Next Up: Creativity and Self-Awareness

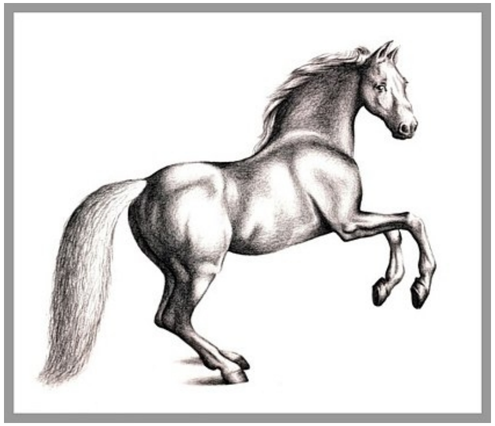
- The symbol grounding problem
- Copycat
  - a program that makes letter-string analogies
- Metacat
  - a more “self aware” version of Copycat
- A self-modeling robot
  - learns about its own body
  - can recover from damage
- EMI
  - a program that composes music

The Symbol Grounding Problem

- In most formal systems, symbols are **passive tokens**
- Meaning is **extrinsic** to the symbols, not **intrinsic**
- Harnad's paper brings together many ideas we've discussed this semester:
  - formal systems
  - physical symbol systems
  - the Chinese Room
  - connectionist networks
  - embodiment
  - Fodor & Pylyshyn's arguments against connectionism

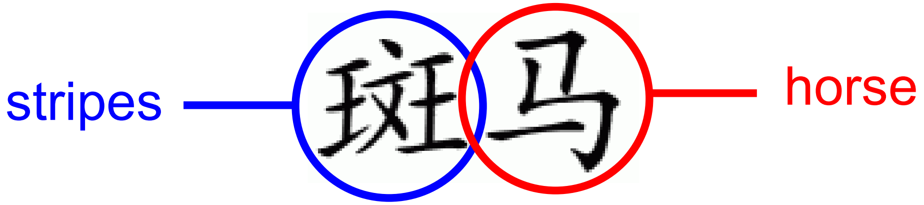
Representations According to Harnad

- **Iconic** and **categorical** representations are based on low-level sensory perceptions
- Example: “horse” concept
  - iconic: retinal image of a particular horse
  - categorical: prototypical image of a generic horse



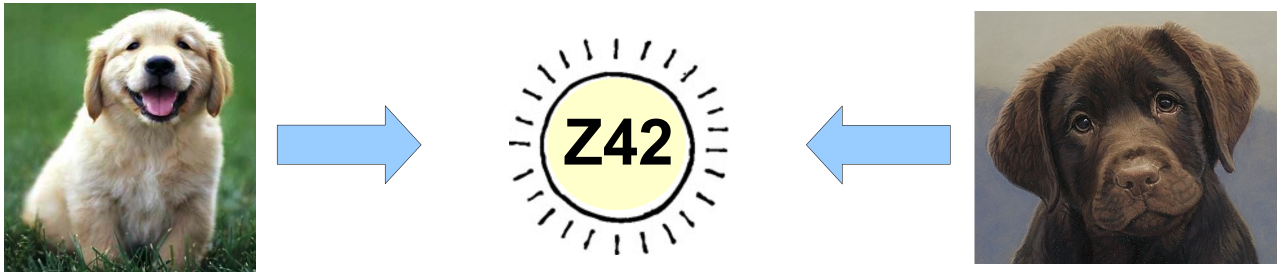
Representations According to Harnad

- **Symbolic** representations are composed of iconic or categorical representations
  - systematically structured
  - NOT based directly on sensory perceptions
- Example: “zebra” concept
  - “zebra” = “horse” & “stripes”
  - “zebra” is grounded by “horse” and “stripes”
  - similar to written form of “zebra” in Chinese:



The Symbol Grounding Problem

- **Intrinsically meaningful** symbols must be more than just arbitrary syntactic tokens
- Symbols must be composed of iconic or categorical representations **grounded in perception**
- Symbols grounded in perception would be associated with particular sensory contexts
- Systematic activation of a symbol by a particular type of sensory context would tie its meaning to that context



Copycat

- An analogy-making program developed by Douglas Hofstadter and Melanie Mitchell
- Operates within an abstract microworld of letter strings
- Designed to be a computer model of
  - analogy-making
  - high-level perception
  - “fluid” concepts
  - creativity
- Takes the symbol grounding problem seriously

Copycat's Microworld

abc ⇒ **abd**

ijk ⇒ **?**

Copycat's Microworld

abc ⇒ **abd**

srqp ⇒ **?**

Copycat's Microworld

aabc ⇒ **aabd**

ijkk ⇒ **?**

Copycat's Microworld

abc ⇒ **abd**

mrrjjj ⇒ **?**

Copycat's Microworld

abc ⇒ abcd

qqq ⇒ ?

