Adaptive Communication and Compositional Modeling

Part 1: Adaptive Communication

Michael Matessa

Mike.Matessa@nasa.gov

NASA Ames Research Center

What Processes Lead to Form of Communication?

- Language-based (static) “Hello”

  (Frazier, 1987; Rosenberg & Cohen, 1966)

- Collaboration-based (dynamic) “Whassup”

  (Clark, 1996; Krauss & Fussell, 1991)

Evidence for Collaboration

- Efficiency in referential communication
  - Partners tend to use fewer words over time
  - Influenced by presence & feedback of partner
- Adaptive nature of communication
  - Partners tend to match behavior of others

This work shows an effect of accommodation on efficiency with a new experiment & computational modeling

Overview

- Background
- Communication task
- Experiment testing effect of matching behavior on efficiency
- Computational modeling
- Conclusions

Efficiency in Communication

- Partners tend to use fewer words over time
  (Clark & Wilkes-Gibbs, 1986; Krauss & Fussell, 1991)
- Partner-related factors influence decrease
  - No decrease for imaginary partner
    (Hupet & Chantraine, 1992)
  - Less decrease with no feedback from partner
    (Krauss & Weinheimer, 1966)
  - Less decrease with non-matching word use?
    (Hupet, Chantraine, & Nef, 1993)

Accommodation

- Partners tend to match behavior of others (accommodation)
  - Word Choice (Garrod & Doherty, 1994)
  - Syntax Choice (Bock, 1986)
  - Non-verbal Behavior (Giles, Mulac, Bradac & Johnson, 1987)
- Motivated by social & efficiency concerns
  (Giles, Mulac, Bradac & Johnson, 1987)

Combining Accommodation & Efficiency

- Accommodation theoretically motivated by efficiency, but only subjective evidence
- Non-matching word use related to decrease in efficiency, but no causal evidence
- Test effect of accommodation on efficiency by manipulating matching or non-matching partner

Agents & Accommodation

- To test effect of accommodation, compare two partners: accom. & non-accom.
- Using simulated computational agents as partners, other factors can be held constant
- Agents created in the cognitive architecture ACT-R

ACT-R

- ACT-R is a computational cognitive architecture incorporating both procedural and declarative knowledge

	Procedural	Declarative
Symbolic	Rules	Facts
Subsymbolic	Rule reliability	Fact Activation

Why ACT-R?

- Framework for explaining cognitive data
  - learning arithmetic facts (Lebiere & Anderson, 1998)
  - mental states of tutored students (Anderson, Corbett, Koedinger, & Pelletier, 1995)
  - errors & RTs in memory (Anderson & Matessa, 1997)
- Computational, so agents can be created

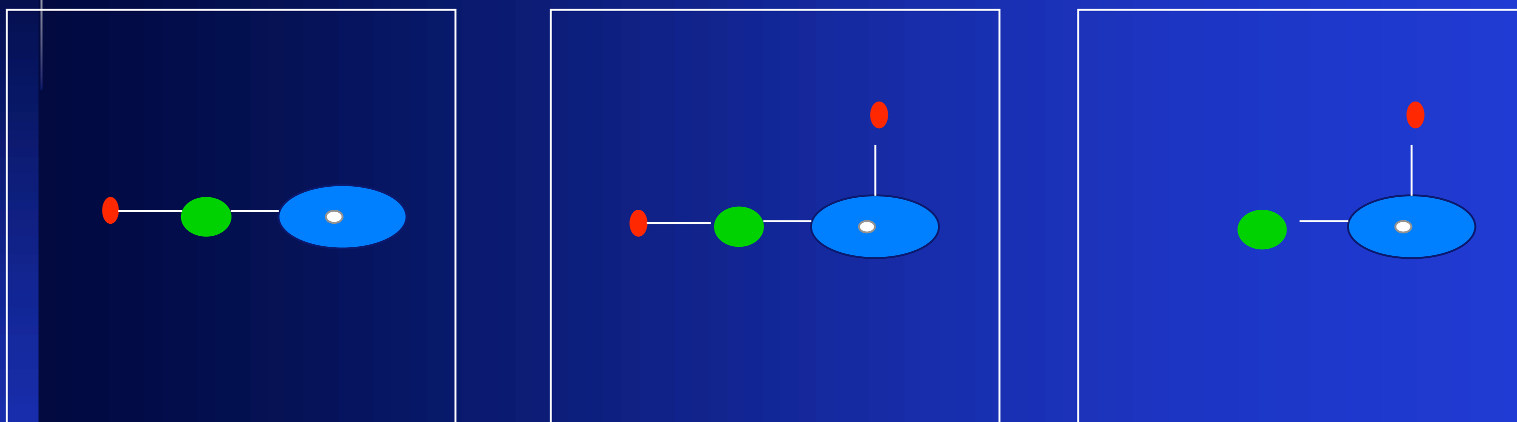
Communication in ACT-R

- Representation
  - Speech Acts (Core & Allen, 1997)
- Rules
  - Obligations (Traum & Allen, 1994)
- Fact Creation
  - Common Knowledge (Clark & Schaefer, 1989)
  - Common Language (Garrod & Anderson, 1987)

Summary

- ACT-R can be used to create agents
- Two agents, accomodating & non-accom., can test effect of accommodation on efficiency
- What communication task?

