Simulating Adaptive Communication

Michael Matessa
Fall, 2000

Department of Psychology
Carnegie Mellon University
Pittsburgh, PA

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Thesis Committee:

John R. Anderson, Chair
Brian MacWhinney
Jill Fain Lehman

This research was sponsored by the Office of Naval Research under contract number N00014-95-10223 to John Anderson at Carnegie Mellon University. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the ONR or the U.S. government.

Abstract

In a collaborative view of communication, the meaning of a message is determined not solely by words and syntax, but rather is negotiated by conversational partners using words and syntax. This collaborative nature has been demonstrated by research on the increased efficiency (Hupet & Chantraine, 1992) and the adaptive behavior (Giles, Mulac, Bradac, & Johnson, 1987) of communicating pairs, but these two lines of research have never been explicitly related. This dissertation combines and extends these lines of research with empirical results showing that adaptively matching word use can increase communication efficiency and also gives an ACT-R (Anderson & Lebiere, 1998) modeling account of the processes involved.

Keywords: ACT-R, modeling, communication, accommodation, efficiency

Dedicated to my family.

Table of Contents

Acknowledgments

I would like to thank John Anderson for his many years of guidance and support. He is without a doubt the best advisor anyone could ask for. I would also like to thank my committee members: Brian MacWhinney for his many ideas (such as the experimental manipulation which forms the backbone of this thesis), and Jill Lehman for understanding the whole point of what I was trying to do.

The psychology graduate students were essential in getting me through graduate school. I have made many friends I will value for the rest of my life.

"I don't know what you mean by 'glory,'" Alice said.

Humpty Dumpty smiled contemptuously. "Of course you don't -- till I tell you. I meant 'there's a nice knock-down argument for you!'"

"But 'glory' doesn't mean 'a nice knock-down argument,'" Alice objected.

"When I use a word," Humpty Dumpty said, in rather a scornful tone, "it means just what I choose it to mean -- neither more nor less."

"The question is," said Alice, "whether you can *make words mean so many different things."*

"The question is," said Humpty Dumpty, "which is to be master -- that's all."

Lewis Carroll

Chapter 1: Introduction

What processes are involved when people decide how to phrase a message? One group of research has focused on the effects of the language being used (Frazier, 1987; Rosenberg & Cohen, 1966), while another has focused on the effects of the collaborative nature of communication (Clark, 1996; Krauss & Fussell, 1991). From the first point of view, meaning is determined by words and syntax, while from the second point of view, words and syntax are used by conversational partners to negotiate what meaning is.

The research presented in this dissertation will extend and combine two lines of research that support the collaborative view of communication. The first finding deals with efficiency: two partners tend to use fewer words over time to establish mutual reference, while people who are asked to create referential phrases for an imagined partner do not decrease their word use over time (Hupet & Chantraine, 1992). The second finding deals with adaptive behavior: people show an ability to accommodate to their partner's communication style by tending to converge to that style in many situations (Giles, Mulac, Bradac, & Johnson, 1987). These findings have both been theorized to be a result of motivations to increase communication efficiency, but no detailed theory of how this efficiency comes about has been presented, nor has any previous research explicitly related the two findings.

The novel communication task used in this dissertation is to describe simple graphical objects through a computer chat window. The partners communicate through a restricted interface which constrains the words and phrases that can be typed in the chat window. In previous work with referential communication, partners have been given roles as directors of action and followers of directions, and stimuli have typically been difficult to name so that partners are forced to come up with creative ways of describing the stimuli. In contrast, the current referential communication task gives both partners equivalent goals and abilities, and the stimuli are simple graphical objects that vary on dimensions of size, shape, and color. The variance of creativity is reduced by having the interface provide names for the values on the dimensions. The communication task and interface are used in an experiment that tests the effect of accommodation on the efficiency of referential communication. The accommodation of a conversational partner is experimentally manipulated, and it is found that subjects with an accommodating partner create references with fewer words than subjects with a non-accommodating partner. This result is replicated with a computational theory of human cognition (ACT-R) that explains the result as the use of rules for creating efficient communication (rules for skipping words or whole messages) that are sensitive to the cooperative actions of a conversational partner. Unknown to the subjects, the accommodating and non-accommodating partners are actually interactive agents created with the ACT-R theory. Having computational agents as partners allows a very clean manipulation of accommodation, since all behavior besides the accommodating or non-accommodating communication is known to be the same. These agents perform comparably to human subjects and are able to trick about half of the subjects into thinking they are communicating with a human.

Chapter 1 of this dissertation will present past research on referential communication, accommodation, and the ACT-R theory. Chapter 2 will describe a pilot study where subjects used either a restricted or non-restricted interface to solve a simple communication task. Chapter 3 will describe the main experiment which found an effect of accommodation on communication efficiency. Chapter 4 will describe how the ACT-R theory and theories of communication are combined to create accommodating and non-accommodating interactive models, as well as a "human" model that represents human subjects interacting with those models. Chapter 5 will provide conclusions for the empirical evidence found, the modeling effort, and dissertation contributions.

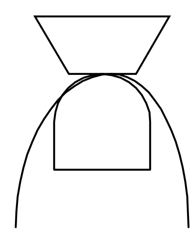
"There is more than a verbal tie between the words common, community, and communication..."

John Dewey

Referential Communication

Imagine that two people have to communicate a number of times about abstract figures that are difficult to name. Typically, the pair will initially use a long referential phrase and with subsequent references shorten that phrase to one or two words (Clark & Wilkes-Gibbs, 1986; Krauss & Fussell, 1991; Krauss & Weinheimer, 1966). This process is evidence of the collaborative nature of communication since subsequent phrases tend to make reference to previous phrases and since the phrase eventually agreed on to describe the object would not likely be able to describe the object without the benefit of the prior history of the evolution of the phrase.

Trial	Words	Reference Phrase
1	10	Looks like a Martini glass with legs on each side
2	5	Martini glass with the legs
3	4	Martini glass shaped thing
4	2	Martini glass
5	1	Martini



(from Krauss & Fussell (1991))

Several partner-related factors have been shown to influence the number of words used in the referential communication task. If subjects are asked to create referential phrases for an imagined partner who will later read the phrases, the phrases tend not to decrease over time (Hupet & Chantraine, 1992). If a partner is present but not allowed to give feedback, the rate of decrease is slowed (Krauss & Weinheimer, 1966). Also, pairs of elderly partners tend to use more words than younger pairs (Filer & Scukanec, 1995; Hupet, Chantraine, & Nef, 1993). In the Hupet et al. study, the elderly group was matched to the younger group by language and working memory capacity measures, and pairs were selected within groups to perform a repeated referential task. Both groups used a decreasing number of words over time, but the younger group used significantly fewer words than the elderly group. Detailed analysis showed that younger subjects were more likely to take into account previously shared information while elderly subjects were more likely to be idiosyncratic in their descriptions. In the terms of Giles, Mulac, Bradac, & Johnson. (1987), the younger subjects accommodated by converging their behavior while the elderly subjects were diverging.

"*Do as most do and men will speak well of thee*."

Thomas Fuller

Accommodation

In this discussion, accommodation is the matching of partner behavior in a conversational setting. These behaviors can include lexical choice (Fais, 1998; Garrod & Anderson, 1987; Garrod & Doherty, 1994) and syntactic choice (Bock & Griffin, 2000; Bock, 1986; Fais, 1994) as well as speech styles, dialect, non-verbal behavior, vocal intensity, prosody, speech rate and duration and pause length (Giles et al., 1987). The motivations for accommodating can be thought of in terms of normative and informational motivations (Deutsch & Gerard, 1955). Conforming to the behavior of others for normative reasons would include social motivations such as avoiding rejection, maintaining acceptance, or gaining approval. Conforming for informational reasons includes motivations such as efficiency, the desire to perform correct behaviors in ambiguous situations, or to learn the behaviors of a new culture. In their Communication Accommodation Theory, Giles et al. (1987) describe the conditions for convergence of communication behaviors as follows:

(1) People will attempt to converge toward the speech and nonverbal patterns believed to be characteristic of their message recipients, be the latter defined in individual, relational, or group terms, when speakers:
  (a) desire recipients' social approval (and the perceived costs of acting in an approval-seeking manner are proportionally lower than the perceived rewards);
  (b) desire a high level of communicational efficiency;
  (c) desire a self-, couple-, or group-presentation shared by recipients;
  (d) desire appropriate situational or identity definitions;

  when the recipients'
  (e) actual speech in the situation matches the belief that the speakers have about recipients' speech style;
  (f) speech is positively valued, that is, nonstigmatized;
  (g) speech style is appropriate for the speakers as well as for recipients.

