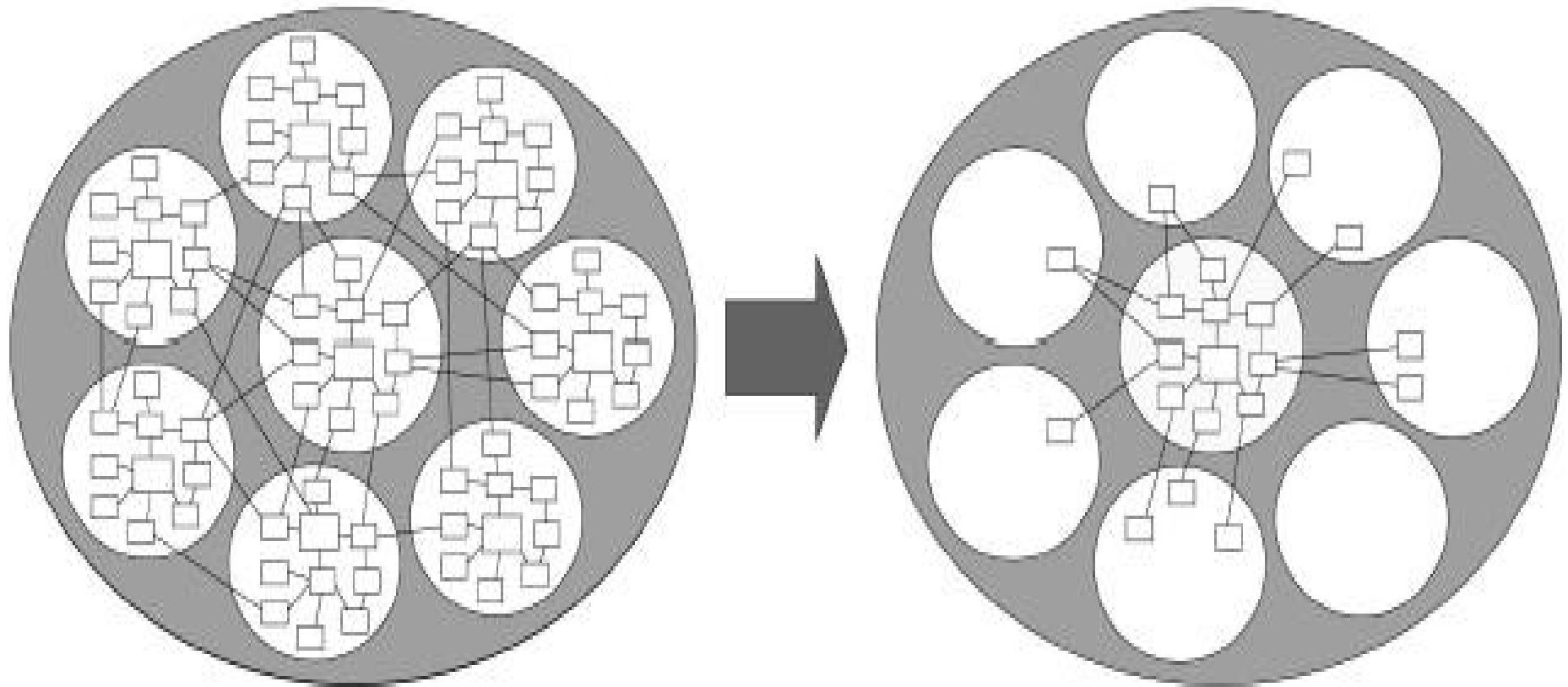
Complexity management (# elements in diagram » cognitive overload)



