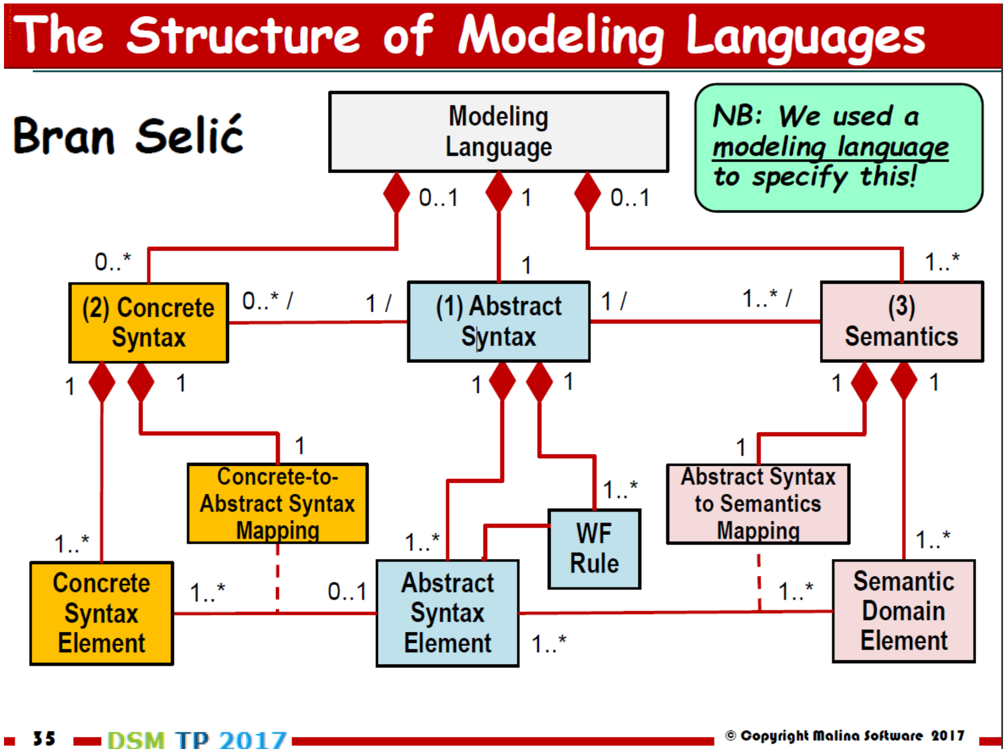
# Modelling Languages:
# (mostly) Concrete (Visual) Syntax

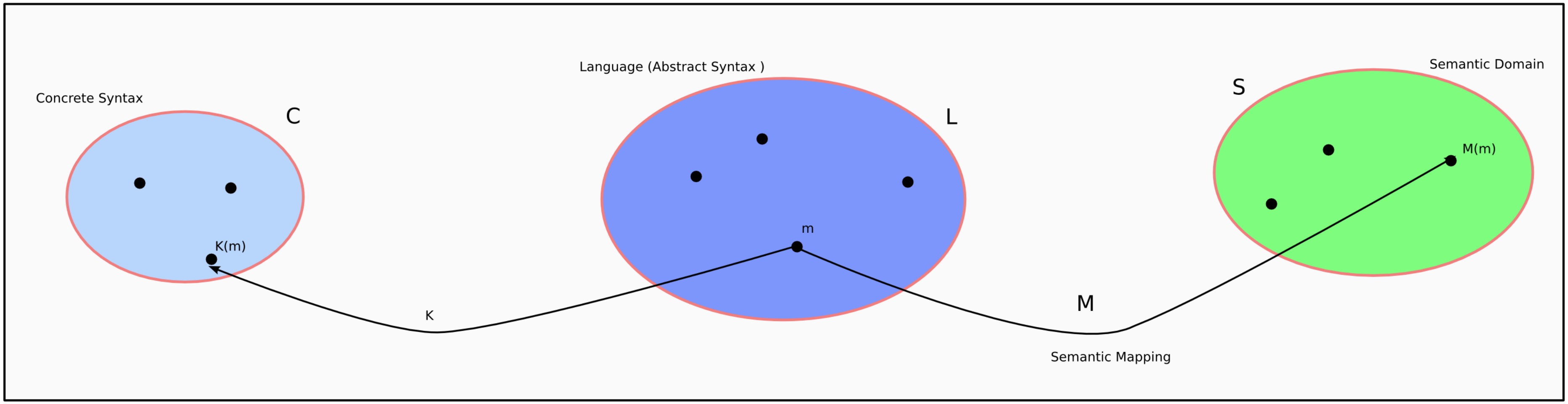
**Hans Vangheluwe**

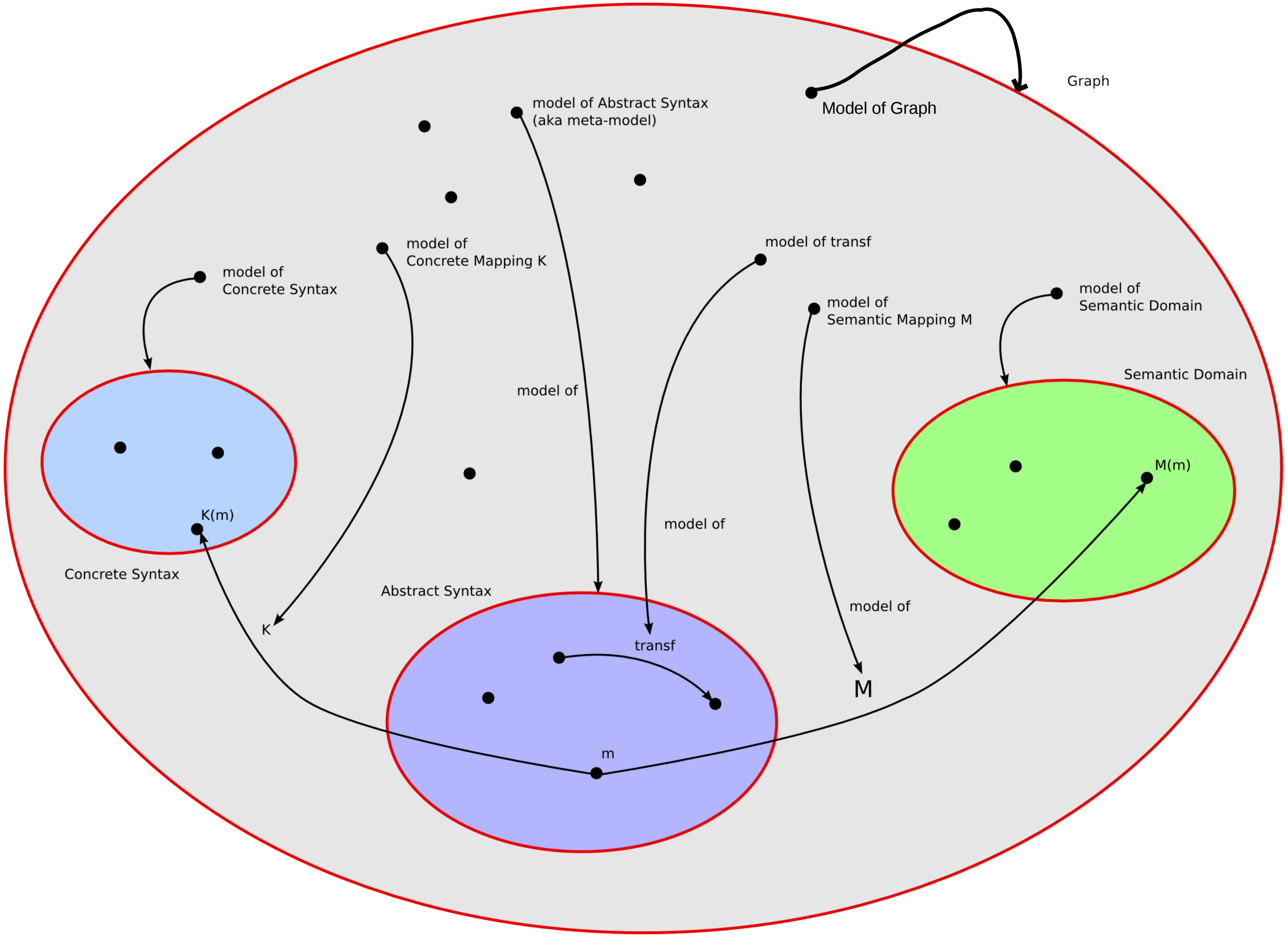
http://msdl.cs.mcgill.ca/

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




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 letters in words *specified* by regular expressions
- Valid words in a language *specified* by a grammar

- letters/words are combined by “is to the right of”

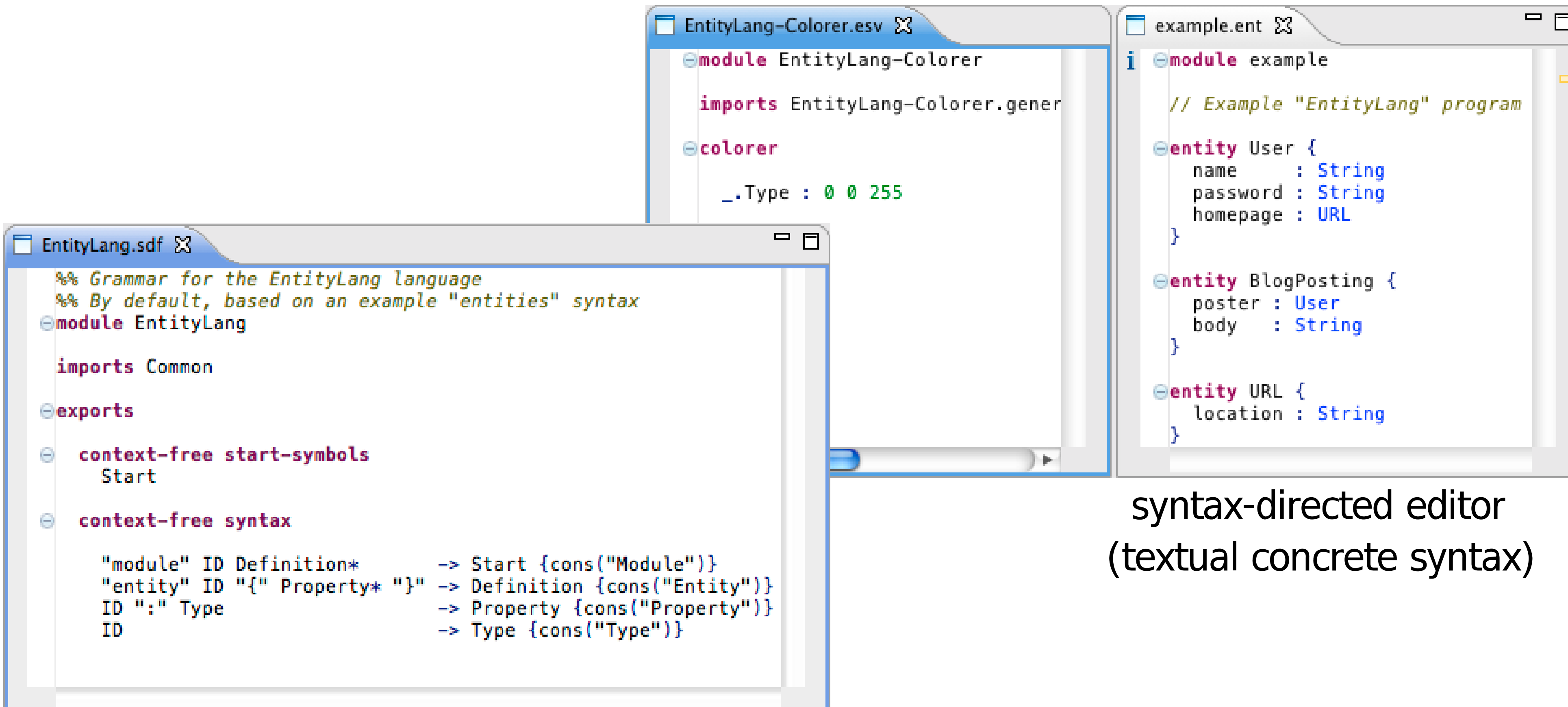
# The Spoofax Language Workbench

Report TUD-SERG-2010-014a

## Rules for Declarative Specification of Languages and IDEs

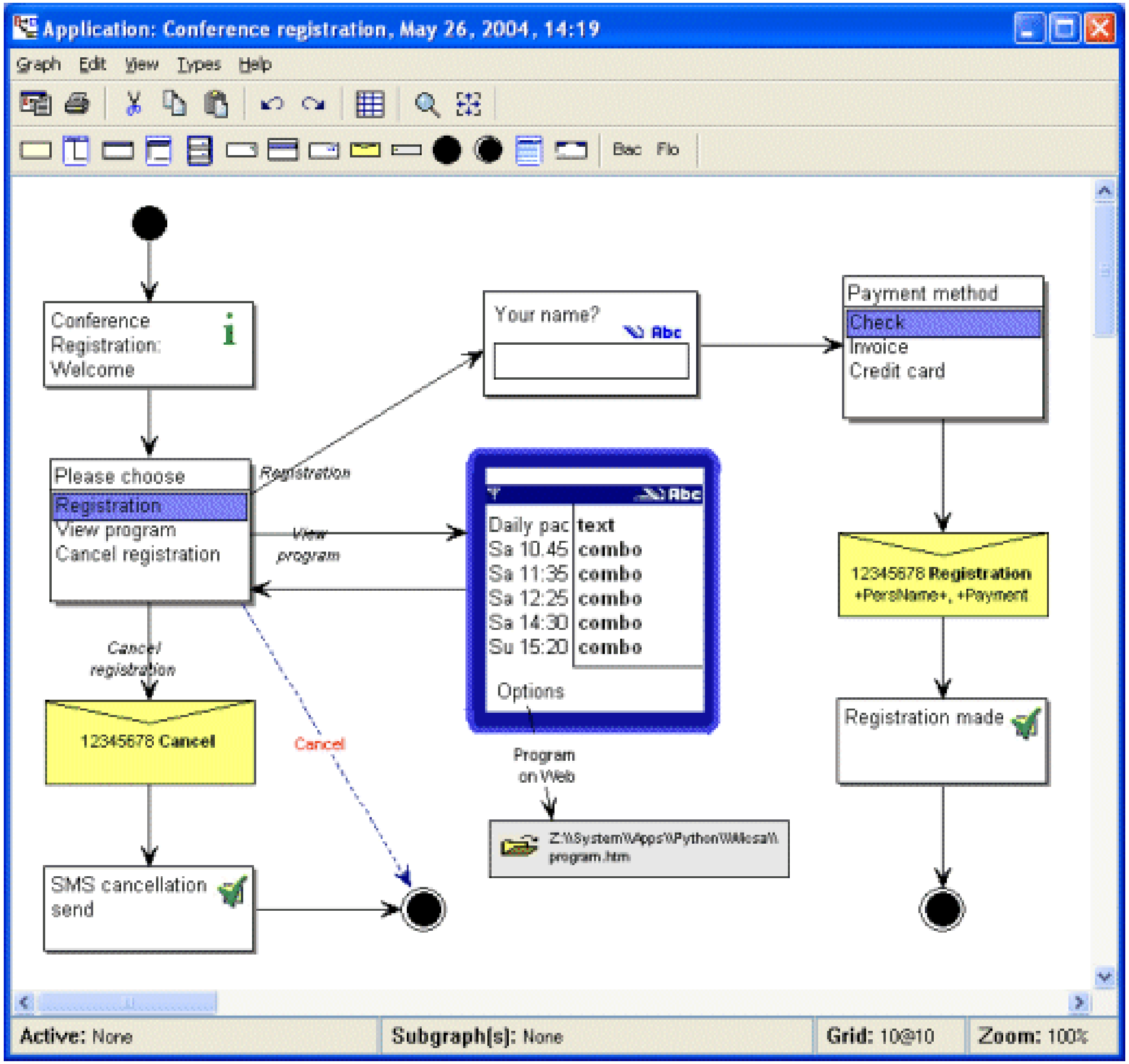
Lennart C. L. Kats
Delft University of Technology
l.c.l.kats@tudelft.nl

