Modelling Languages:
(mostly) Concrete (Visual) Syntax

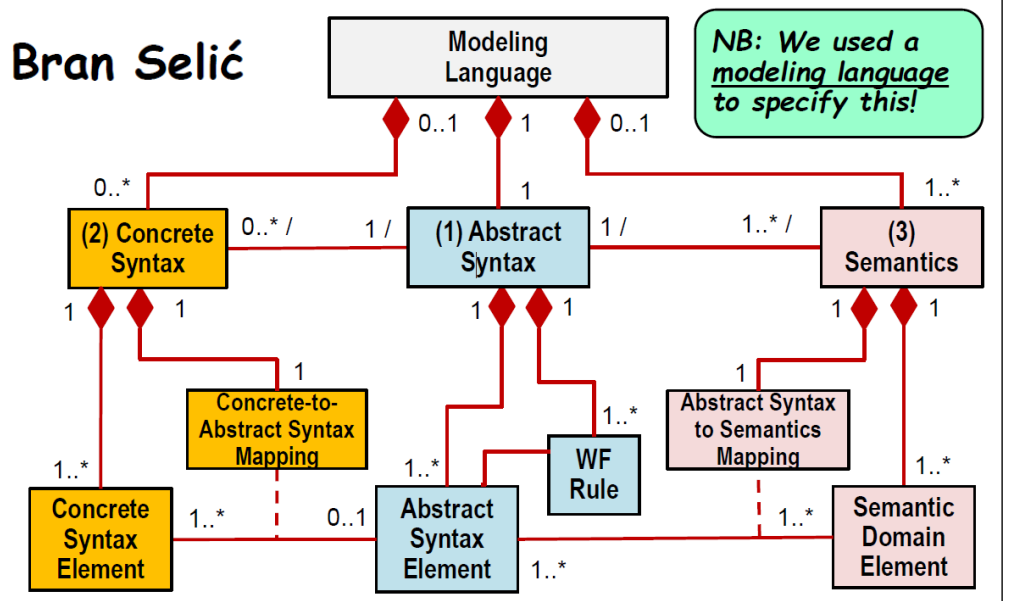
Hans Vangheluwe

http://msdl.cs.mcgill.ca/

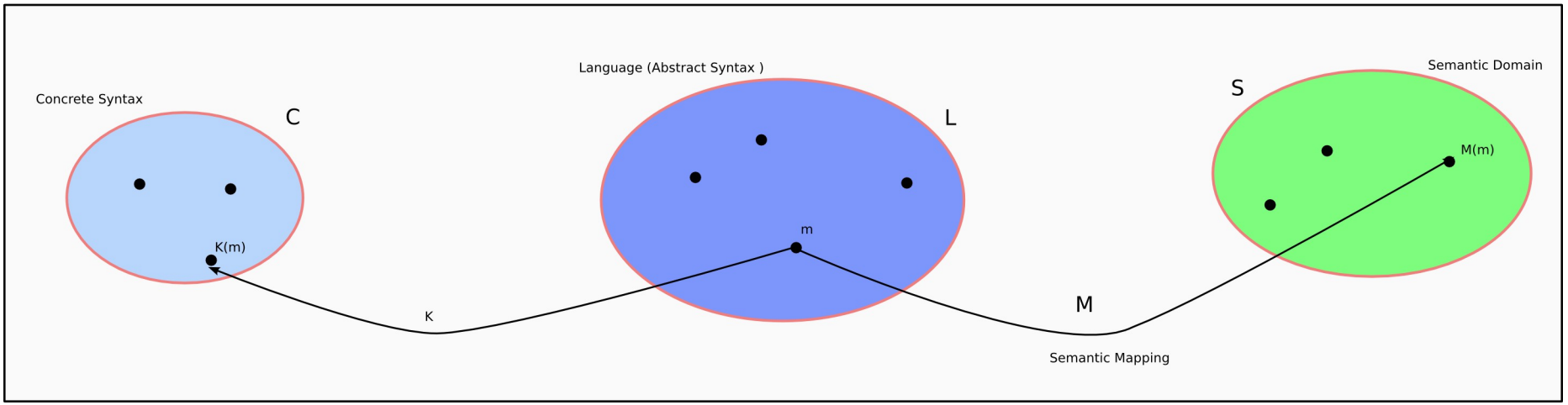
Universiteit Antwerpen
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp
Nexor
Cyber-Physical Systems
University of Antwerp
FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK
McGill

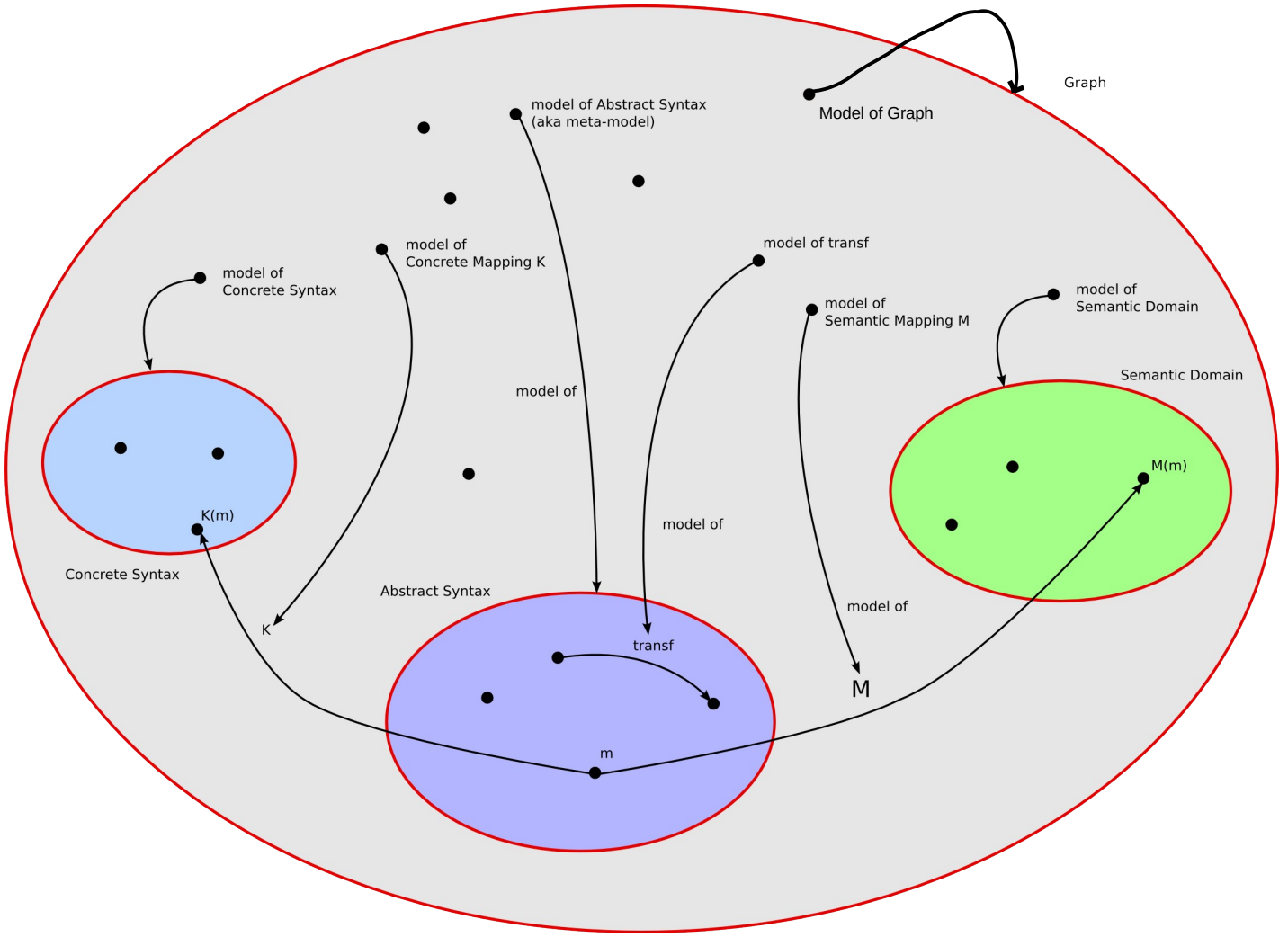
The Structure of Modeling Languages

Bran Selić



Concrete Formalism F




Graph
Model of Graph
model of Abstract Syntax
(aka meta-model)
model of
Concrete Mapping K
model of transf
model of
Concrete Syntax
model of
Semantic Mapping M
model of
Semantic Domain
Semantic Domain
model of
M(m)
K(m)
model of
Concrete Syntax
Abstract Syntax
model of
K
transf
M
m

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 “is to the right of” spatial relationship

The Spoofax Language Workbench

Rules for Declarative Specification of Languages and IDEs

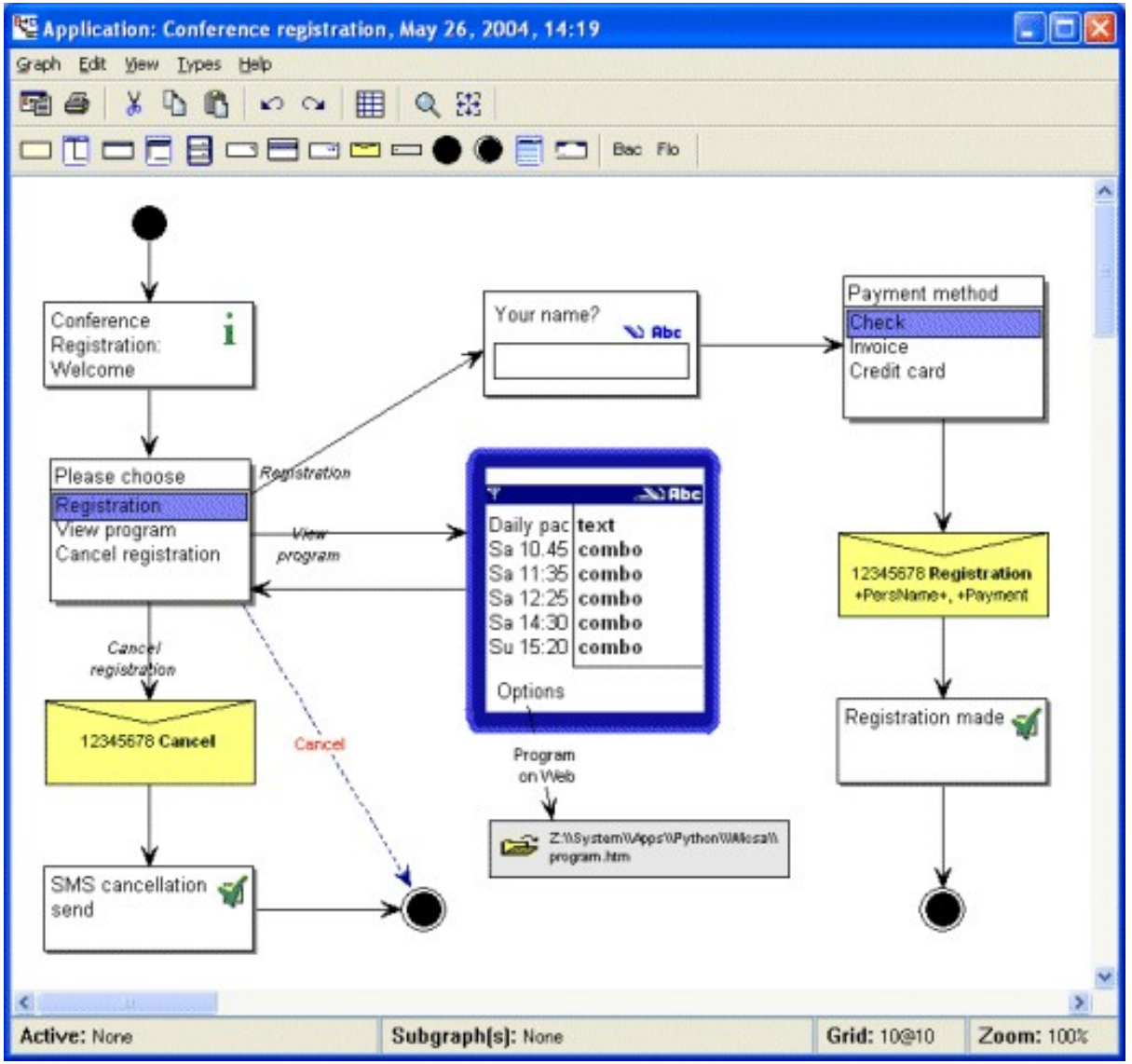
Report TUD-SERG-2010-014a

Lennart C. L. Kats
Delft University of Technology
l.c.l.kats@tudelft.nl

Eelco Visser
Delft University of Technology
visser@acm.org

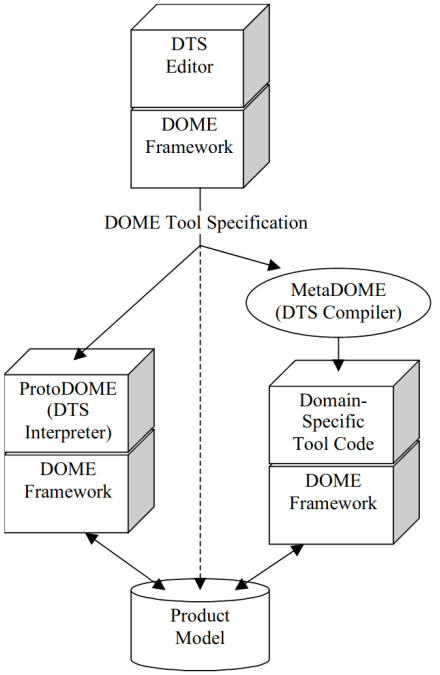
syntax-directed editor
(textual concrete syntax)

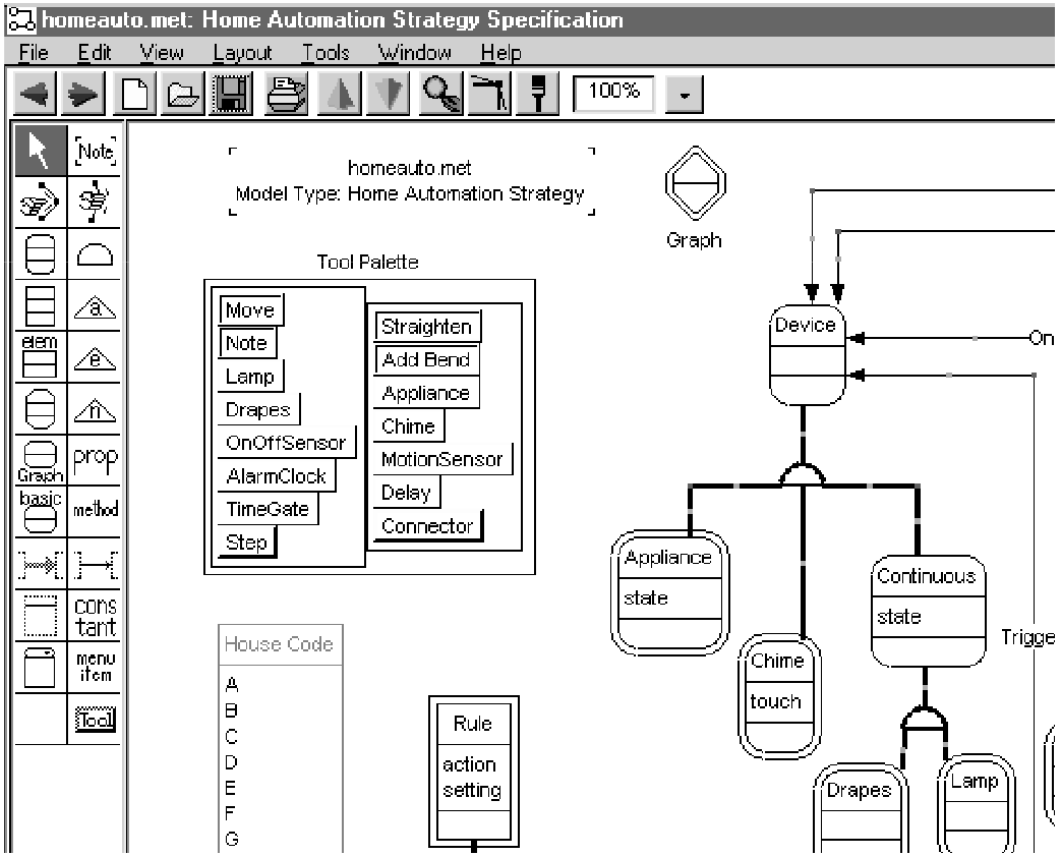
syntax-directed editor
(visual concrete syntax)