Communication Task



Initial (A) Final (both) Initial (B)

Object Dimensions

	Size	Shape	Color
Obj1	small	thin	red
Obj2	medium	round	green
Obj3	large	fat	blue

Communication Menu

Communication
Connections A) An object is above another object. B) What is above that object? C) Please put an object above that object. D) Let’s put an object above that object. Numbers E) I have four objects. F) How many objects do you have? G) Please make sure you have four objects. H) Let’s see how many objects we have. Assessment I) OK.

Message Template

A The No Our	topmost bottomost leftmost rightmost middle northern southern western eastern	small medim large	thin round fat	red green blue	dot blob object shape	is	above below left of right of north of south of west of east of	a the no our	topmost bottomost leftmost rightmost middle northern southern western eastern	small medim large	thin round fat	red green blue	dot blob object shape
A	topmost	small	thin	red	dot	is	above	our	middle	large	fat	blue	dot
A				red		is	above	our					dot

Accommodation

A	topmost	small	thin	red	dot	is	above	a	topmost	small	thin	red	dot
The	bottomost	medim	round	green	blob		below	the	bottomost	medim	round	green	blob
No	leftmost	large	fat	blue	object		left of	no	leftmost	large	fat	blue	object
Our	rightmost				shape		right of	our	rightmost				shape
	middle						north of		middle				
	northern						south of		northern				
	southern						west of		southern				
	western						east of		western				
	eastern								eastern				

Accommodation

Initial:													
A	topmost	small	thin	red	dot	is	above	our	middle	large	fat	blue	dot
Accom:													
A	leftmost	medium	round	green	dot	is	left of	our	middle	large	fat	blue	dot
Non-accom:													
A	western	medium	round	green	object	is	west of	our	middle	large	fat	blue	object

Non-Accommodation

Initial:													
	topmost		thin	red	dot	is	above	our	middle		fat	blue	dot
Non-accom:													
	western	medium	round		object	is	west of	our	middle	large	fat		object

Initial:													
	topmost			red	dot	is	above	our	middle			blue	dot
Non-accom:													
	western	medium			object	is	west of	our	middle	large			object

Agent Communication

- Task obligations
  - Get info, give info
  - Confirm patterns are same
  - Go to next problem
- Communication obligations
  - Answer questions
  - If message is unclear, tell partner