(2) The magnitude of such convergence will be a function of
  (a) the extent of speakers' repertoires, and

(b) individual, relational, social, and contextual factors that may increase the needs for social comparison, social approval, and or high communicational efficiency.

Examples of accommodation can be seen in the maze game of Garrod and Doherty (1994), where subjects must decide how to describe their positions in a two-dimensional maze. Some subjects came to describe their positions in a line notation, giving first the line and then their location in that line:

A: Third row two along.
B: Second row three along.

Other subjects developed a matrix notation, giving horizontal and vertical locations:

A: Correct, I'm presently at C5.
B: E1.

People will even accommodate to computer systems. Many studies (Lehman, 1989; Slator, Anderson, & Conley, 1986; Zoltan-Ford, 1991) have shown that people will adapt their vocabulary and syntax to match the output of computer programs. For example, Lehman (1989) shows an example from a computerized scheduling system where given the following confirmation output by the computer:

```
Do you want:
     12:00-1:30     lunch, Andy
```

one user started to use a similar syntax to schedule appointments:

```
12:00 - 1:30 June 11 lunch with Andy
```

In fact, even when given explicit instructions that the computer could not understand its own output, another user saw the following output:

```
JUNE 10  10:00am - 12:00pm MEETING in/at OFFICE with AISYS
```

and started to imitate the output with her own commands:

```
cancel 10:00am Meeting in/at Office with AISys June 10
```

In addition to accommodating by converging on similar communication behavior, people can also diverge to different communication behavior. This process is called non-accommodation.

"You say potato, and I say po-tah-to
You say tomato, and I say to-mah-to
Potato, po-tah-to
Tomato, to-mah-to
Let's call the whole thing off"

George & Ira Gershwin

Non-Accommodation

A conversational partner may be non-accommodating by failing to match the behaviors of a partner. This non-accommodation may be motivated or unmotivated. The same normative and informational motivations for accommodation (Deutsch & Gerard, 1955) can be considered for non-accommodation. Socially, nonconformity could be motivated by an intention to show disapproval or assert power. Informationally, nonconformity could be motivated by non-confrontational disagreement on appropriate behavior or an intention to deceive. Additionally, humor can be a motivation for nonconforming behavior. In their Communication Accommodation Theory, Giles et al. (1987) describe the "antecedents" for divergence of communication behaviors as follows:

(3) Speakers will attempt to maintain their communication patterns, or even diverge away from their message recipients' speech and nonverbal behaviors when they
- (a) desire to communicate a contrastive self-image;
- (b) desire to dissociate personally from the recipients or the recipients' definition of the situation;
- (c) define the encounter in intergroup or relational terms with communication style being a valued dimension of their situationally salient in-group or relational identities;
- (d) desire to change recipients' speech behavior, for example, moving it to a more acceptable level;

when recipients:
- (e) exhibit a stigmatized form, that is, a style that deviates from a valued norm, which is
- (f) consistent with speakers' expectations regarding recipient performance.

(4) The magnitude of such divergence will be a function of
- (a) the extent of the speakers' repertoires, and
- (b) individual, relational, social, and contextual factors increasing the salience of the cognitive and affective functions in (3) above.

Most research on accommodation has focused on dependent measures of converging/diverging behavior or recipient evaluations of that behavior. Giles et al. (1987) describe the "consequences" of convergence and divergence as follows:

(5) Convergence will be positively evaluated by message recipients, that is, will lead to high ratings for friendliness, attractiveness, and solidarity when recipients perceive
    (a) a match to their own communicational style;
    (b) a match to a linguistic stereotype for a group in which they have membership;
    (c) the speaker's convergence to be optimally distant sociolinguistically, and to be produced at an optimal rate, level of fluency, and level of accuracy;
    (d) the speaker's style to adhere to a valued norm;

    especially when
    (e) perceived speaker effort is high;
    (f) perceived speaker choice is high;
    (g) perceived intent is altruistic or benevolent.

(6) Divergence will be negatively rated by recipients when they perceive
    (a) a mismatch to their own communicational style;
    (b) a mismatch to a linguistic stereotype for a group in which they have membership;
    (c) the speaker's divergence to be excessively distant, frequent, fluent, and accurate;
    (d) the speaker's style to depart from a valued norm;

    especially when
    (e) perceived speaker effort is high;
    (f) perceived speaker choice is high;
    (g) perceived intent is selfish or malevolent.

In addition to affecting evaluations, some research has shown that diverging communication behavior can also lead to disagreement of propositions (Connor-Linton, 1999) or discrimination (Chick, 1985). One hypothesis of this dissertation is that non-accommodation can also reduce the efficiency of referential communication. Support for this hypothesis would be shorter messages for subjects interacting with accommodating partners as compared to subjects interacting with non-accommodating partners.

Partial support of this hypothesis comes from the Hupet et al. (1993) study where the referring phrases of an elderly population were longer and more idiosyncratic than those of a younger population. However, the convergence or divergence of the phrases were not manipulated, and so it is not clear if the idiosyncratic choices actually caused longer phrases to be formed. To do this manipulation with human partners, either confederate partners or partners motivated with positive and negative social group pressures would need to converge or diverge to communication behavior. Either choice would introduce extraneous social complications into a question about informational processing. Ideally, the decision to diverge or converge should be independent of other communication processing in the partner. One solution is to use computational agents as partners. Two agents could be created that would either converge to or diverge from word choice of a human partner, with other communication processing being exactly the same. If both agents accommodated to the message length used by the human partner, then message length could be used as a dependent measure of efficiency. This would then test the effect of lexical accommodation on message length. The generality of the results would be greater if the agents were psychologically plausible. One line of research involving computational theory of human cognition is ACT-R.

"*It is easy to build a philosophy -- It doesn't have to run.*"
Charles Kettering

ACT-R

ACT-R (Anderson & Lebiere, 1998) is a computational theory of human cognition incorporating both declarative knowledge (e.g., addition facts) and procedural knowledge (e.g., the process of solving a multi-column addition problem) into a production system where procedural rules act on declarative chunks. At a subsymbolic level, facts have an activation attribute which influences their probability of retrieval and the time it takes to retrieve them. Rules have a reliability attribute which influences their probability of being used.

	Declarative Knowledge	Procedural Knowledge
Symbolic	Facts	Rules
Subsymbolic	Activation	Reliability

Support for this declarative/procedural viewpoint has been found in many ACT-R language projects. One project emphasizing declarative representation is Boyland and Anderson's (1997) model of syntactic priming. Research has shown that the use of a specific syntax can be primed in experimental settings if a subject repeats presented sentences (Bock & Griffin, 2000; Bock, 1986). Boyland and Anderson created a model that explained this phenomenon as priming of declarative structures built from the comprehension of sentences. Anderson and Matessa (1997) showed that a model using a hierarchical declarative representation of lists could explain effects in the list memory literature such as serial position, list length, and positional confusion. This same hierarchical structure was used by Lewis (1999) to represent the list nature of sentences in a model of sentence processing that accounts for difficulty in relative clause embeddings in a number of different languages. Budiu and Anderson (2000) used a model to show the effect that declarative facts have on metaphor understanding, processing of semantic illusions, and text memory.

With a procedural representation, Matessa and Anderson (2000) showed that the ACT-R rule reliability learning mechanism predicts a blocking effect in cue learning where the use of highly available cues can block the learning of more reliable cues since the sequential nature of productions allows only one cue to be chosen at a time. This prediction was supported by experimental evidence of blocking for linguistic actor choice cues such as word order, case marking, and verb/noun matching. Taatgen and Anderson (2000) used a model that combined both declarative and procedural learning to explain the U-shaped learning of irregular verbs.