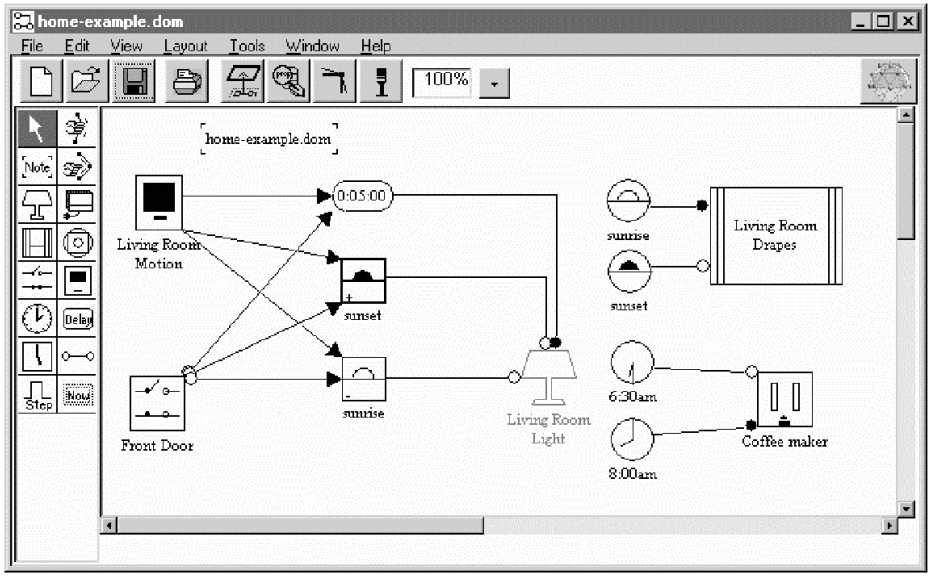


1990s: Honeywell's DoME

Domain Modeling Environment







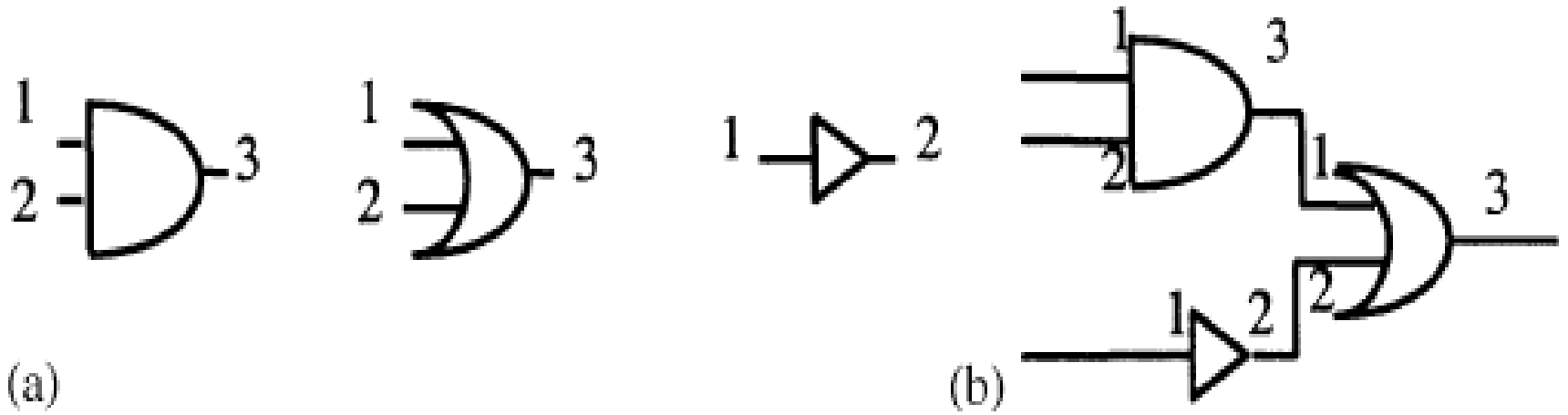
http://msdl.uantwerpen.be/people/mosterman/campam/cca01/cacsd00a/index.html/ek.pdf

Journal of Visual Languages and Computing (2002) **13,** 573–600
doi:10.1006/S1045-926X(02)00025-3 available online at http://www.idealibrary.com on IDEAL®

A Classification Framework to Support the Design of Visual Languages

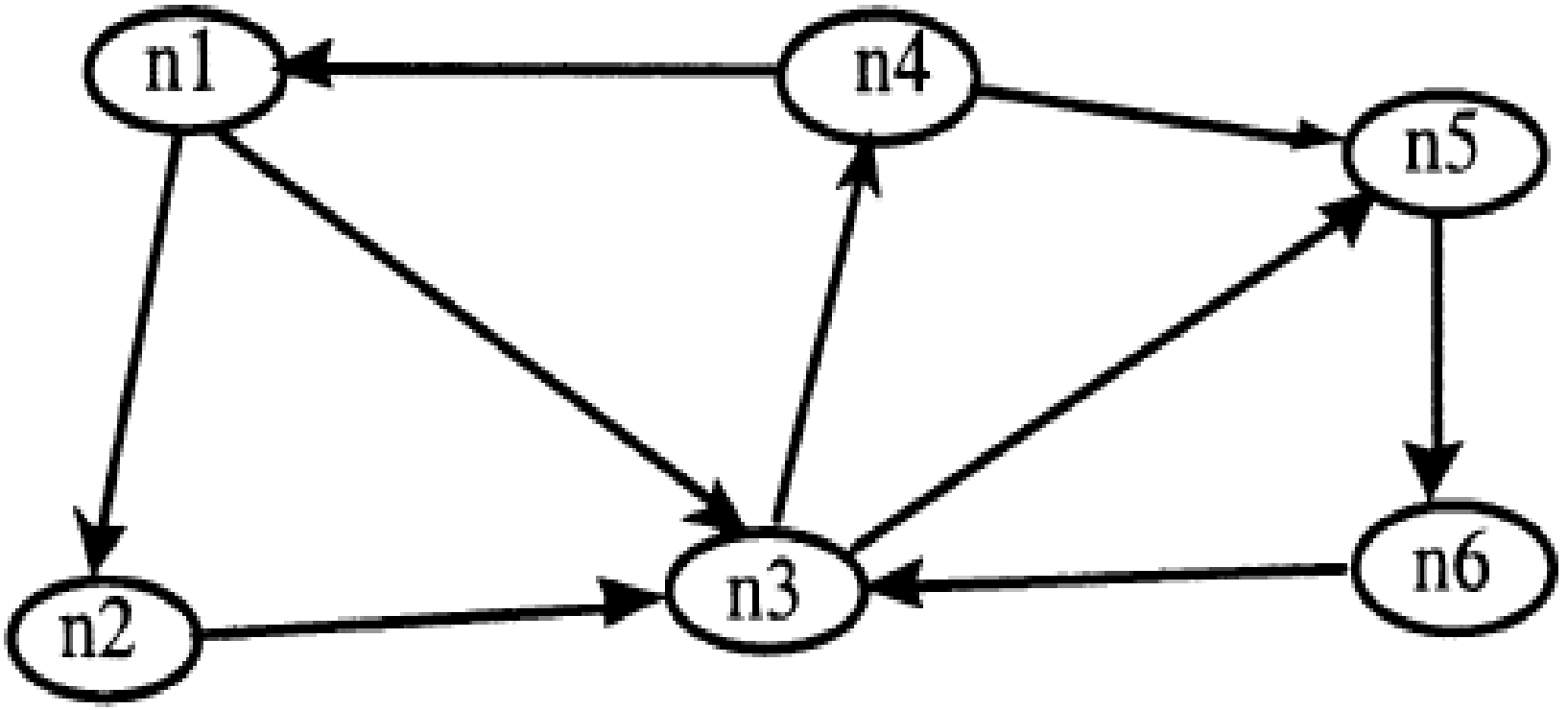
G. Costagliola*, A. Delucia†, S. Orefice‡ and G. Polese*

Plex

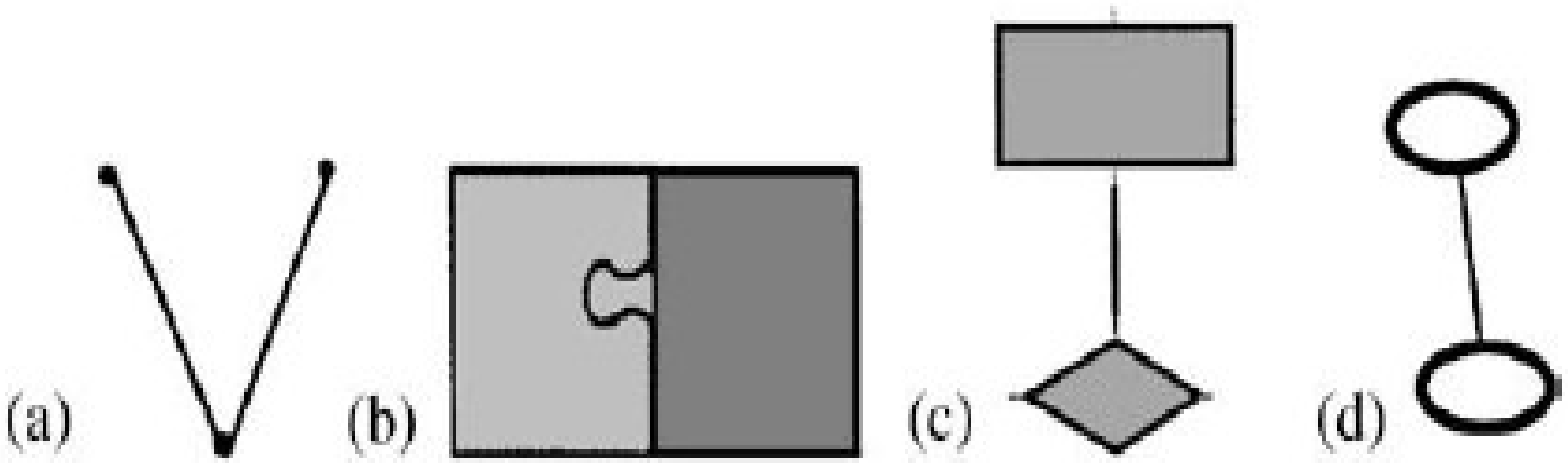


meta-model of network of blocks with “ports”?