Copycat's Microworld

abced ⇒ abcde

ppqqrrrss ⇒ ?

Copycat's Microworld

acde ⇒ **abcde**

nnxn ⇒ **?**

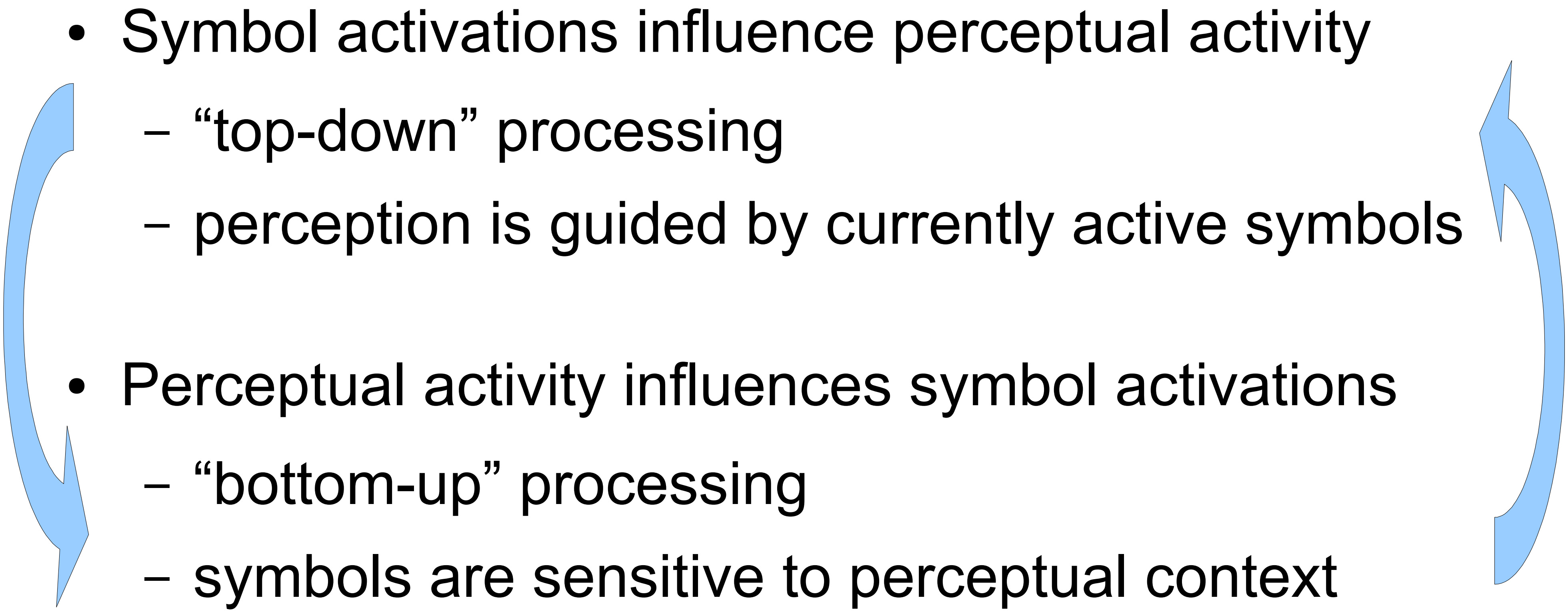
Main Program Components

- **Workspace**
  - locus of perceptual processing
  - all processing carried out by “codelets”
  - all codelet decisions are made probabilistically
  - computational temperature
- **Slipnet**
  - a network of nodes representing concepts that the program understands (*letter*, *group*, *successor*, etc.)
  - spreading activation
- **Coderack**
  - current pool of available codelets waiting to run