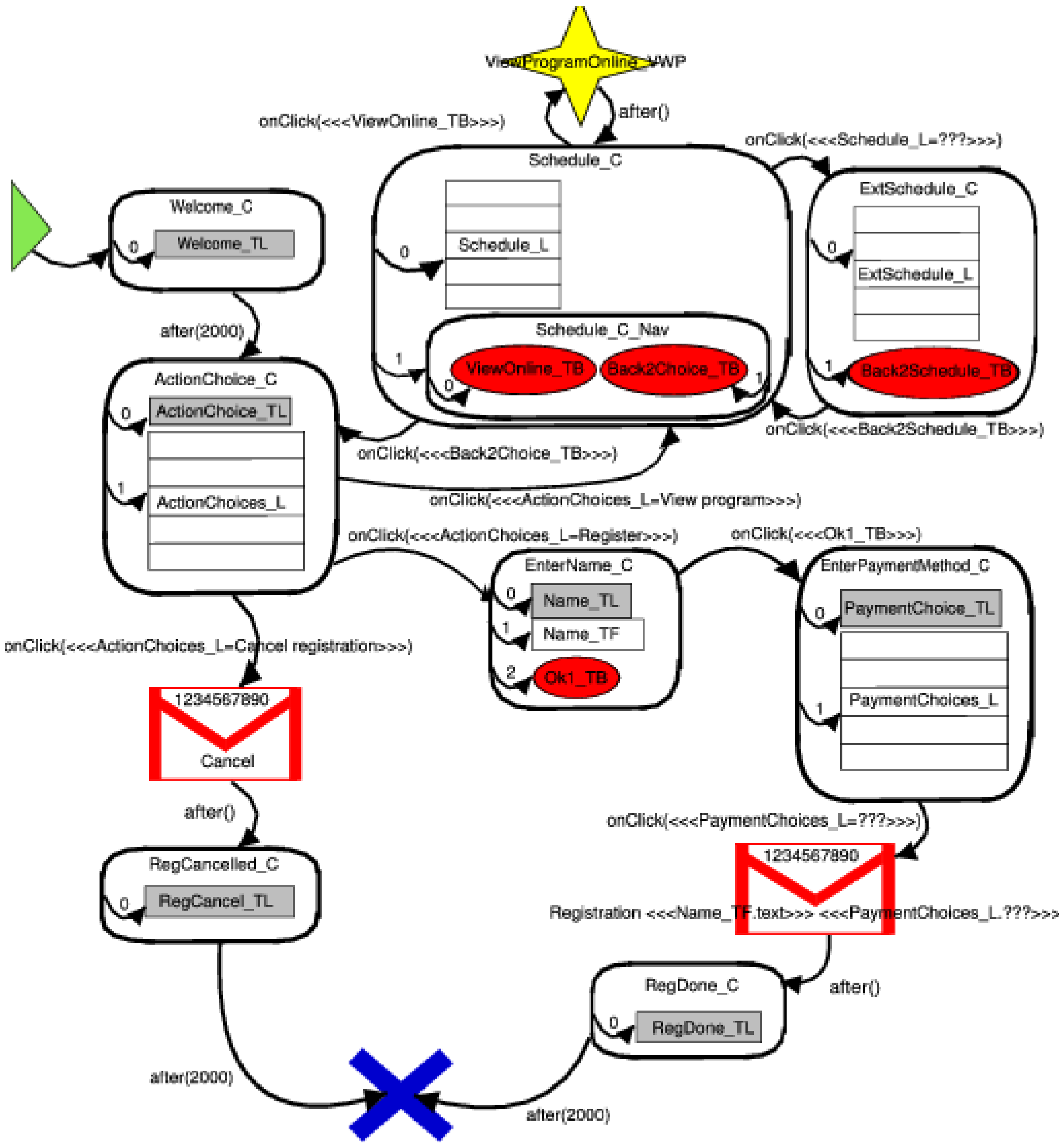
Eelco Visser
Delft University of Technology
visser@acm.org




syntax-directed editor
(textual concrete syntax)

syntax-directed editor
(visual concrete syntax)



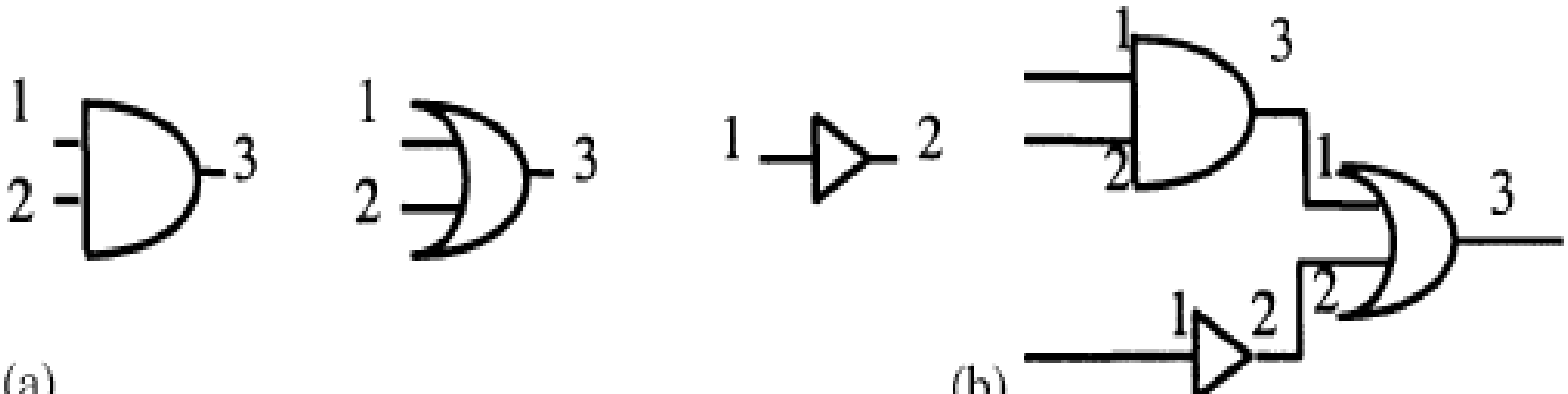

ViewProgramOnline_VWP
after()
onClick(<<<ViewOnline_TB>>>)
onClick(<<<Schedule_L=???>>>)
Schedule_C
ExtSchedule_C
Welcome_C
Welcome_TL
Schedule_L
ExtSchedule_L
after(2000)
Schedule_C_Nav
ViewOnline_TB
Back2Choice_TB
Back2Schedule_TB
ActionChoice_C
ActionChoice_TL
onClick(<<<Back2Schedule_TB>>>)
onClick(<<<Back2Choice_TB>>>)
ActionChoices_L
onClick(<<<ActionChoices_L=View program>>>)
onClick(<<<ActionChoices_L=Register>>>)
onClick(<<<Ok1_TB>>>)
EnterName_C
EnterPaymentMethod_C
Name_TL
PaymentChoice_TL
Name_TF
onClick(<<<ActionChoices_L=Cancel registration>>>)
Ok1_TB
1234567890
PaymentChoices_L
Cancel
after()
onClick(<<<PaymentChoices_L=???>>>)
RegCancelled_C
1234567890
RegCancel_TL
Registration <<<Name_TF.text>>> <<<PaymentChoices_L.???>>>
RegDone_C
after()
RegDone_TL
after(2000)
after(2000)

*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*

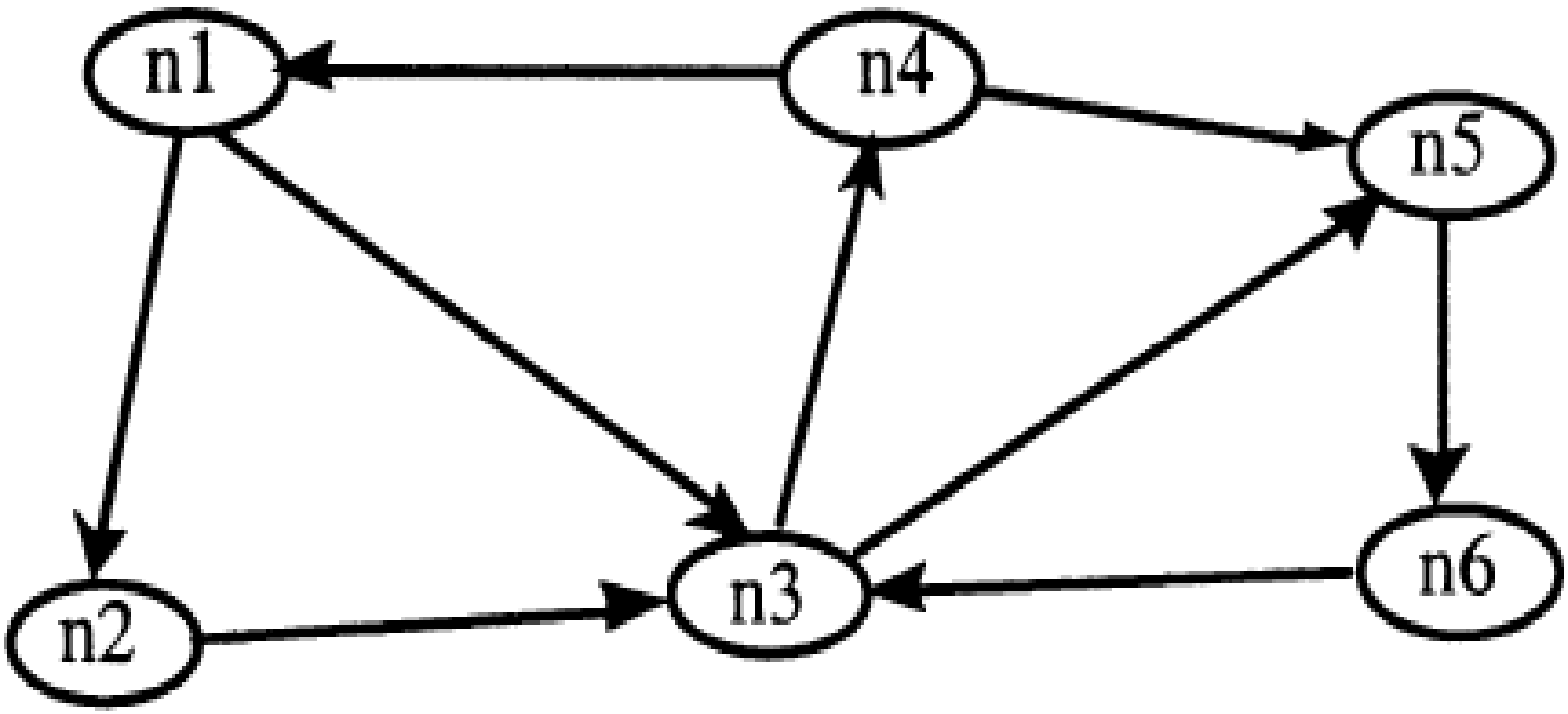
# Plex



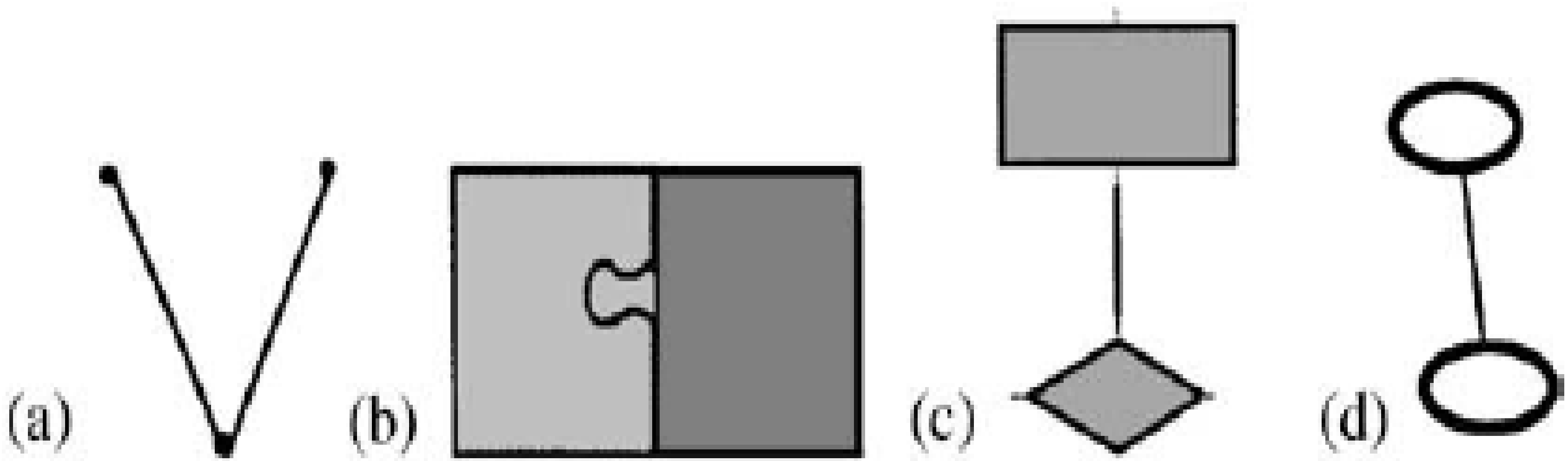

(a)

(b)