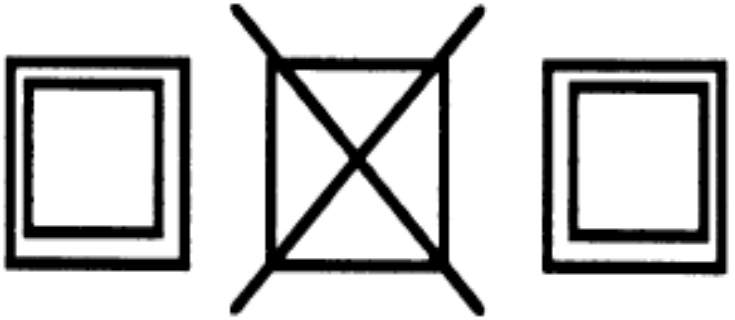
Graph

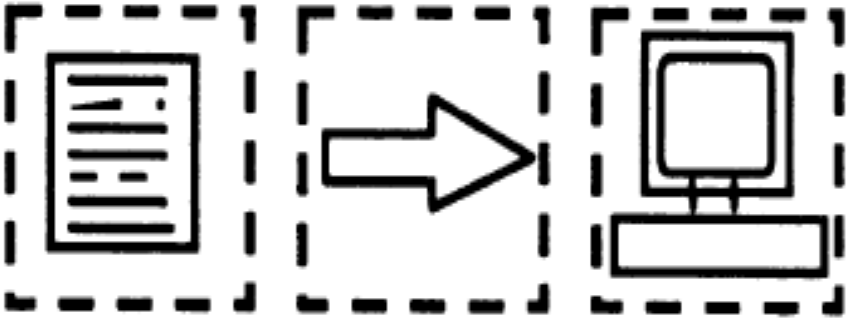


Connection Types

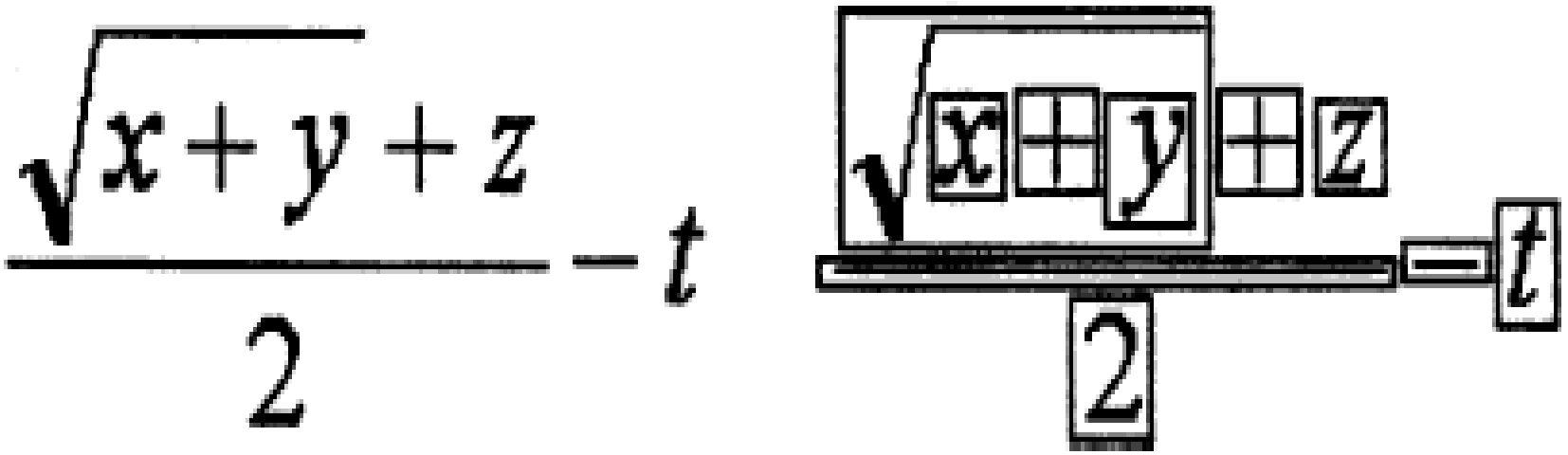


Iconic

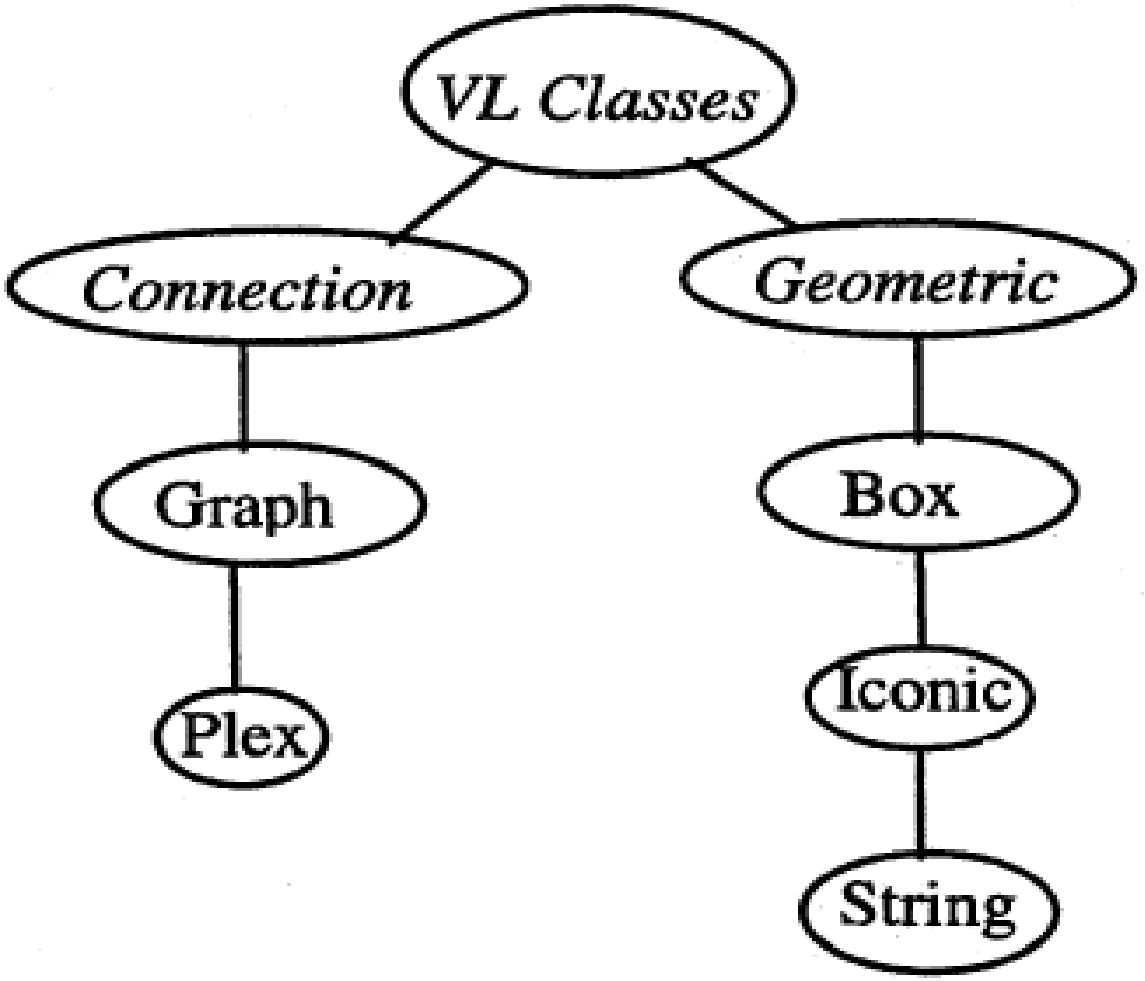




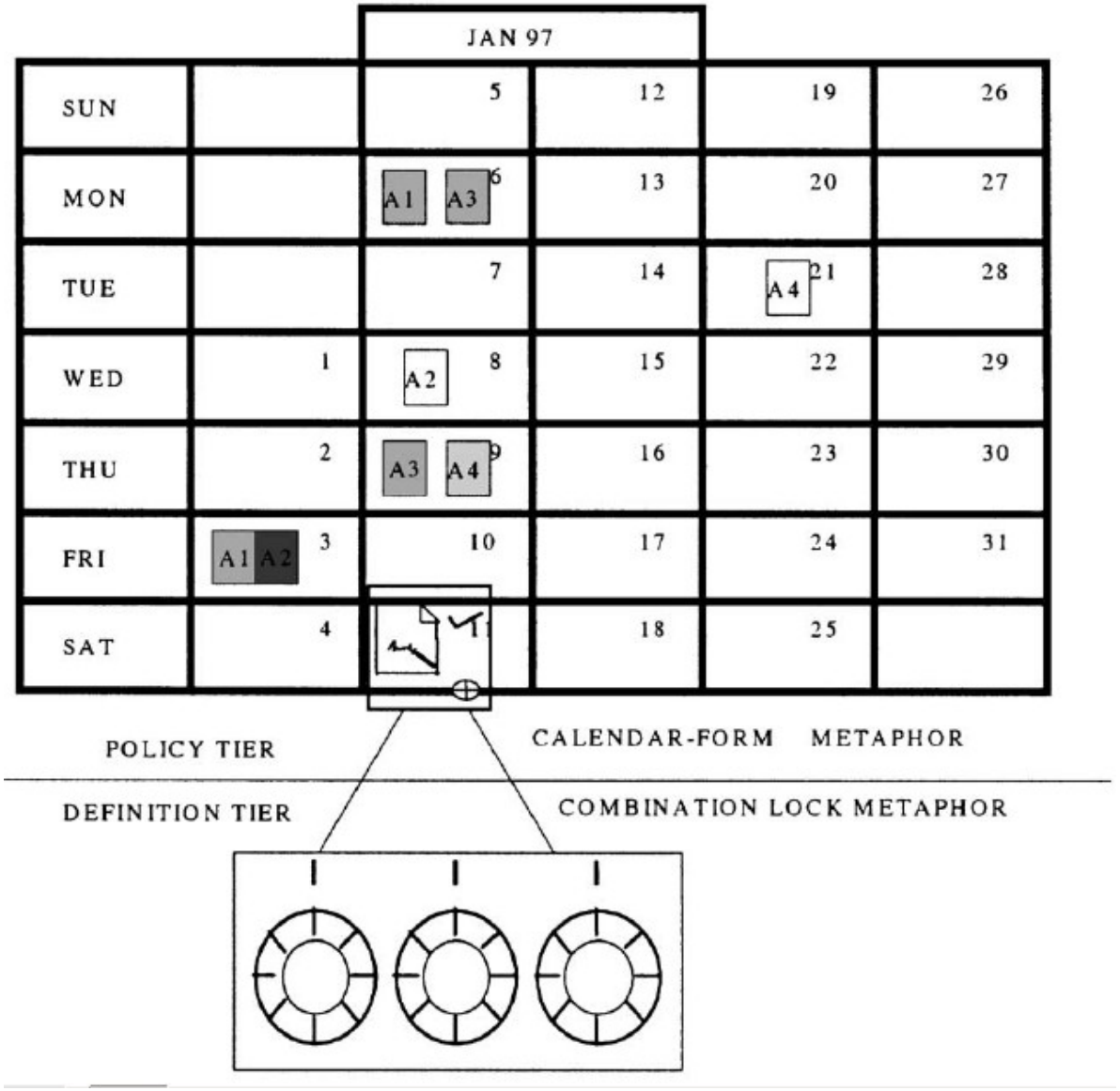
Box



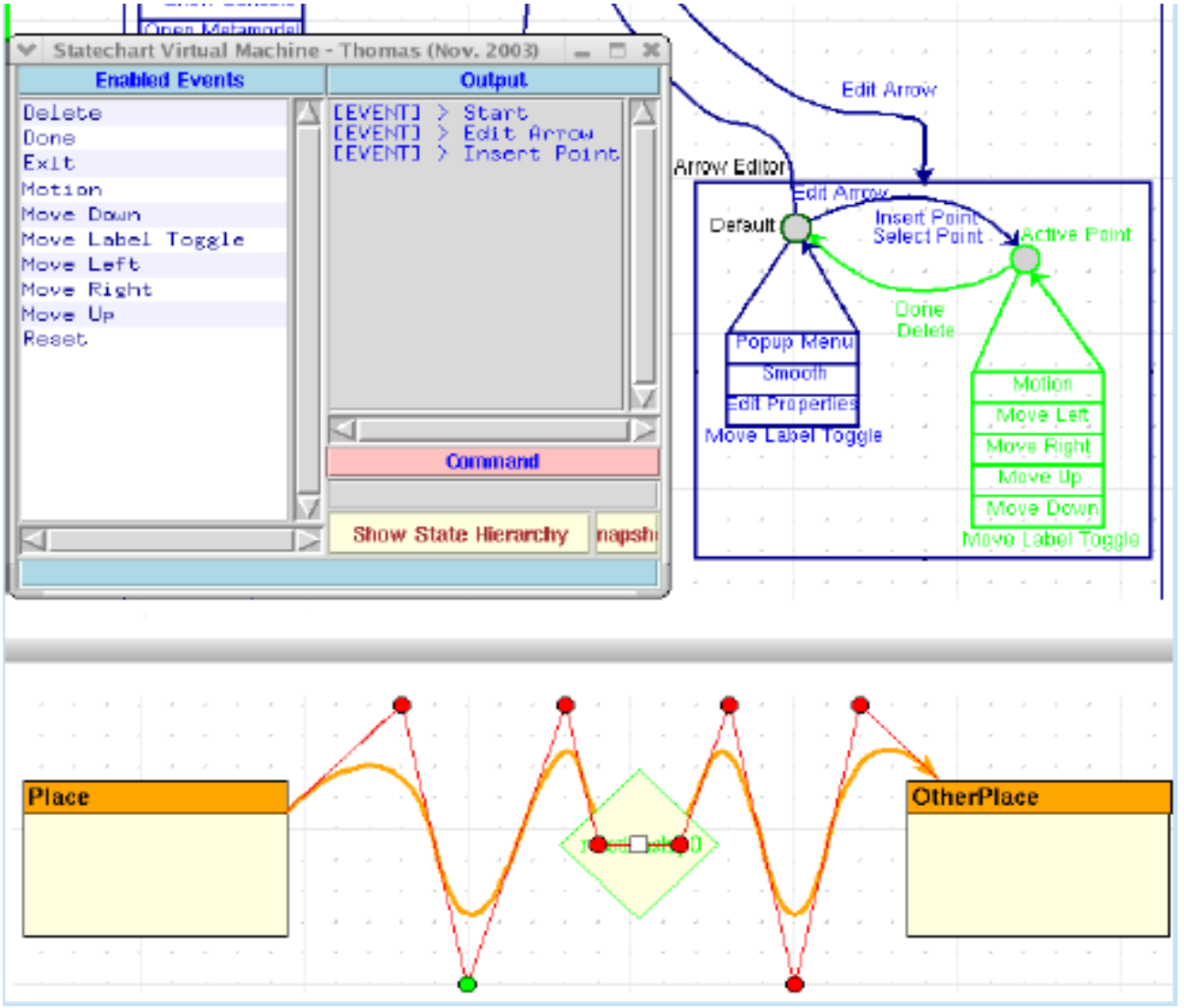
Visual Language Classes



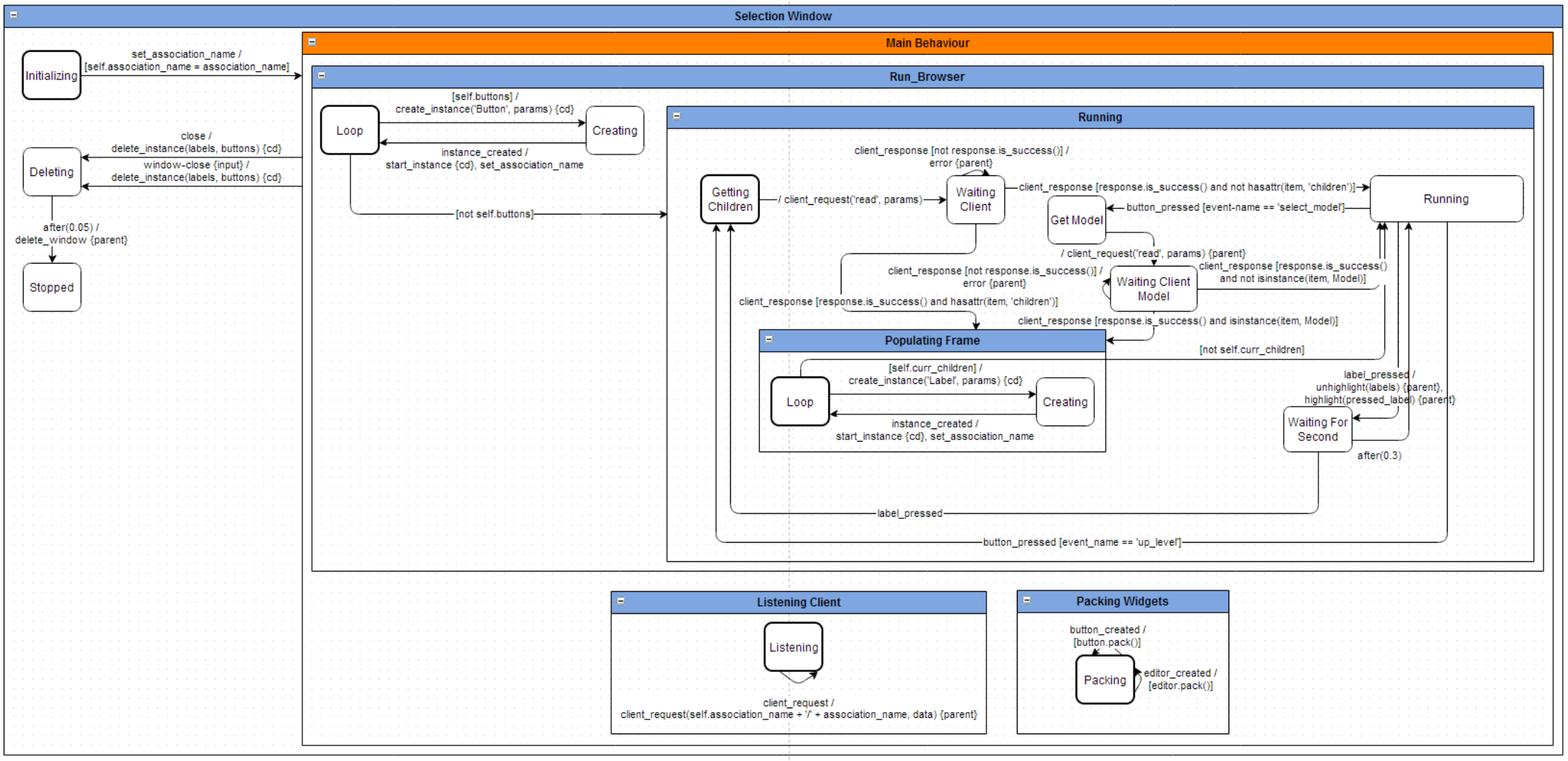
Hybrid Languages



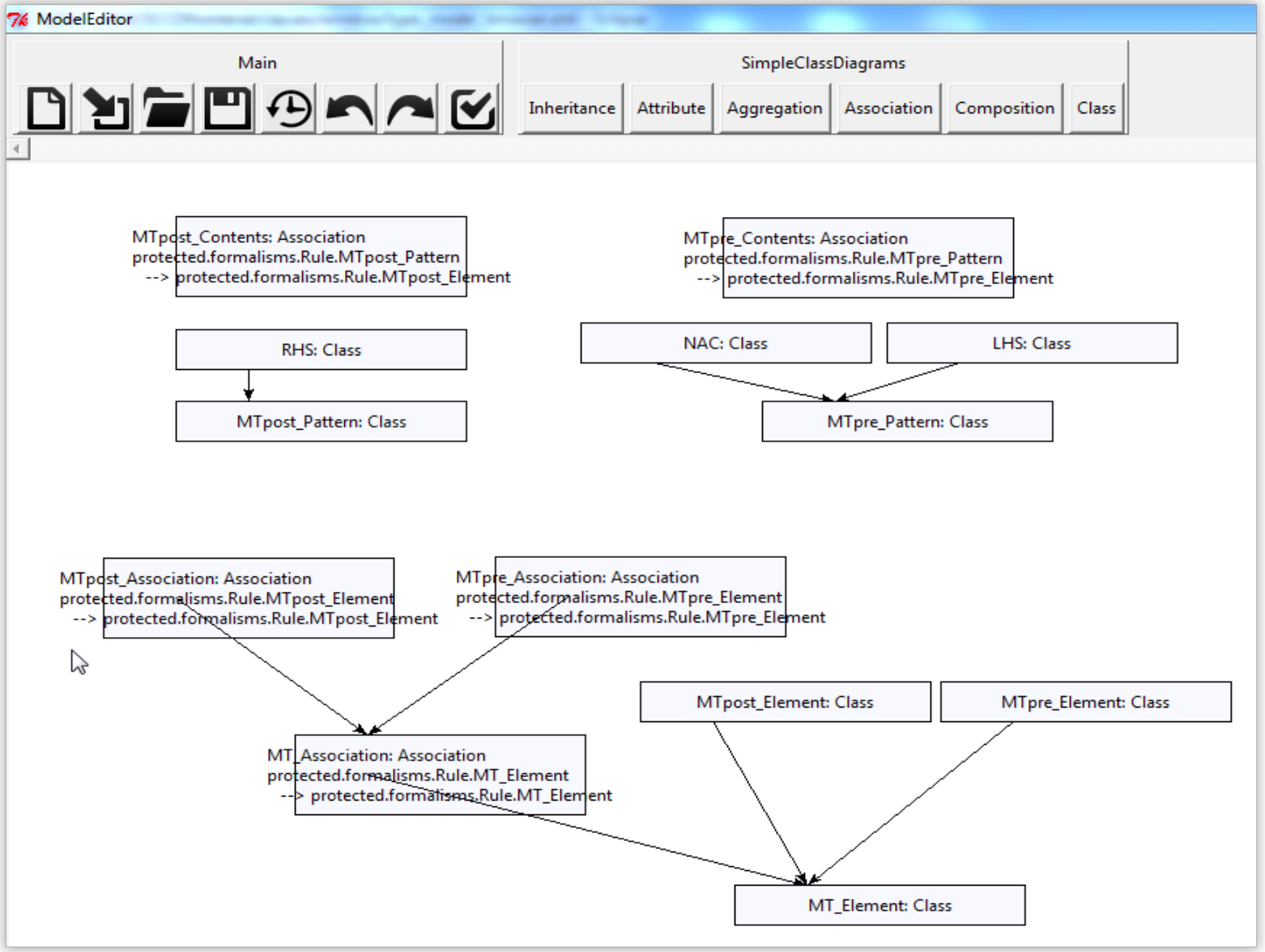
Syntax-directed Visual Editors: model behaviour



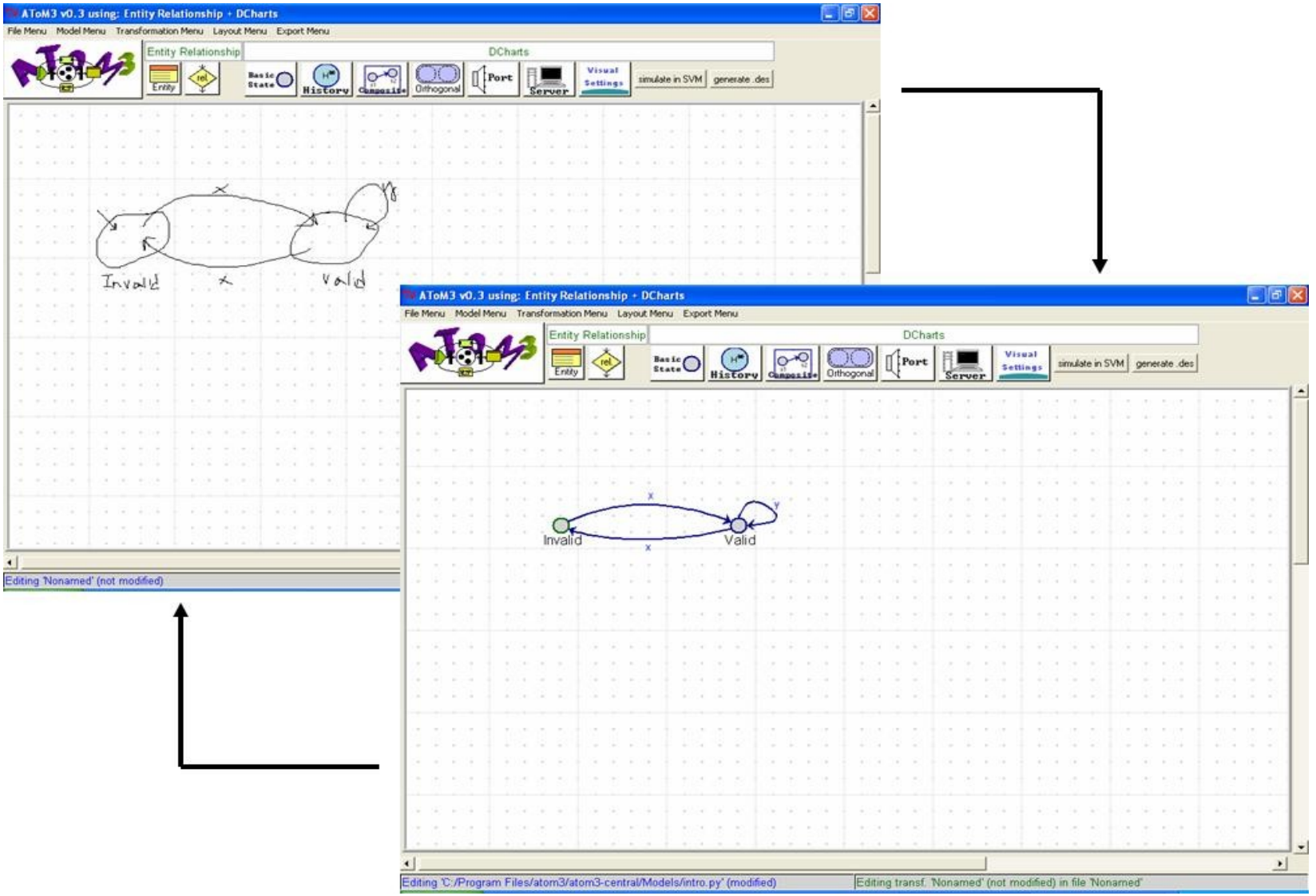
Syntax-directed Visual Editors: model behaviour