Lebiere and Anderson's (1998) model of addition fact learning provides a good example of the ACT-R theory of learning new declarative chunks. The theory stipulates that there are only two sources of new chunks: from perception and from completed goals. The goal in addition is to find the sum of two numbers and this can be accomplished by computing the answer (e.g., by counting) or retrieving the answer from memory. ACT-R accounts for the creation of addition fact chunks as follows: initially, the goal of an addition problem is completed by computing the answer and storing the answer in the goal. Once this goal is completed, it is then available as an addition fact. This process of creating new declarative chunks can also be applied in the domain of communication, where the declarative knowledge assumed to be shared by participating individuals is known as mutual knowledge or common ground. Before describing an ACT-R model of communication, a communication task will first be described, along with an evaluation of the task with subjects in a pilot study.

Chapter 2: Pilot Study

Communication Task

A simple two-participant task was required where both participants have the same abilities and unique knowledge to be communicated. At first, a letter sequence task (Novick, Hansen, & Lander, 1994) was considered where subjects are given different sequences of letters with some missing letters with the goal of creating a whole letter sequence. Any letter that is missing for one subject is known by the other subject, and some letters are known by both subjects. So one interesting aspect of this task is that initially there is some information that is mutually known, some that is only known by one subject, and some that is only known by the other subject.

The idea of subjects talking about shared and unique information was appealing, but the one-way linear constraint of reading a sequence seemed to make the task too simplistic, so a two-dimensional task was created where subjects are given parts of a graph with the goal of creating a whole graph. The graphs are colored circles connect by lines (similar to those used by Levelt (1982) to study communicative reference) and are designed so that similarly colored circles on the parts can overlap and form a larger graph (see Figure 2.1). So like the letter sequence task there is common information and information unique to each subject, but unlike the letter sequence there is no linear constraint to the information and so subjects must agree on how to communicate information about their graph parts and how the parts of the graph overlap. Communication using text is more conducive to modeling, so the subjects send messages by way of a chat window from two different computers. In addition to creating a whole graph from two parts, subjects also have the goal of confirming each of the circles. This is done by each subject selecting one circle at a time -- if the circles are the same, their score is increased, but if the circles are different, the score is decreased. This confirmation goal gives an objective measure of task

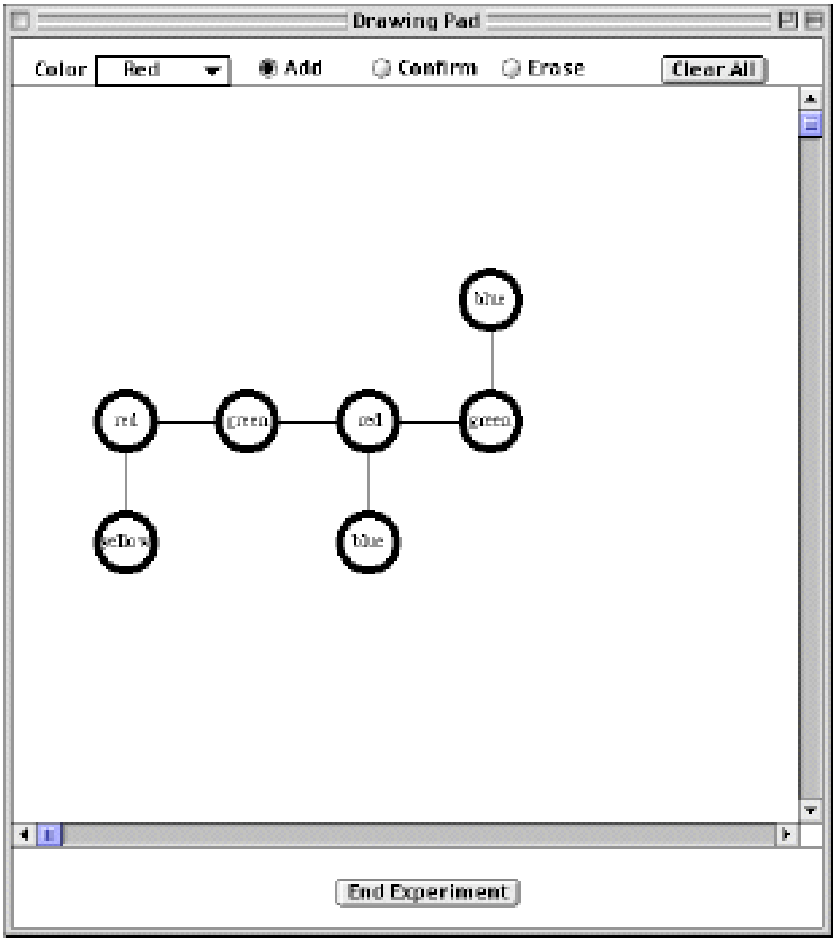


Figure 2.1: Drawing pad

performance in terms of a score, and it allows for the use of more complicated dialogue acts such as requesting that the other person confirm a circle or committing to confirming a circle. In order to facilitate coordination of confirmation, confirmation status messages appear in a status window. If a subject selects a circle to confirm and their partner has not yet confirmed, the message "Waiting for partner..." appears. If their partner confirms a circle first, the message "Waiting for you..." appears.

Communication Interface

In a similar spirit to the COLLAGEN project (Rich & Sidner, 1998), this modeling effort is not aimed towards the processing of unrestricted English syntax but in modeling the higher-level communicative acts accomplished with English. So like the COLLAGEN project the models interact with people with a restricted set of English phrases. This restricted interface need not drastically hinder the communication process or task performance. In a study comparing a restricted interface to an unrestricted interface for students solving physics problems, Baker and Lund (1997) showed that the restricted communication interface did not interfere with task performance. In fact, it promoted a more task-focused and reflective interaction. Still, for the current task, unrestricted and restricted communication were compared to see if the restricted interface had any effect on task performance. The restricted interface allows the composition of a text message by first choosing a topic of discussion and dialogue act to address the topic. The topics of conversation are paired connections (how one circle relates to another), multiple connections (rows or columns of circles), numbers (how many of a specific kind of circle there are), correspondences (what circle in one person's graph corresponds to in the other person's graph), confirmations (talking about mutually confirming a circle), and experiment phases. Figure 2.2 shows the communication window where these choices can be made. For paired connection, multiple connection, and number topics, the Assert, Info-request, and Answer dialogue acts can be initiated with the Make Statement, Ask Question, and Answer buttons (respectively). For correspondences, the Assert and Agreement dialogue acts can be initiated with the Propose and Assess buttons, and for confirmations and experiment phases, the Action-directive and Agreement dialogue acts can be initiated with the Request and Assess buttons. These choices bring up sentence templates where words can be chosen for the sentences from pull-down menus. Sample words from a choice to make a statement about paired connections can be seen in Figure 2.3. Each column represents a list of words in a menu of which one word can be chosen. For example, the sentence "My leftmost red circle is above my rightmost orange circle" can be created with these menus.

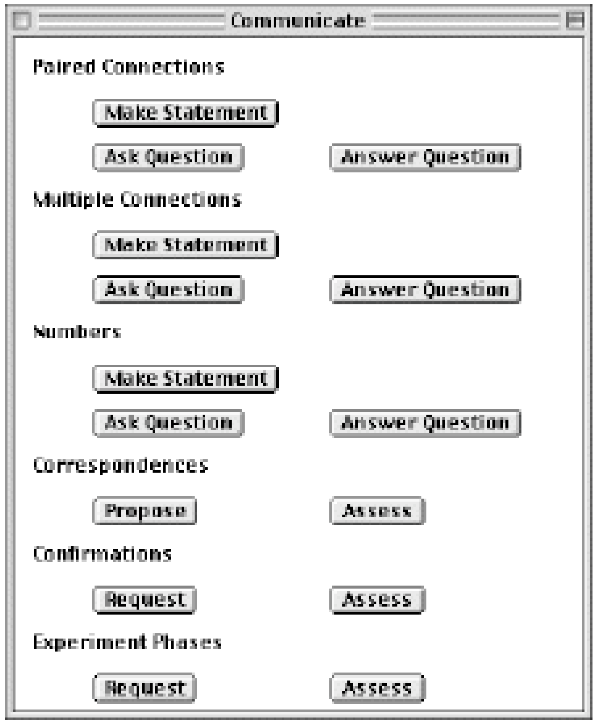


Figure 2.2: Communication window

My	leftmost	red	circle	is above	my	leftmost	red	circle.
Our	rightmost	orange		is below	our	rightmost	orange	
	topmost	yellow		is left of		topmost	yellow	
	bottommost	green		is right of		bottommost	green	

Figure 2.3: Words to make statements about paired connections

Figure 2.1 shows the drawing pad on which circles can be added, erased, and confirmed. Subject initially see only their part of the problem, and so any information about their partner's circles must be received from the chat window.