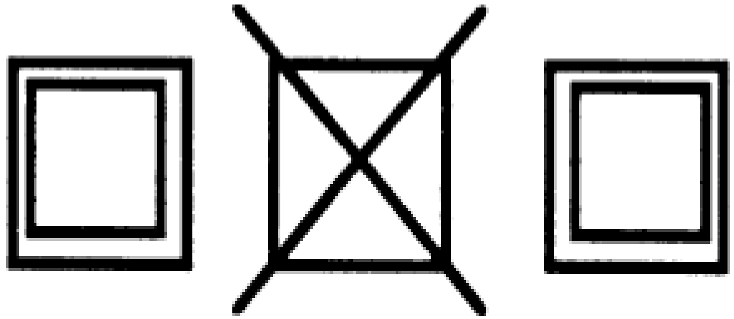
Graph

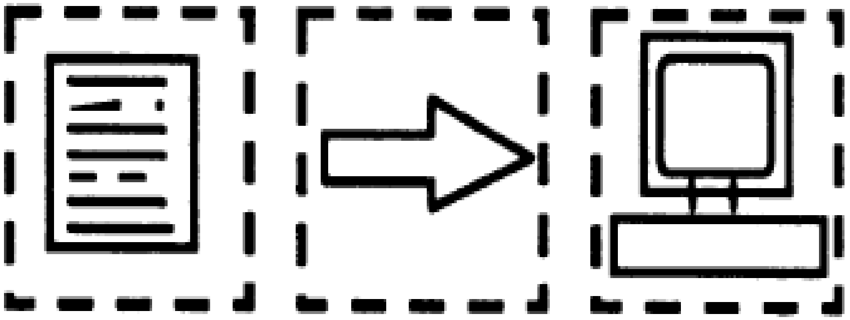


# Connection Types



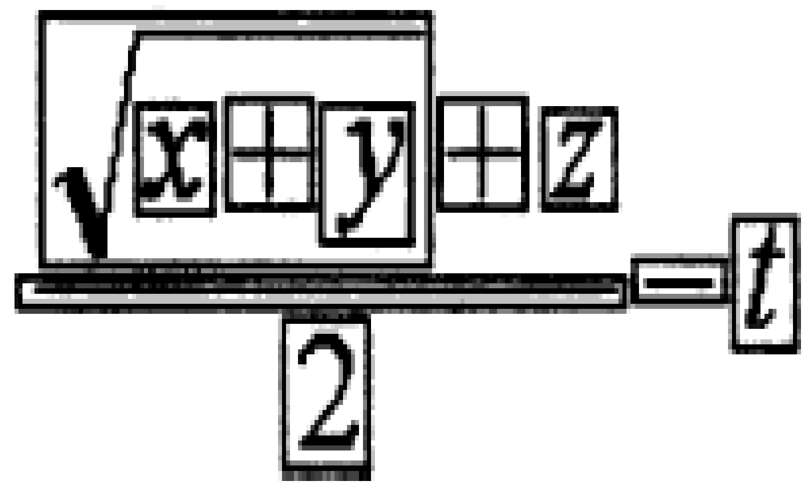
# Iconic



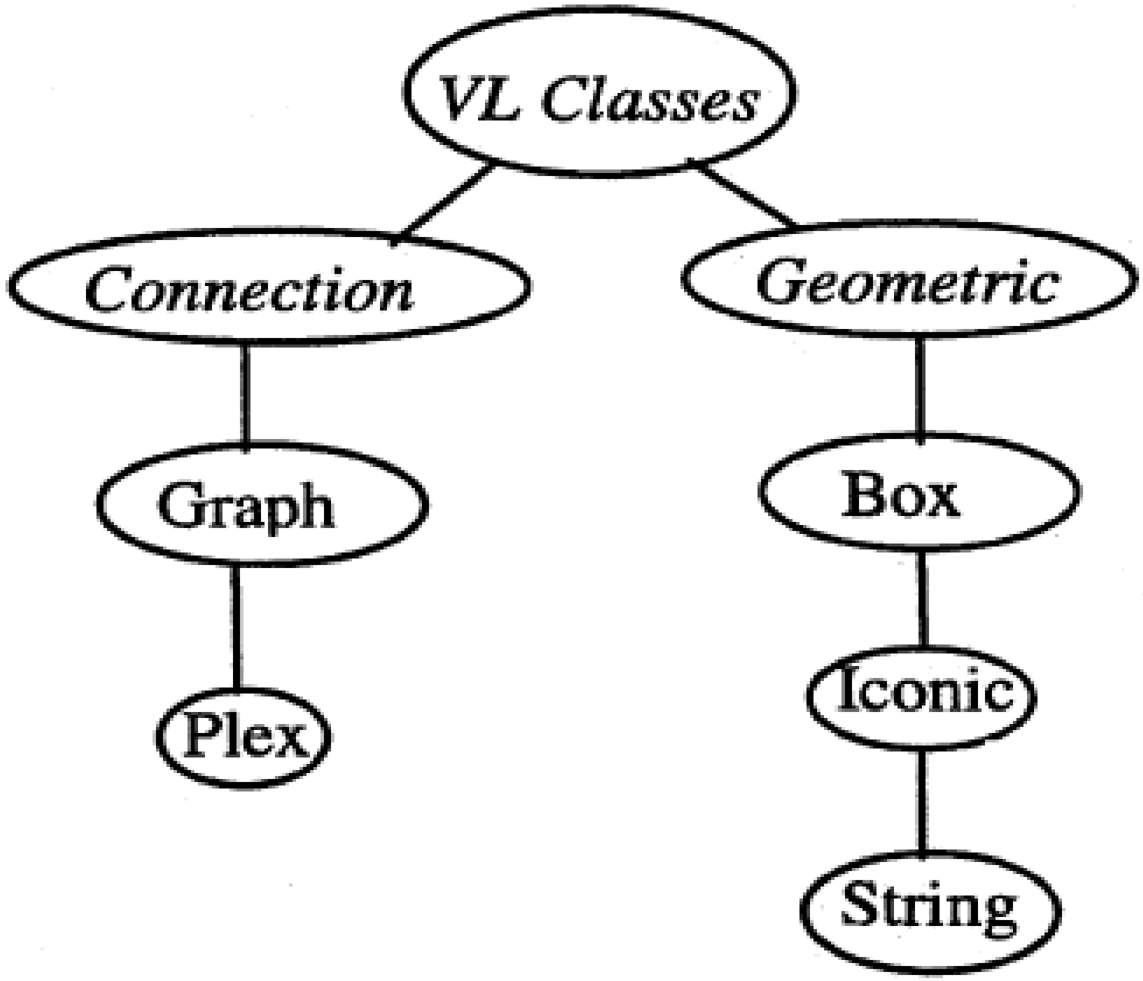


Box

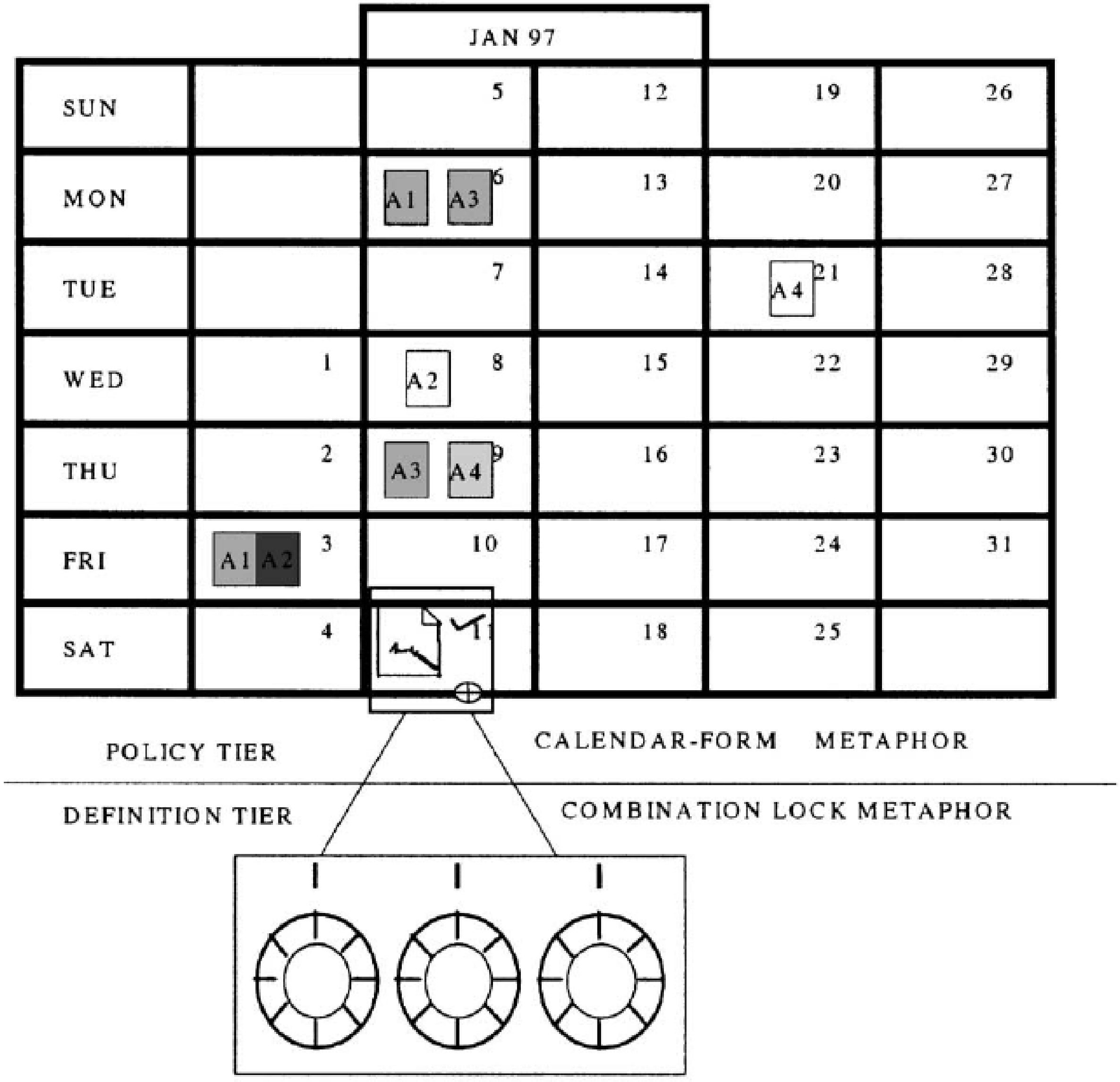
$$\frac{\sqrt{x+y}+z}{2}-t$$



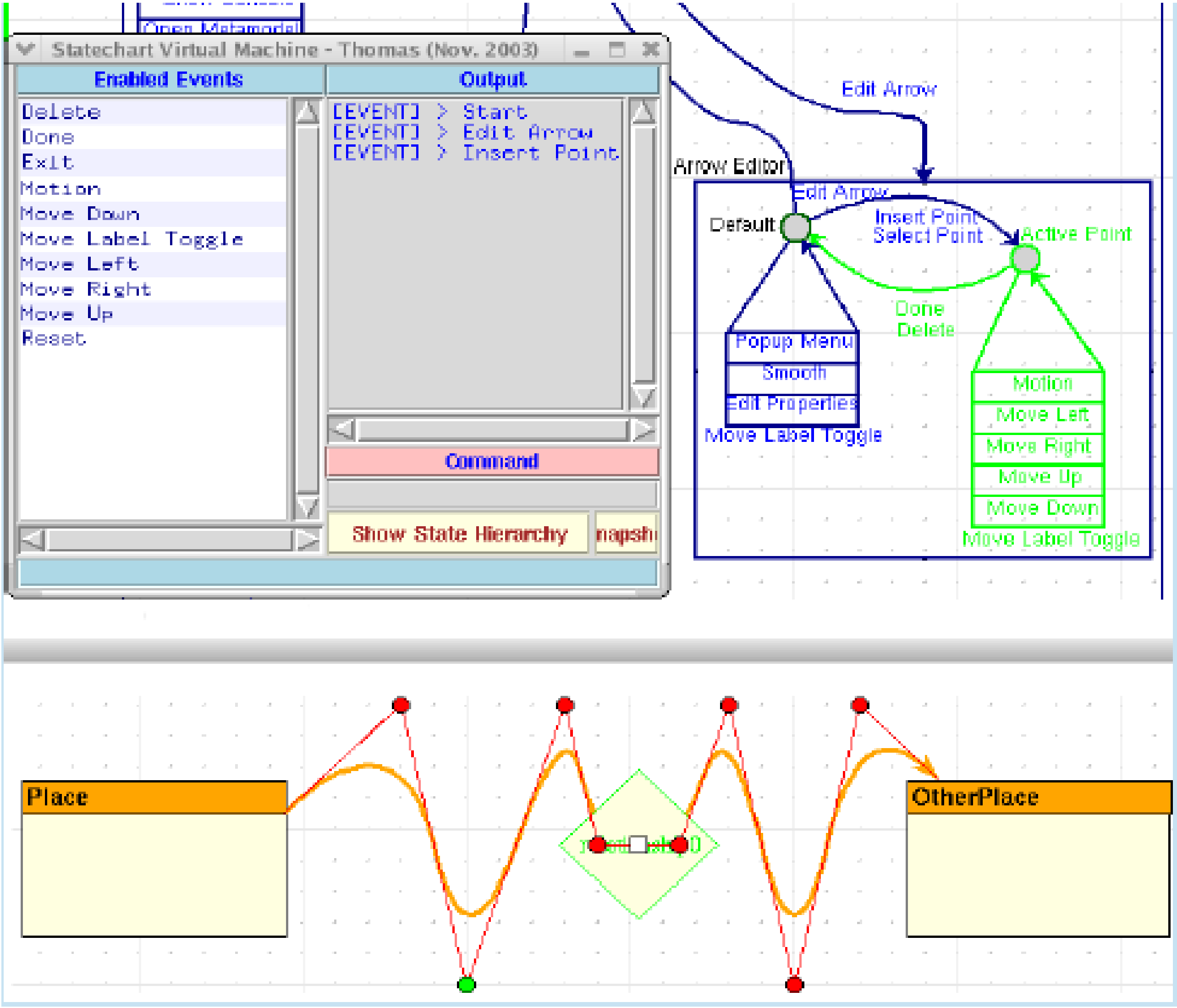
Visual Language Classes



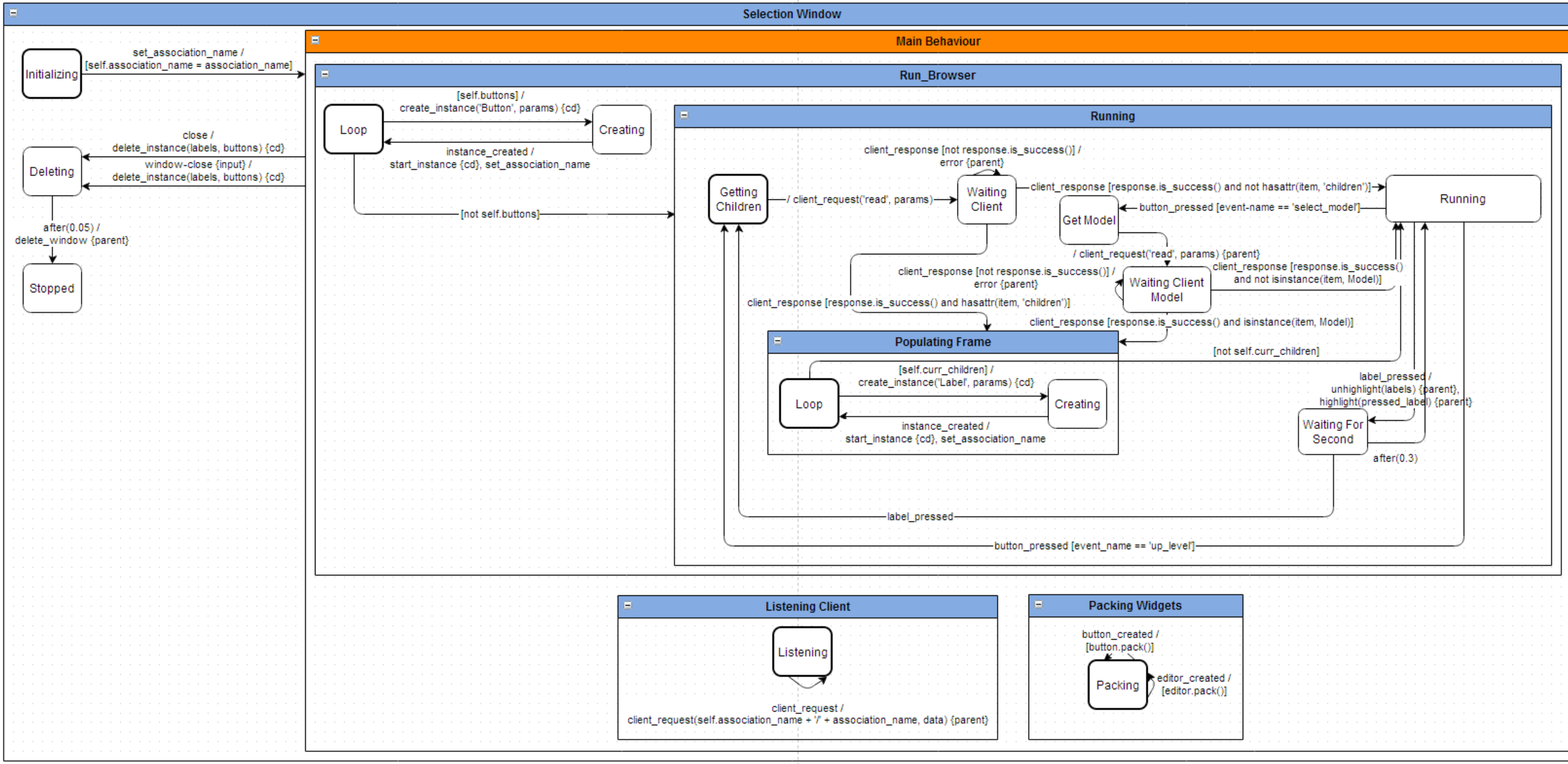
## Hybrid Languages



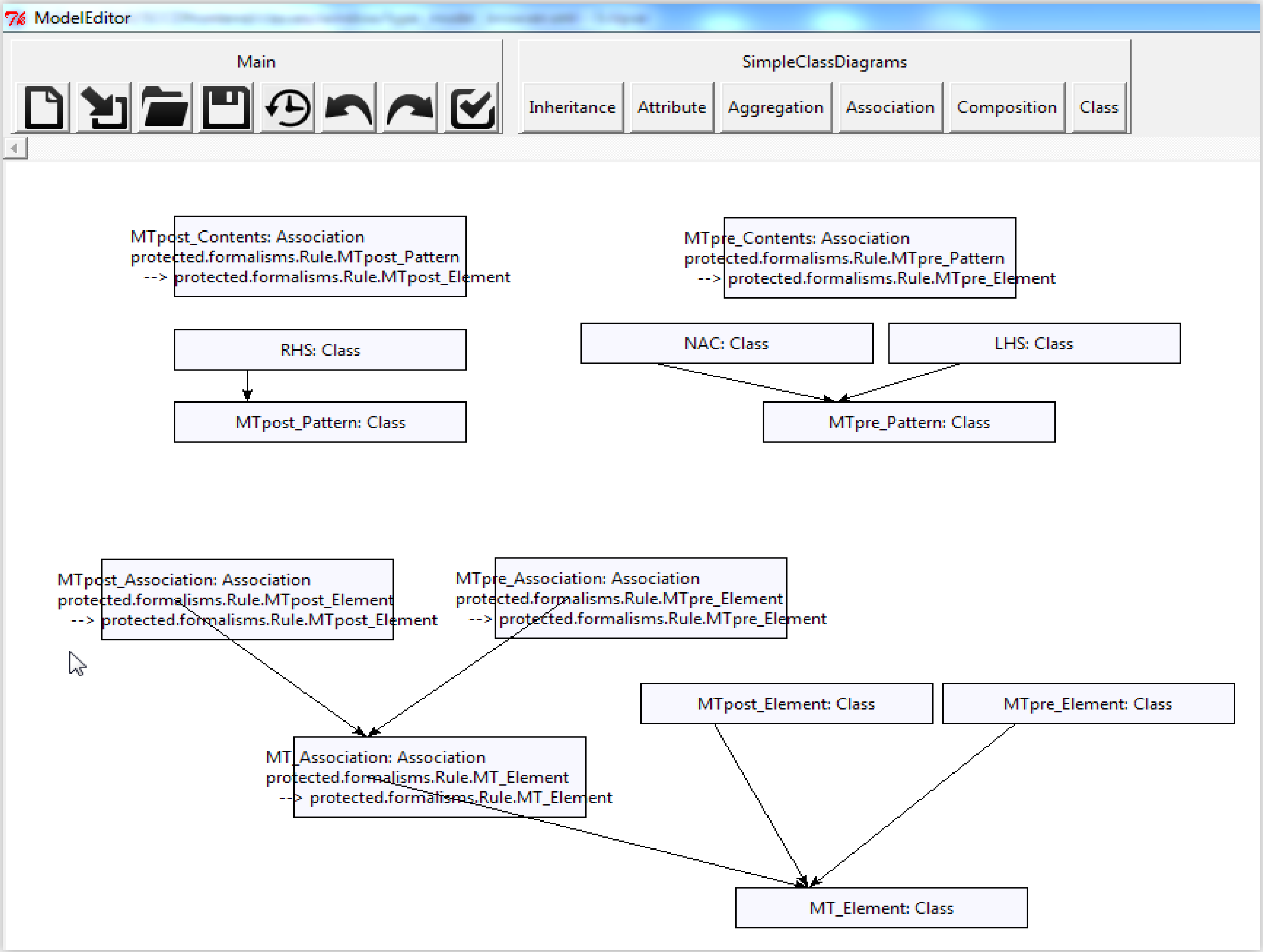
# Syntax-directed Visual Editors: model behaviour