Generate Syntax-directed Visual Editors

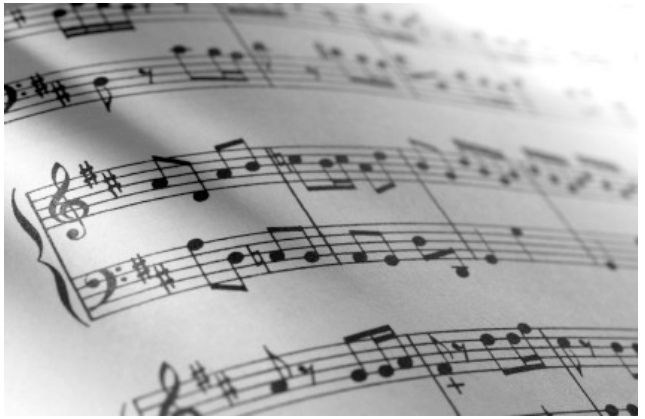


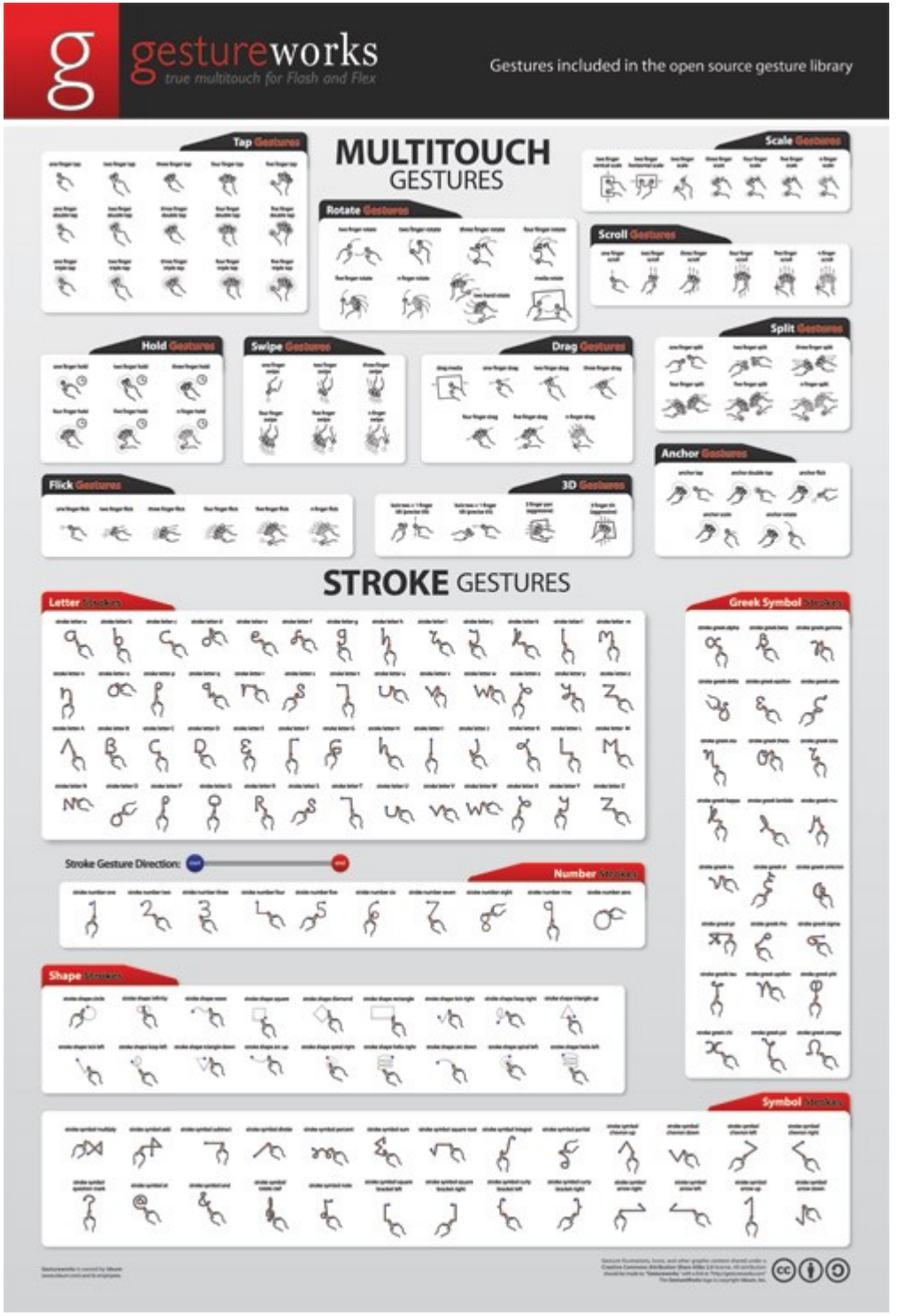
Syntax-directed Visual Editors: freehand (early stages of multi-domain project)



Different Media:

Gestural Interaction, Sound, …





IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. 5, NOVEMBER-DECEMBER 2009

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

Daniel L. Moody, *Member, IEEE*

Introduction

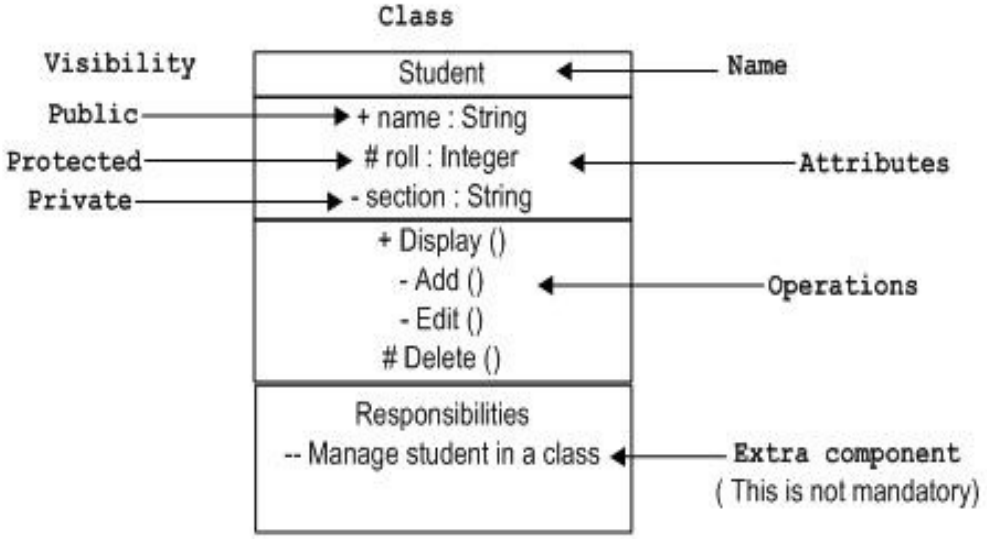
- Visual notations pre-date textual ones
- Textual is special case of Visual
- Visual notations are important for Modelling and Software Engineering
- Humans are excellent pattern recognizers
- Need cognitively efficient and effective notations.

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

a DSVL @ Lascaux

Introduction/Rationale

Visual notations are often introduced without underlying theory or rationale

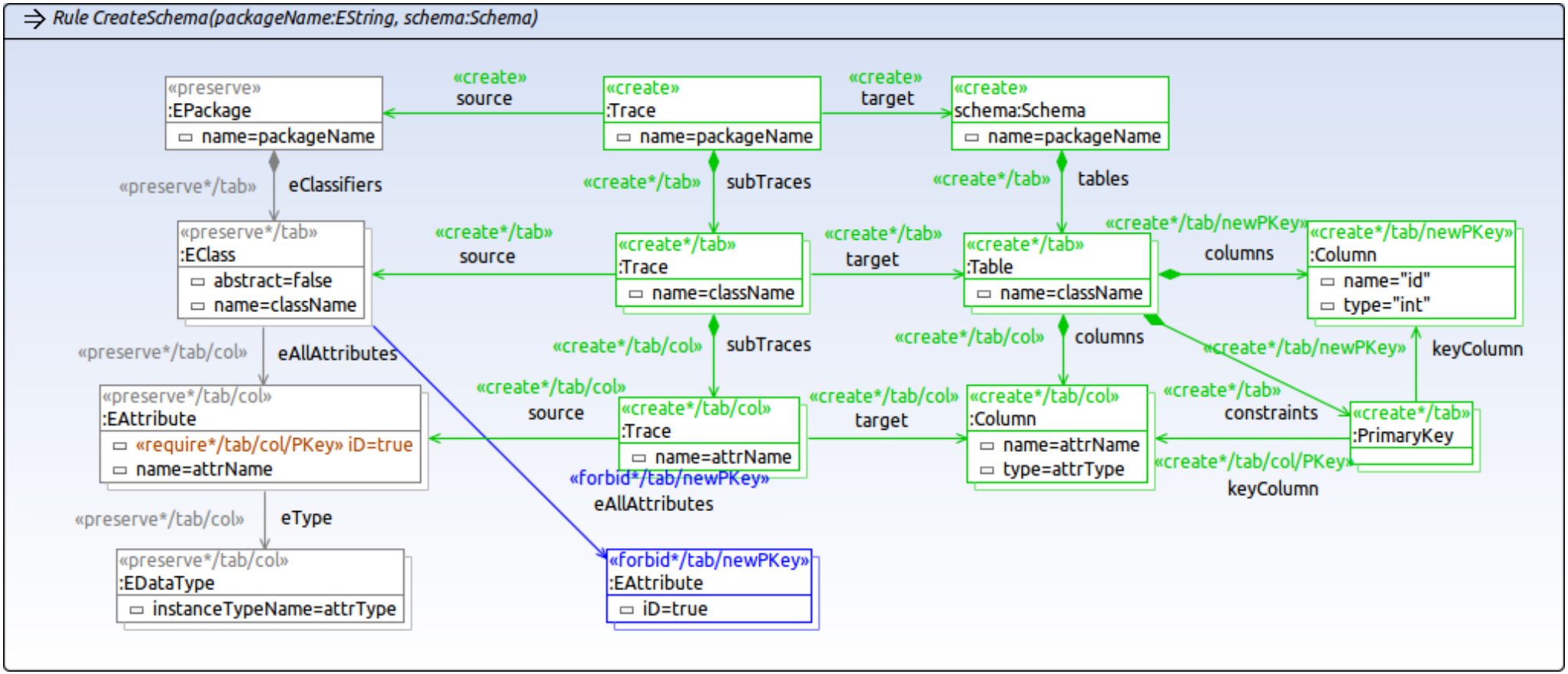


Many visual notations for same concepts.

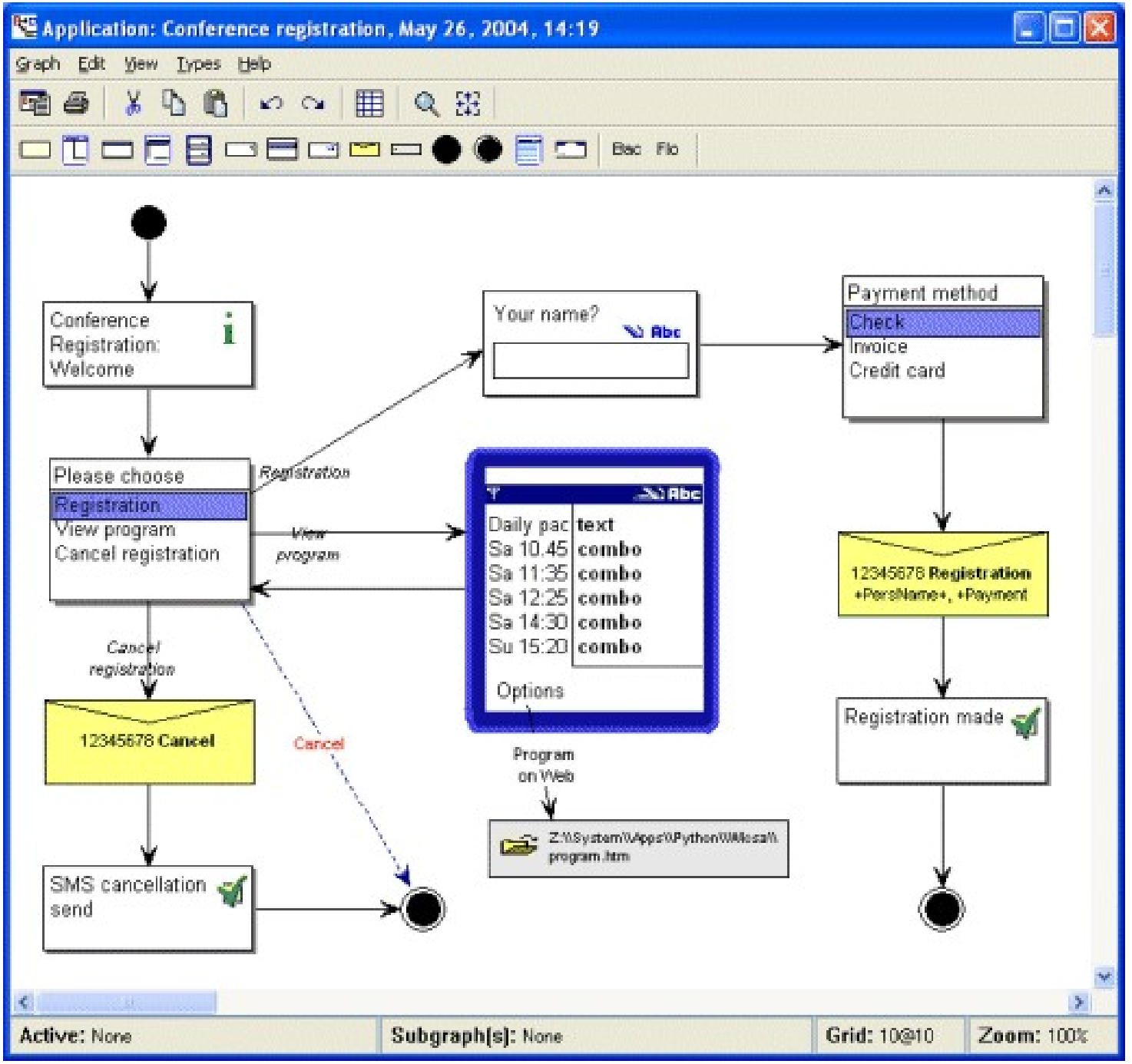
Chen notation | Information Engineering (IE) notation | Bachman notation | IDEF1X notation

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

Cognitive Effectiveness?