Subjects

Fourteen pairs of Carnegie Mellon University undergraduate and graduate students attempted the graph completion task, with seven pairs using an unrestricted interface and seven pairs using a restricted interface.

Method

Each pair was told that they would each be given part of a graph and their goal was first to create a whole graph as a result of circles overlapping from each part of the graph, and then to confirm each circle in the whole graph. They were told they would be sitting in different rooms and would be using a chat window to talk to each other. They were shown a drawing pad which contained an example graph part consisting of connected colored circles, and were shown how to add and erase circles representing circles from the partner's graph. They were also shown a chat window which could send eighty-character messages and only displayed the partner's last message. In the restricted interface condition, subjects were told that messages were composed in a communication window that allowed the creation of restricted sentences and were led through the creation of each kind of message. After making sure subjects understood the task, they were then given individual practice problems which used the adding, erasing, and confirming functions of the drawing pad. Finally, the subjects were given their graph parts and were told there were no time constraints in solving the problem.

Results

Of fourteen total pairs, one pair in each of the unrestricted and restricted conditions were unable to complete the task in the hour provided. To compare task performance between the unrestricted and restricted interface conditions, the number of turns to complete the task, the time to complete the task, and the final score were measured (Table 2.1). There was no significant difference in the number of turns (t=0.798), time (t=0.1551), or final score (t=1.185). The low number of students gives these results a low power, but as a pilot result it appears there are no drastic differences between the two interfaces.

	Unrestricted			Restricted			T	df	p
	mean	SD	min,max	mean	SD	min,max			
turns	21.5	(5.8)	[11,28]	24.7	(7.8)	[12,33]	0.798	10	0.44
time	22.3	(6.6)	[15,34]	28.3	(6.8)	[15,33]	1.551	10	0.152
score	90.0	(16.7)	[60,100]	98.3	(4.1)	[90,100]	1.185	10	0.264

Table 2.1: Performance in Unrestricted and Restricted conditions

To examine detailed subject performance, paired dialogue acts in the two conditions were examined (Table 2.2). The obligation theory of Traum and Allen (1994) predicts that all Info-request dialogue acts will be followed by Answer dialogue acts. This prediction is supported by the data, where in both restricted and unrestricted interface conditions, 97% of Info-request dialogue acts were followed by Answer dialogue acts. The grounding theory of Clark and Schaefer (1989) predicts that statements are known to be in common ground only after acceptance from a partner. Here, the prediction is that confirmation Action-directive dialogue acts will be followed by Agreement dialogue acts (which include Accept, Reject, Maybe, and Hold acts) before confirmation actions are made. This prediction is supported in the restricted interface condition, where 95% of the confirmation Action-directive dialogue acts were followed directly by explicit Agreement acts (a text message) or implicit Agreement acts (confirmation of the circle by the other subject). In the unrestricted interface condition, only 54% of the Action-directive acts were followed directly by explicit or implicit Agreement acts. Part of the reason for this low number is that 70% of these Action-directive acts occurred after an explicit plan had been made on the sequence of circles to be confirmed. This planning was not supported by the restricted interface or the model. But even with this planning, all (three) of the incorrectly confirmed circle errors in the unrestricted interface occurred as a result of a subject not waiting for an explicit or implicit Agreement act after a confirmation Action-directive.

	Restricted	Unrestricted
Info-request	(n=6)	(n=12)
->Answer	97%	97%
->Statement	3%	1%
->Directive	0	2%
Action-directive (confirm)	(n=66)	(n=50)
->explicit Accept	47%	44%
->implicit Accept	41%	10%
->(speaker action)	5%	46%
->implicit Reject	5%	
->implicit Hold	1%	
->explicit Reject	1%	

Table 2.2: Paired speaker/listener dialogue acts

The only error in the restricted interface condition occurred as a result of a "group hallucination" when both partners created and confirmed a circle that neither of them had as part of their original graph. Since each problem had ten circles to confirm and twelve pairs of subjects completed the task, there were 120 chances overall to incorrectly confirm circles. An example of how waiting for an Agreement act prevented an error in the restricted condition can be seen in the following example: Subject A produced an Action-directive act with "Let's confirm our third green circle." Subject B produced an implicit Reject act with "I have two green circles." Subject A then did not confirm the circle but produced an Accept act with "OK."

The conceptual content found in the two interface conditions can be seen in Table 2.3. Pluses indicate that the restricted interface supported the expression of the content, while minuses indicate the restricted interface did not support expression. Generally, the conceptual content was similar for the two interface conditions with a few exceptions. Under the Multiple Connections topic, the restricted interface supported discussion of "blank" circles, or circles that did not exist in a certain location. For example, the third row of Figure 2.1 could be described as yellow, blank, blue, blank. Prompted by the option to use the word "blank", 100% of the pairs in the restricted interface condition used this blank circle concept, while only 33% of the pairs in the unrestricted condition discovered and used this concept. Under the Correspondences topic, 50% of the pairs in the unrestricted condition directed their partner to add a circle, while none did in the restricted condition because it was not an option. Under the Confirmations topic, 50% of the pairs in the unrestricted condition used a sequence plan (e.g. left-to-right) to confirm circles without having to send explicit messages to do so. Finally, under the Experiment Phases condition, 67% of pairs in the unrestricted condition sent messages saying they were done giving information, while none did in the restricted condition.

	Restricted Interface	Restricted Pairs (n=6)	Unrestricted Pairs (n=6)
Paired Connections			
statement	+	83%	100%
question	+	67%	100%
Multiple Connections			
statement	+	100%	100%
question	+	50%	67%
row	+	100%	100%
column	+	33%	17%
blank	+	100%	33%
Numbers			
total	+	50%	50%
colors	+	67%	33%
row	+	50%	33%
column	+	33%	17%
in row	-	0	17%
in column	-	0	17%
Correspondences			
explicit	+	67%	33%
implicit	+	33%	67%
direct to add	-	0	50%
commit to add	-	0	33%
Confirmations			
explicit	+	100%	100%
sequence plan	-	0	50%
Experiment Phases			
more info?	-	0	17%
done info	-	0	67%
end experiment	+	100%	100%

Table 2.3: Conceptual content

Conclusions

Since subjects' performance in the graph completion task (as measured by score, turns to completion, and time to completion) was not unusually different between the restricted and unrestricted interface conditions, the restricted interface seems to be an appropriate tool in studying this task. For subjects using this restricted interface, the obligation theory of Traum and Allen (1994) was successful in its predictions of the obligation to answer questions, and the grounding theory of Clark and Schaefer (1989) was successful in its prediction of subjects waiting for an Agreement dialogue act before confirming circles. For subjects using the unrestricted interface, these theories were only successful in their prediction of the obligation to answer questions. Most subjects using the unrestricted interface who did not wait for agreement before confirming circles had previously agreed on a sequential plan to confirm circles, and this strategy was not supported in the restricted interface or the model. Subjects using this strategy apparently assume their reference to a particular circle and their decision to confirm that circle will be acceptable to their partner because of their previous plan.

Task and Interface Modifications

In order to incorporate this goal planning strategy into the restricted interface, an improved restricted interface (Figure 2.4) was created that allowed discussion of this topic. Other topics that were discussed in the unrestricted interface but not available in the old restricted interface were included, such as directives to do actions and the end of giving information as a phase in the experiment. Instead of choosing from buttons representing dialogue acts, subject in the new interface could choose a sample sentence that would bring up a template which could be filled in with pertinent information. The Statement, Info-query, Action-directive, and Proposal dialogue acts were represented in the sample sentences for each topic.