Modularization/Hierarchy

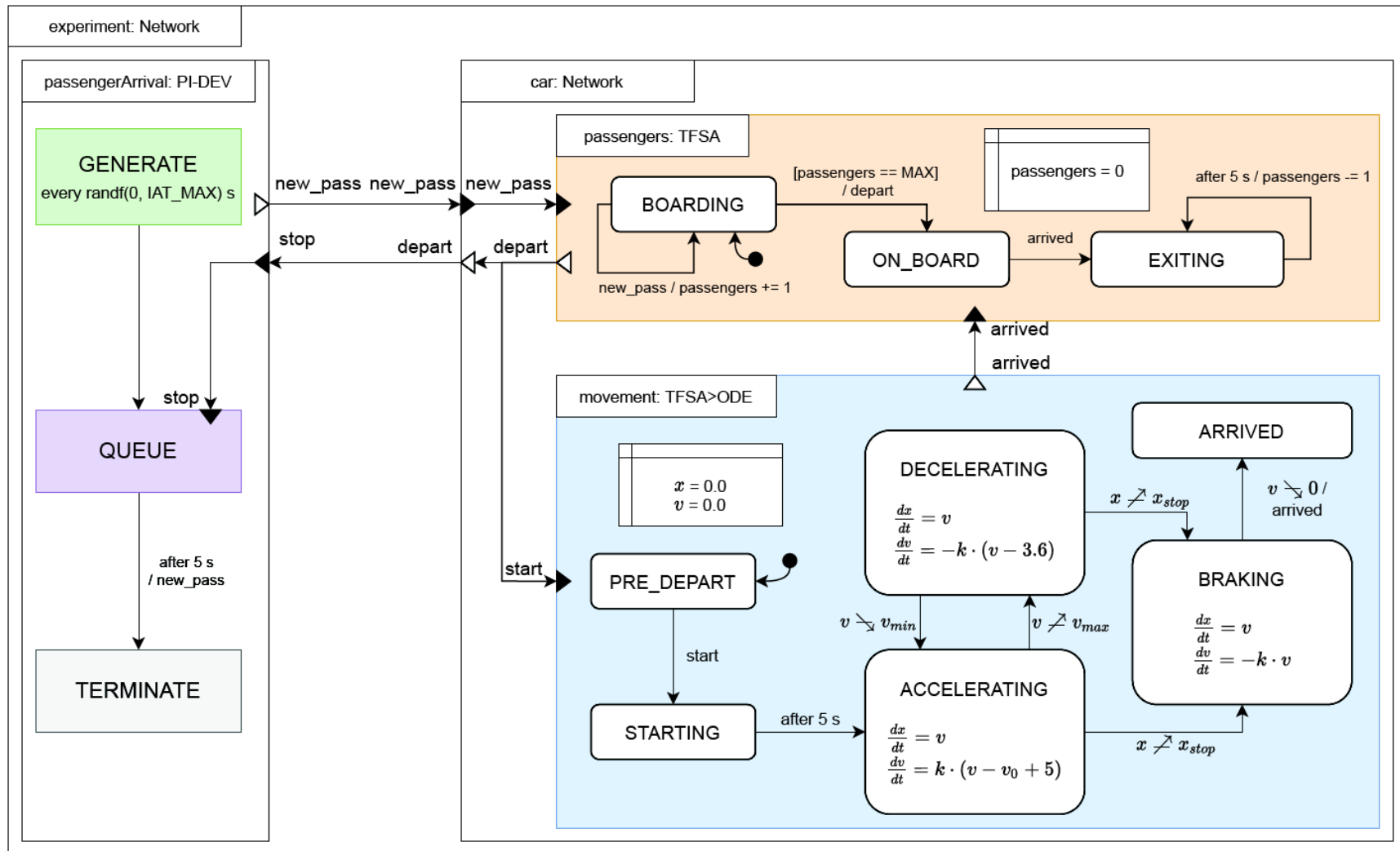


Cognitive Integration (different notations)



- Conceptual integration (**coherent** mental model)
- Enable **navigation** and **transition** between notations

Cognitive Integration (different notations)