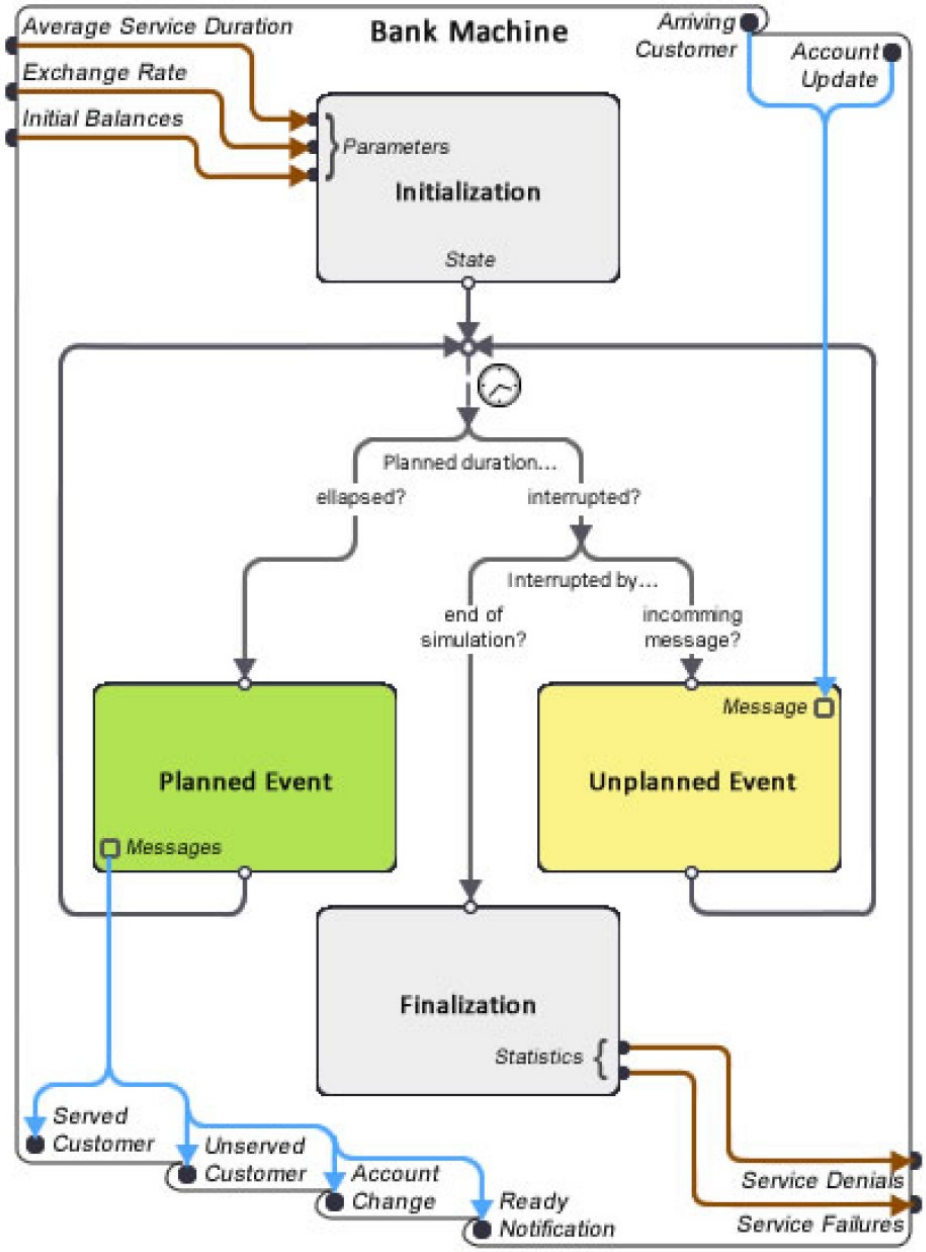


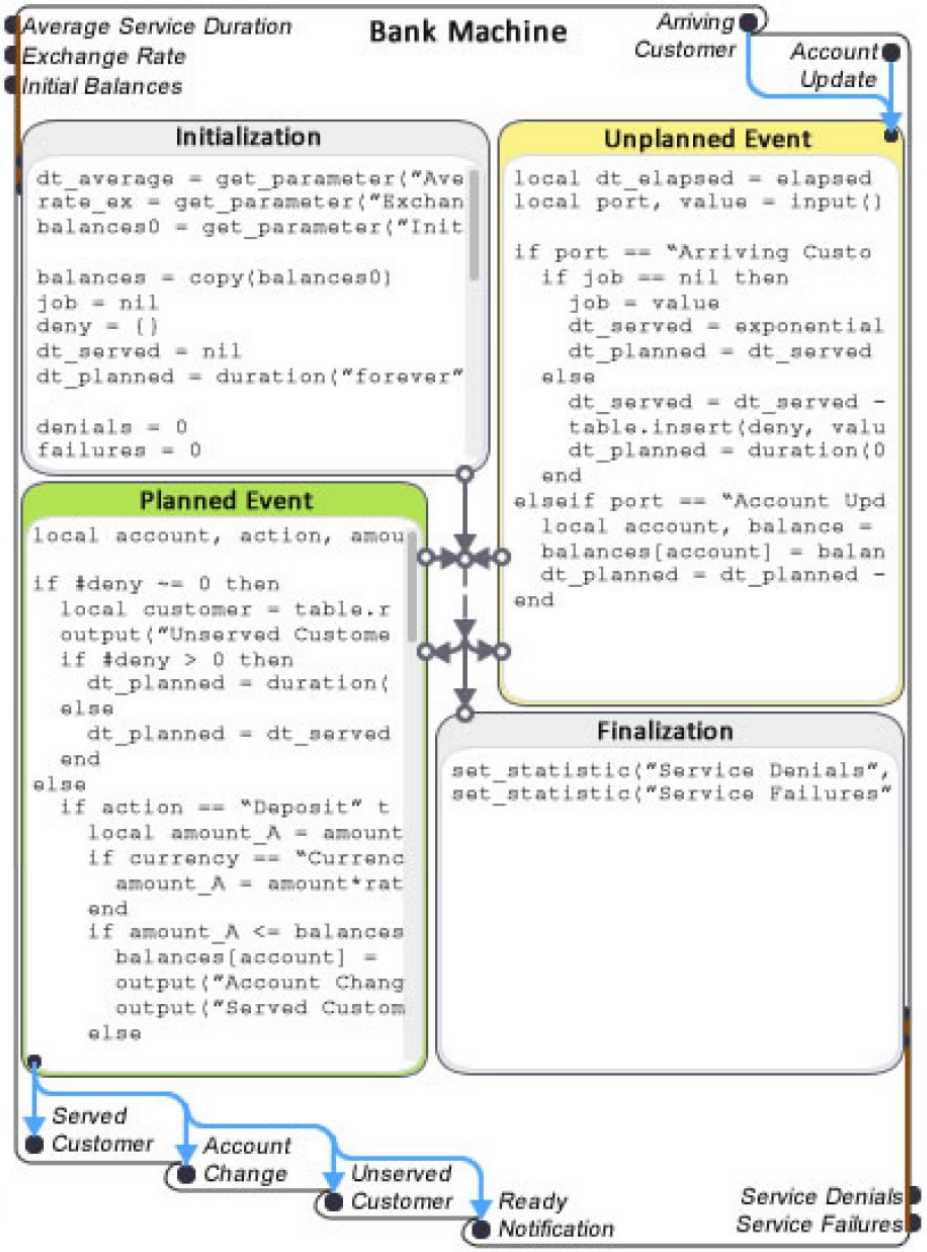
Cognitive Effectiveness?



Cognitive Effectiveness?

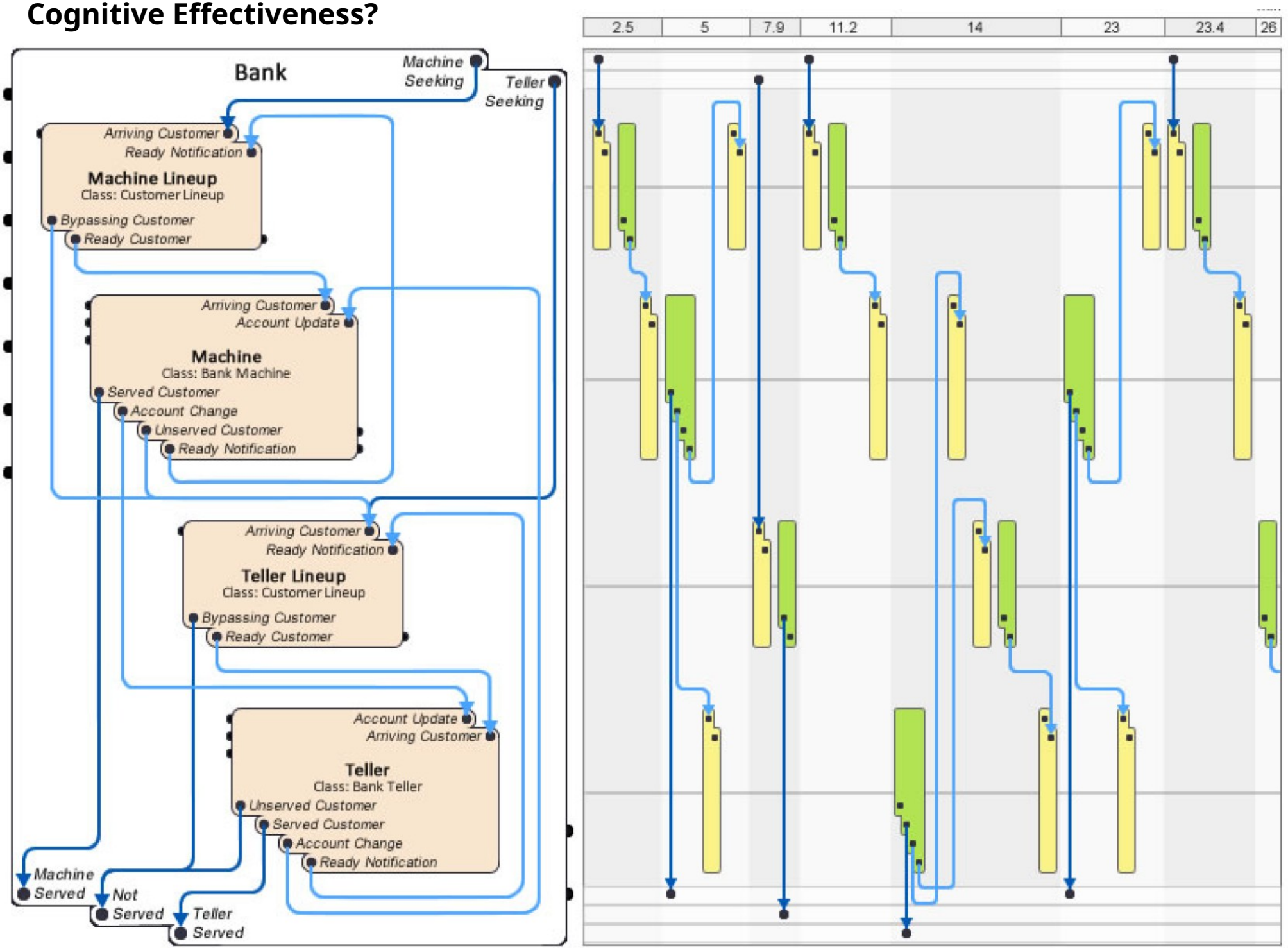
Cognitive Effectiveness?



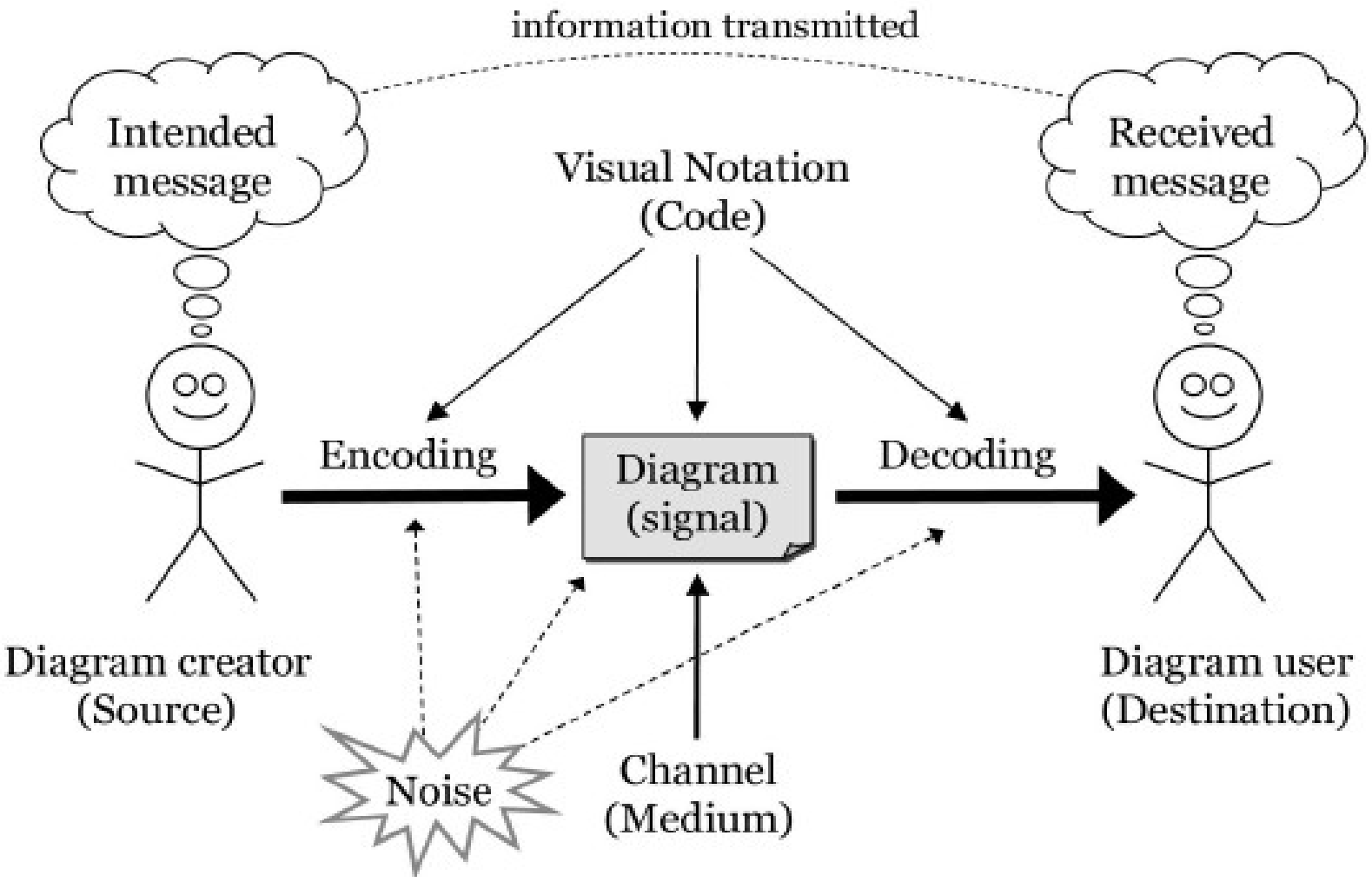


Maryam M. Maleki, Robert F. Woodbury, Rhys Goldstein, Simon Breslav, Azam Khan.
Designing DEVS visual interfaces for end-user programmers. Simulation 91(8): 715-734 (2015)

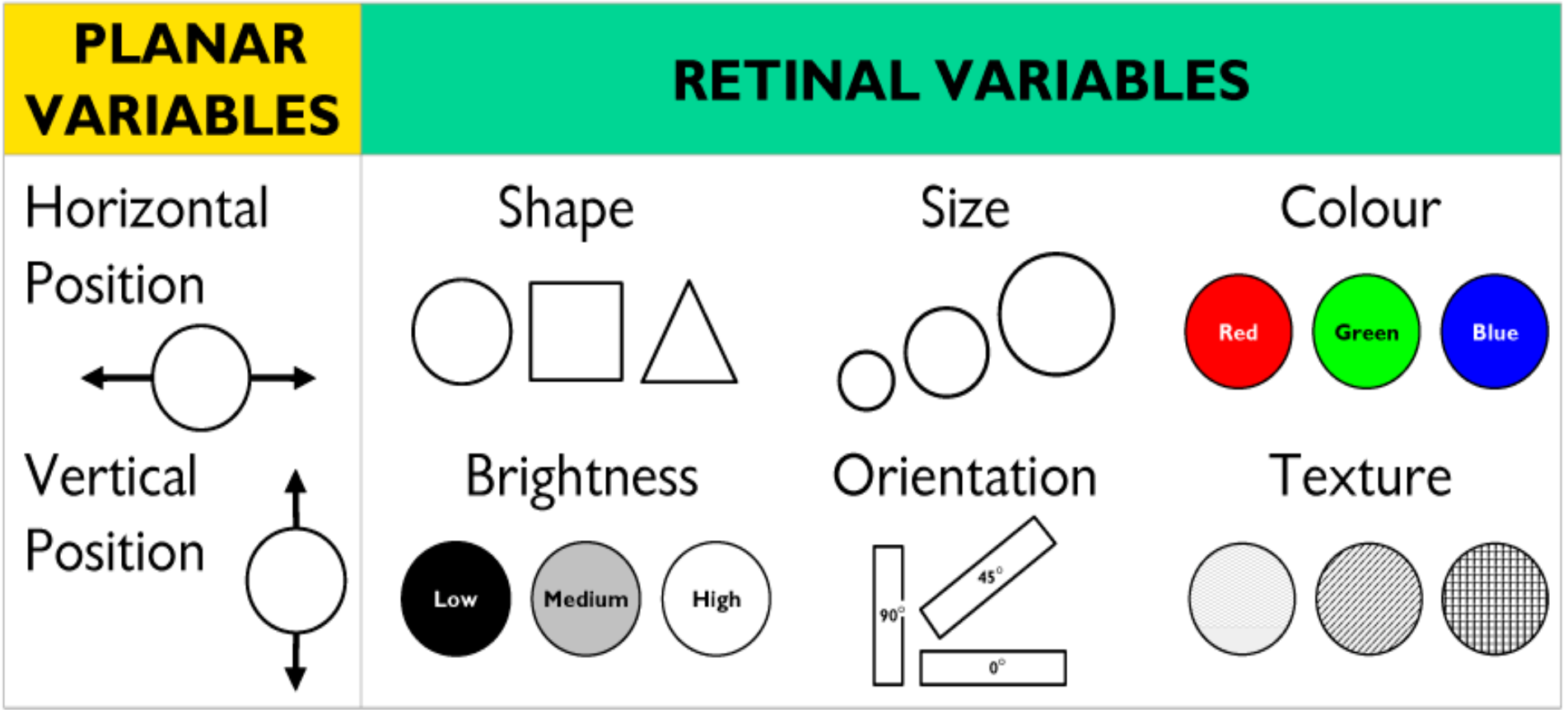
Cognitive Effectiveness?