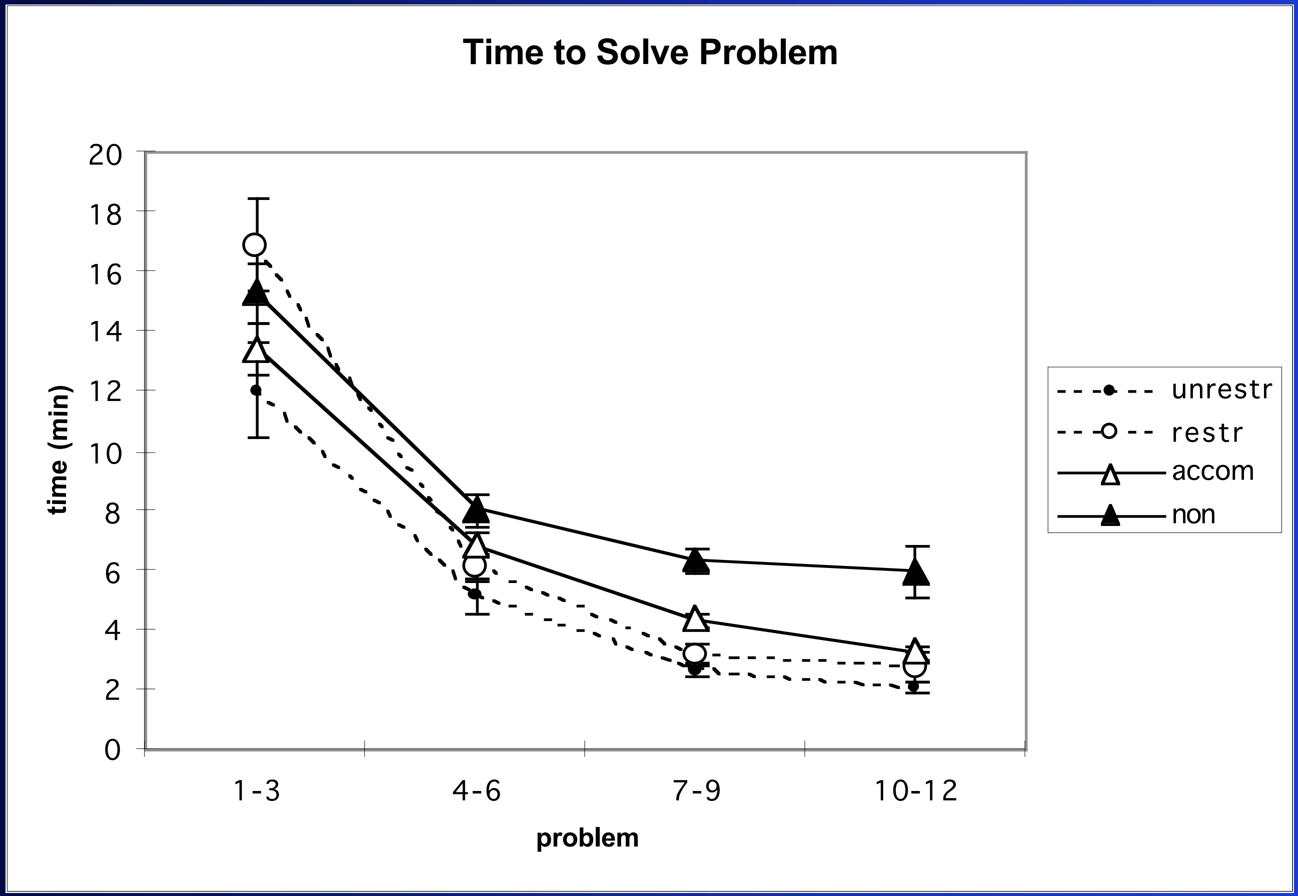
Agent Goal Structure

Goal	Productions Number	Productions Examples
solve	61	(do-wait-for-turn, ...)
decide	45	(decide-request-confirm, ...)
seeif	6	(seeif-done-puzzle-yes, ...)
count	8	(count-color, ...)
global-ground	12	(global-topmost, ...)
present	82	(type-next, fill-color, ...)
understand	90	(read-next, ...)
phrase	2	(parse-lex, ...)
obligation	6	(understand-statement, ...)
ground	17	(get-topmost, ...)
l2r	5	(lower-y-l2r, ...)
stall	4	(stall-see-done, ...)
	338	

Experiment

- Conditions
  - Humans with models (accom. & non-accom.)
  - Humans with humans (restr. & unrestr. interf.)
- Dependent Measures
  - Errors
  - Time
    - Message Length

Results



Time to Solve Problem
20
18
16
14
12
10
8
6
4
2
0
time (min)
1-3
4-6
7-9
10-12
problem
unrestr
restr
accom
non

“Human” Agent

- Accom. & non-accom. agents both match message length so it can be used as a measure of efficiency
- To model a subject changing message length, a third agent is needed to explain word-skipping

Skipping Words for Efficiency

- IF word is not part of core meaning THEN do not type word
- Rule unlikely to be used because not standard English
- Once word is removed, common language use prevents it from returning

Skipping with Evidence

- IF word is not part of core meaning AND cooperative memory exists THEN do not type word
- Rule more likely to be used because cooperative partner should accept change
- Cooperative memory comes from accommodation goal