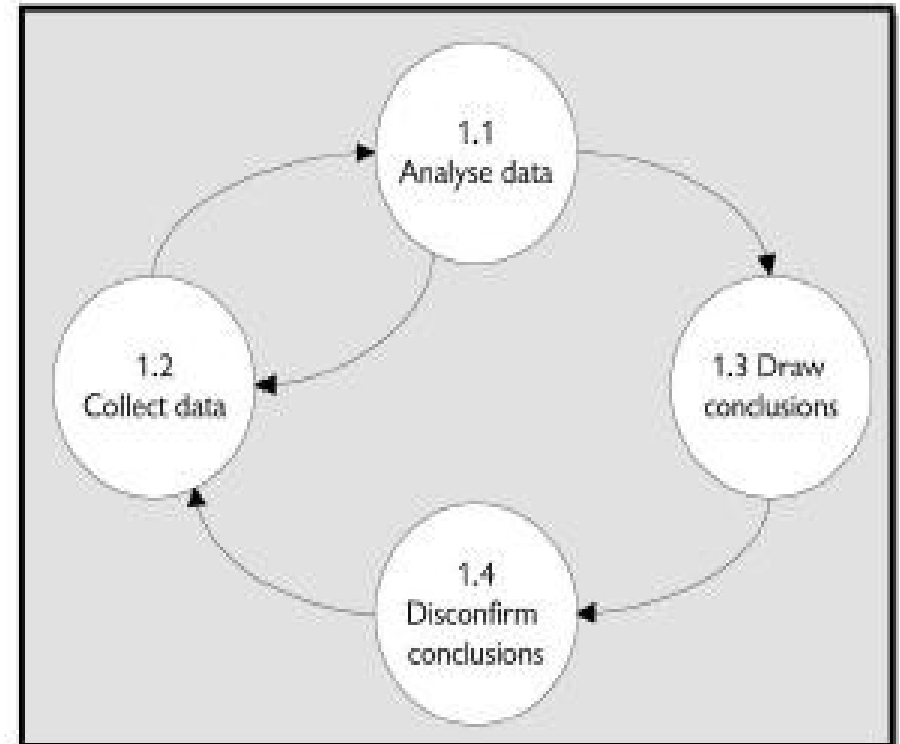
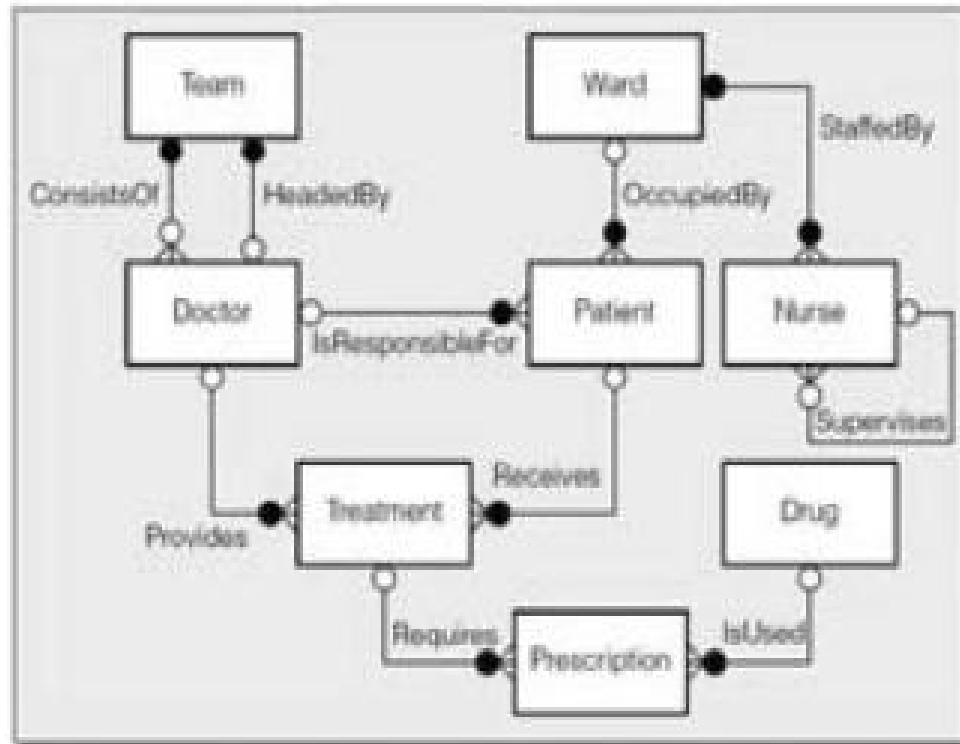


- Conceptual integration (**coherent** mental model)
- Enable **navigation** and **transition** between notations

Visual Expressiveness

Number of visual variables used (UML, mostly shape, no colour)

8 degrees of visual freedom (0 = non-visual – 8 = visually saturated)



Visual Expressiveness

Different visual variables have **different capacity** to encode information

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

Dual Encoding

Combine **Textual** and **Visual**

Supplement rather than duplicate (e.g., multiplicity values)

Graphical encoding



Textual encoding



Dual coding
(graphics+ text)