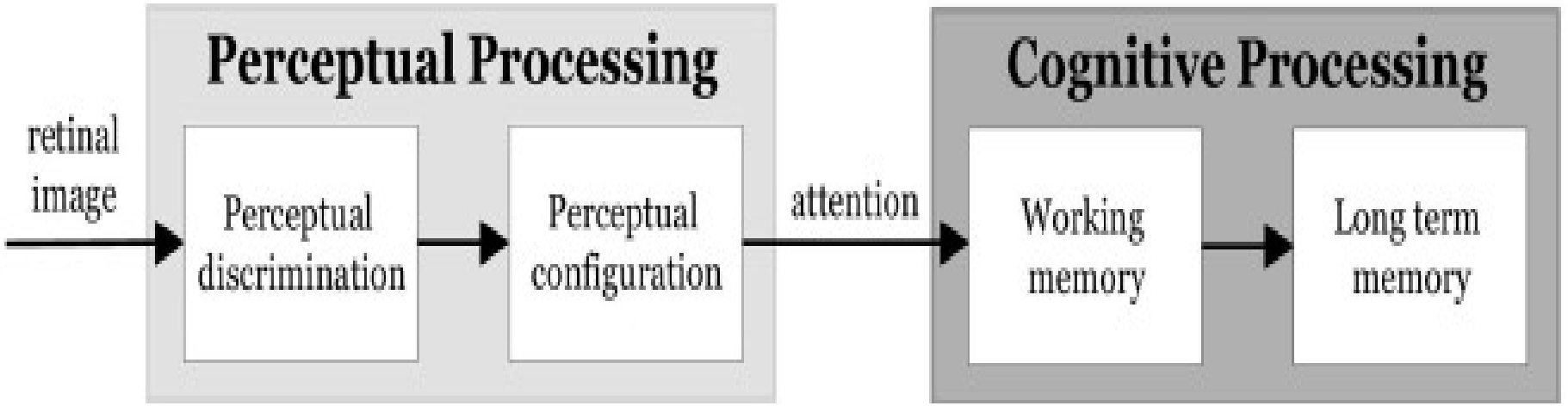
Communication Theory



Encoding: 8 visual variables to (graphically) encode information



Decoding



automatic, fast, parallel

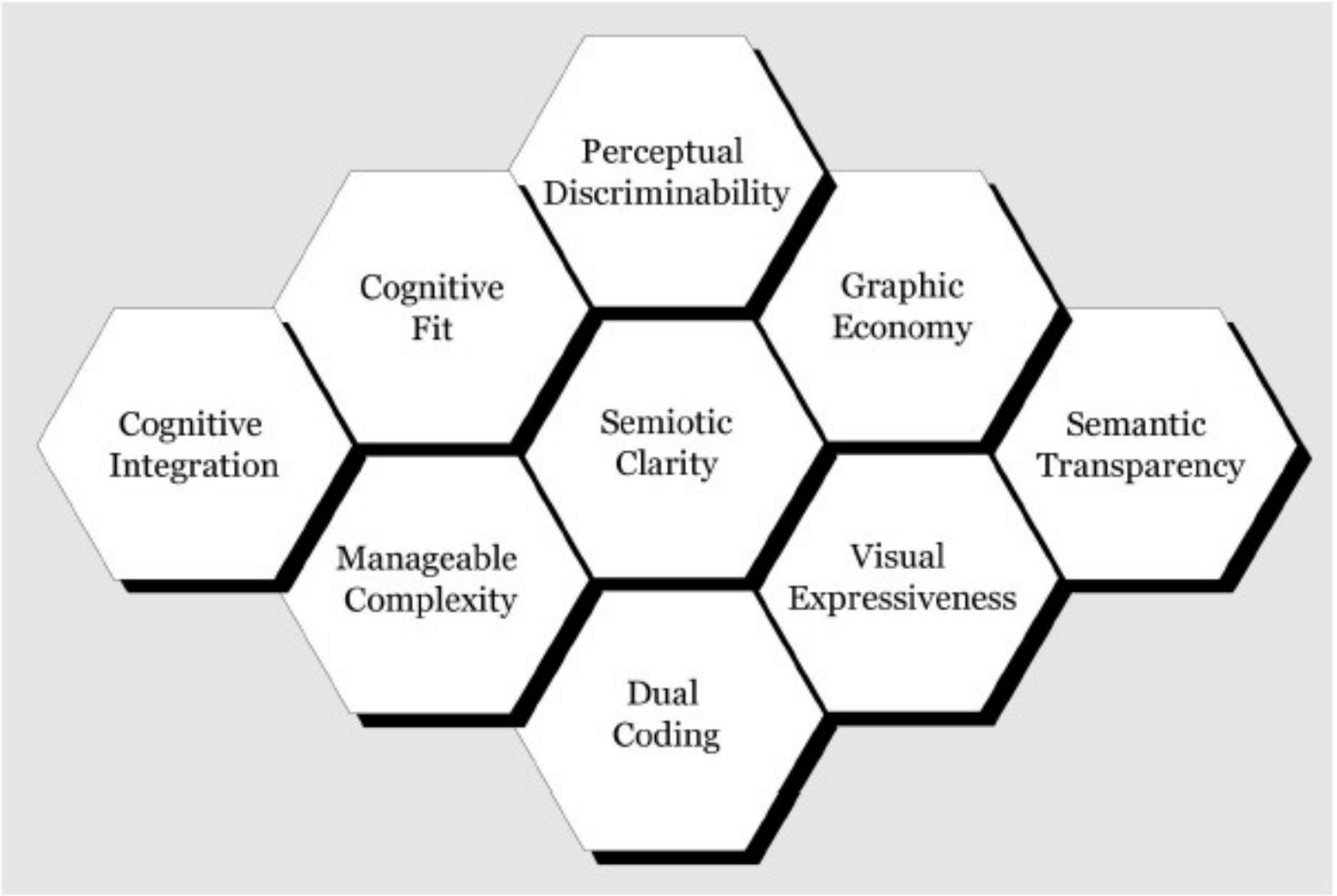
slow, large effort, sequential

Appropriate notations →
offload some of the burden from cognitive to perceptual

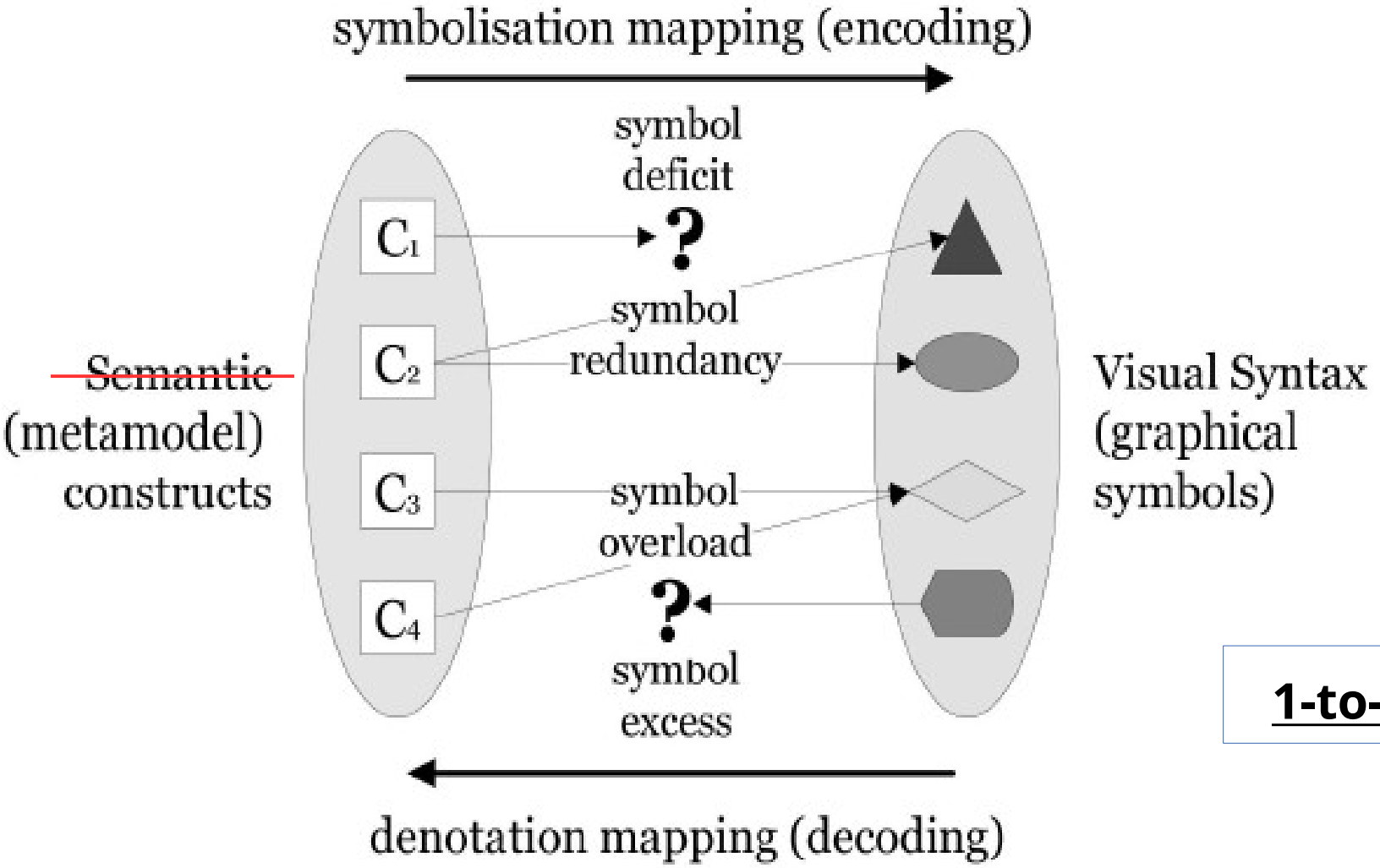
Note: “dual channel theory”:
auditory/verbal channel and visual/pictorial channel are processed in parallel

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

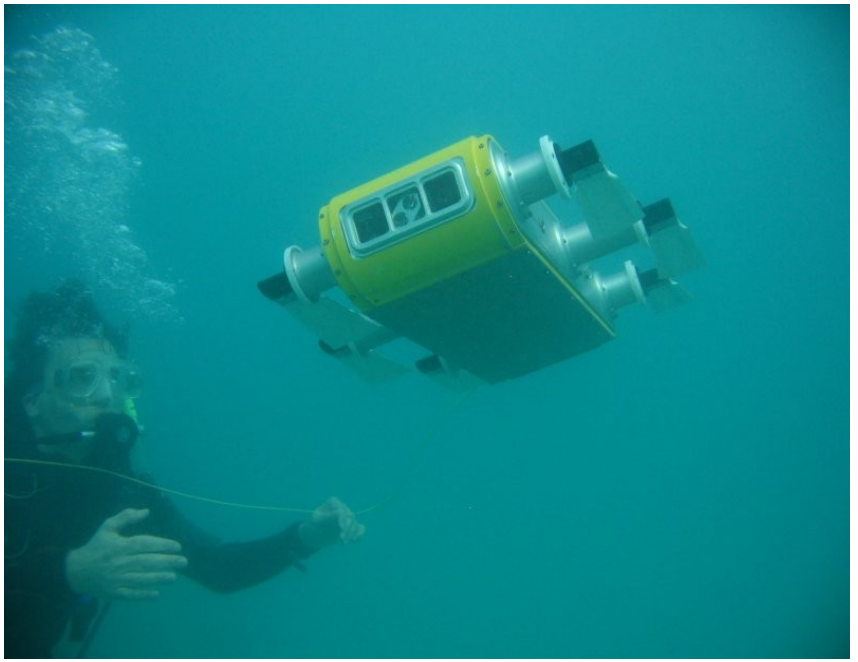


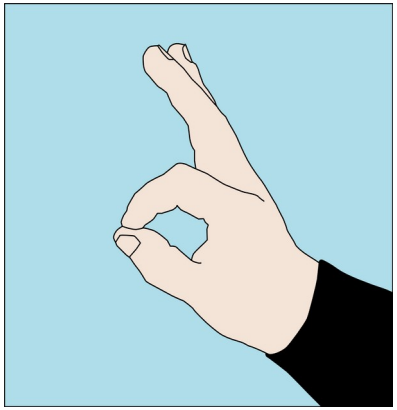
Semiotic Clarity (semiotics = study of signs and sign processes)

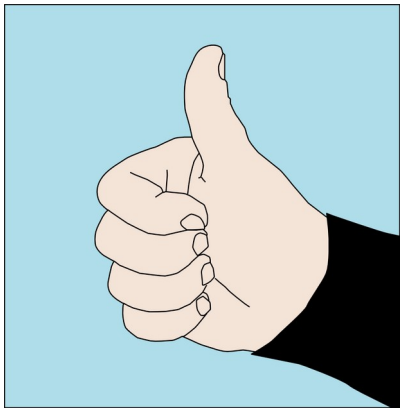


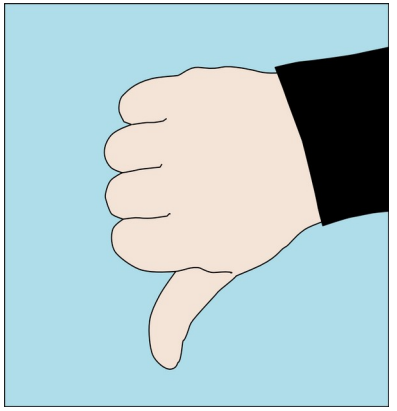
1-to-1

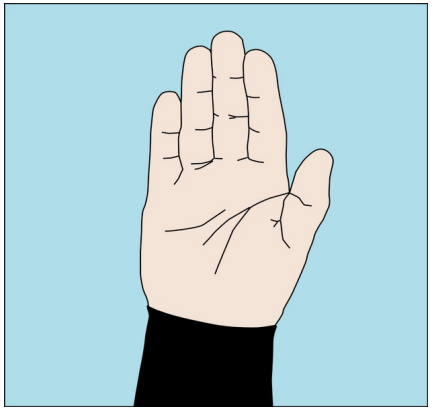
Perceptual Discriminability





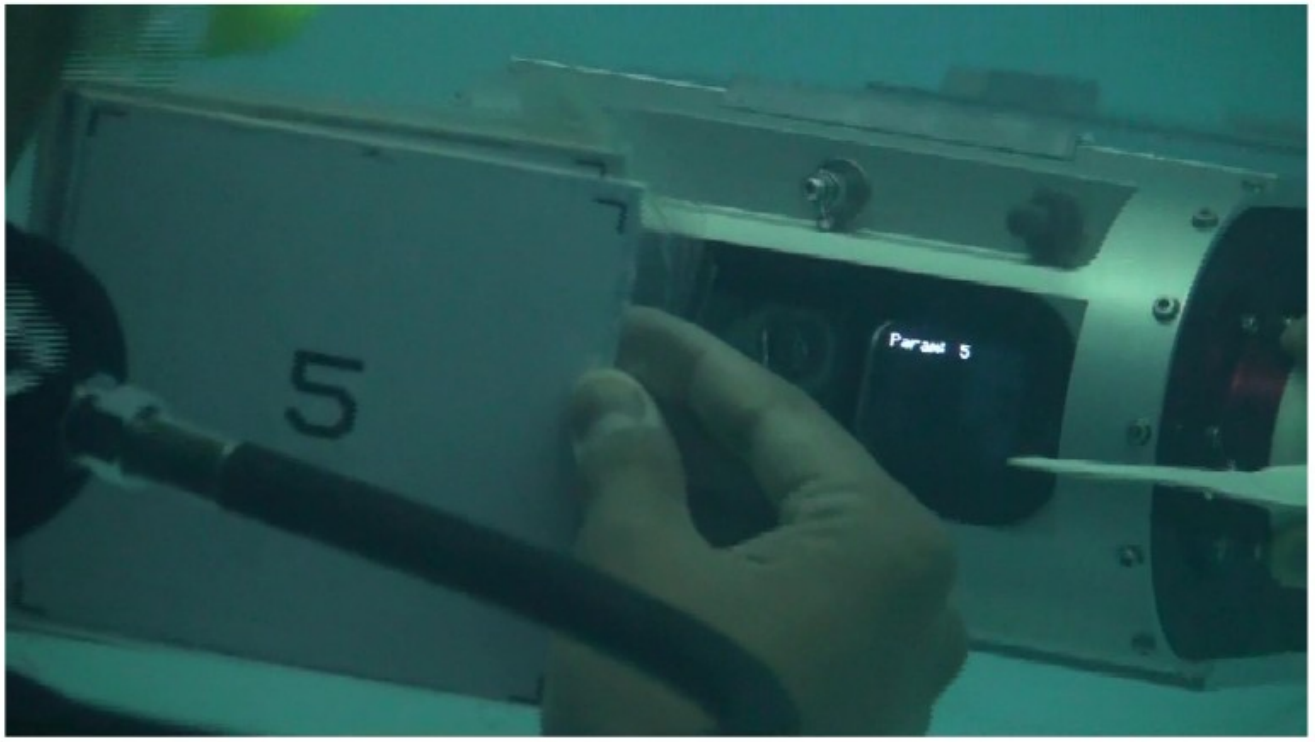






(a) Divers programming Aqua2 during pool trials.

(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.

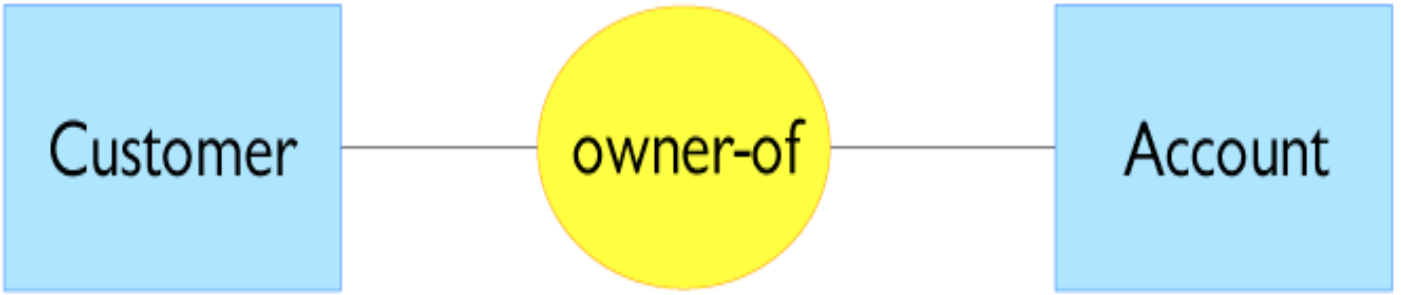
Junaed Sattar, Gregory Dudek. Reducing Uncertainty in Human-Robot Interaction: A Cost Analysis Approach. ISER 2010: 81-95.