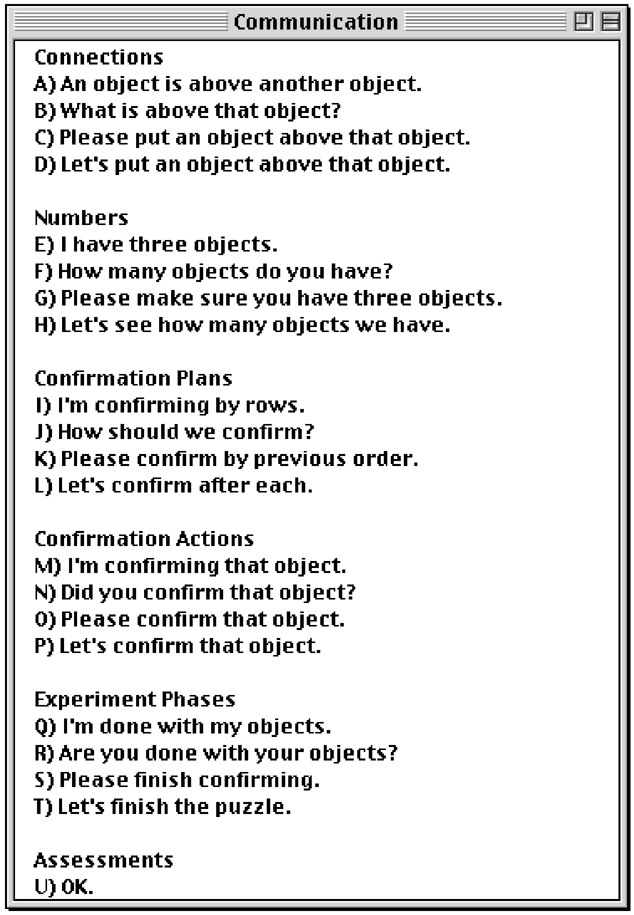


Figure 2.4: New communication window

Also, to allow more problems to be solved in a single experimental session which would allow the development of communication over time, the problems were simplified to have six total objects with one marked as common. From previous research (Clark & Wilkes-Gibbs, 1986; Krauss & Fussell, 1991; Krauss & Weinheimer, 1966) it was expected that the message length would decrease over time. To facilitate this decrease in the restricted interface, the manner of composing messages in the template was changed from choosing words from a pull-down menu to typing words that were displayed in a menu. The menu for the word choice could be skipped over with the Tab key, and in this way shorter messages could be produced. This new method permits a closer correspondence to the unrestricted interface (unrestricted typing) and gives a time benefit to skipping words by not having to spend time in typing them. Additional dimensions of size and shape were added to the color dimension of the circles in order to provide more redundant information in the problem that could later be left out of messages, resulting in a shorter message length. These dimensions were redundant, so that red objects were always small and thin, green objects were always medium and round, and blue objects were always large and fat. Examples of these shapes can be seen in Figure 2.5.

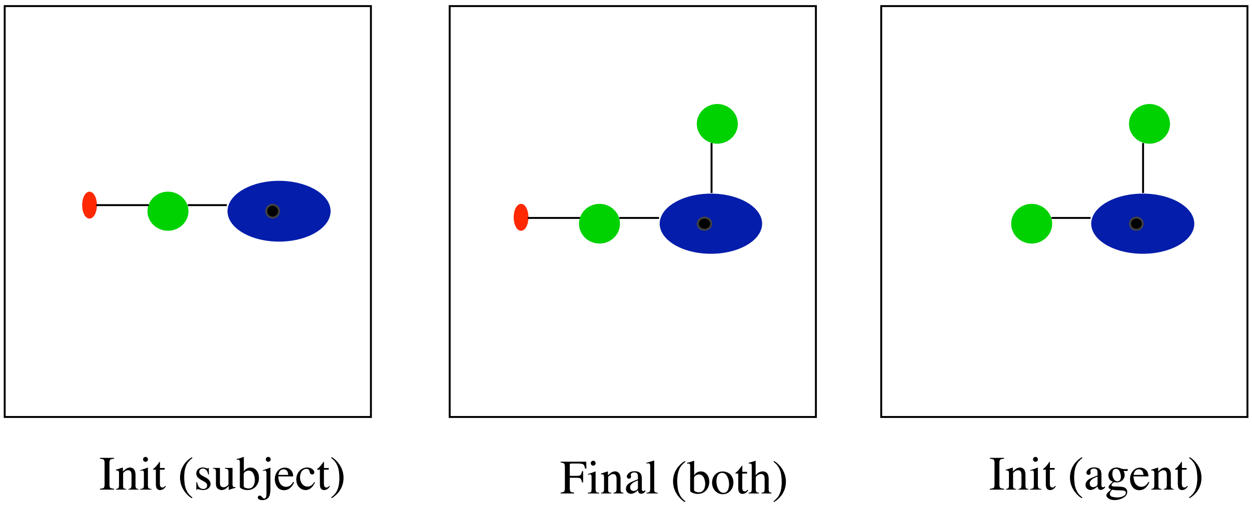


Figure 2.5: Initial and final drawing pads

The restricted interface was also modified to allow the testing of the effect of non-accommodation on subject behavior by providing different but functionally equivalent word choices. Figure 2.6 shows the words used to make sentences that discuss connections between pairs of objects. For example, the sentences "A topmost small thin red dot is above our middle medium round green dot" or "red above our dot" could be created. The functionally equivalent words include object description words (object, shape, dot, or blob), and words for directional mode (north, south, east, or west; above, below, right, or left). Accommodation can be accomplished by matching the word use of a partner. Non-accommodation can be accomplished by using different but functionally equivalent words.

A	topmost	small	thin	red	dot	is	above	a	topmost	small	thin	red	dot
The	bottommost	medium	round	green	blob		below	the	bottommost	medium	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				

Figure 2.6: Words for sentences discussing connected pairs of objects

Chapter 3: Main Experiment

The changes to the pilot task and interface allow the testing of the main hypothesis of this dissertation: subjects communicating with accommodating partners that match word use will solve problems more quickly than subjects who communicate with non-accommodating partners that use functionally equivalent but different words. From previous research (Clark & Wilkes-Gibbs, 1986; Krauss & Fussell, 1991; Krauss & Weinheimer, 1966) it is expected that some of this efficiency will be due to subjects using shorter messages over time. These accommodating and non-accommodating partners are (unknown to the subjects) ACT-R models of communication, to be described in the next chapter.

Subjects

One hundred Carnegie Mellon University undergraduates attempted the newer graph completion task. Twenty-two were paired and used the unrestricted interface, thirty-two were paired and used the restricted interface, twenty-two were paired with an accommodating ACT-R model, and twenty-four were paired with a non-accommodating ACT-R model. This created eleven pairs in the unrestricted interface condition, sixteen pairs in the restricted interface condition, twenty-two pairs in the accommodating model condition (pairs consisting of a subject and a model), and twenty-four pairs in the non-accommodating model condition.

Method

The procedure was the same as the pilot study except that multiple problems were used and subjects were told that there would be a cash incentive for completing as many problems as possible in the time given (one hour and forty minutes). Problems were smaller than those in the pilot study and initial views consisted of a common object that was marked with a black dot along with two or three other objects. The objects had three dimensions, color (red, blue, or green), size (small, medium, or large), and shape (thin, round, or fat). The dimension values were redundant so that any red object was small and thin, any green object was medium and round, and any blue object was large and fat.

Results

Figure 3.1 shows the number of pairs (either subject/subject or subject/model) that complete each problem. Pairs that solved problems faster were able to complete more problems in the one hour and forty minutes available, so slower pairs do not appear in later problem counts and the total count decreases with the problem number. More subjects were placed in the ACT-R accommodating and non-accommodating conditions to increase the statistical reliability of results in those conditions.