Copycat's Symbols

- **Slipnet concepts** serve as the program's symbols

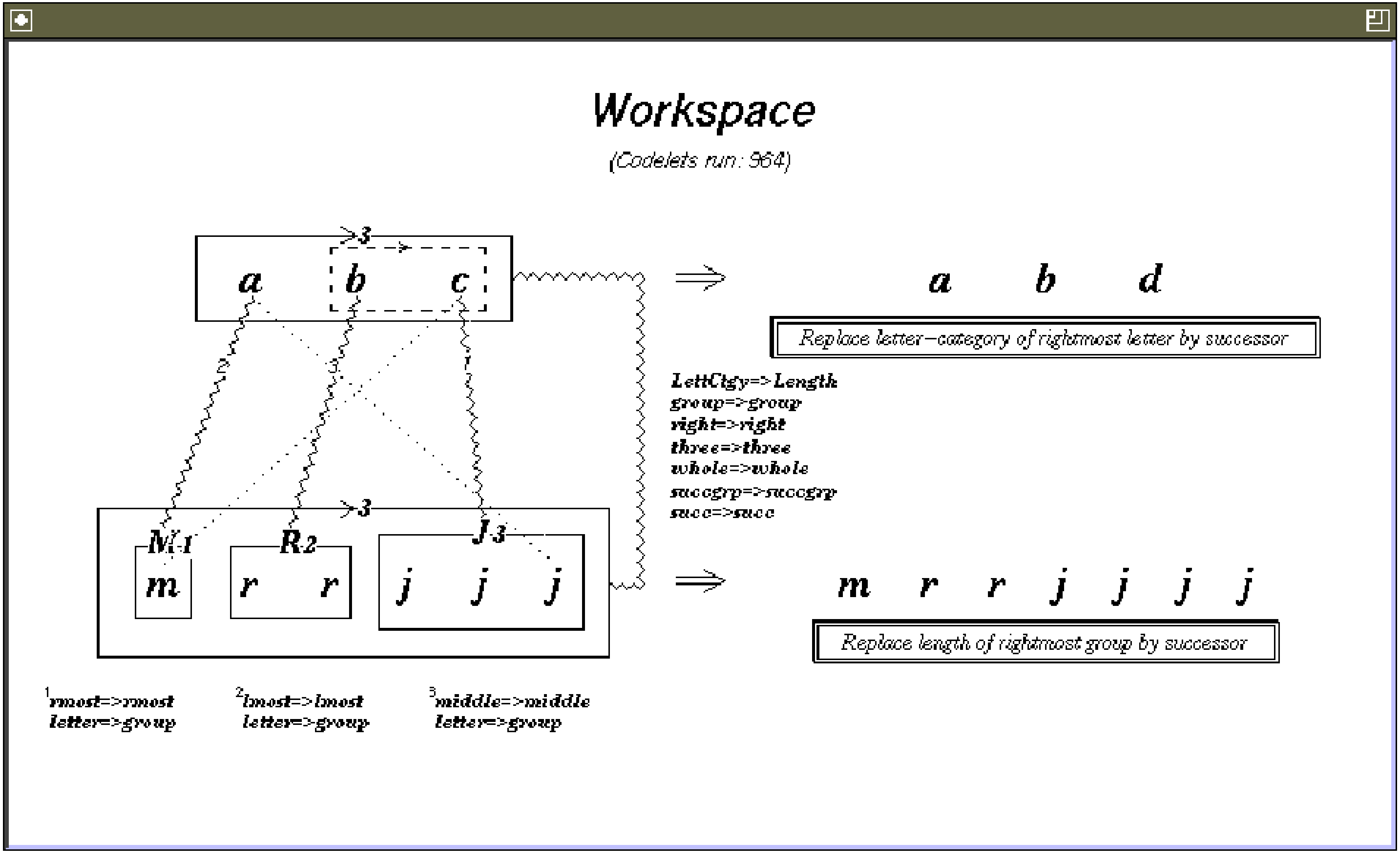
- Symbols are **active**, not passive



- Symbol activations influence perceptual activity
  - “top-down” processing
  - perception is guided by currently active symbols

- Perceptual activity influences symbol activations
  - “bottom-up” processing
  - symbols are sensitive to perceptual context

The Workspace



Workspace

(Codelets run: 964)

Replace letter–category of rightmost letter by successor

LettCtgy=>Length
group=>group
right=>right
three=>three
whole=>whole
succgrp=>succgrp
succ=>succ