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)

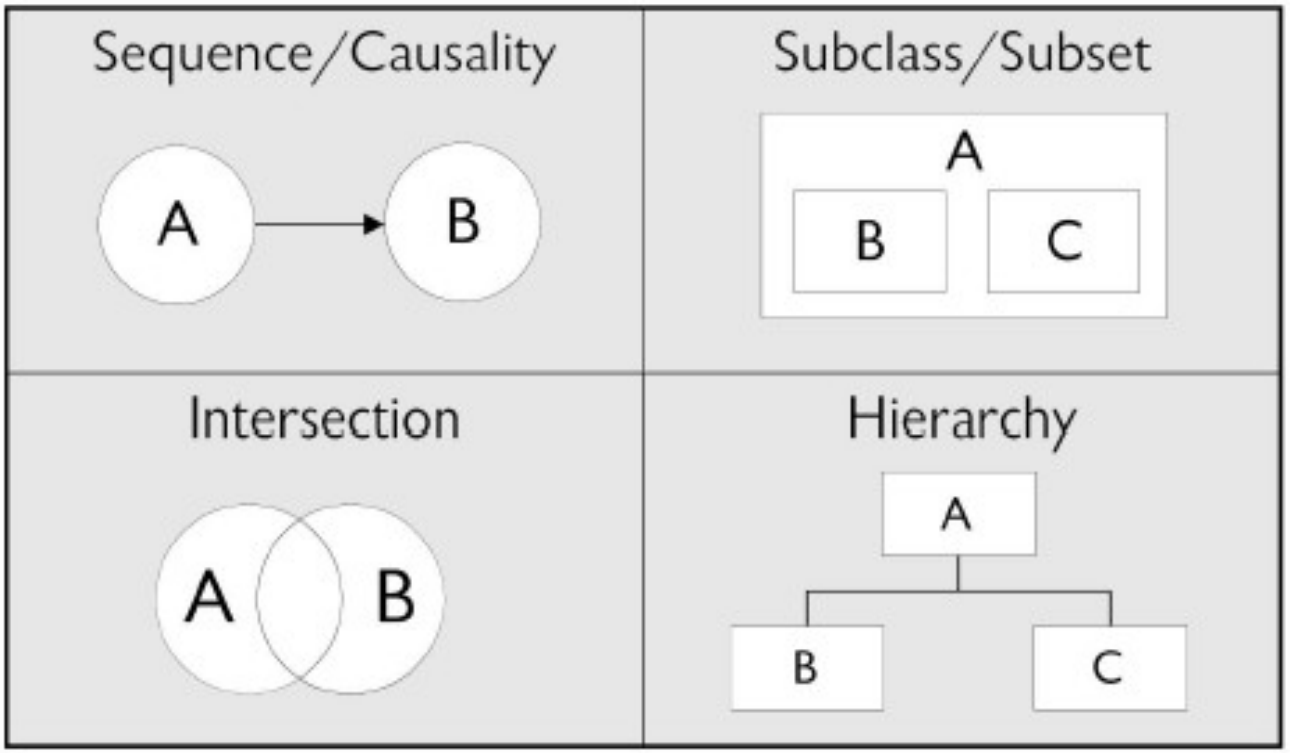
Semantic Transparency

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

Symbols can be:

Semantic Transparency: semantically **immediate** symbols

Semantic Transparency

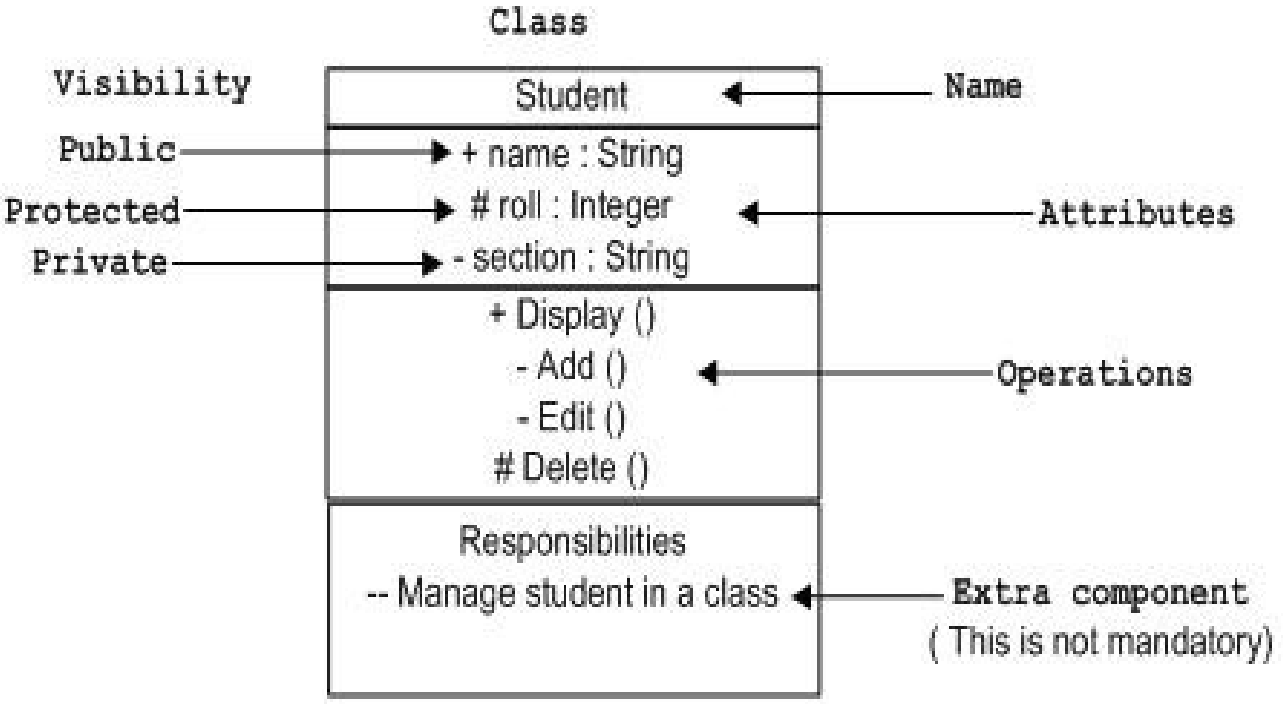


Semantic Transparency

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

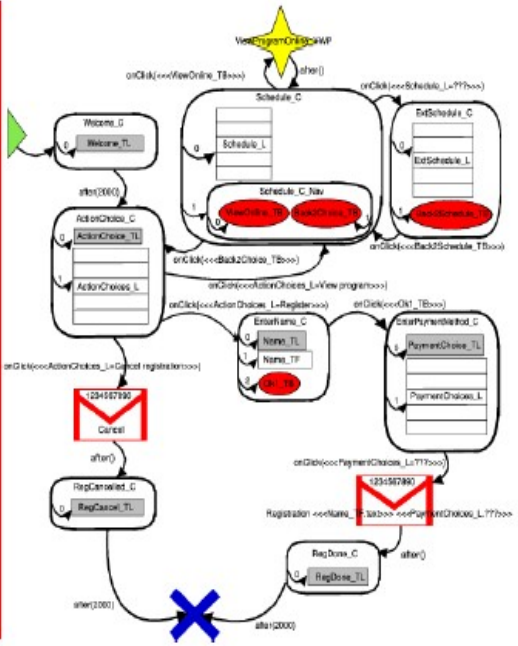
Semantic Transparency: semantically **perverse** symbols

Semantic Transparency

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

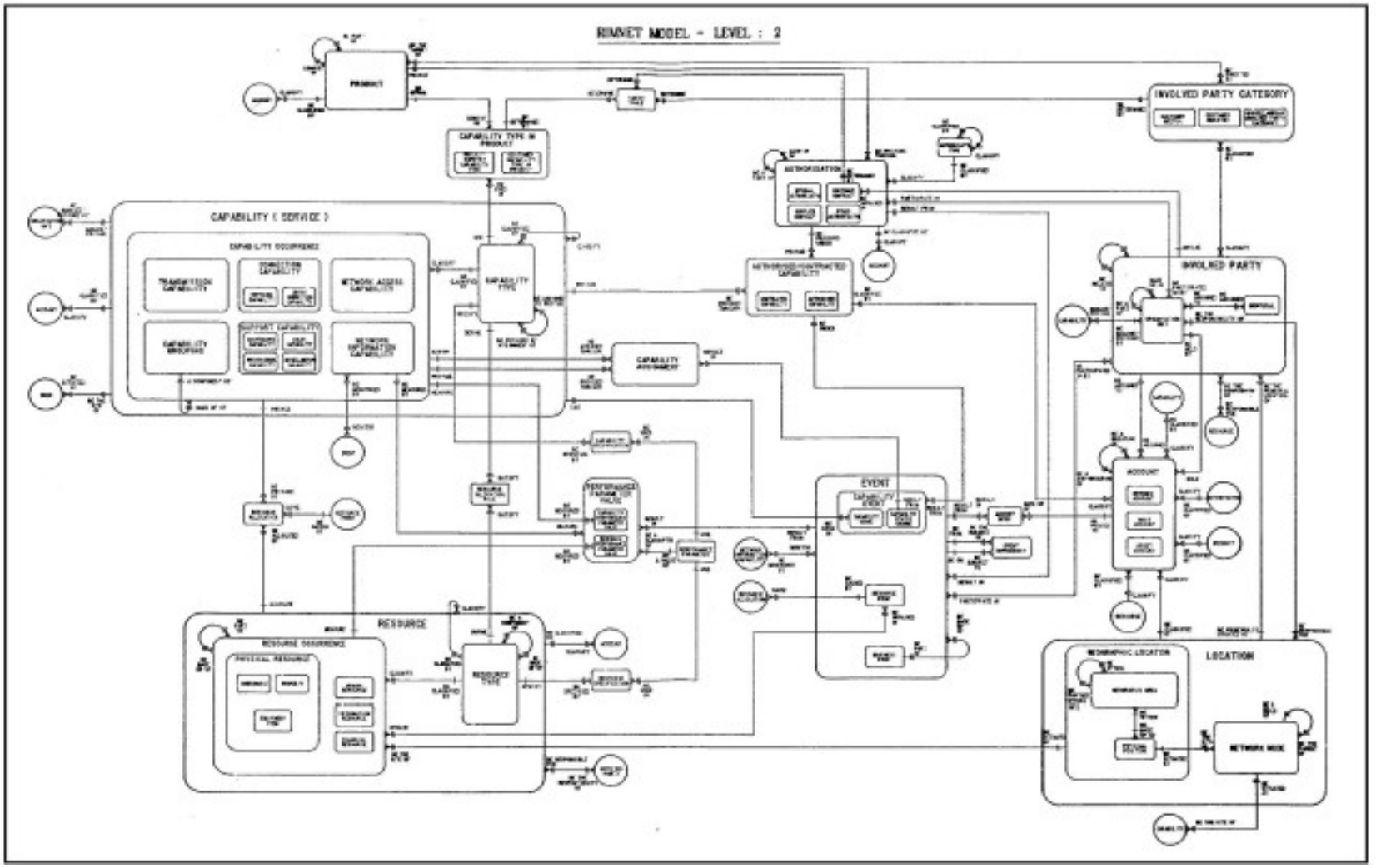
Symbols can be:

- Semantically Immediate
- Semantically Opaque
- Semantically Perverse

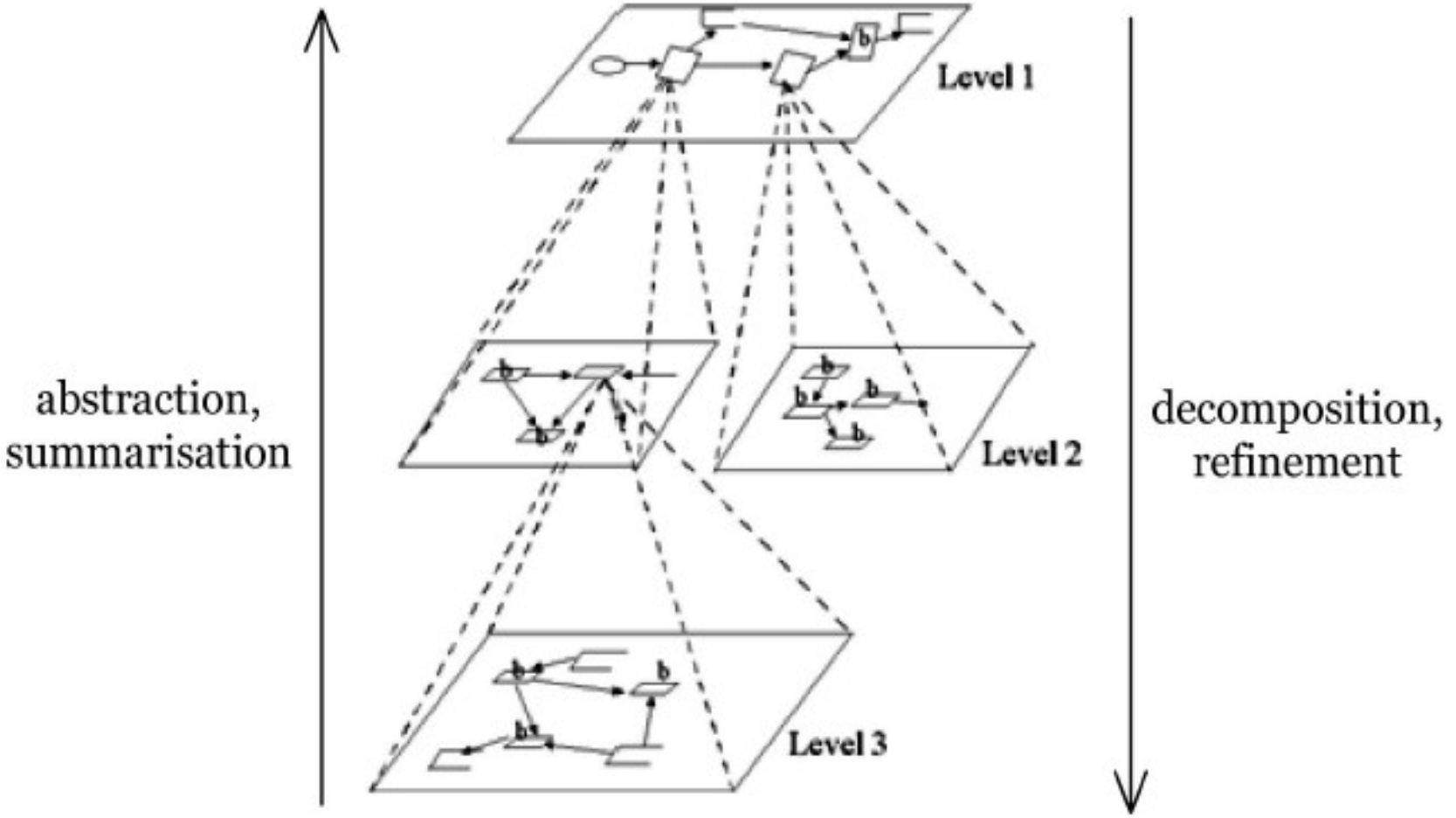


Domain-specific icons and visual arrangement should be intuitive

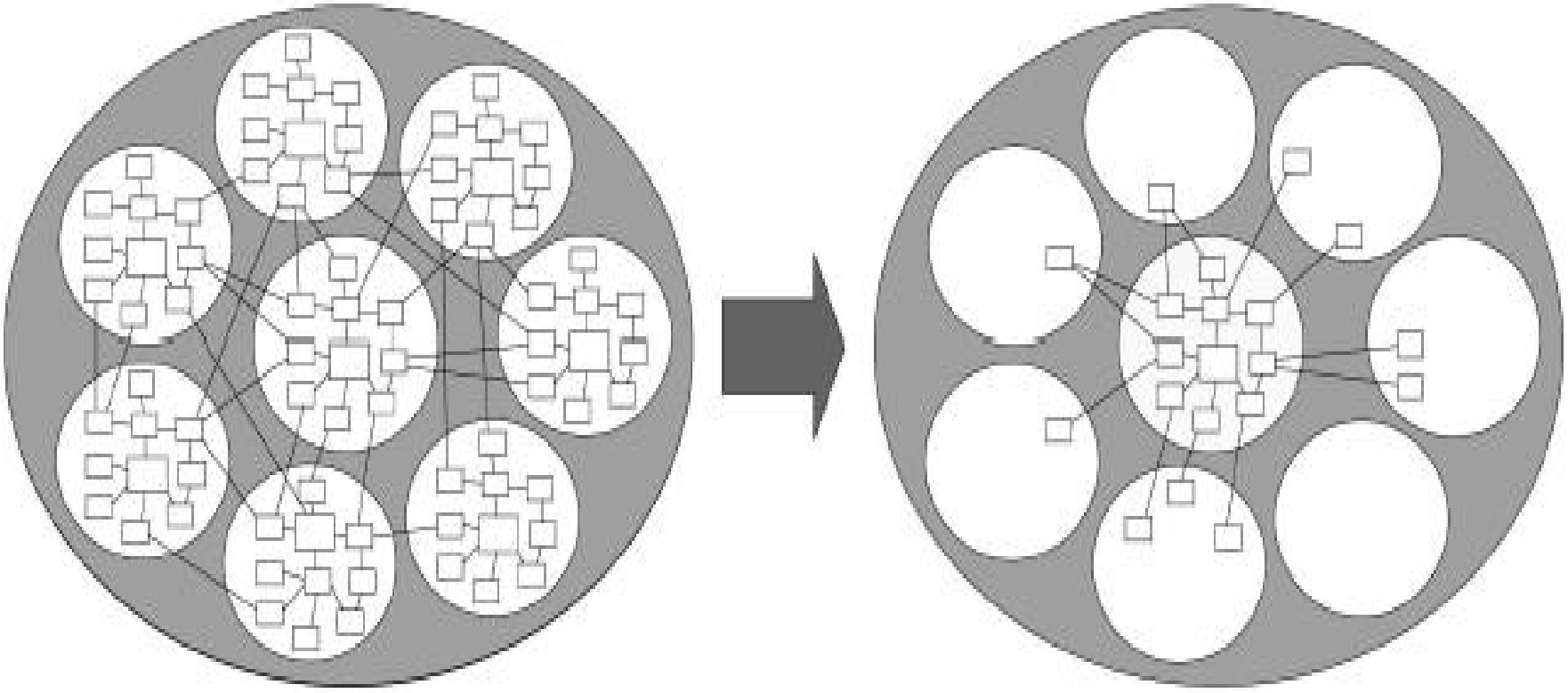
Complexity management (# elements in diagram » cognitive overload)



Modularization/Hierarchy



Cognitive Integration (different notations)