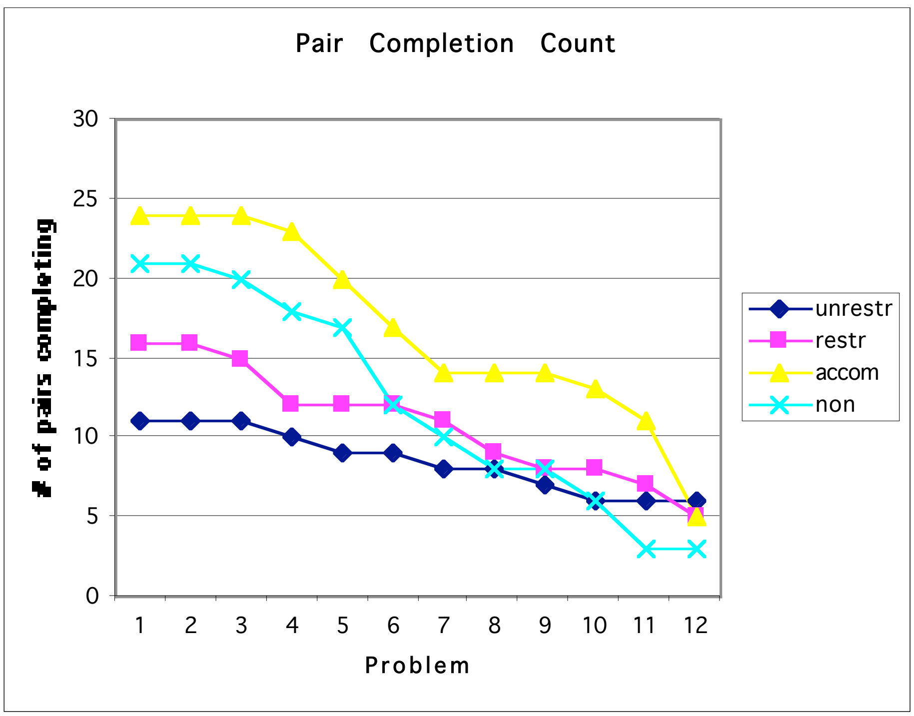


Figure 3.1: Number of pairs completing problems

Figure 3.2 shows the percent of pairs that solve each problem. Since the total number of pairs in each condition are different, this allows a comparison of pairs completing problems in the various conditions. In all conditions, all pairs were able to complete at least two problems. Around 55% of pairs in the unrestricted condition were able to complete 12 problems, compared to around 30% in the restricted condition, around 20% in the accommodation condition, and around 14% in the non-accommodation condition. The general trend after five problems is that more pairs in the unrestricted condition were able to complete problems than those in the restricted condition, and more pairs in the accommodation condition were able to complete problems than those in the non-accommodation condition. Due to the smaller number of pairs that complete the later problems, subsequent results are averaged over three-problem groups for greater statistical reliability.

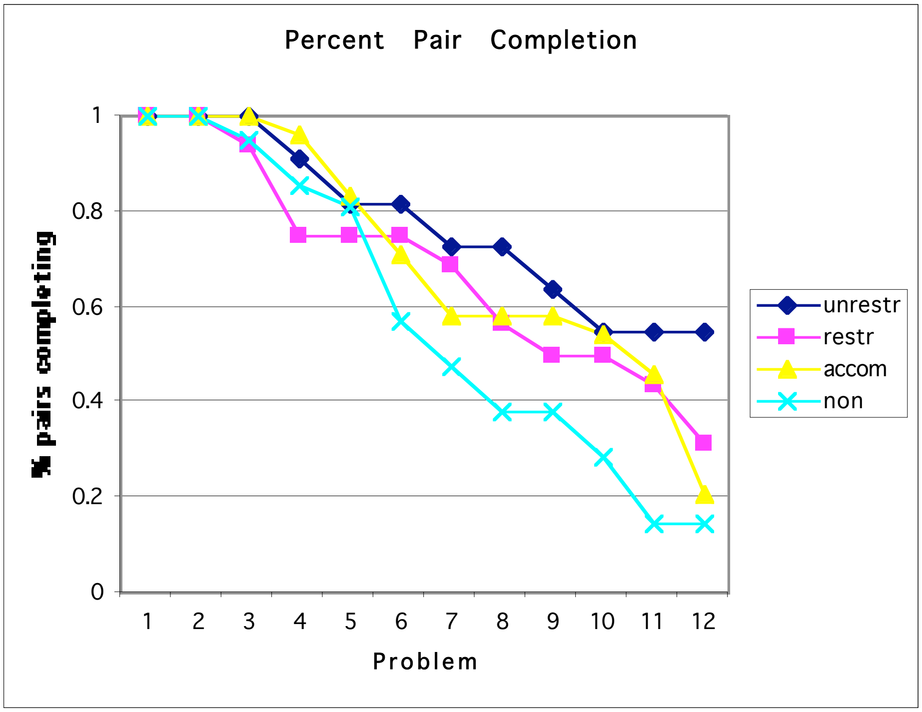


Figure 3.2: Percent of pairs completing problems

Figure 3.3 shows errors in performance for pairs in the experimental conditions. Each problem had a final pattern consisting of six objects, and points were given for correct confirmation of objects and removed for incorrect confirmation of objects. Error bars in this and subsequent figures represent standard error. Results are averaged for the first three problems, the second three, third, and fourth. Since there were an unequal number of pairs in each condition for any particular problem, statistics are performed on each group of three problems. There was no significant effect of experimental condition on errors for the first group of three problems ($F(3,65)=0.85$)) or any other group of three problems ($F(3,56)=1.25$; $F(3,37)=2.28$; $F(3,28)=0.38$).

In some problems the performance in the restricted condition (human/human) is better than performance in the accommodation or non-accommodation conditions (human/model). This is due to model difficulties with referencing objects that were not explicitly grounded by subjects and where no previous plan for referencing (e.g. sequentially by rows) had been established (for example, "a small thin red object is above that"). The model had some simple rules for understanding this kind of anaphoric reference, but sometimes incorrect references were made and errors were produced as a result.

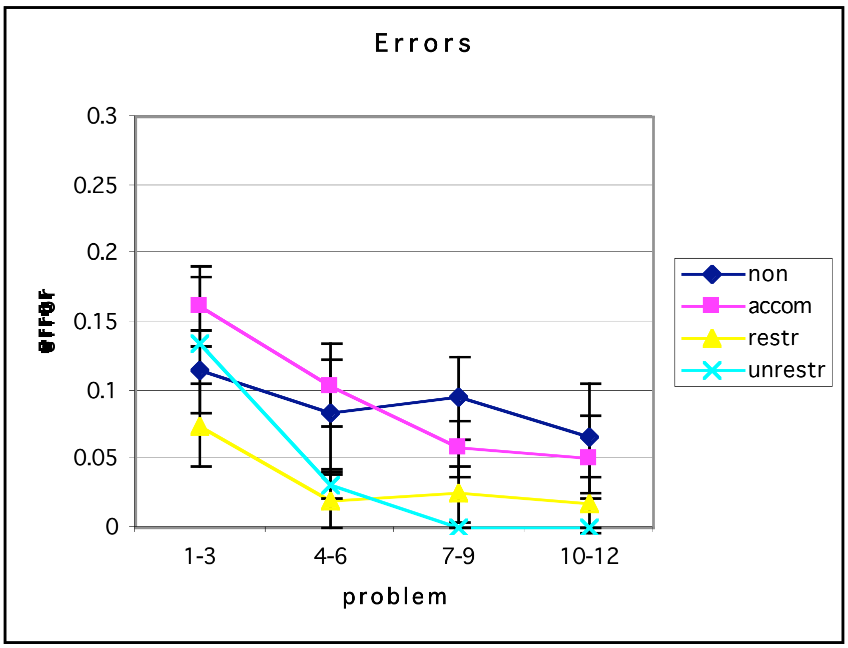


Figure 3.3: Errors in experimental conditions

Figure 3.4a shows the average time that pairs in each condition took to solve problems. There was a significant effect of condition on time to solve problems ($F(3,186)=5.91$, $p<.001$) but no significant interaction between condition and problem groups ($F(9,186)=0.96$). There was no significant effect of condition in the first two groups of three problems ($F(3,65)=2.12$; $F(3,56)=1.72$), but there was an effect in the last two groups of three problems ($F(3,37)=8.58$, $p<.0005$; $F(3,28)=5.76$, $p<.005$). This effect is driven by slower times in the non-accommodation condition, which was significantly slower than the accommodation condition in the last two groups of three problems ($t(24)=3.71$, $p<.001$; $t(18)=2.61$, $p<.05$). Looking at the unrestricted and restricted conditions, subjects in the restricted condition were significantly slower for the first three problems (t(22)=2.03, p<.05) but not significantly different for the remaining problem groups (t(18)=1.08; t(15)=1.86; t(10)=1.12). The initial difference and subsequent similarity between the conditions suggests a gradual adaptation to the restricted interface until by the second group of three problems the time to solve problems is not statistically different. Since the results were similar between the restricted and unrestricted conditions for errors and time to solve problems, the unrestricted condition will not be included in subsequent discussions.