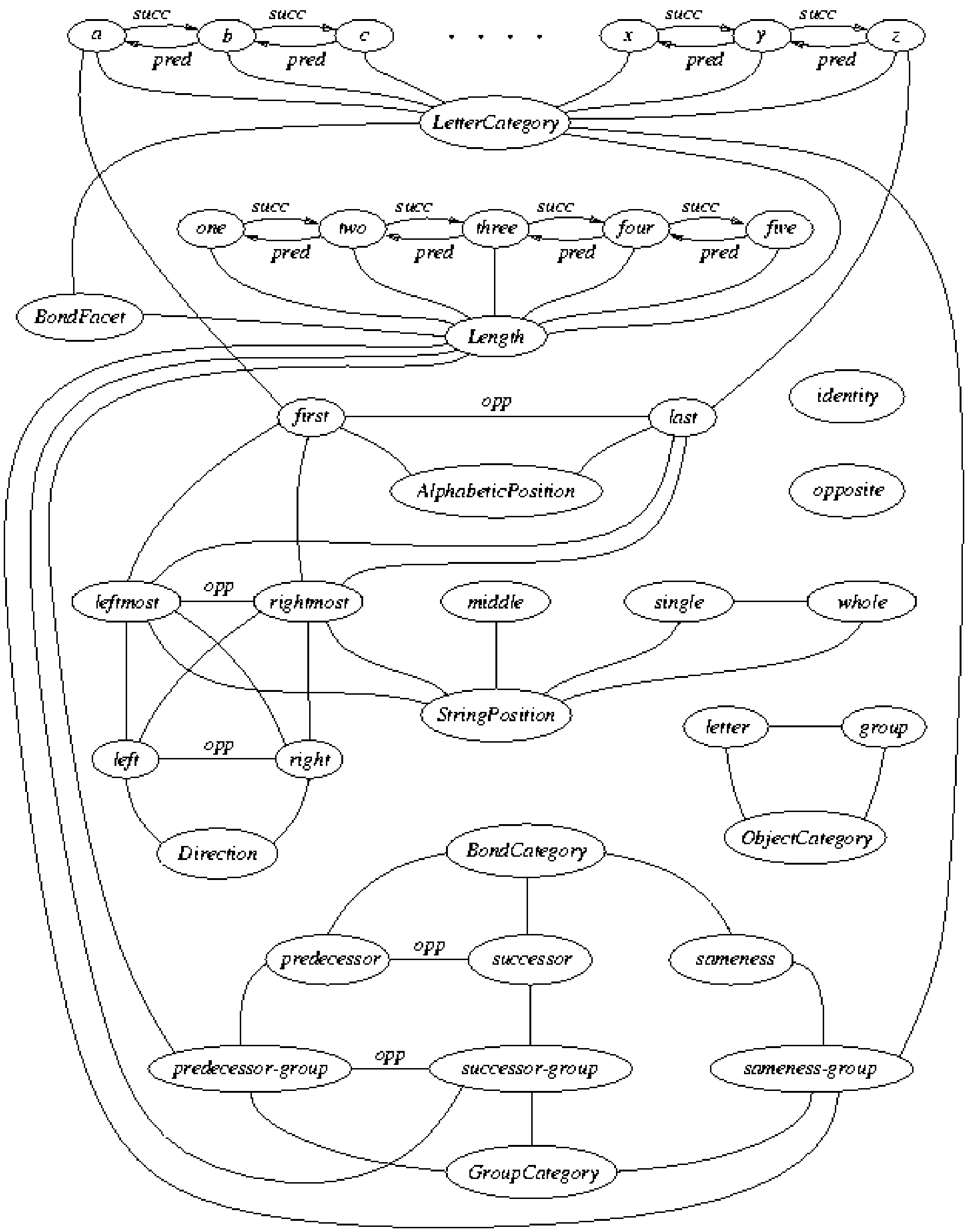
Replace length of rightmost group by successor