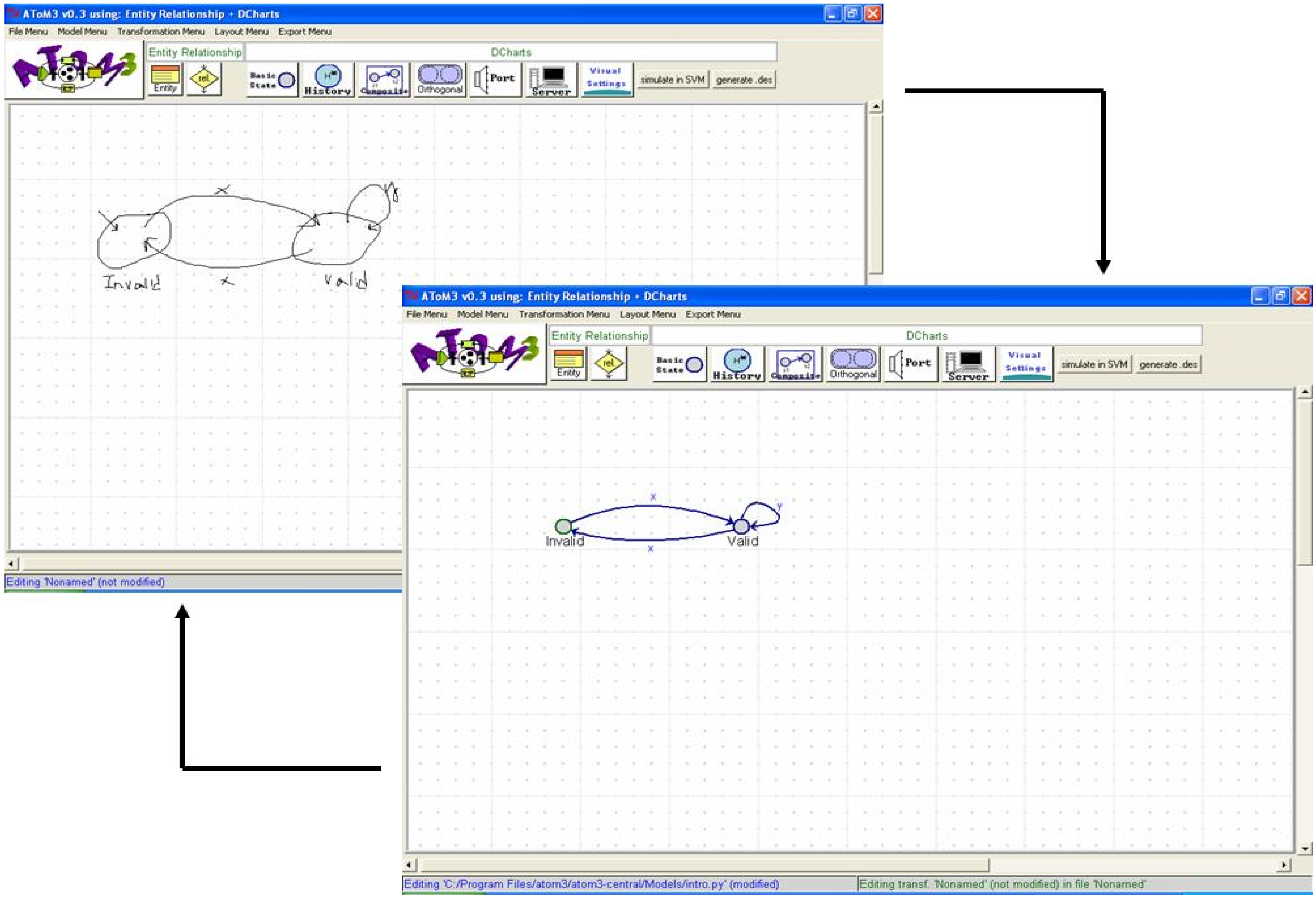
# Syntax-directed Visual Editors: model behaviour



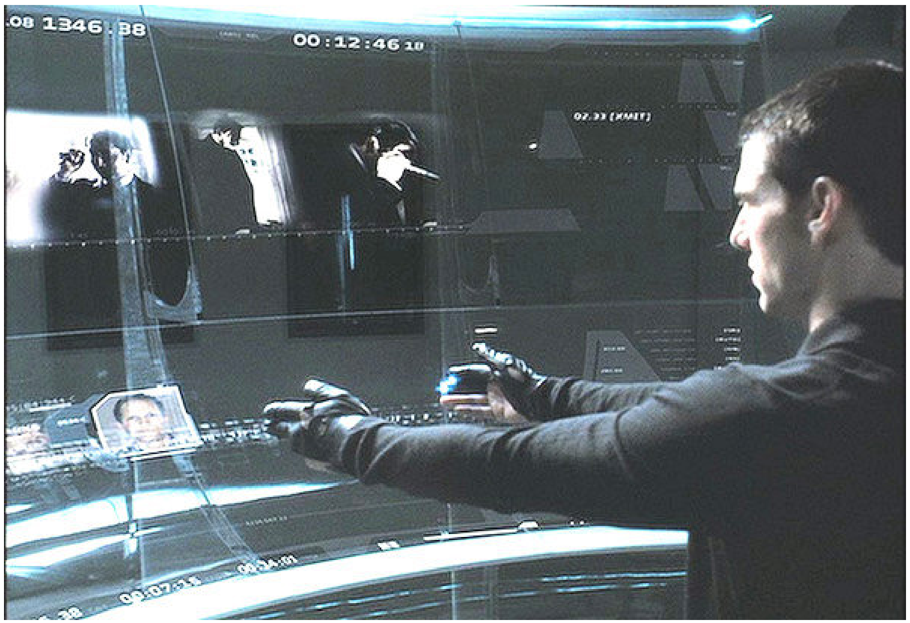
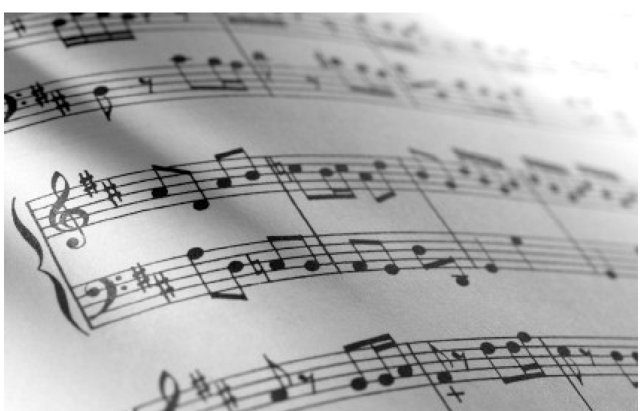
# Generate Syntax-directed Visual Editors

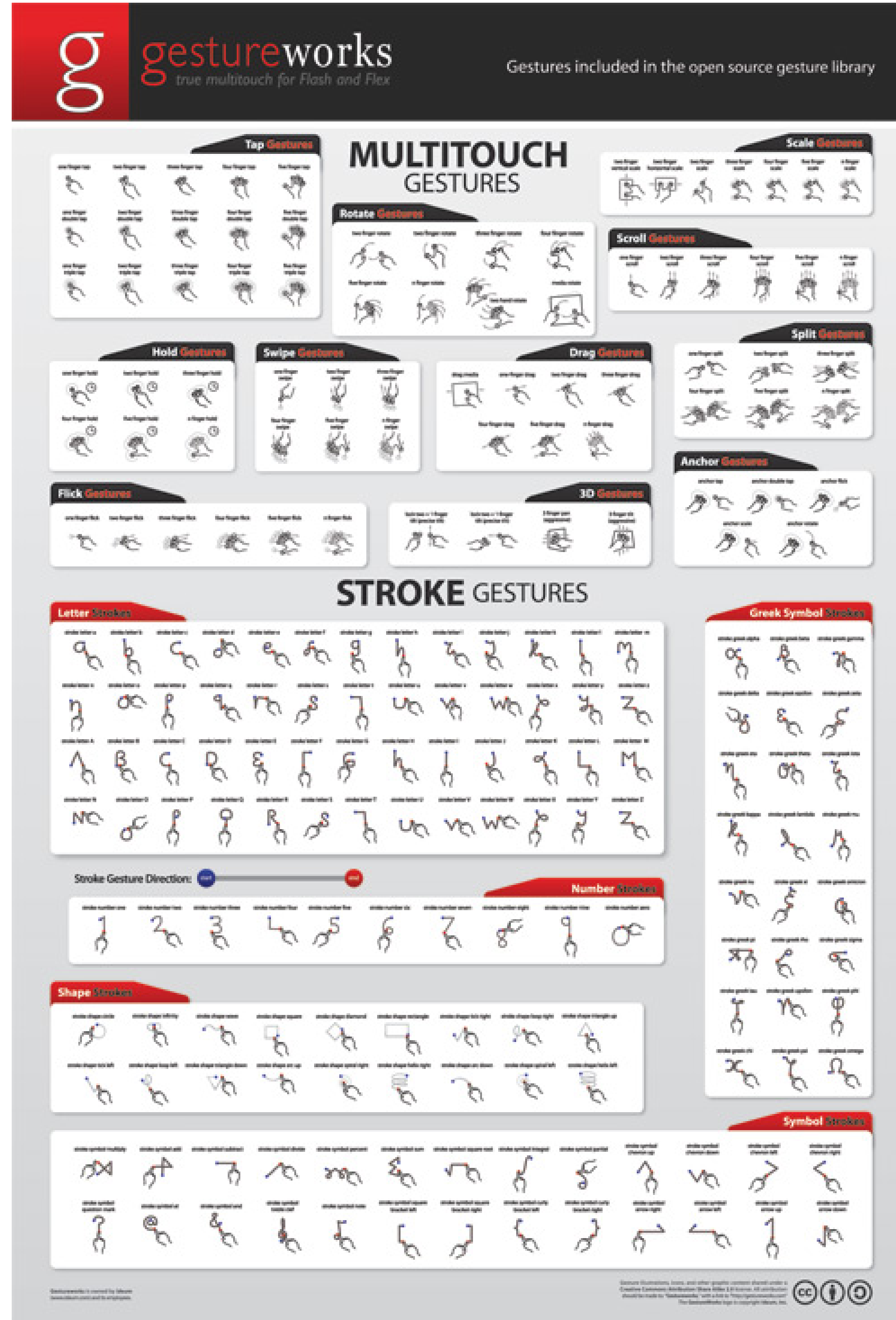


# Syntax-directed Visual Editors: freehand (early stages of multi-domain project)



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

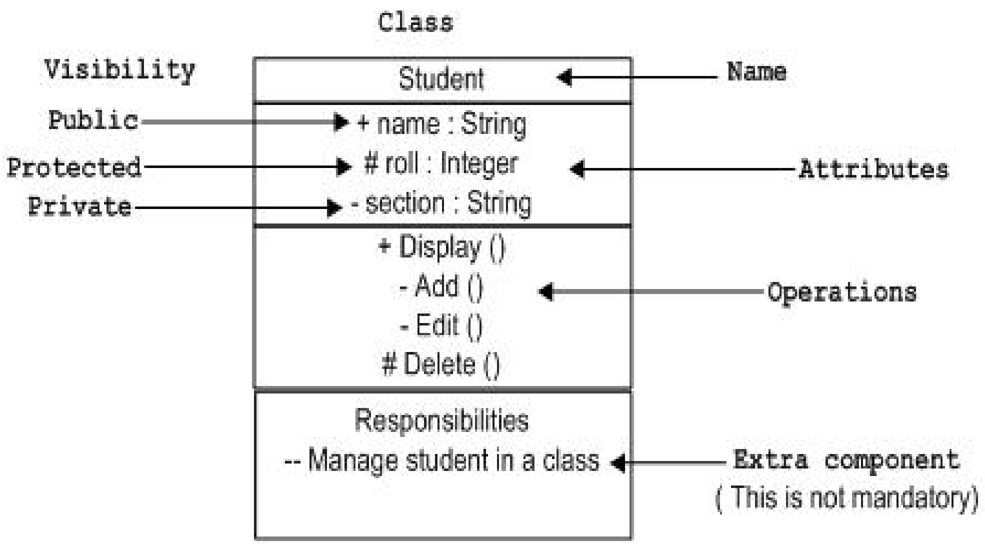
- Visual notations pre-date textual ones
- Visual notations are important for Modelling and Software Engineering
- Humans are excellent pattern recognizers
- Need cognitively efficient and effective notations.

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

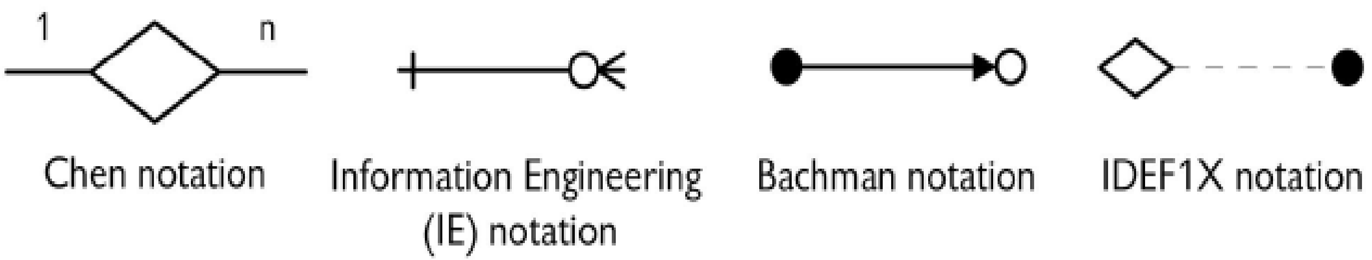
a DSVL @ Lascaux

## Introduction/Rationale

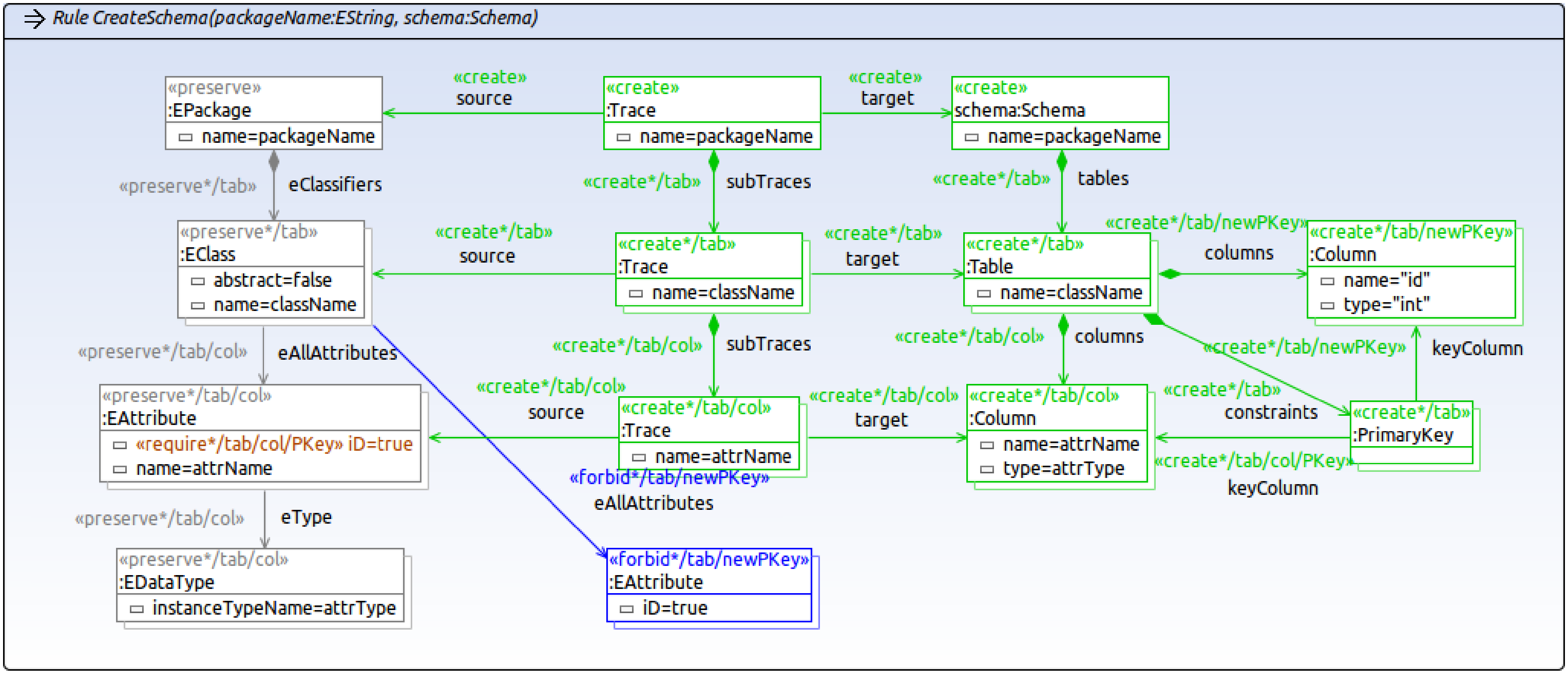
Visual notations are often introduced without underlying theory or rationale