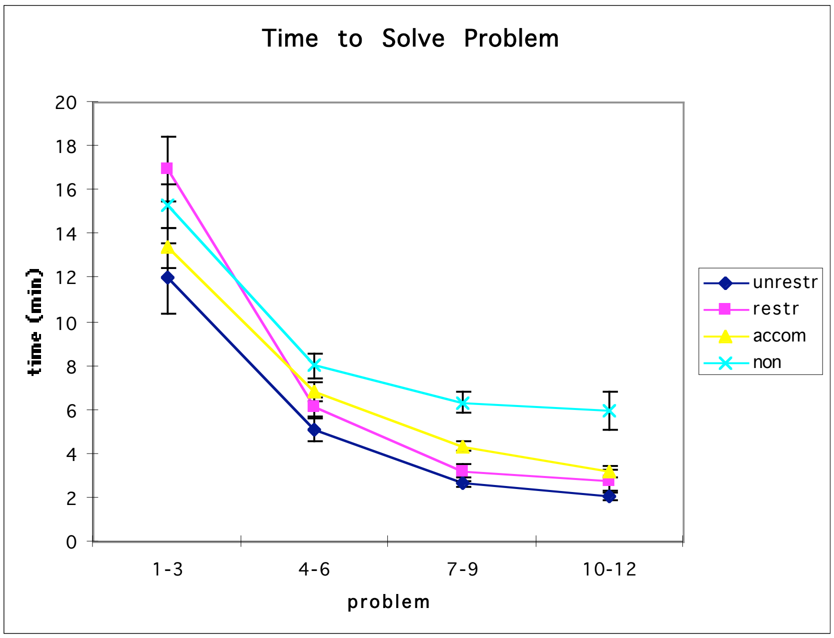


Figure 3.4a: Time to solve problems for all subjects

Since not all subjects complete all twelve problems, the trend of decreasing time with problem could be due to a selection effect where subjects who solve fewer problems always solve problems slowly and subjects who solve more problems always solve problems quickly. Figure 3.4b shows that this is not the case. This figure contains only subjects solving ten or more problems, and shows the trend of decreasing time with problem is not due to subjects dropping out of the analysis.

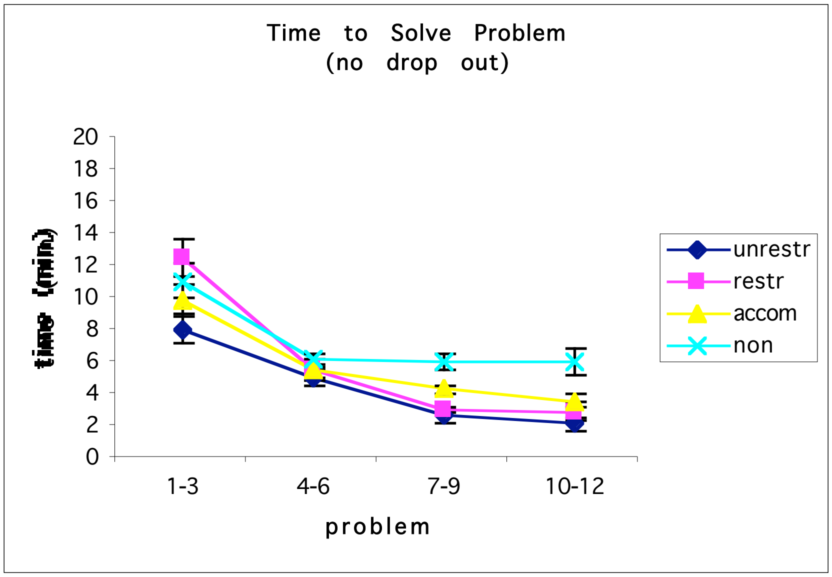


Figure 3.4b: Time to solve problems for subjects solving ten or more problems

The time involved in solving a problem is comprised of time from several phases: typing, drawing, reading, and waiting for a partner's response. Typing time is defined as the time between choosing a template and hitting return to send the message to the partner, drawing time is the time between choosing attributes of an object to the final placement of the object, waiting time is the time between sending a message to the partner and receiving a message back, reading time is the time between the arrival of a partner's message and the first action taken by the subject. As Figures 3.5a, b, and c show, typing takes up the majority of the active time spent in solving the problems in all of the conditions, so drawing and reading time will not included in subsequent discussions. The results in these figures are taken from partner "A" in the experimental conditions, which is the human subject in the accommodation and non-accommodation conditions.

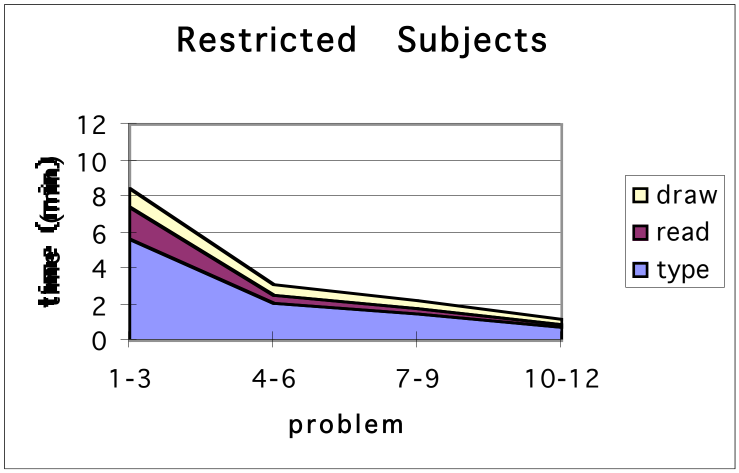


Figure 3.5a: Phase time for subjects in restricted interface condition

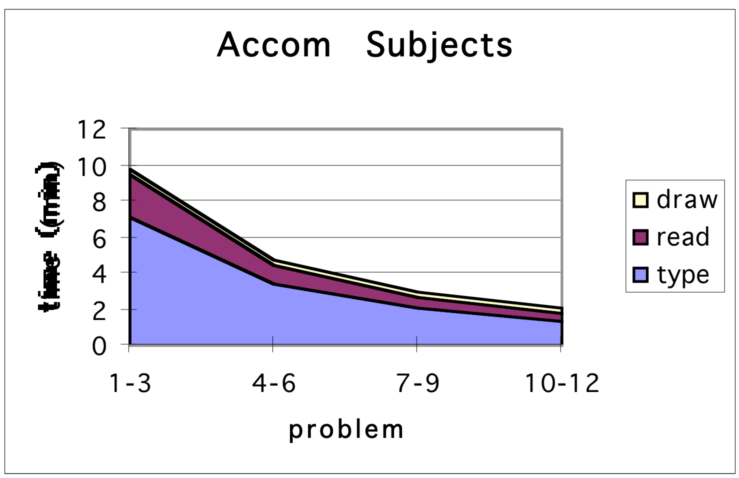


Figure 3.5b: Phase time for subjects in accommodating model condition

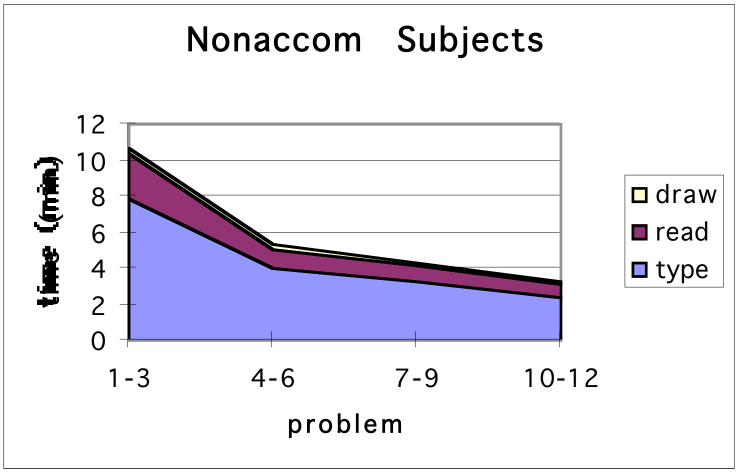
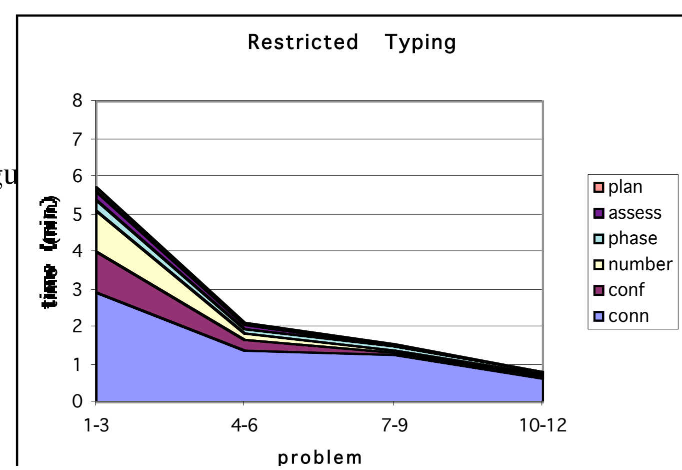


Figure 3.5c: Phase time for subjects in non-accommodating model condition

When choosing a message template for typing, subjects could choose from six different topics: connection of objects, number of objects, confirmation plan, confirmation action, experiment phase, or assessment. Figures 3.6a, b, and c show the time spent typing about these topics for each experiment condition. As the figures show, most of typing time is spent on the connection of objects, followed by confirmation actions, then number of objects. Since these topics take up the majority of typing time, time spent on the topics of confirmation plan, experiment phase, and assessment will not be included in subsequent discussions.