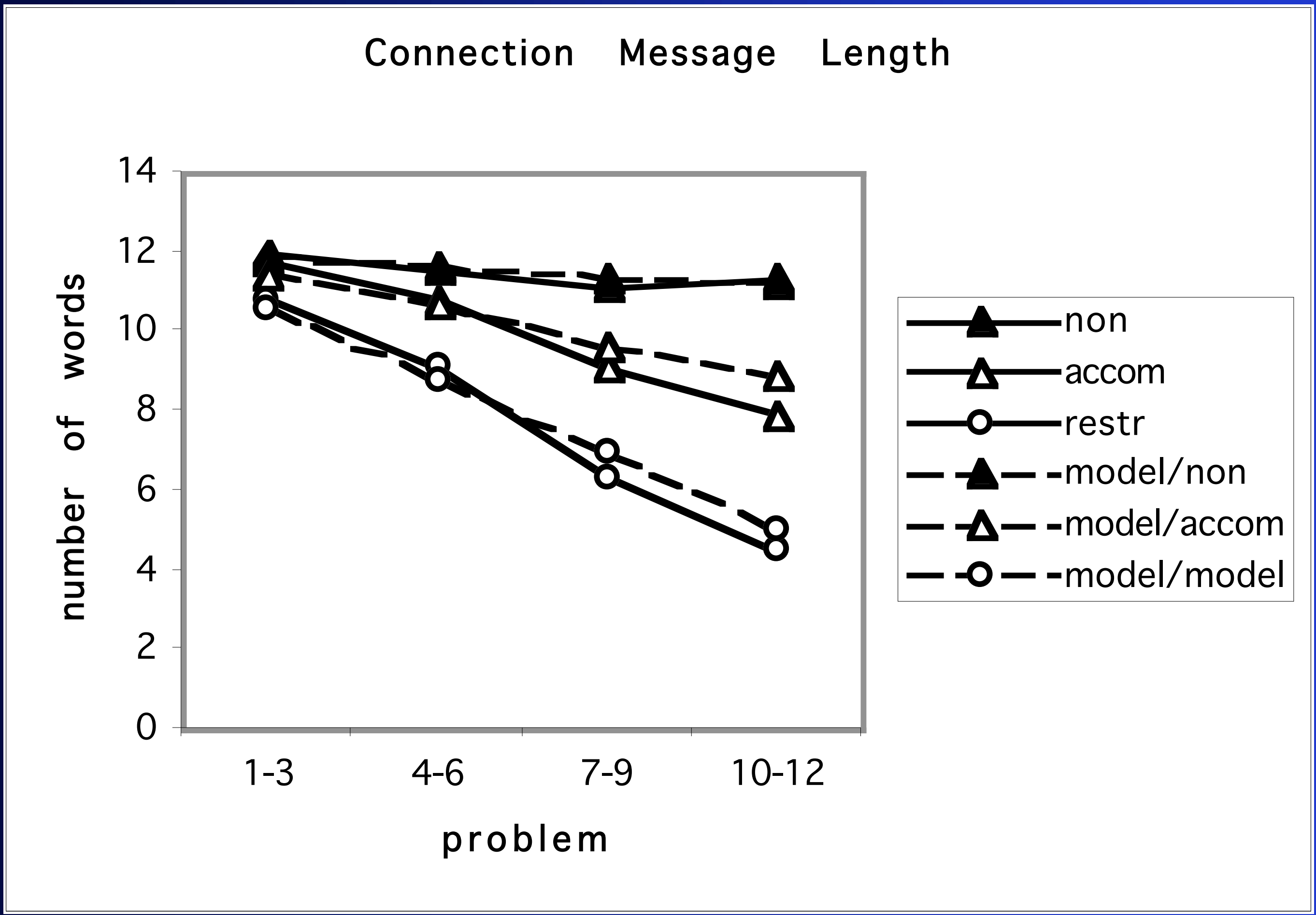
Skipping Rules

Rule	prob	Description
skip-word-match-eff	.52	skips a nonessential word if partner matches word
skip-word-nomatch-eff	.48	skips a nonessential word

Agent Properties

	Partner	Word Choice	Message Length
Accom. Agent	human	match	match
Non-Accom. Agent	human	different	match
“Human” Agent	agents	match	match and decrease

"Human" Agent Results



Results Summary

- Effect of accom. on efficiency: Subjects use shorter descriptive messages and skip coordination messages with accom. agent
- “Human” agent more efficient with accom. agent because memories of cooperation influence word & message skipping

Research Contributions

- Referential communication task
- Restricted interface
- Empirical evidence of accom. effect
  - Subjects more efficient with matching partner
- Computational theory of collaborative communication