[1]rmost=>rmost
letter=>group

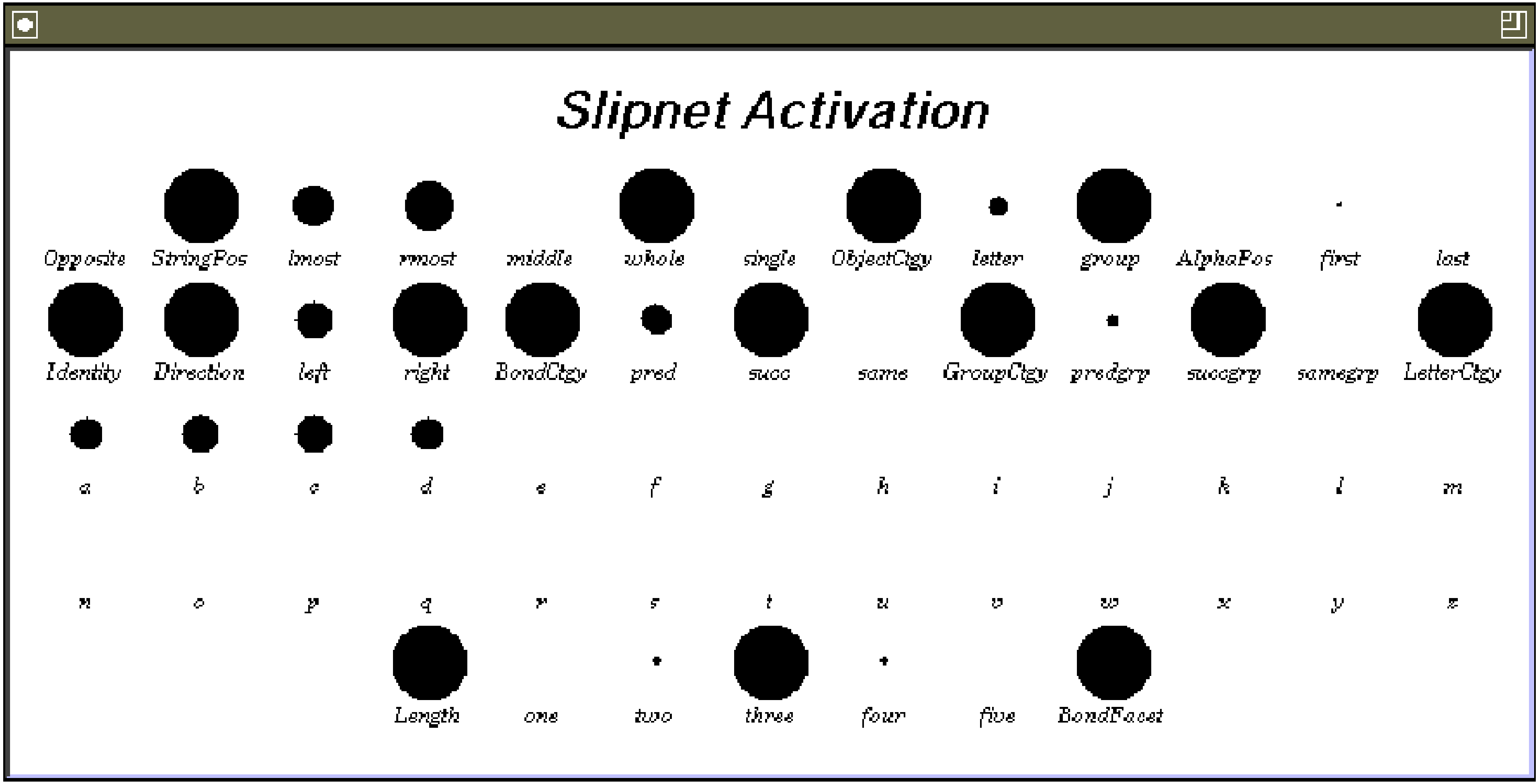
[2]lmost=>lmost
letter=>group

[3]middle=>middle
letter=>group

The Slipnet


succ
succ
succ
succ
a
b
c
. . . .
x
y
z
pred
pred
pred
pred
LetterCategory
succ
succ
succ
succ
one
two
three
four
five
pred
pred
pred
pred
BondFacet
Length
first
opp
last
identity
AlphabeticPosition
opposite
leftmost
opp
rightmost
middle
single
whole
StringPosition
letter
group
left
opp
right
ObjectCategory
Direction
BondCategory
predecessor
opp
successor
sameness
predecessor-group
opp
successor-group
sameness-group
GroupCategory

The Slipnet


Slipnet Activation
Opposite
StringPos
lmost
rmost
middle
whole
single
ObjectCtgy
letter
group
AlphaPos
first
last
Identity
Direction
left
right
BondCtgy
pred
succ
same
GroupCtgy
predgrp
succgrp
samegrp
LetterCtgy
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
Length
one
two
three
four
five
BondFacet

Context-Sensitive Symbols

- Copycat's *successor* symbol can be activated by many different strings under the right circumstances