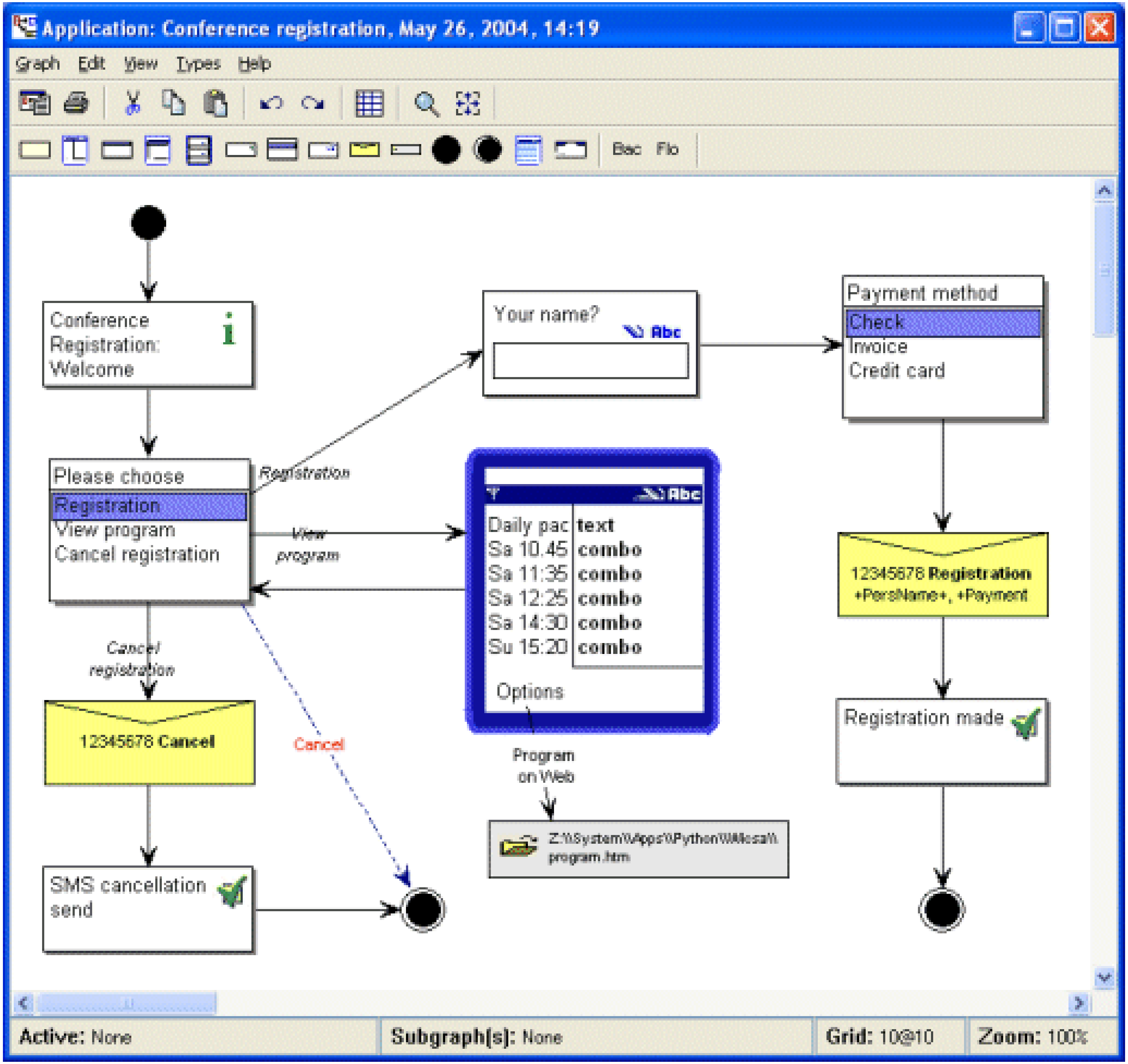



Many visual notations for same concepts.



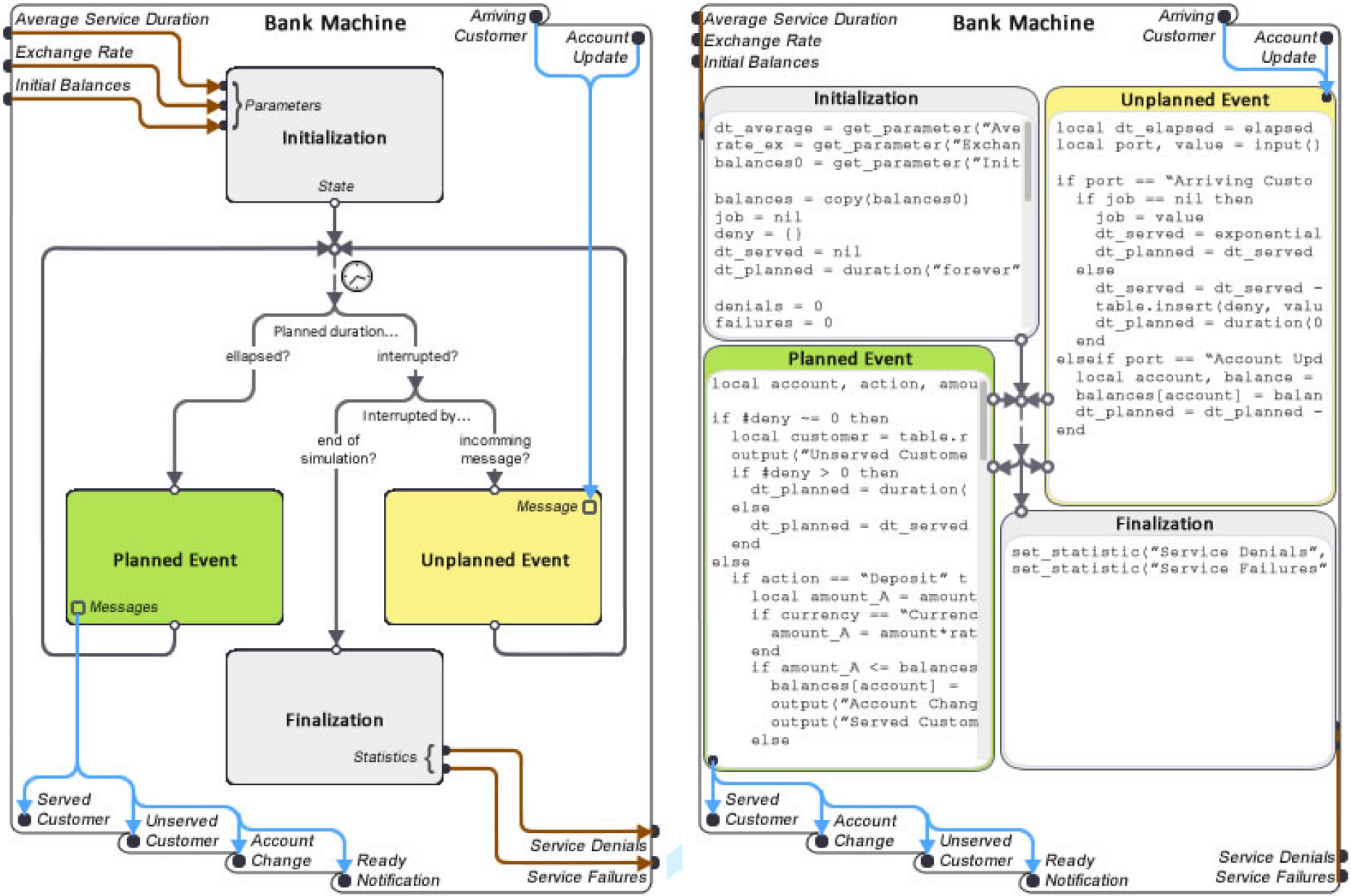

No rigorous way to compare effectiveness and hence no clear design goal.


Rule CreateSchema(packageName:EString, schema:Schema)
«preserve»
:EPackage
name=packageName
«create»
source
«create»
:Trace
name=packageName
«create»
target
«create»
schema:Schema
name=packageName
«preserve*/tab»
eClassifiers
«create*/tab»
subTraces
«create*/tab»
tables
«preserve*/tab»
:EClass
abstract=false
name=className
«create*/tab»
source
«create*/tab»
:Trace
name=className
«create*/tab»
target
«create*/tab»
:Table
name=className
«create*/tab/newPKey»
columns
«create*/tab/newPKey»
:Column
name="id"
type="int"
«preserve*/tab/col»
eAllAttributes
«create*/tab/col»
subTraces
«create*/tab/col»
columns
«create*/tab/newPKey»
keyColumn
«preserve*/tab/col»
:EAttribute
«require*/tab/col/PKey» iD=true
name=attrName
«create*/tab/col»
source
«create*/tab/col»
:Trace
name=attrName
«create*/tab/col»
target
«create*/tab/col»
:Column
name=attrName
type=attrType
«create*/tab»
constraints
«create*/tab»
:PrimaryKey
«create*/tab/col/PKey»
keyColumn
«forbid*/tab/newPKey»
eAllAttributes
«preserve*/tab/col»
eType
«preserve*/tab/col»
:EDataType
instanceTypeName=attrType
«forbid*/tab/newPKey»
:EAttribute
iD=true

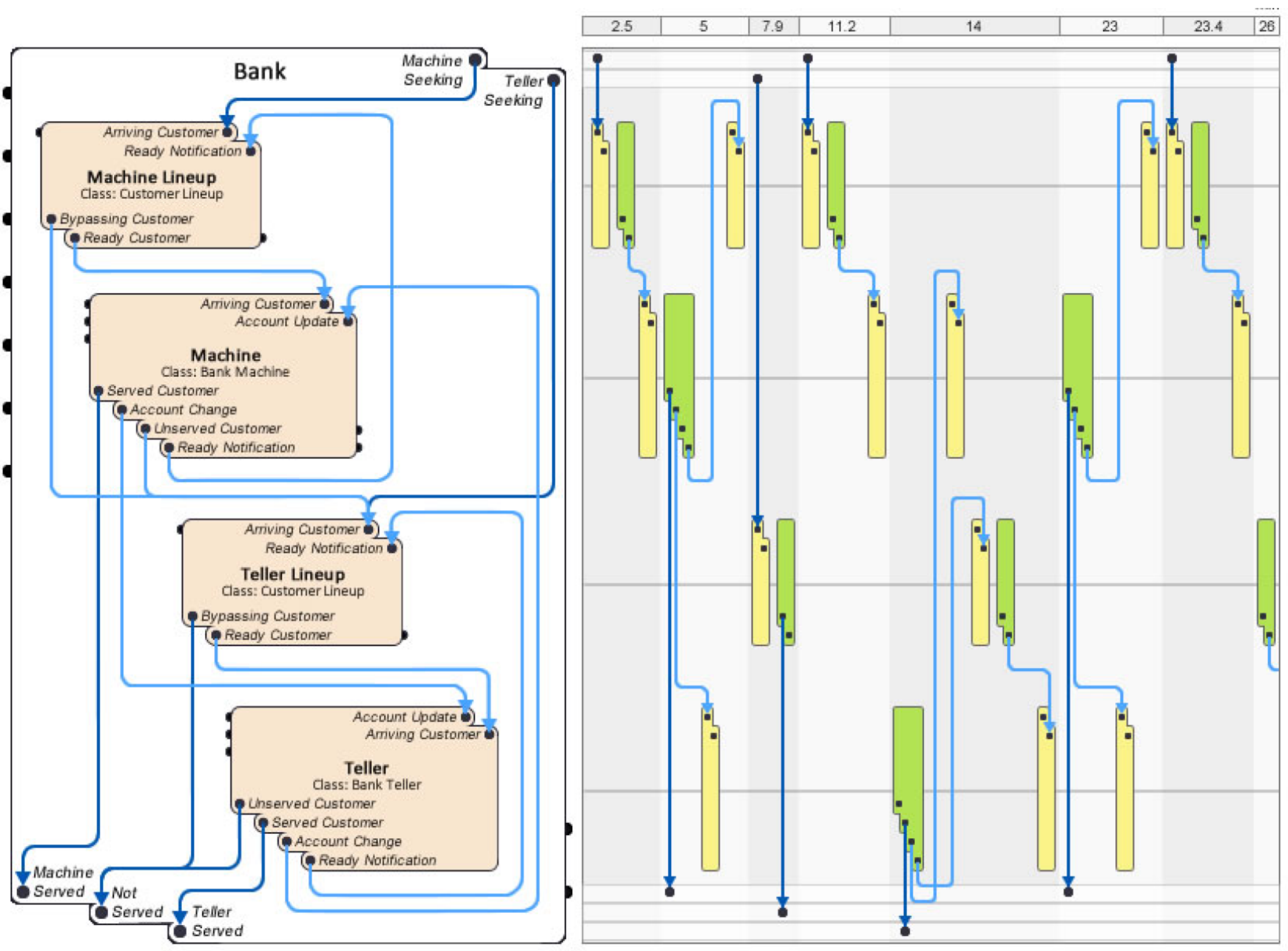


Application: Conference registration, May 26, 2004, 14:19
Graph Edit View Types Help
Bac Flo
Conference Registration: Welcome
Please choose
Registration
View program
Cancel registration
Registration
View program
Cancel registration
Your name?
Abc
Payment method
Check
Invoice
Credit card
Abc
Daily pac text
Sa 10.45 combo
Sa 11:35 combo
Sa 12:25 combo
Sa 14:30 combo
Su 15:20 combo
Options
Program on Web
Z:\\System\\Apps\\Python\\Wilosal\\ program.htm
12345678 Registration +PersName+, +Payment
Registration made
12345678 Cancel
Cancel
SMS cancellation send
Active: None
Subgraph[s]: None
Grid: 10@10
Zoom: 100%


Conference
registration
Daily pa...
995,- USD
Sa 10.4...
T1: Smith
T2: Jackson
T3: Holland
Sa 11:3...
T4: Cohen
Sa 12:2...
T7: Dawson
Ok
Cancel
ABC
C

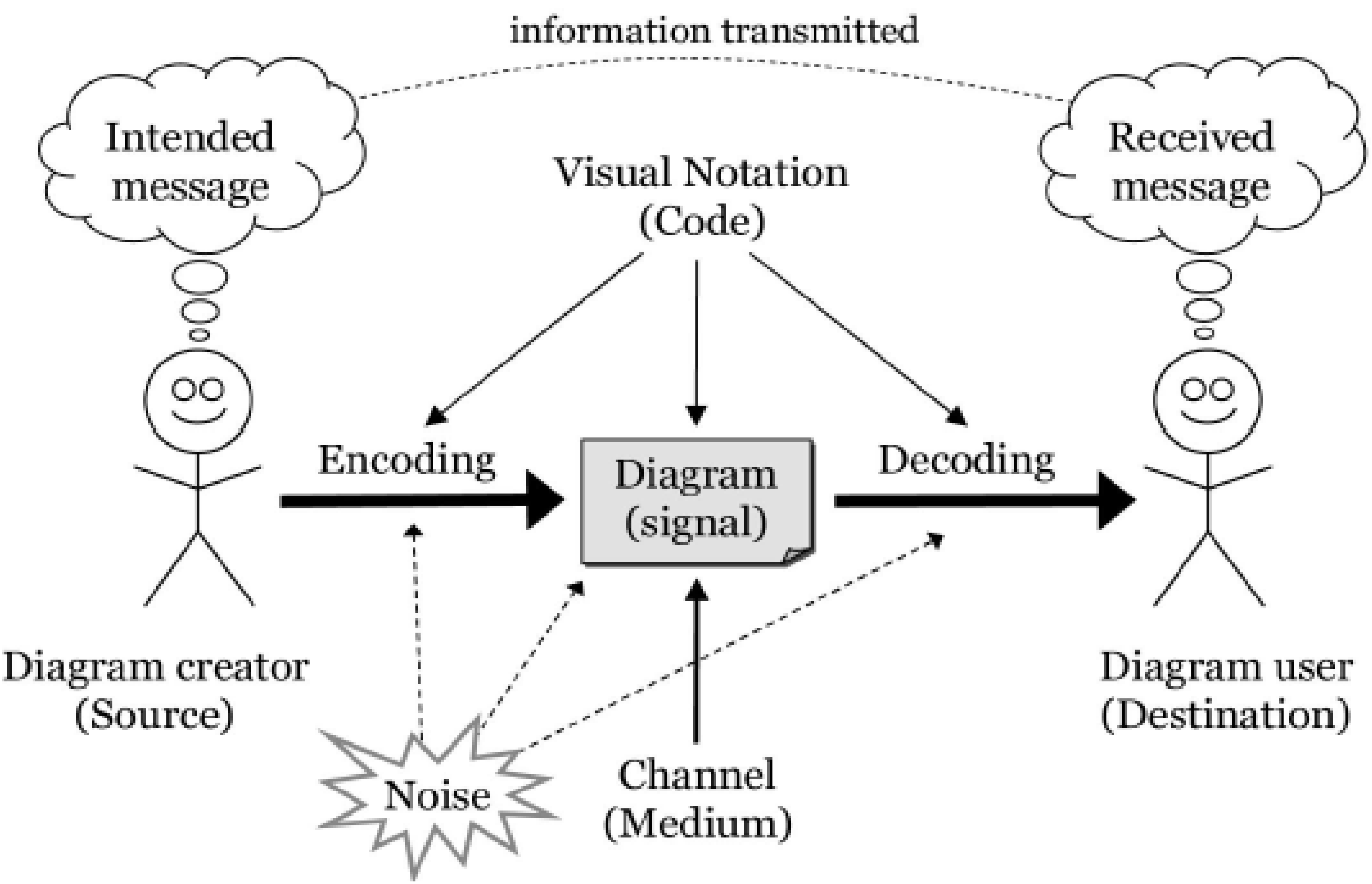



Maryam M. Maleki, Robert F. Woodbury, Rhys Goldstein, Simon Breslav, Azam Khan.
Designing DEVS visual interfaces for end-user programmers. Simulation 91(8): 715-734 (2015)