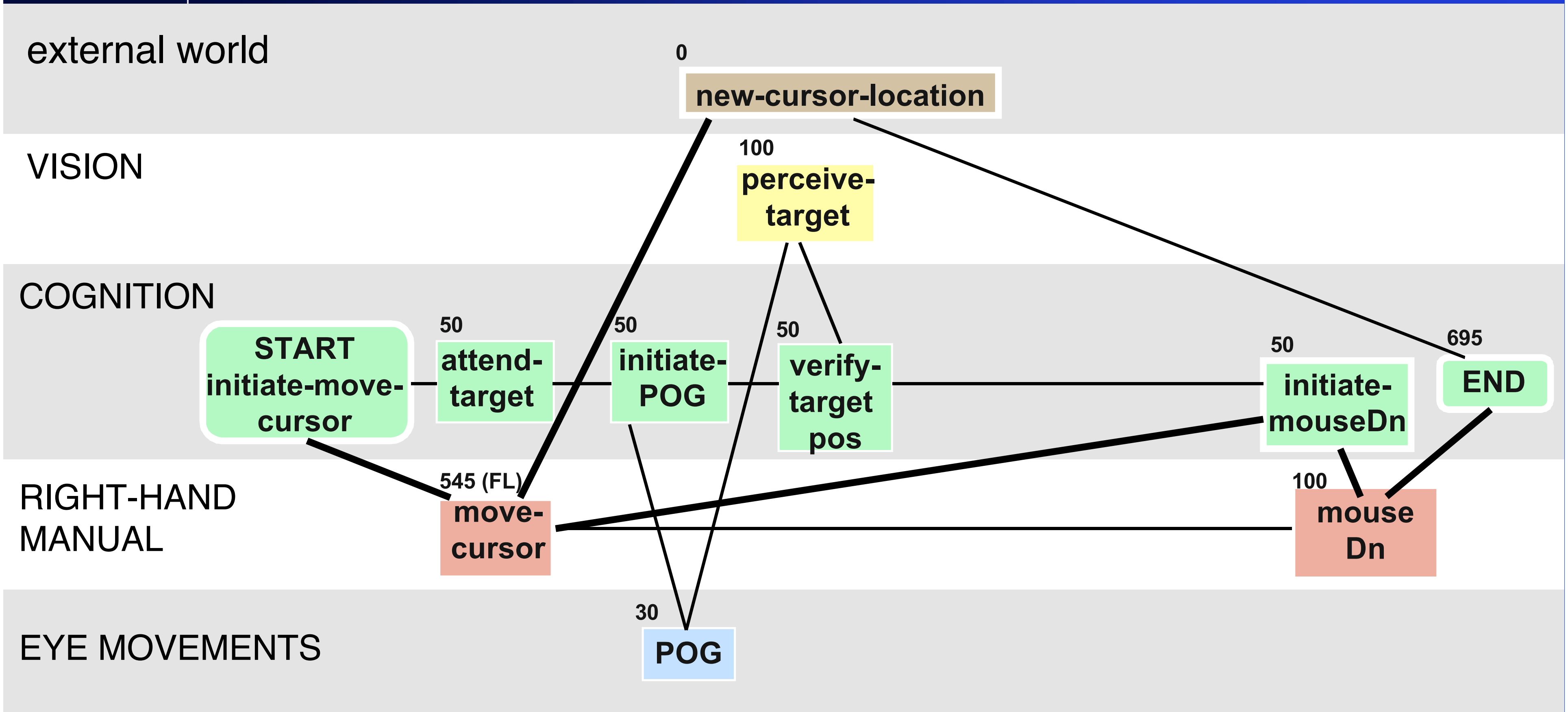
Part 2:
Compositional Modeling

Bonnie John	CMU
Richard Lewis	UMich
Michael Matessa	NASA
Roger Remington	NASA
Alonso Vera	NASA

Compositional Modeling

Cognitive modeling can scale affordably, and be routinely applied to complex tasks, only if it becomes an exercise of composing new behaviors from existing reusable behavior components.

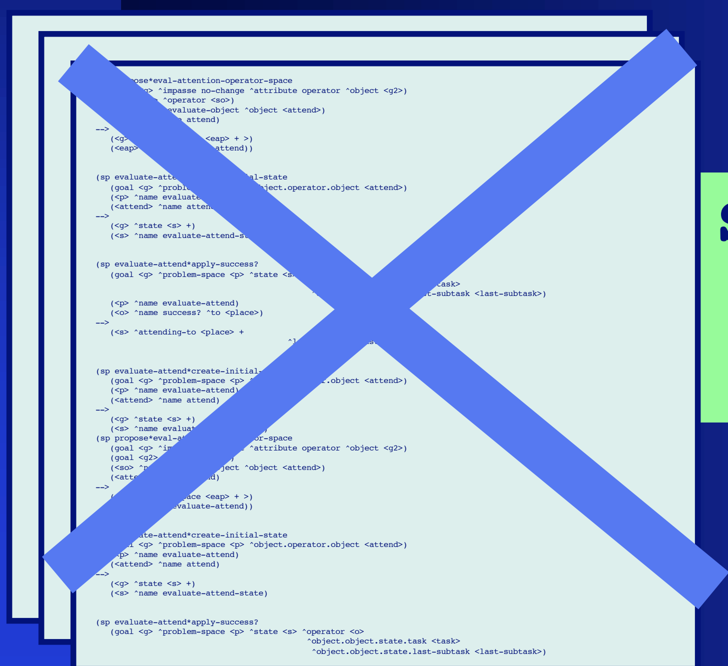
Even simple mouse actions have complex details