abc

tsrqp

ijk

iijjkk

mrrjjj

wwxyzz

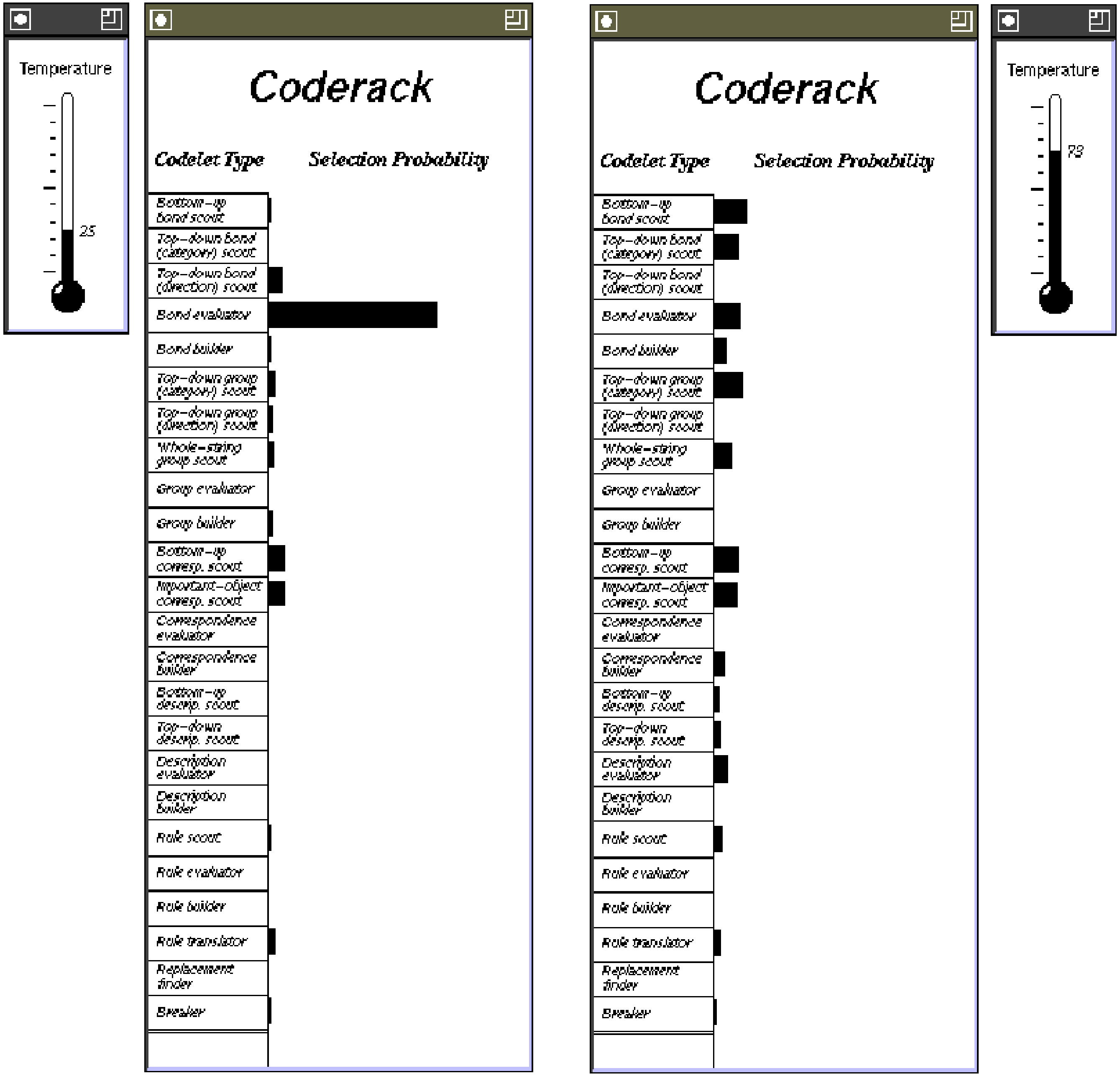
pqabcijkl

aababc

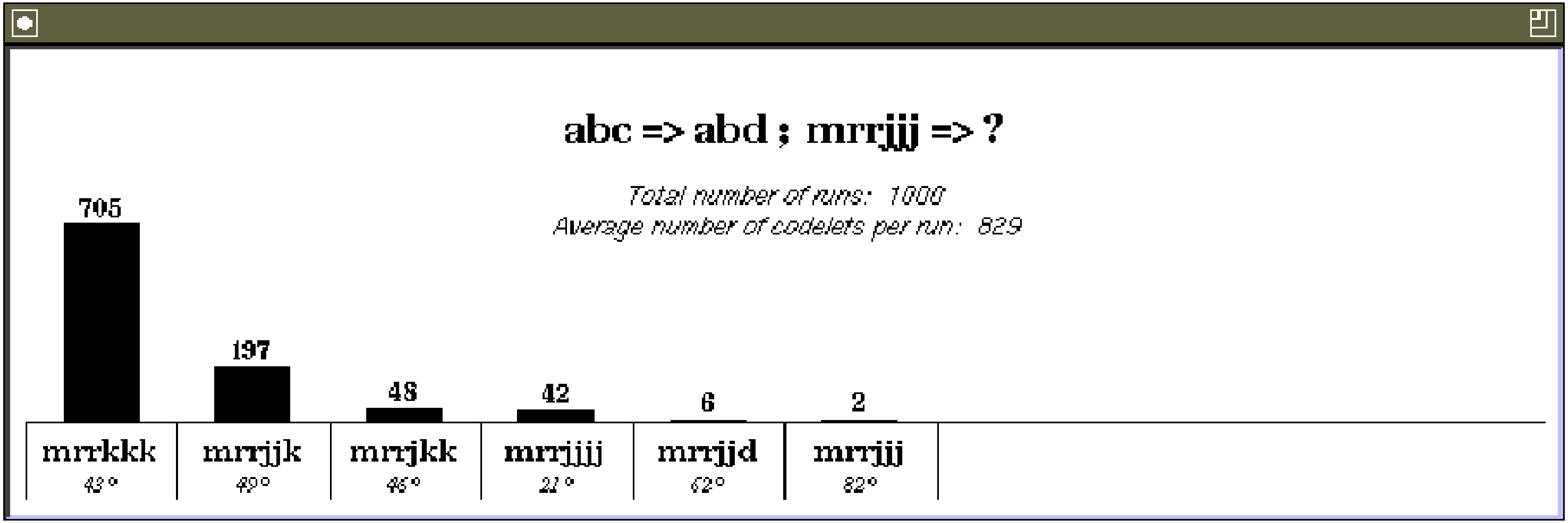
bbbbyyyqq

cba

Temperature and the Coderack



Nondeterministic Behavior



abc => abd ; mrrjjj => ?

Total number of runs: 1000
Average number of codelets per run: 829

Limitations of Copycat

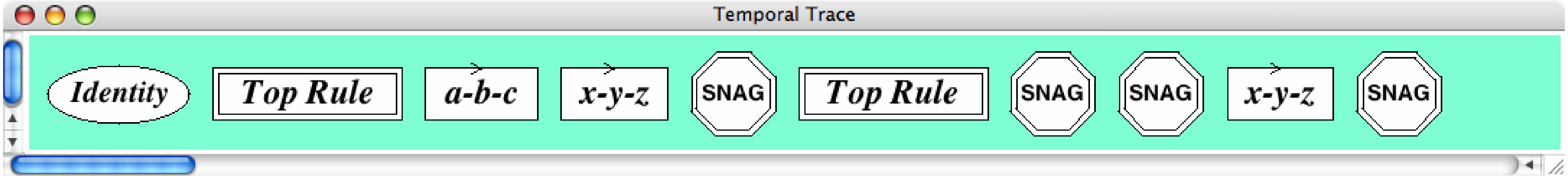
- No explicit “awareness” of what it is doing
- May get stuck in a rut when solving a problem
- Cannot remember more than one answer at a time
- Cannot compare different answers
- Cannot evaluate answers suggested to it
- Many more...

Metacat

- Includes mechanisms for **self-watching**