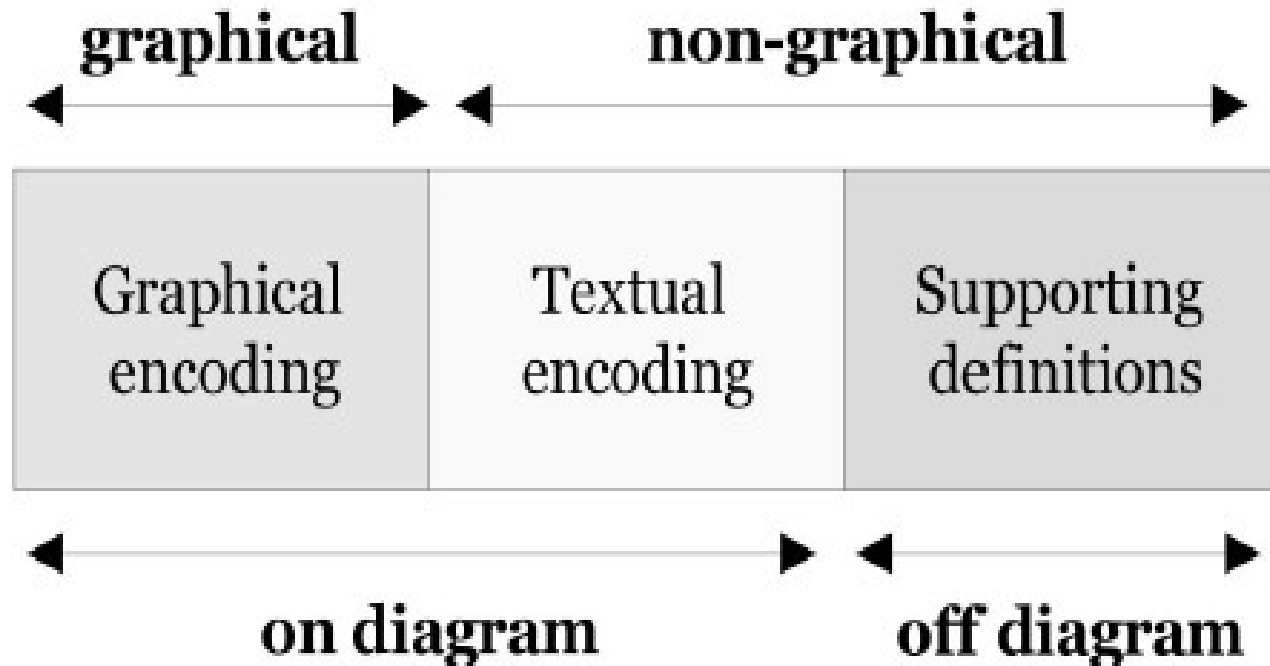
Reinforce
meaning



Graphic Economy

- Not too many symbols. If many, provide **legend**
- Limit on human discrimination capability (6 levels per variable)
- Upper limit on graphic complexity

How?



Cognitive Fit

Adapt choice of visual notation to

- Task
- Audience (e.g., novices vs. experts)

Adaptation may be dynamic (“learn” about Task/User proficiency)

Representation medium matters

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									

Ultimately, need empirical studies!

FEATURE DIAGRAMS NOTATIONAL SET

PoN-Enabled Feature Diagrams Notation [62]	Semantic Construct	Original Notation [54]
	Root	
	Feature	
	Feature	
	Mandatory	
	Optional	
	And	
	Alternative	
	Or	
	Require	
	Exclude	
	Feature cardinality	[n,m]
	Group cardinality	<n,m>
	Attribute of feature	
	Dead Feature	
	Refer feature	
	Generalization	
	Implementation	

UML STATECHART DIAGRAMS NOTATIONAL SET

PoN-Enabled UML Statecharts Notation [16]	Semantic Construct	Original Notation [53]
	Ordinary State	
	Threatened State	
	Vulnerable State	
	Defensive State	
	Compromised State	
	Quarantined State	
	Recovery State	

Ultimately, need empirical studies!

THE DEPENDENT VARIABLES AND THEIR CORRESPONDING HYPOTHESES FOR BOTH EXPERIMENTS

Dependent Variable	Null Hypothesis (Ho):	Alternative Hypothesis (Ha):
<i>Feature Diagrams Experiment</i>		
<i>Errors Committed</i>	$E(\text{FD-ON}) = E(\text{FD-PoN})$	$E(\text{FD-ON}) \neq E(\text{FD-PoN})$
<i>Completion Time</i>	$T(\text{FD-ON}) = T(\text{FD-PoN})$	$T(\text{FD-ON}) \neq T(\text{FD-PoN})$
<i>Misuse Case Diagrams Experiment</i>		
<i>Errors Committed</i>	$E(\text{MUCD-ON}) = E(\text{MUCD-PoN})$	$E(\text{MUCD-ON}) \neq E(\text{MUCD-PoN})$
<i>Completion Time</i>	$T(\text{MUCD-ON}) = T(\text{MUCD-PoN})$	$T(\text{MUCD-ON}) \neq T(\text{MUCD-PoN})$
<i>UML Statechart Diagrams Experiment</i>		
<i>Errors Committed</i>	$E(\text{SC-ON}) = E(\text{SC-PoN})$	$E(\text{SC-ON}) \neq E(\text{SC-PoN})$
<i>Completion Time</i>	$T(\text{SC-ON}) = T(\text{SC-PoN})$	$T(\text{SC-ON}) \neq T(\text{SC-PoN})$

The qualitative data within this category also indicate that while users in general enjoy using colors in comparison to just using black and white notations, users were annoyed by diagrams that have too many colors.

However, speed and effort in using PoN-enabled notations is compensated for once the initial learning curve is completed (i.e., the symbol meanings are now loaded into the working memory) and when users verify their diagrams.