From *Gray and Boehm-Davis (2000)*

We want *task-level reuse* to abstract away from these details

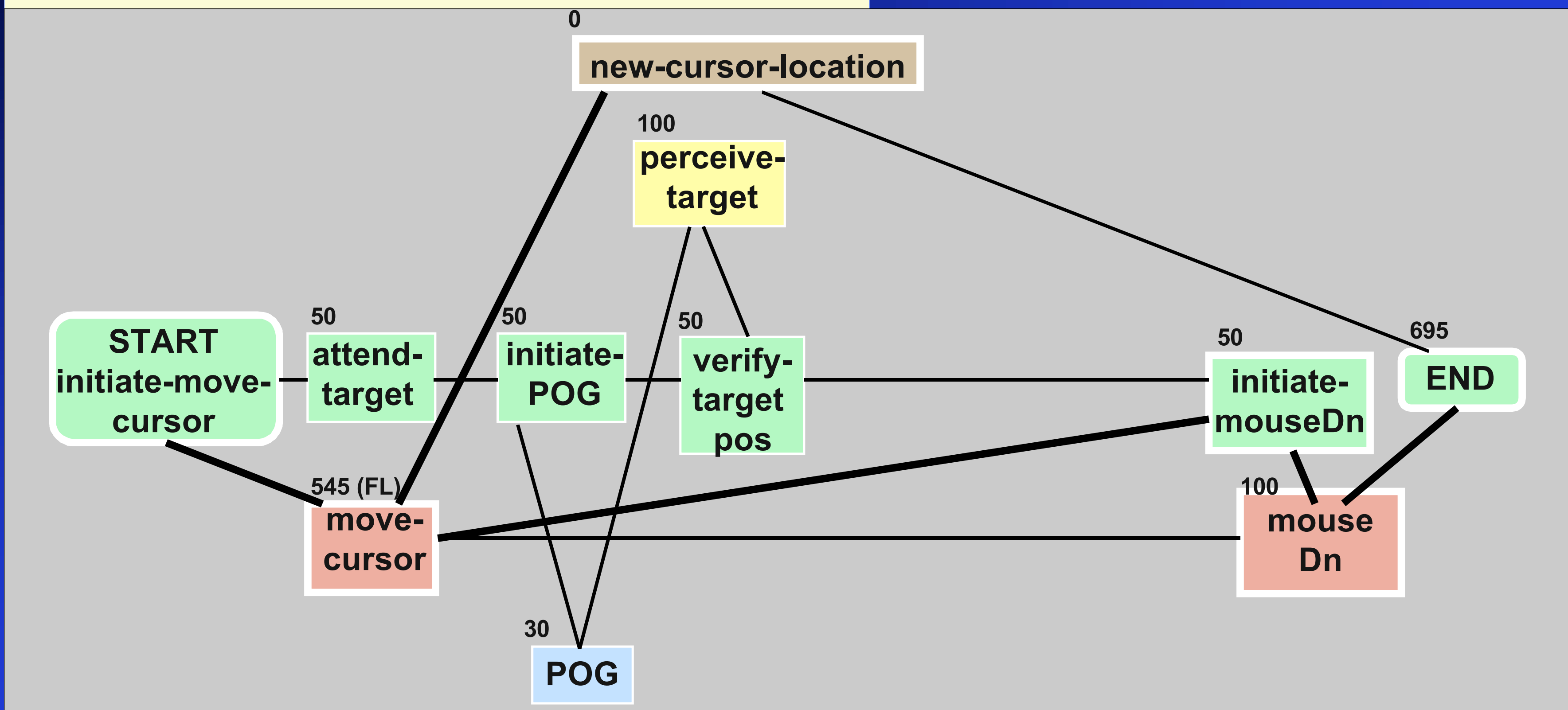
- We want modelers to work as close as possible to the *task level*, not at the level of the architecture
- With building blocks that can be reused to piece together more complex behaviors in a psychologically valid way



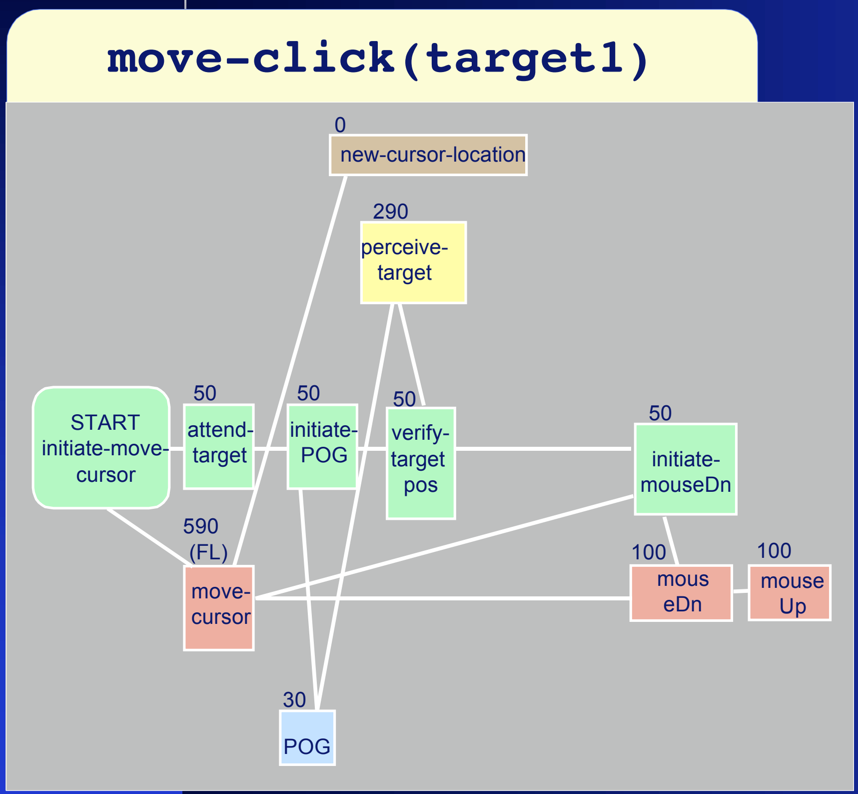
```
Select(target)
   move-click(target)
   move-click("OK")
```