- Builds an explicit temporal trace of its actions while solving a problem
- Can notice when it has fallen into a repetitive pattern of behavior by examining its “train of thought”, and can respond accordingly
- Capable of a high degree of self-control
- Can compare answers based on the temporal information gleaned from self-watching

The Temporal Trace

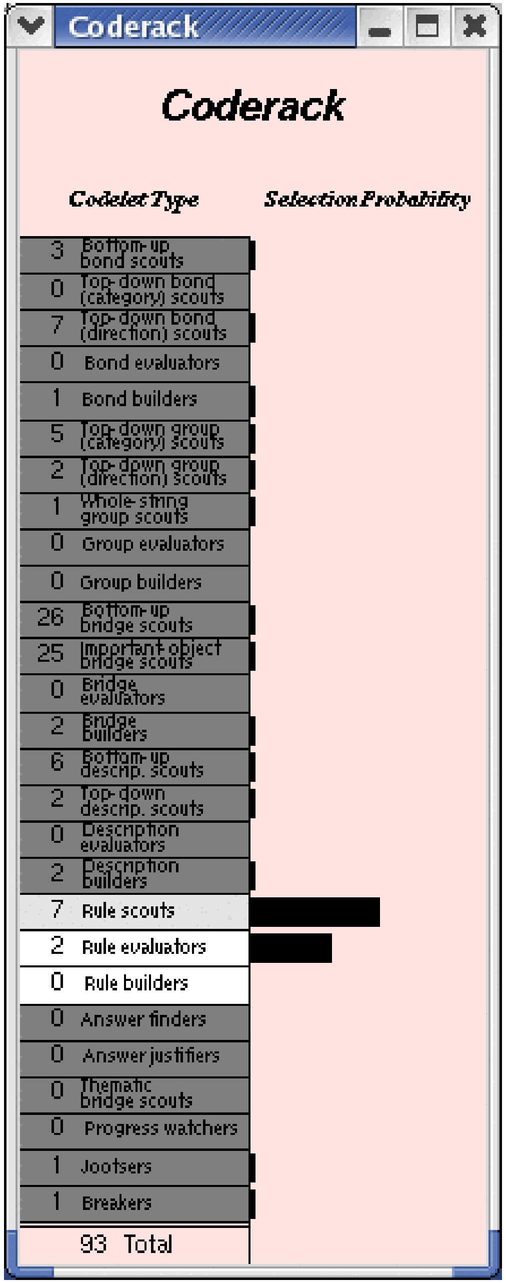
- Gives a high-level picture of the most important events that happen during a run
- Example: 10 *events* versus 1,320 *codelets*
- Allows Metacat to explicitly represent its own behavior
- Codelets can examine the Trace for patterns of events