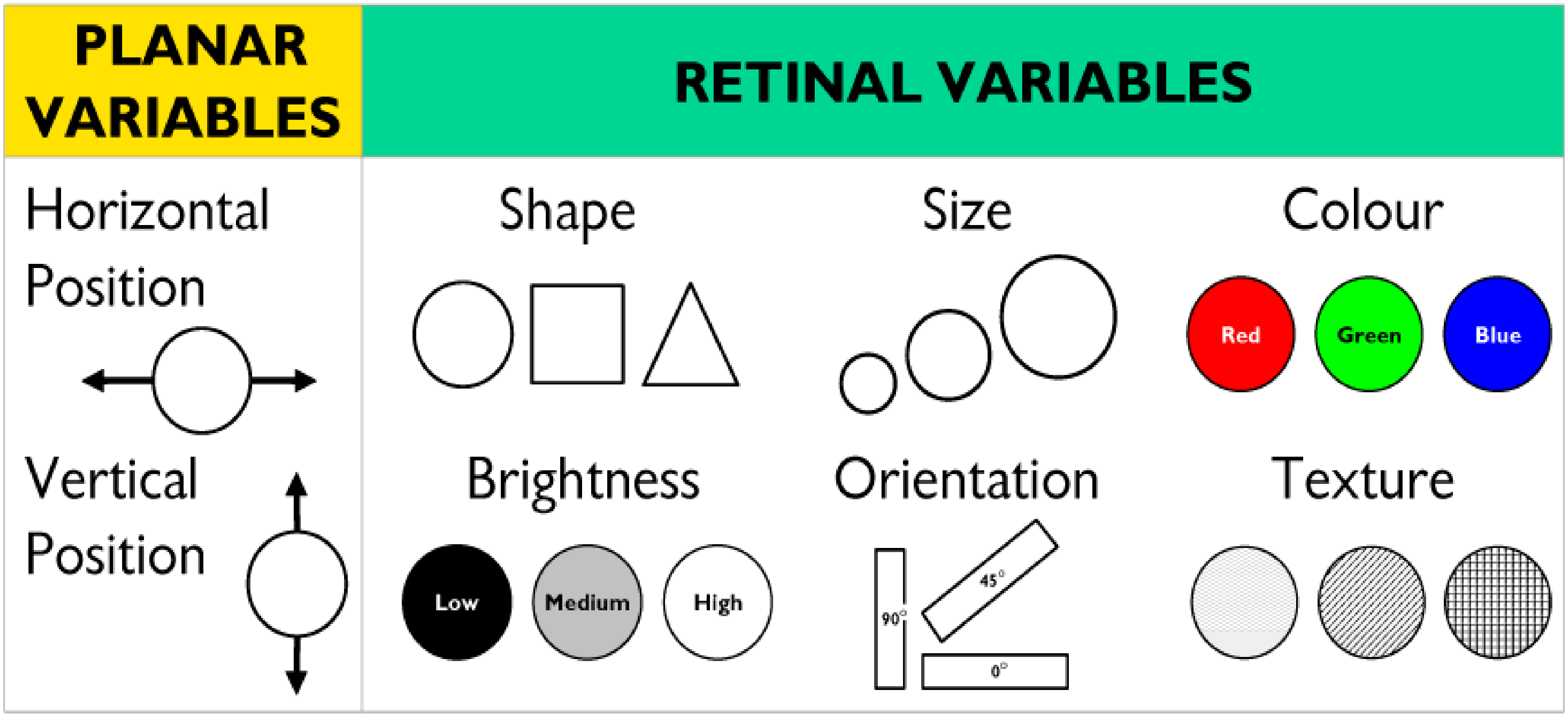


Bank
Machine Seeking
Teller Seeking
Arriving Customer
Ready Notification
Machine Lineup
Class: Customer Lineup
Bypassing Customer
Ready Customer
Arriving Customer
Account Update
Machine
Class: Bank Machine
Served Customer
Account Change
Unserved Customer
Ready Notification
Arriving Customer
Ready Notification
Teller Lineup
Class: Customer Lineup
Bypassing Customer
Ready Customer
Account Update
Arriving Customer
Teller
Class: Bank Teller
Unserved Customer
Served Customer
Account Change
Ready Notification
Machine Served
Not Served
Teller Served
2.5
5
7.9
11.2
14
23
23.4
26

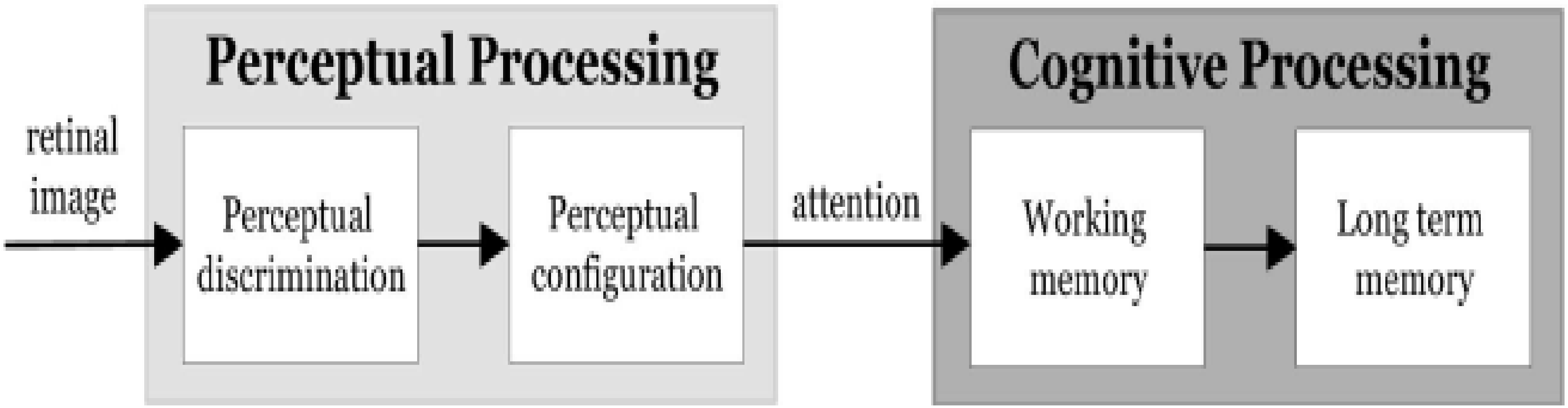
Communication Theory



Encoding: 8 visual variables to (graphically) encode information



# Decoding




automatic, fast, parallel

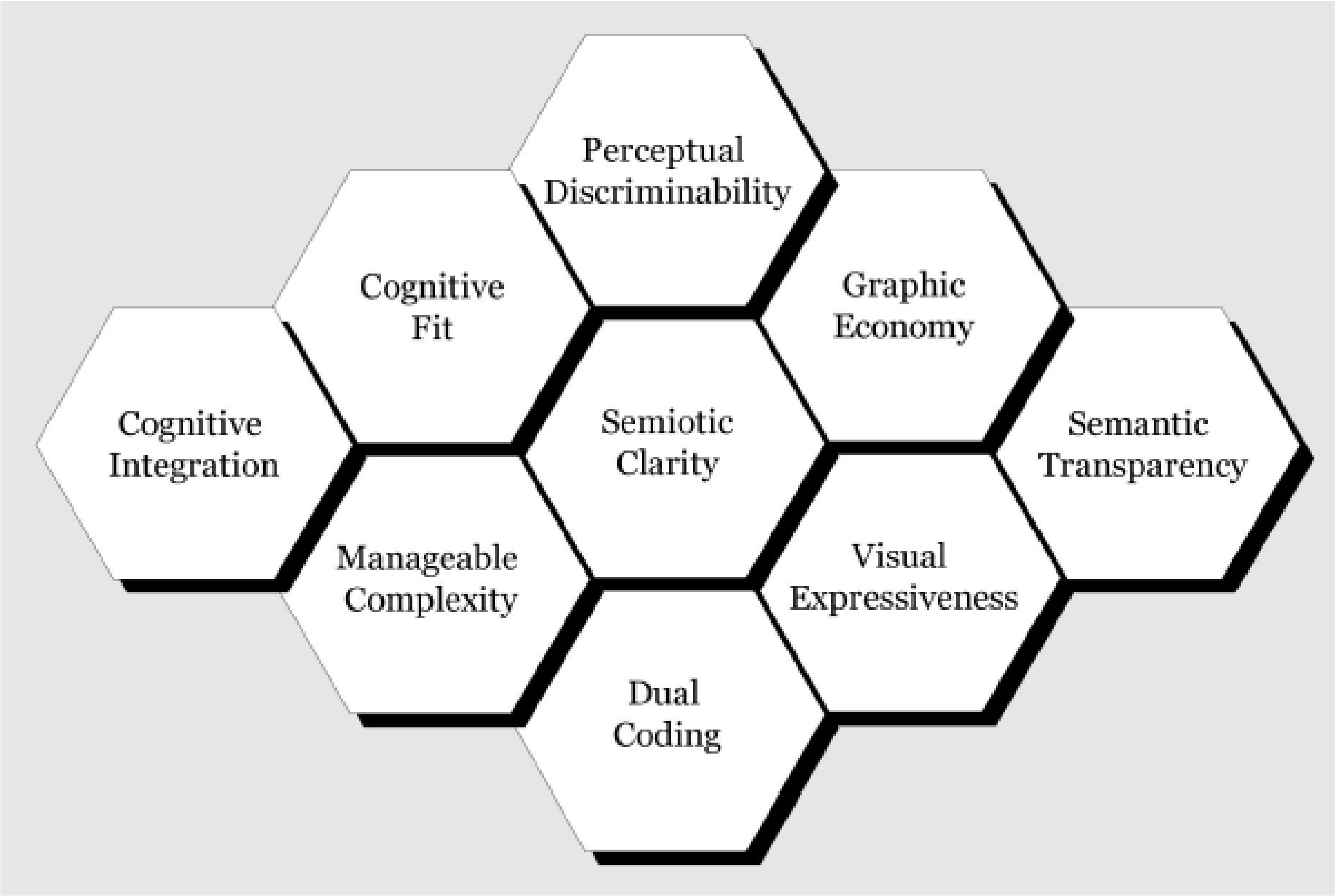
slow, large effort, sequential

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

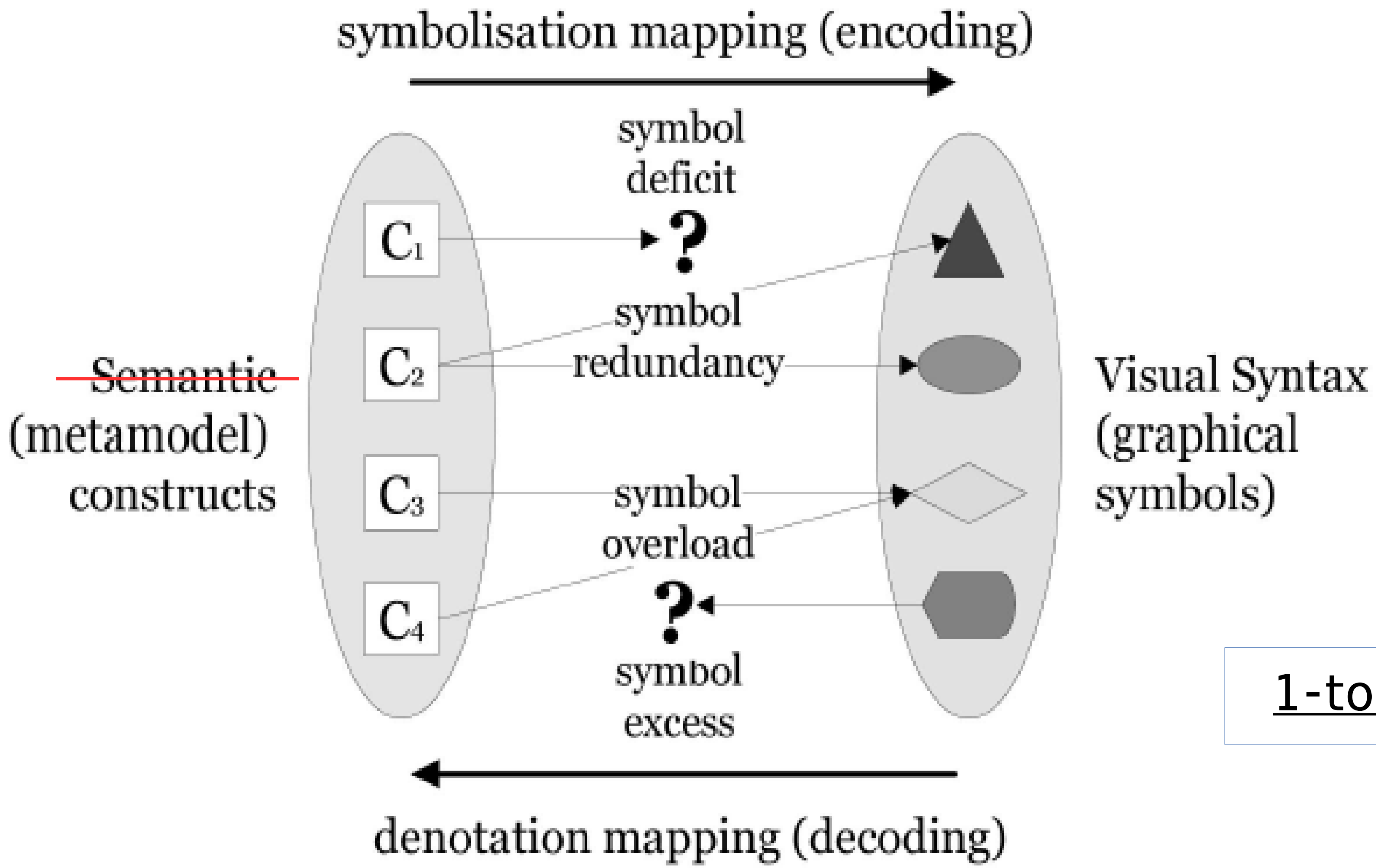
Note: "dual channel theory":
auditory/verbal channel and visual/pictorial channel are processed in parallel

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

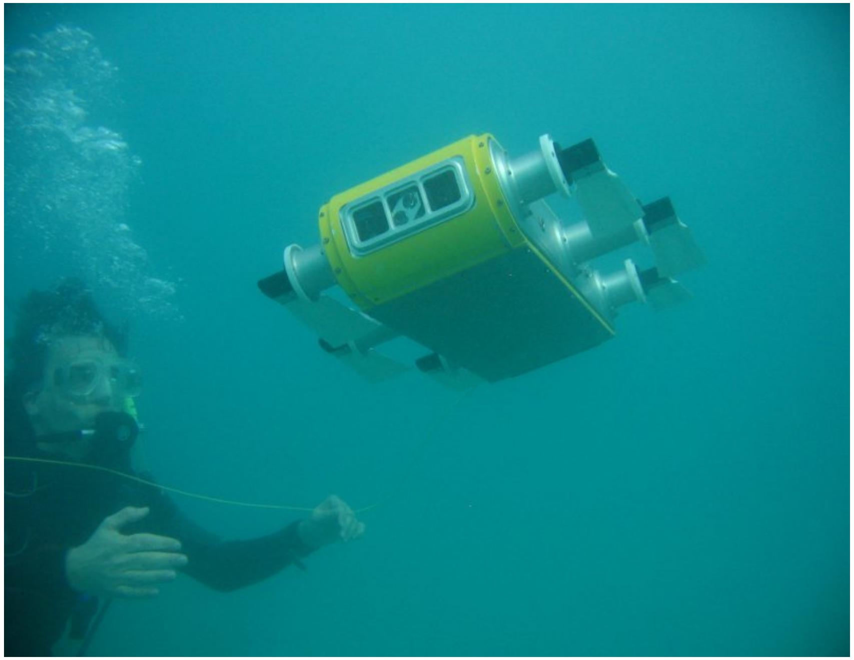


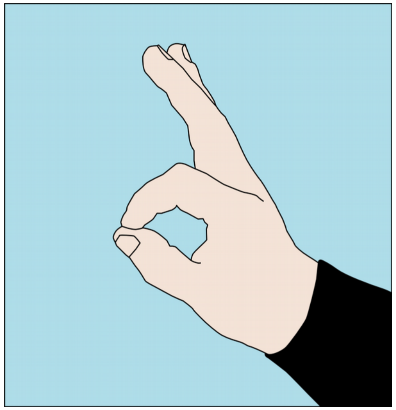
Semiotic Clarity (semiotics = study of signs and sign processes)