Templates: Reusable behavioral building blocks

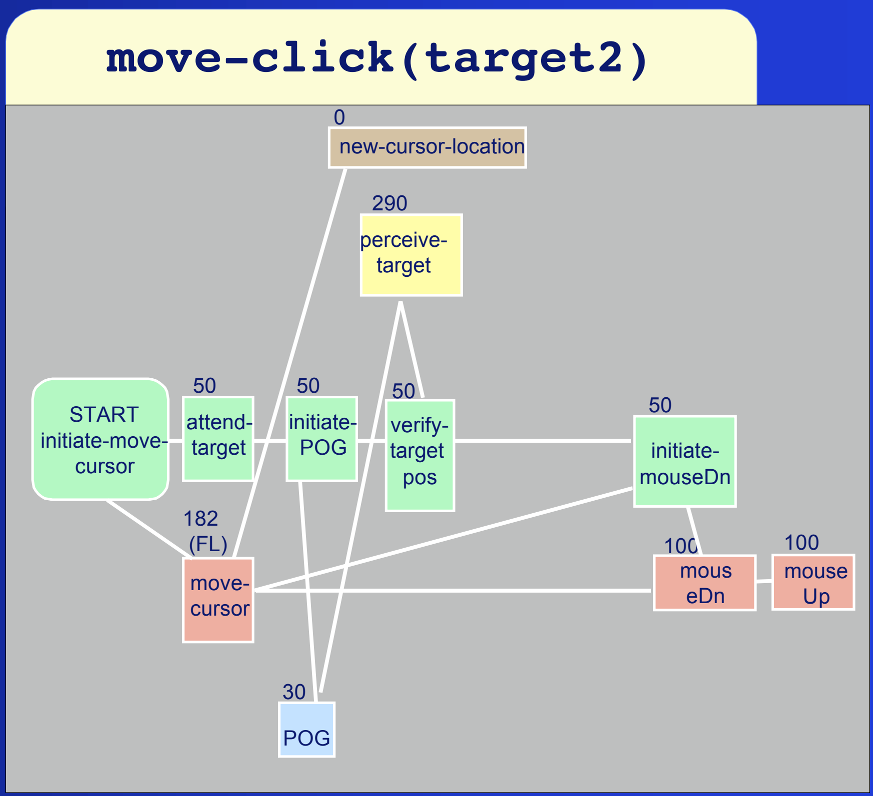
move-click(target)



Can we just stick blocks of behavior together?

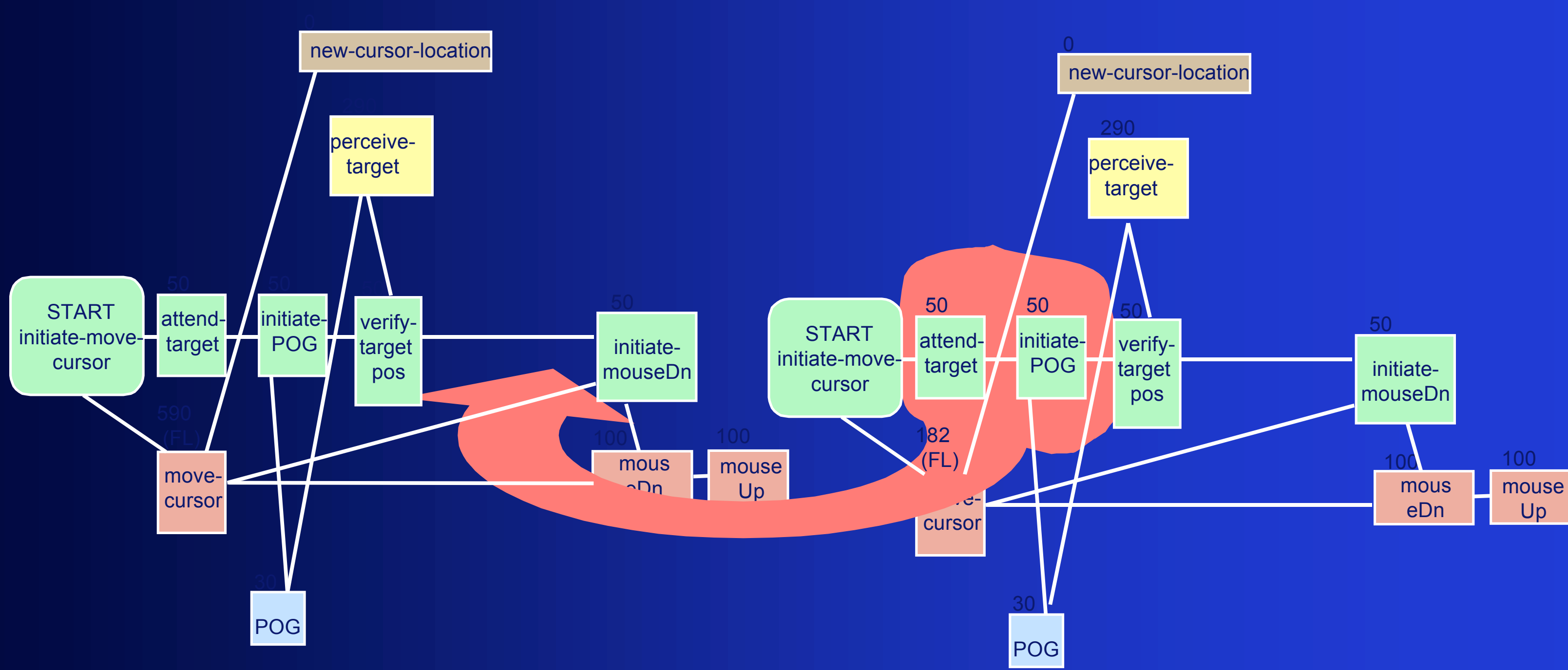


+



NO. The details are important when we compose behavior

- Unfortunately, **we can't abstract away these details when we piece together new stretches of behavior**
- Consider what can happen with two consecutive mouse clicks:



Ongoing Projects

- Models from demonstrated behavior
- Simple language for creating models
- Framework using model constraints
- (Towards a) theory of composition

Modeling by Demonstration

- Demonstrate a task on an html storyboard
- Produces a trace of the motoric operators (both observed and implied) along with implied mental operators
- Compiles into model
- Model interacts with storyboard to give time predictions

Retro modeling language

Simple language → **ACT-R chunks** → **ACT-R productions**

Simple language:

```
Do Task
1. Do A
2. Do B
2. Do C
3. Do D

Do A
1. Move mouse
2. Click
```

ACT-R chunks:

```
(add-dm
  (t24 isa task)
  )
```

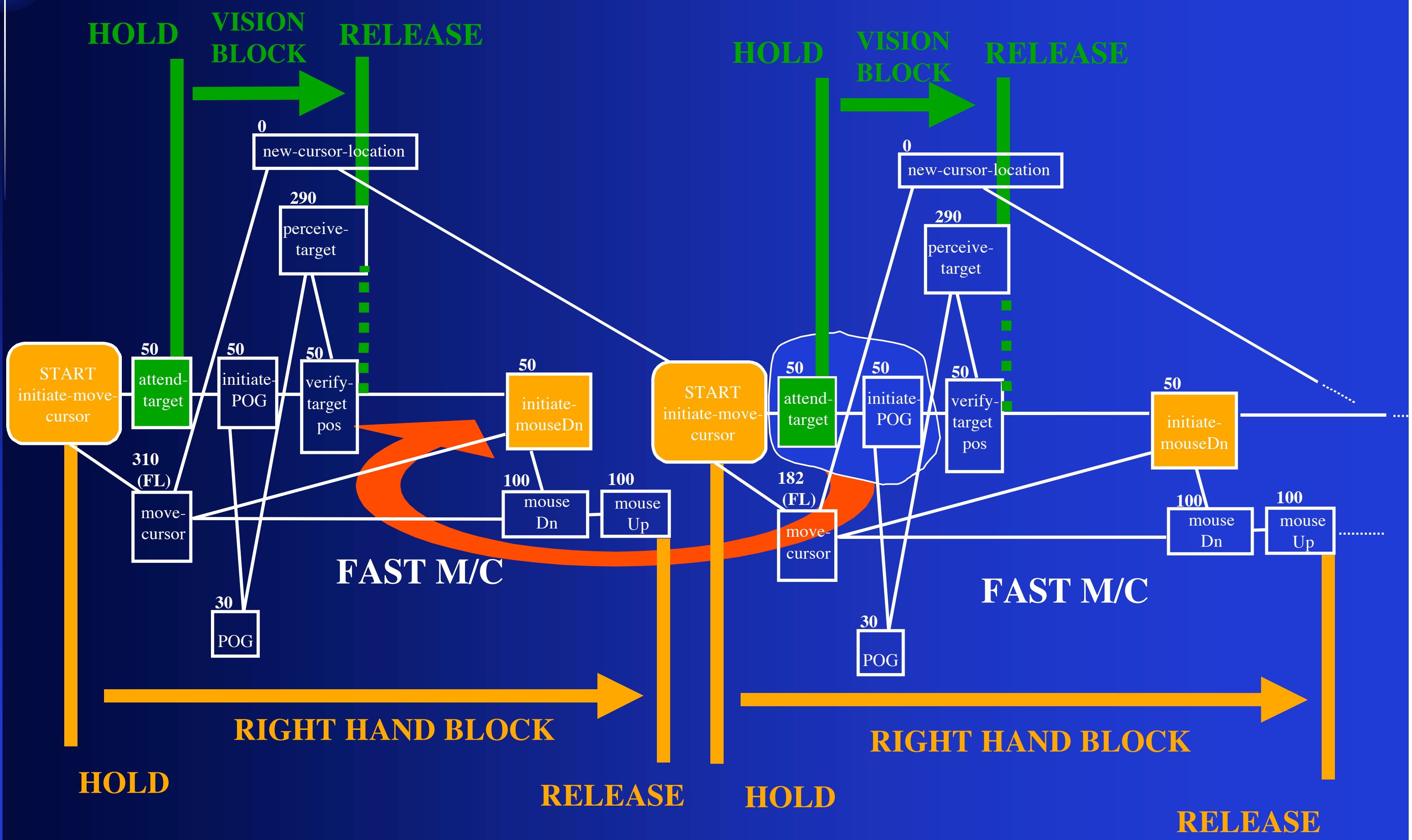
ACT-R productions:

```
(p t53
  =goal>
    isa do-task
==>
)
```

CPM-X Constraint Framework

- Model expressed as task, strategy, and architecture constraints
- These constraints describe temporal relationships on cognitive, perceptual, and motor processes
- This provides a temporal sequence of human behavior

Composition Theory



Composition Results