Clamping Codelets and Concepts

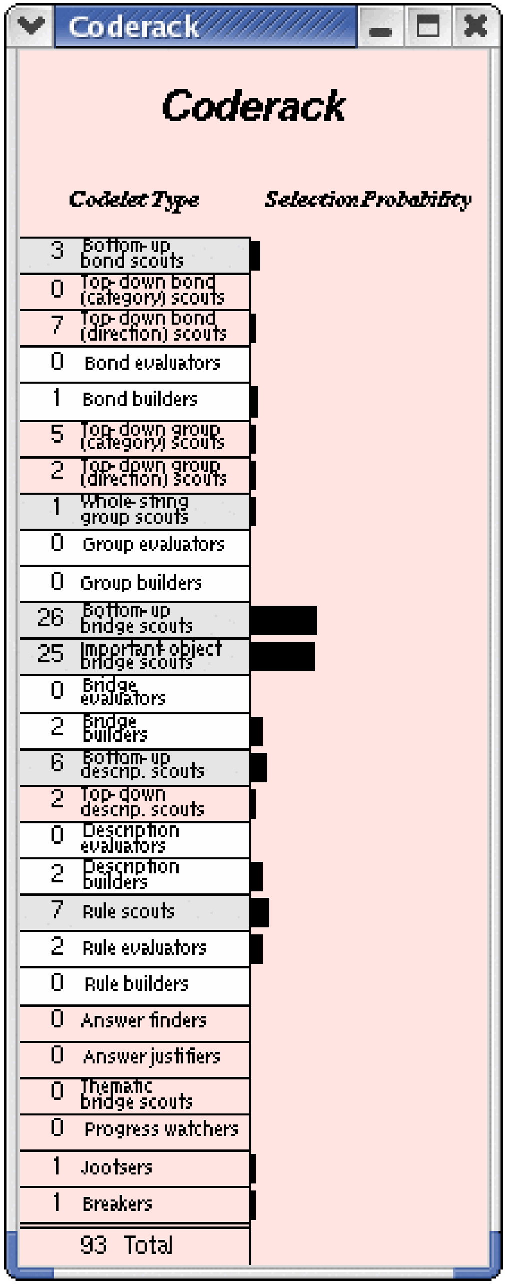
- Codelet urgencies can be **clamped** in order to alter the selection probabilities of different codelet types
- Slipnet concepts can be clamped in a similar fashion
- Metacat itself decides which concepts or codelets to clamp by examining the information in the Trace
- More attention is paid to particular concepts or types of perceptual structures
- This can help the program to discover alternative interpretations of a problem
- Gives the program a high degree of self-control

Clamping Codelets and Concepts



Coderack

Codelet Type	Selection Probability
3 Bottom-up bond scouts	
0 Top-down bond (category) scouts	
7 Top-down bond (direction) scouts	
0 Bond evaluators	
1 Bond builders	
5 Top-down group (category) scouts	
2 Top-down group (direction) scouts	
1 Whole-string group scouts	
0 Group evaluators	
0 Group builders	
26 Bottom-up bridge scouts	
25 Important object bridge scouts	
0 Bridge evaluators	
2 Bridge builders	
6 Bottom-up descrip. scouts	
2 Top-down descrip. scouts	
0 Description evaluators	
2 Description builders	
7 Rule scouts	
2 Rule evaluators	
0 Rule builders	
0 Answer finders	
0 Answer justifiers	
0 Thematic bridge scouts	
0 Progress watchers	
1 Jootsers	
1 Breakers	
93 Total	



Coderack

Codelet Type	Selection Probability
3 Bottom-up bond scouts	
0 Top-down bond (category) scouts	
7 Top-down bond (direction) scouts	
0 Bond evaluators	
1 Bond builders	
5 Top-down group (category) scouts	
2 Top-down group (direction) scouts	
1 Whole-string group scouts	
0 Group evaluators	
0 Group builders	
26 Bottom-up bridge scouts	
25 Important object bridge scouts	
0 Bridge evaluators	
2 Bridge builders	
6 Bottom-up descrip. scouts	
2 Top-down descrip. scouts	
0 Description evaluators	
2 Description builders	
7 Rule scouts	
2 Rule evaluators	
0 Rule builders	
0 Answer finders	
0 Answer justifiers	
0 Thematic bridge scouts	
0 Progress watchers	
1 Jootsers	
1 Breakers	
93 Total	

Metacat Demo