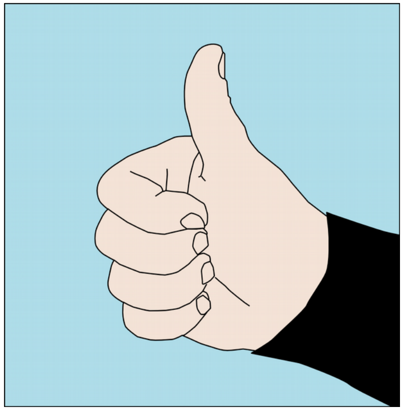


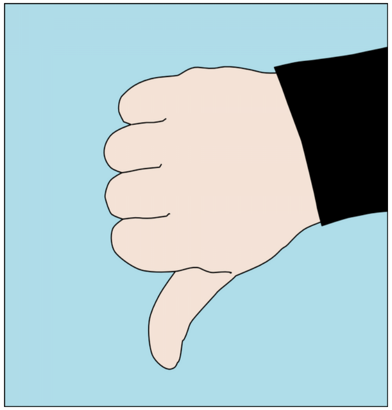

1-to-1

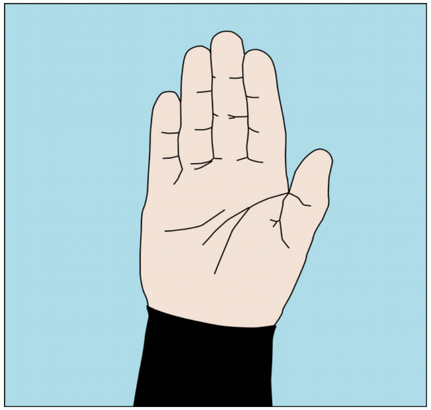
# Perceptual Discriminability







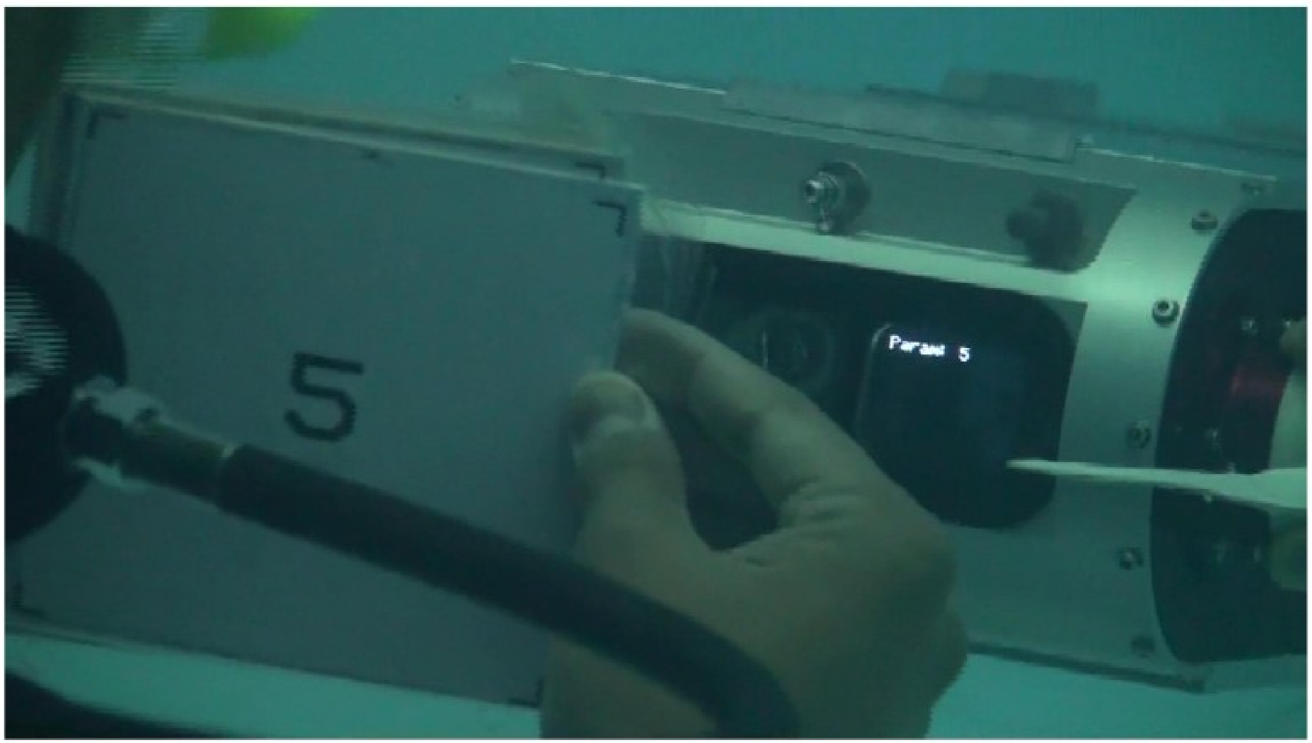




## ``Physics'' of Notations

(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.

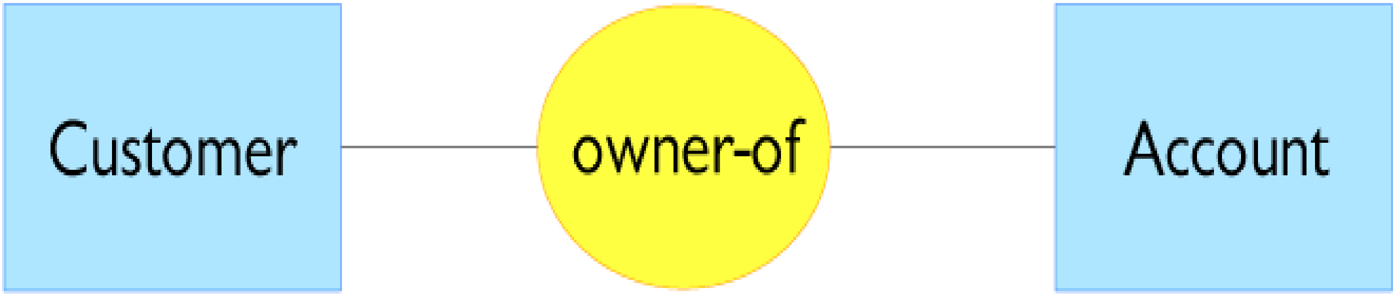
Junaed Sattar, Gregory Dudek. Reducing Uncertainty in Human-Robot Interaction: A Cost Analysis Approach. ISER 2010: 81-95.

## Perceptual Discriminability

should be easy to **distinguish** visual symbols

ability to distinguish is determined by **visual distance**
larger visual distance » faster, more accurate recognition

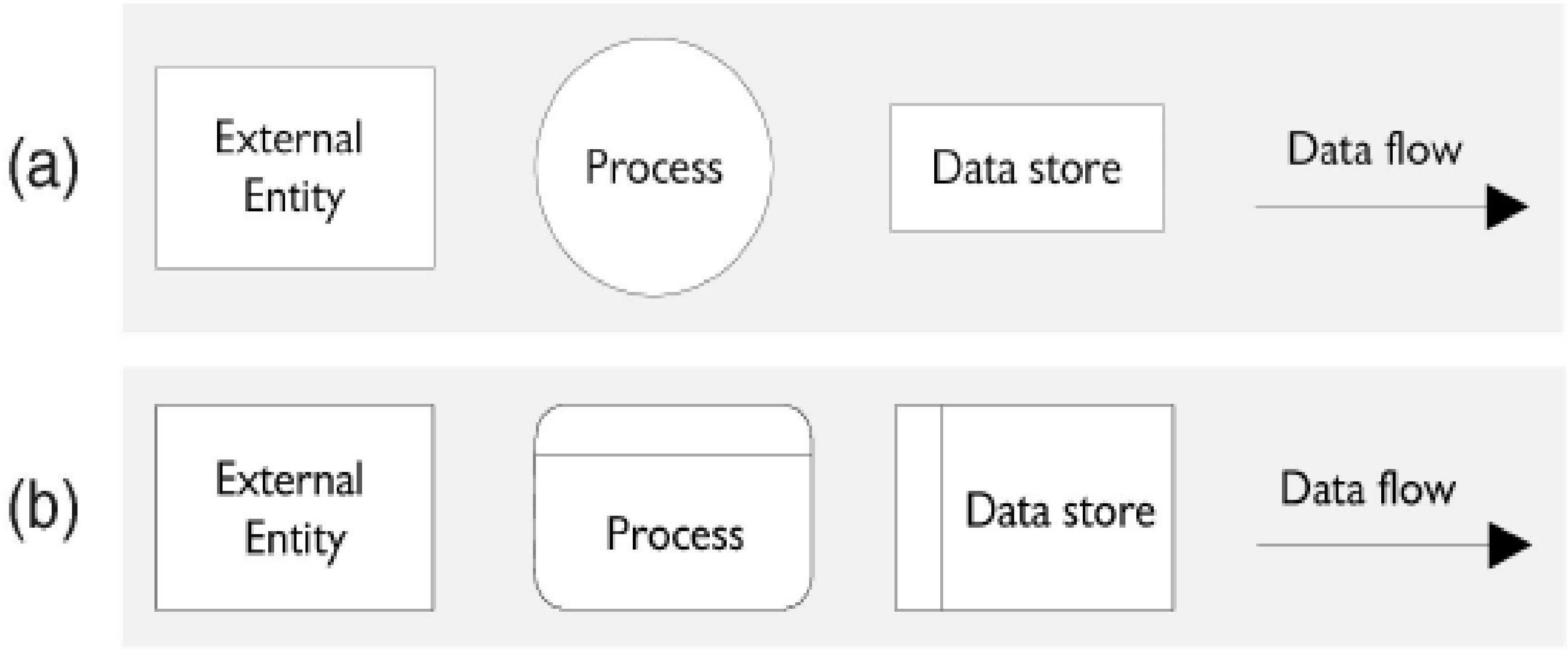
- **number** of visual variables on which they differ and the **magnitude** of the differences
- **shape** is the main visual variable



Perceptual Discriminability

Software Engineering notations mostly use rectangle variants

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



## Semantic Transparency

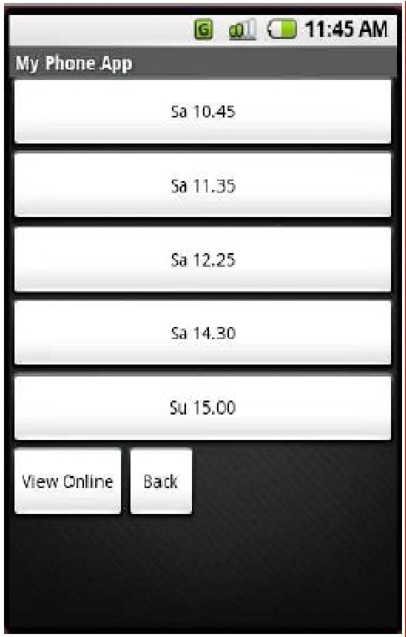
The meaning of a symbol can be inferred from its appearance (intuitive)

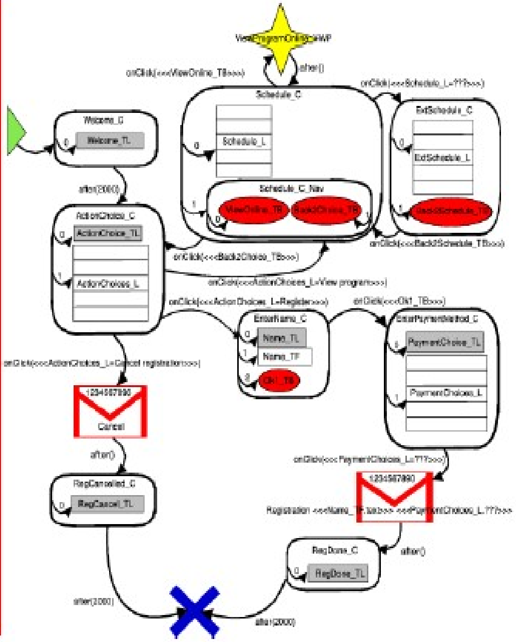
Symbols can be:

- Semantically Immediate
- Semantically Opaque
- Semantically Perverse

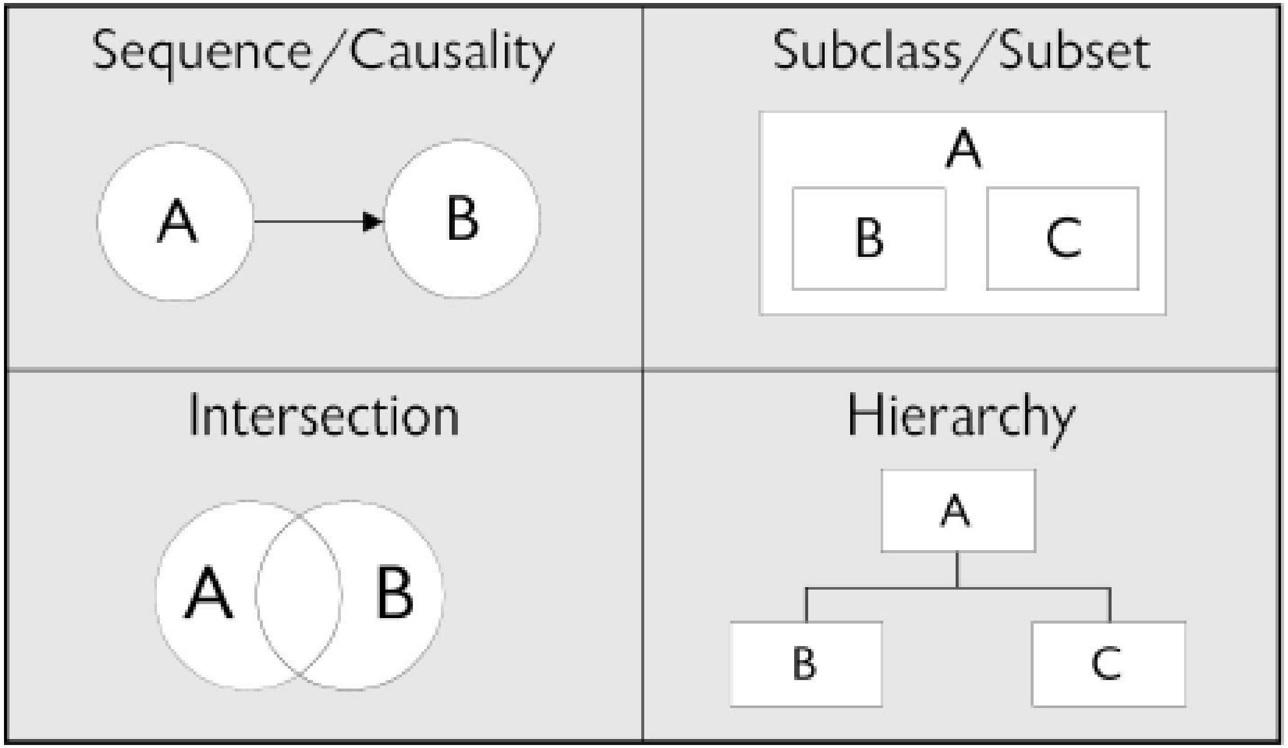
Software Engineering notations are usually abstract (non-intuitive)