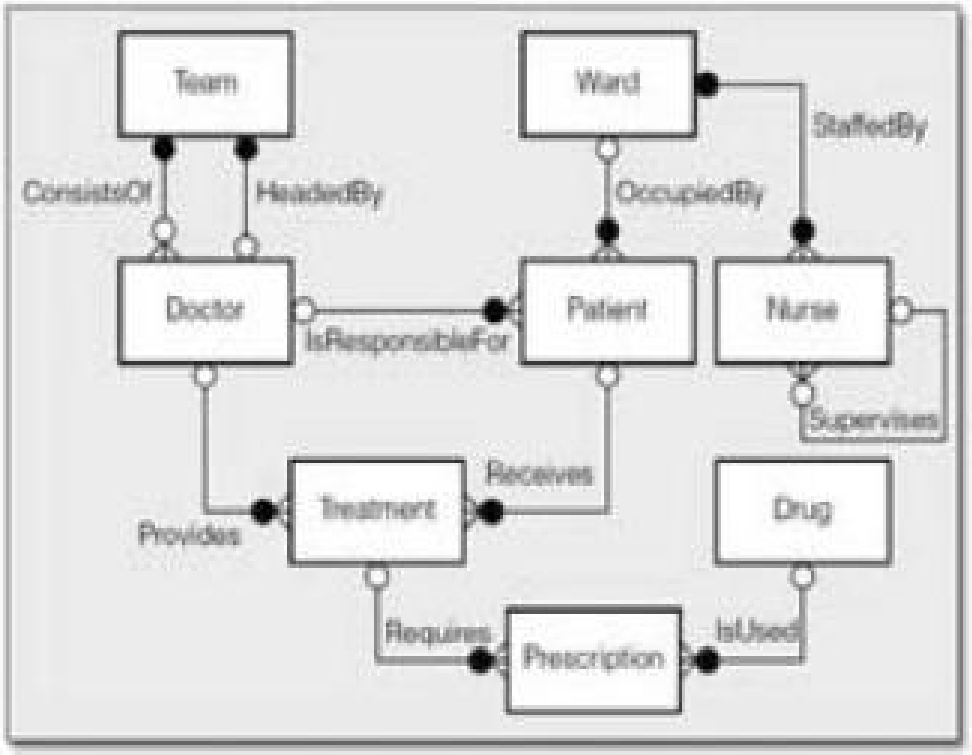


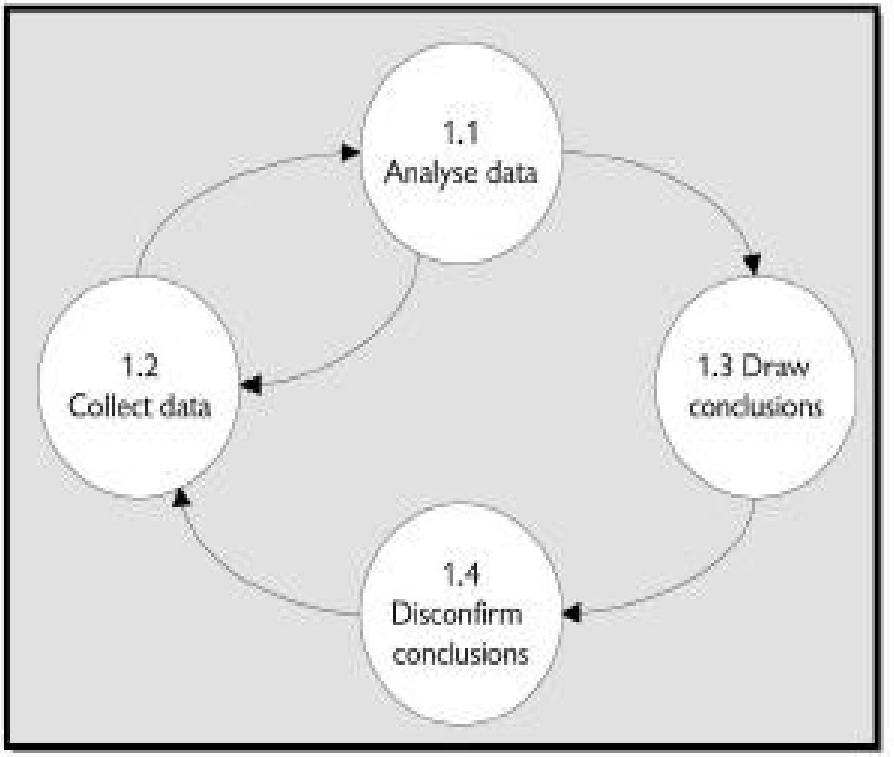
- Conceptual integration (**coherent** mental model)
- Enable **navigation** and **transition** between notations

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

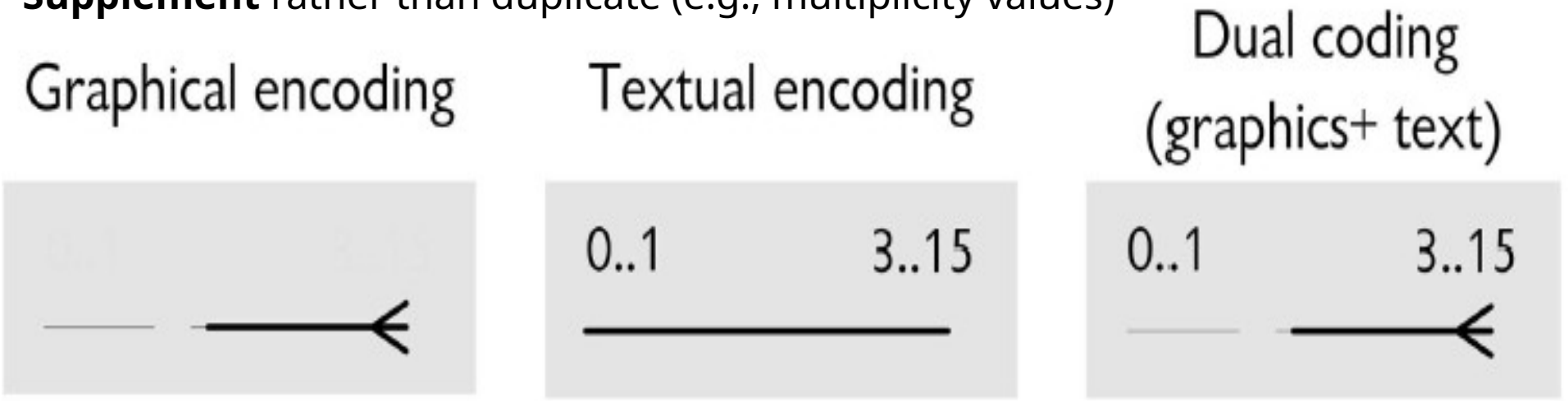
Different visual variables have **different capacity** to encode information

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)

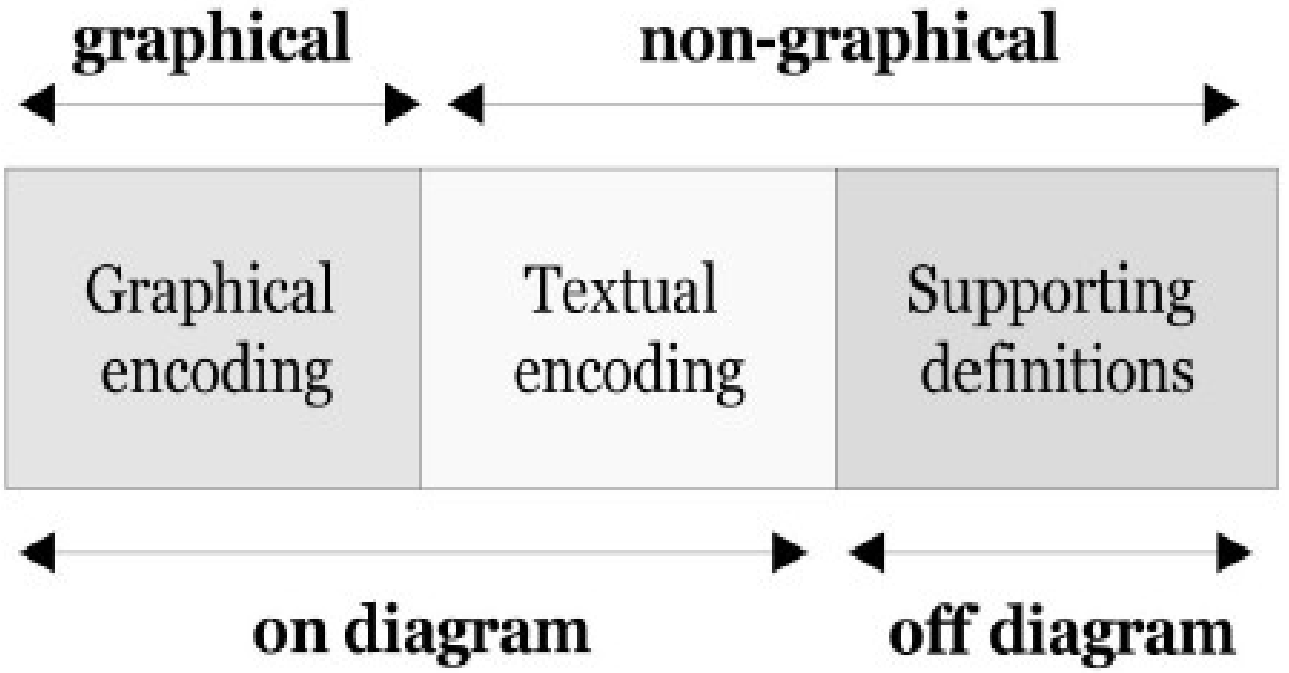


Reinforce
meaning

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

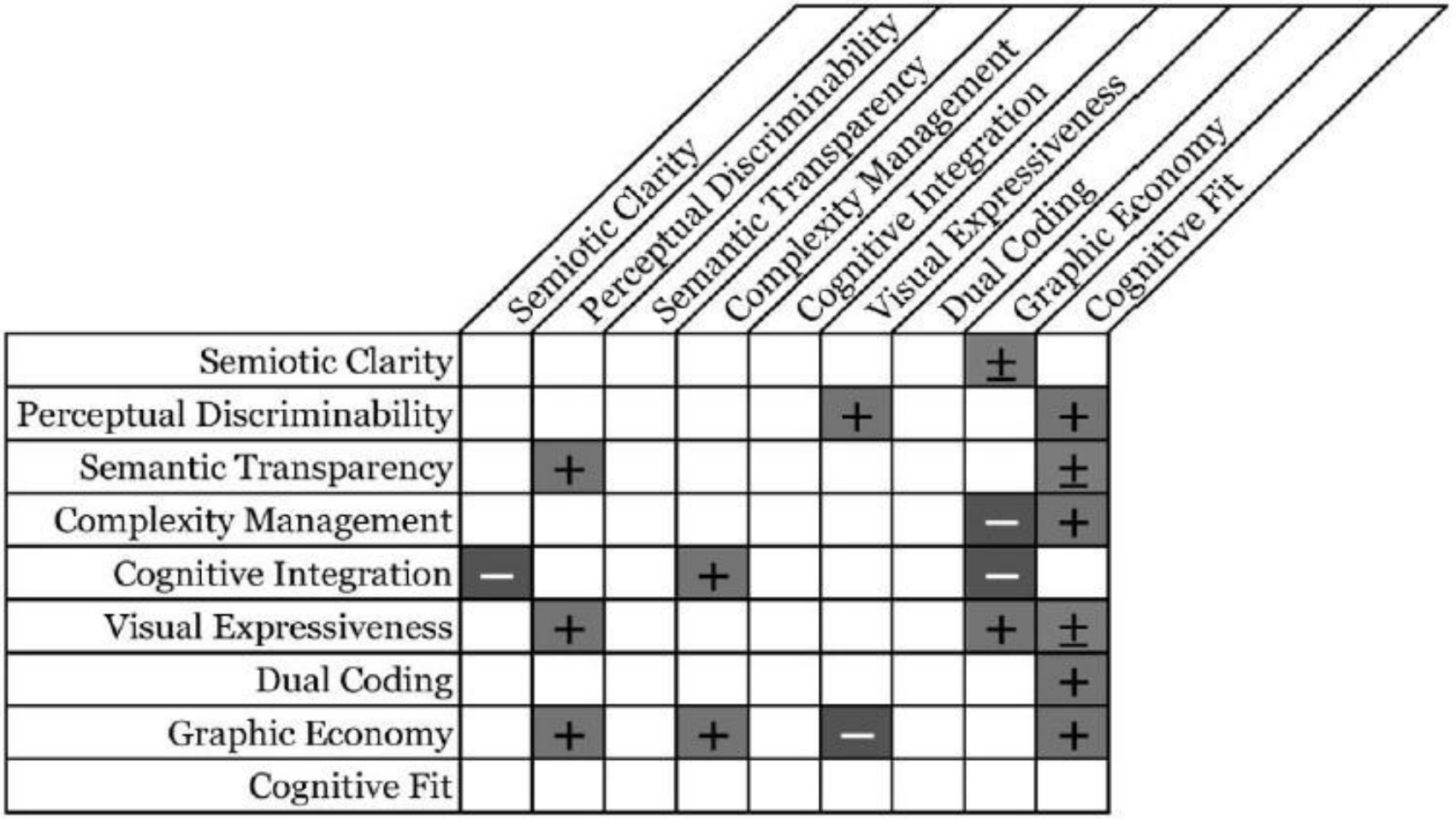
Adapt choice of visual notation to

- Task
- Audience (novices vs. experts)

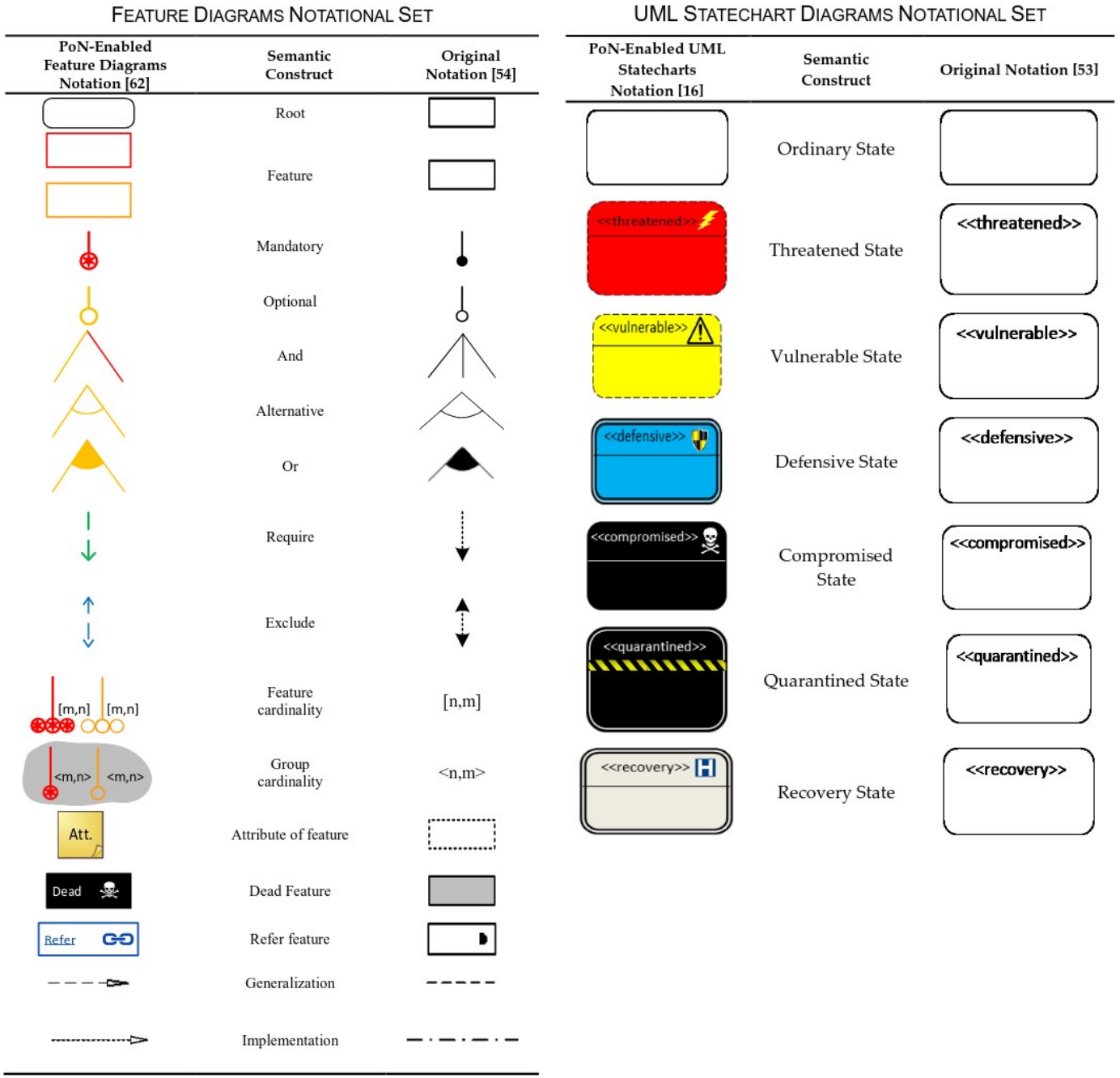
Adaptation may be dynamic (“learn” about Task/User proficiency)

Representation medium matters

Interactions among principles



Ultimately, need empirical studies!



M. El-Attar, "Empirically Evaluating the Effect of the Physics of Notations on Model Construction,"
in IEEE Transactions on Software Engineering, vol. 48, no. 7, pp. 2455-2475, 1 July 2022, doi: 10.1109/TSE.2021.3060344.

Ultimately, need empirical studies!

THE DEPENDENT VARIABLES AND THEIR CORRESPONDING HYPOTHESES FOR BOTH EXPERIMENTS

Dependent Variable	Null Hypothesis (Ho):	Alternative Hypothesis (Ha):
	Feature Diagrams Experiment	
Errors Committed	E (FD-ON) = E (FD-PoN)	E (FD-ON) ≠ E (FD-PoN)
Completion Time	T (FD-ON) = T (FD-PoN)	T (FD-ON) ≠ T (FD-PoN)
	Misuse Case Diagrams Experiment	
Errors Committed	E (MUCD-ON) = E (MUCD-PoN)	E (MUCD-ON) ≠ E (MUCD-PoN)
Completion Time	T (MUCD-ON) = T (MUCD-PoN)	T (MUCD-ON) ≠ T (MUCD-PoN)
	UML Statechart Diagrams Experiment	
Errors Committed	E (SC-ON) = E (SC-PoN)	E (SC-ON) ≠ E (SC-PoN)
Completion Time	T (SC-ON) = T (SC-PoN)	T (SC-ON) ≠ T (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 nota-tions 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.