Domain-specific icons and visual arrangement should be intuitive

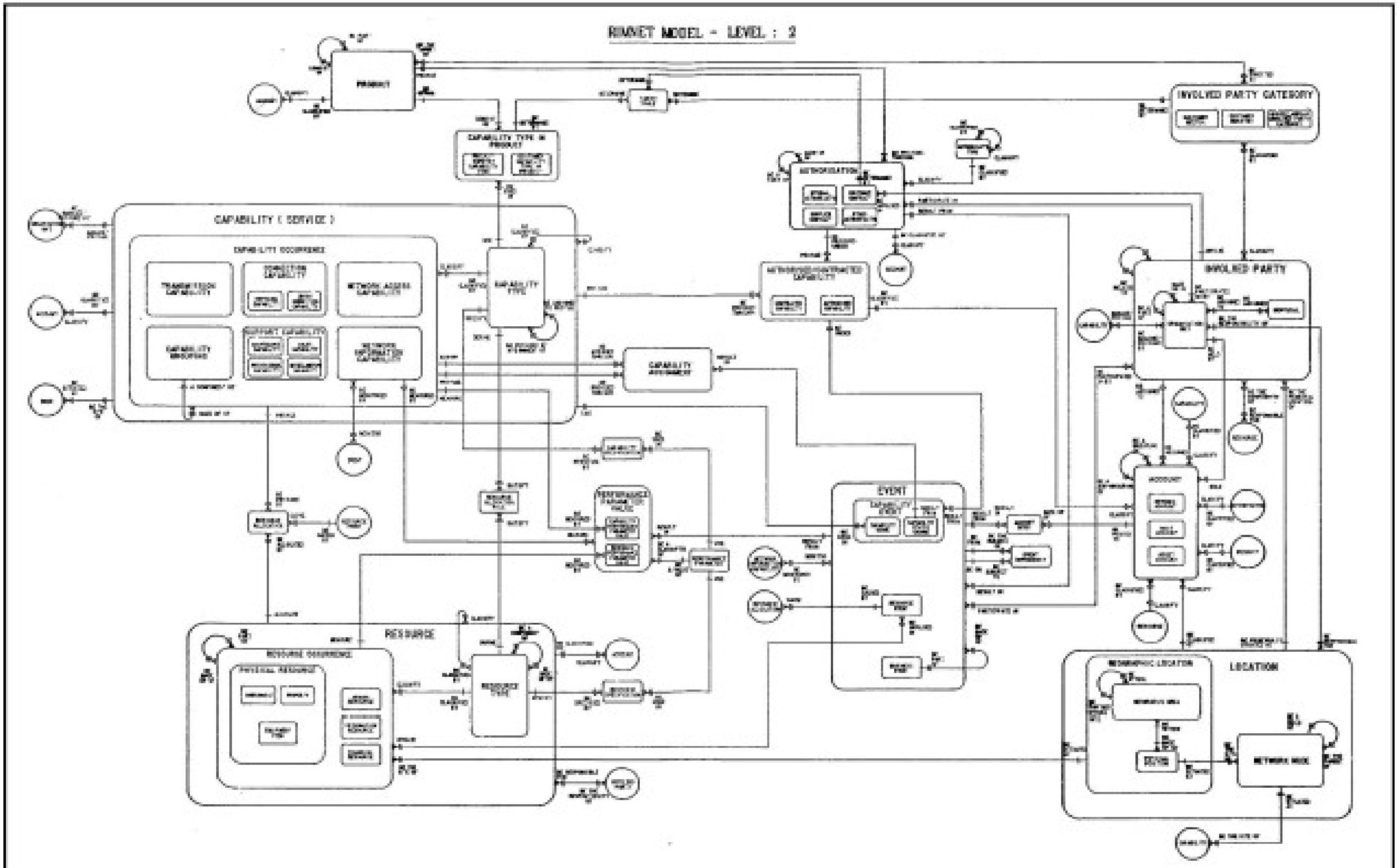


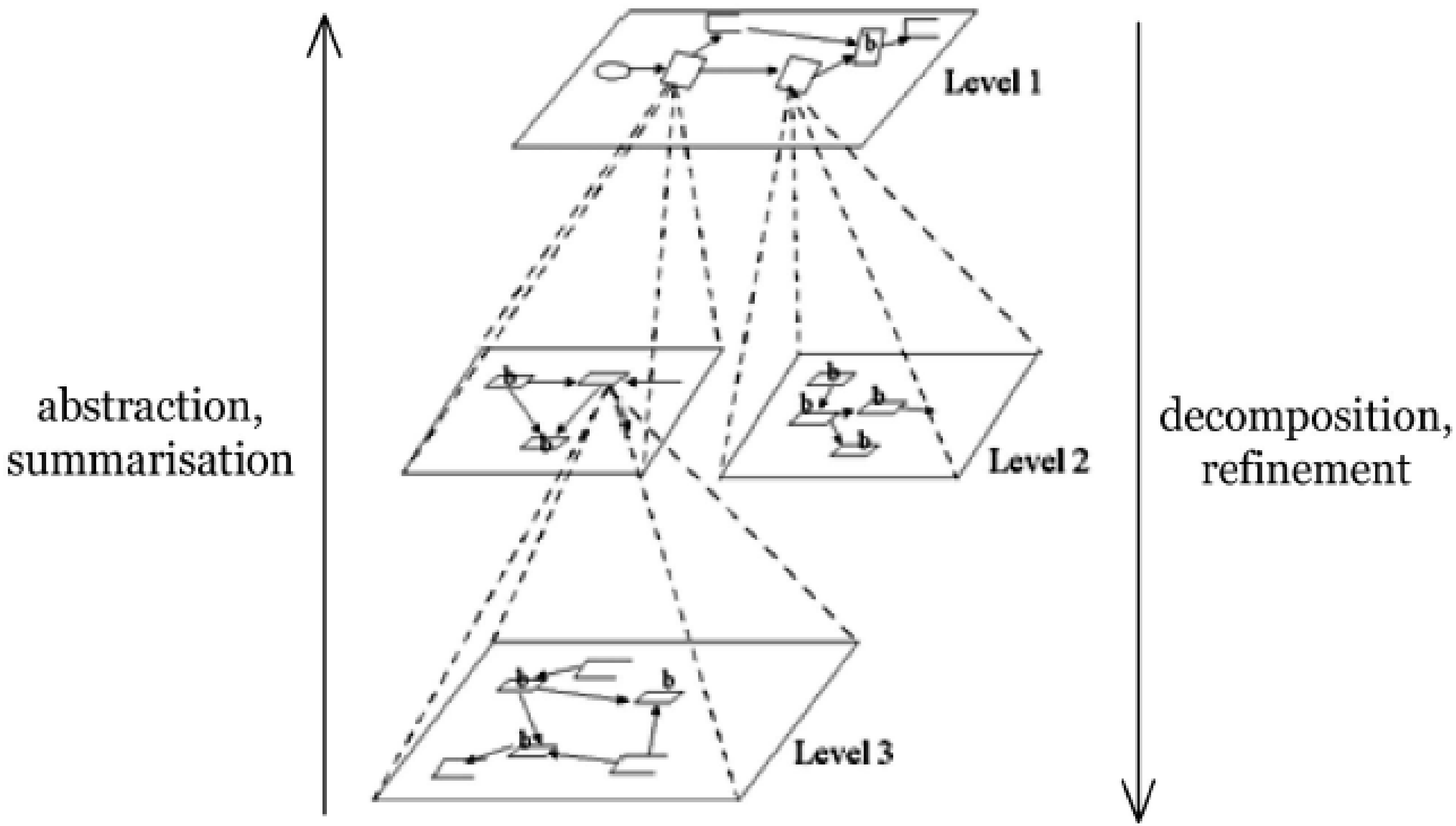
# Semantic Transparency



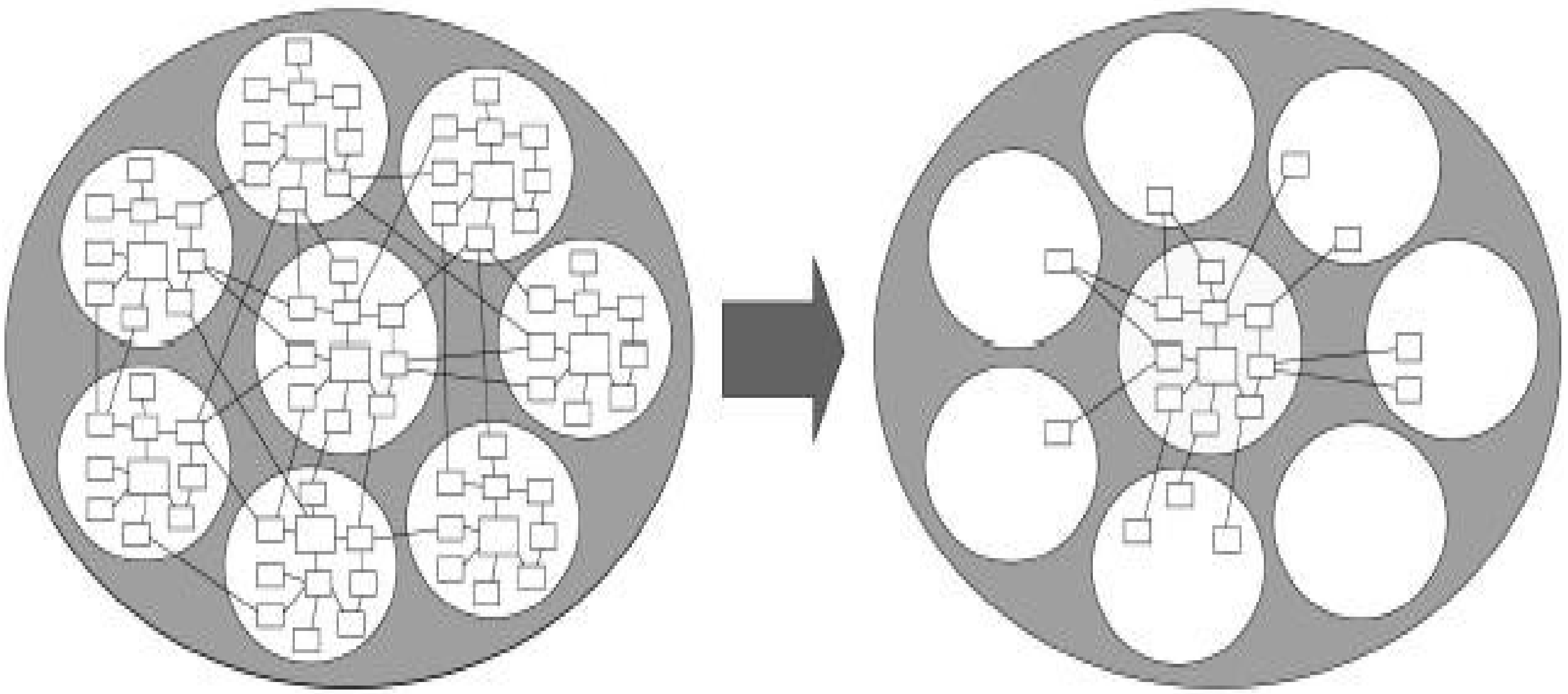
Complexity management (# elements in diagram » cognitive overload)



Modularization/ Hierarchy



## Cognitive Integration (different notations)

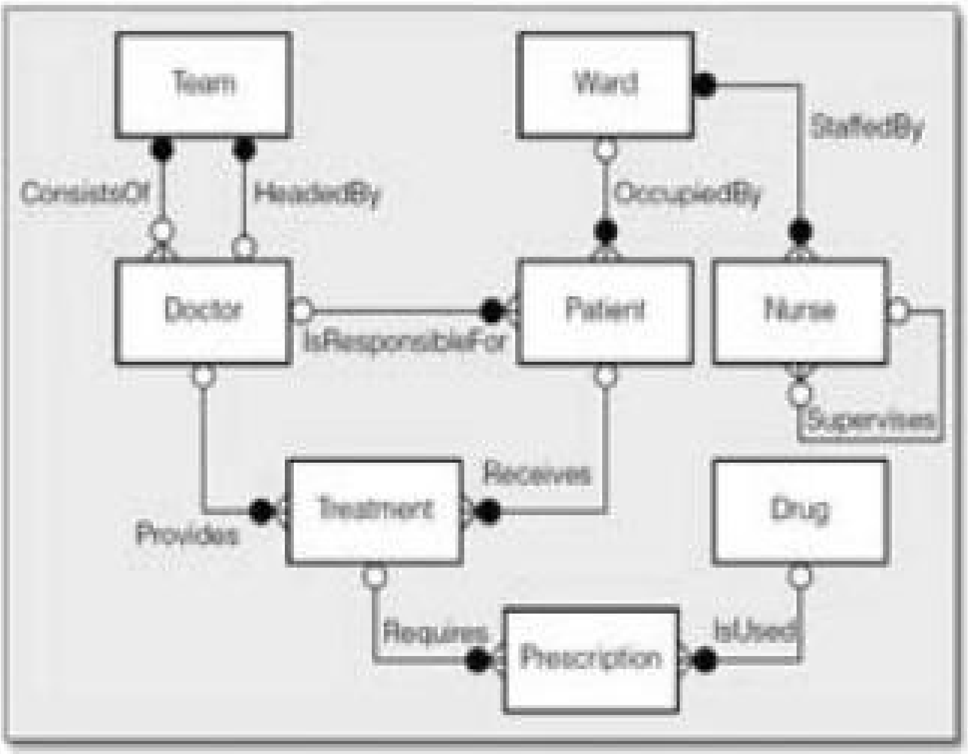


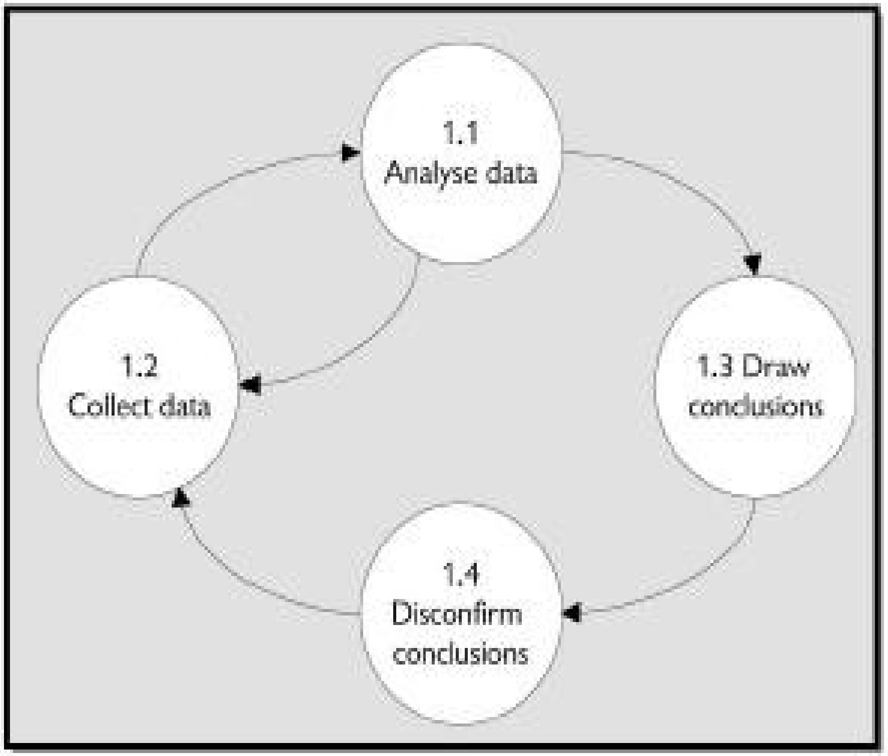
- Conceptual integration (coherent mental model)
- Enable navigation and transition between notations

# 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

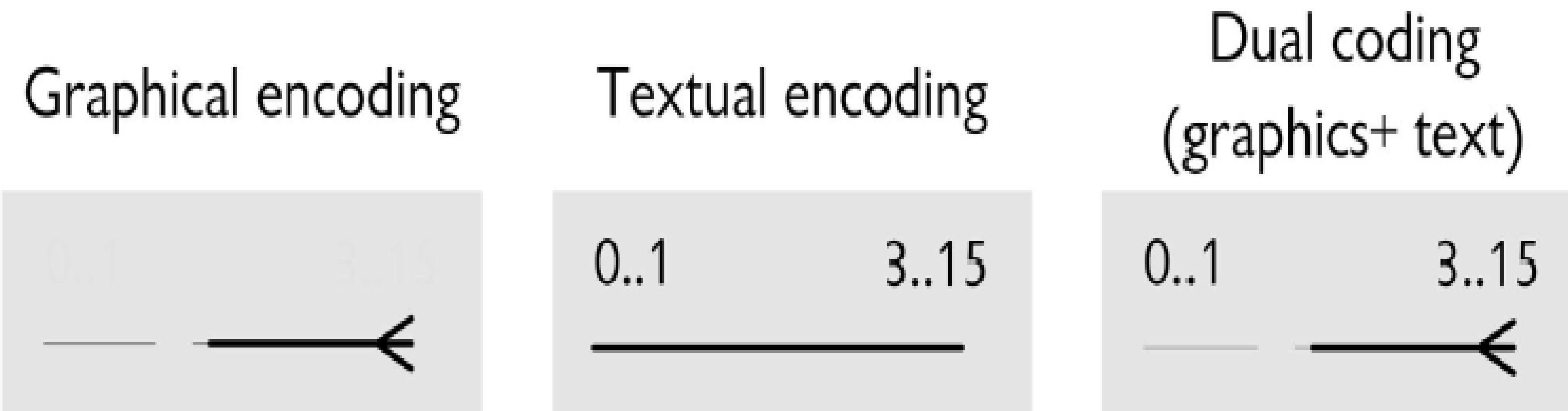
Different visual variables have different capacity to encode information

| 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

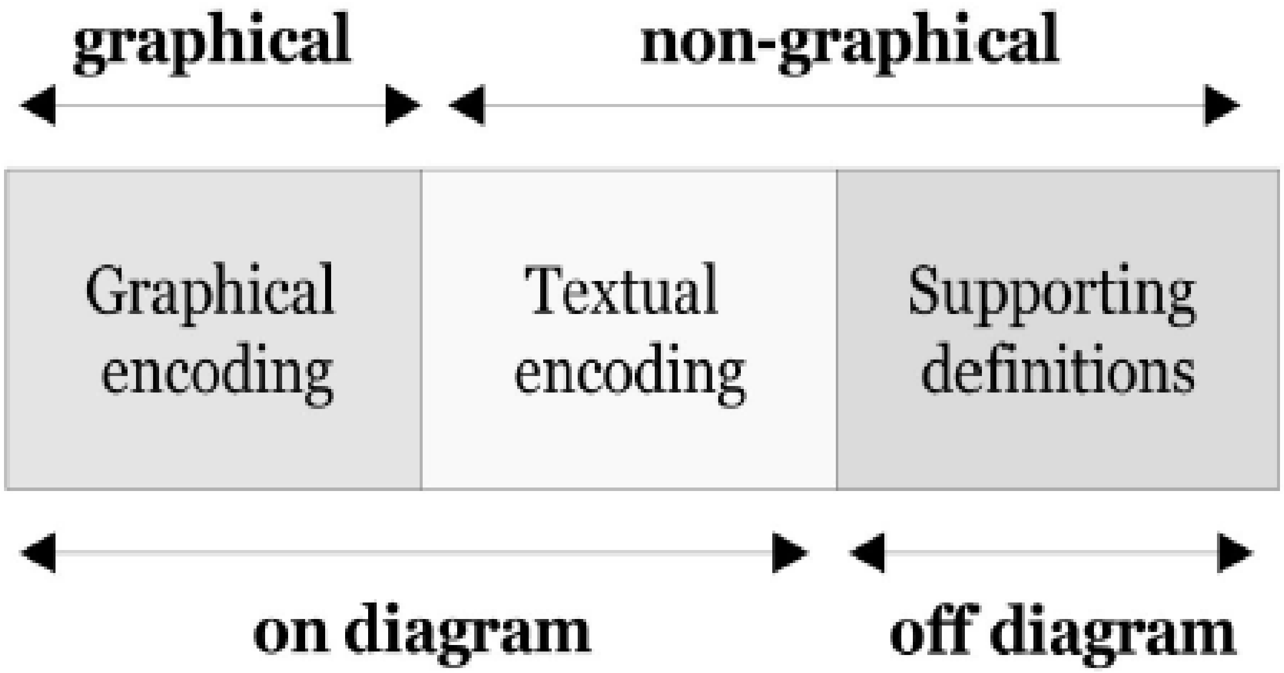
Supplement rather than duplicate (e.g., multiplicity values)
Reinforce meaning



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

| | 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 | | | | | | | | | |