Figu ace condition



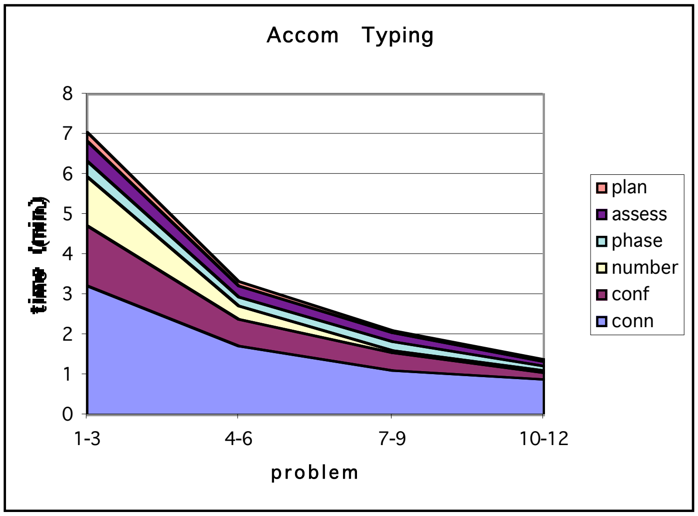


Figure 3.6b: Typing time for subjects in accommodating model condition

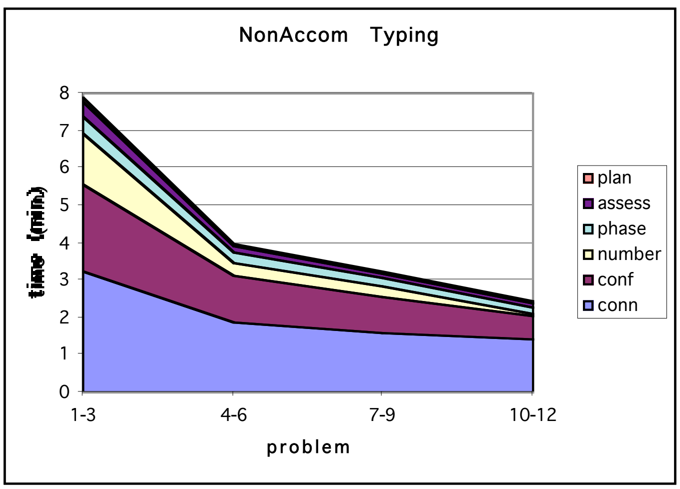


Figure 3.6c: Typing time for subjects in non-accommodating model condition

Connection Typing Time

The time taken in typing messages on the topic of the connection of objects can be compared for the restricted, accommodation, and non-accommodation conditions in Figure 3.7. As with the total time taken to solve problems, there generally appears to be no difference between conditions until problems 7-9, when the non-accommodation condition appears to take longer than the other conditions. Typing time in the non-accommodation condition is not significantly different from time in the accommodation condition for the first two groups of three problems (t(43)=0.29; t(39)=0.69), but is significantly slower for the last two groups of three problems (t(24)=2.36, p<.05; t(18)=2.30, p<.05).

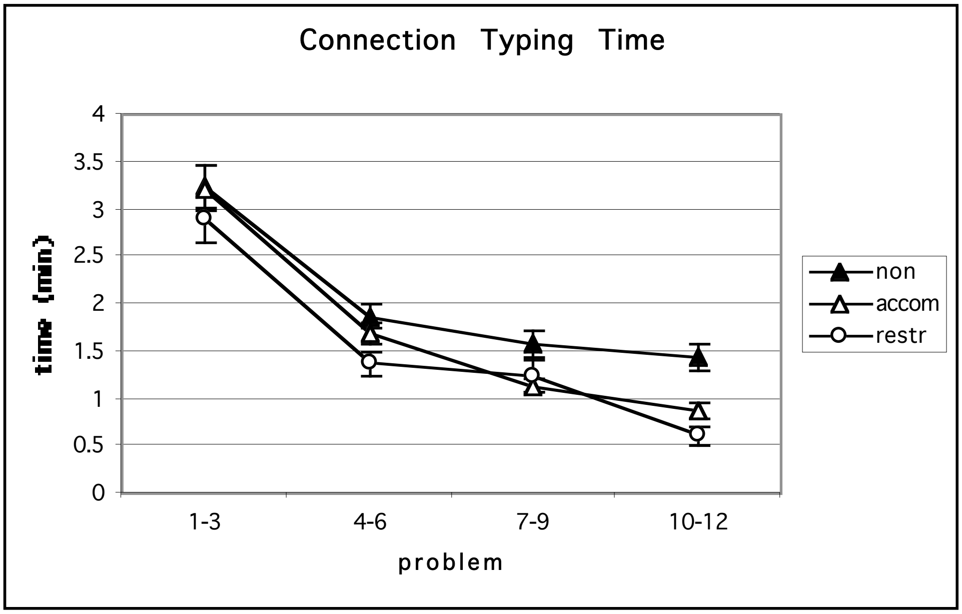


Figure 3.7: Time spent typing messages concerning connection of objects

The total time spent typing messages on a topic depends on two factors: how many messages are typed and the time spent typing each message. Figure 3.8 shows that there is no difference in the number of messages typed on the topic of connections of objects between the accommodation and non-accommodation conditions (t(43)=1.09; t(39)=0.26; t(24)=0.92,; t(18)=0.73).

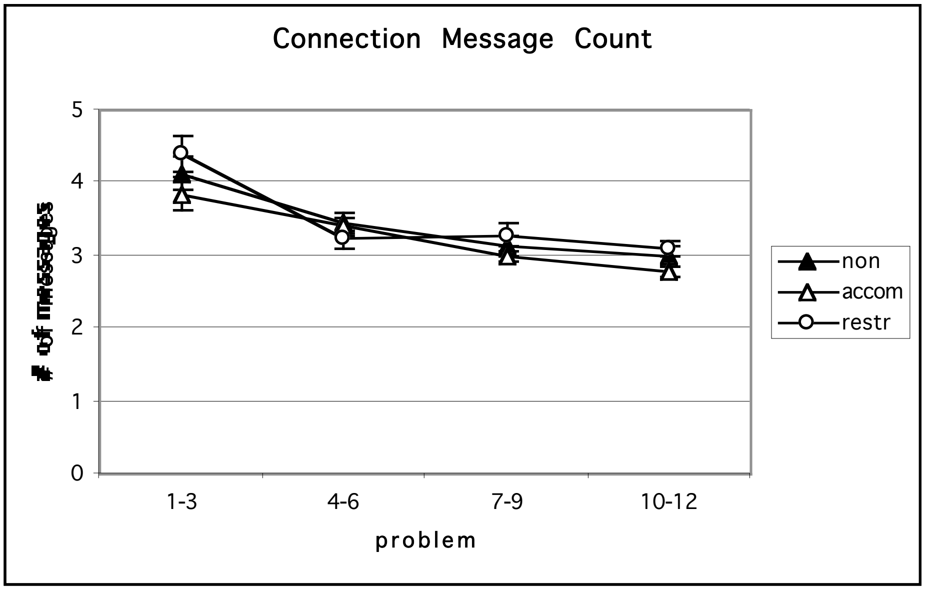


Figure 3.8: Number of messages concerning connections of objects

Figure 3.9 shows that the time to type a connection message in the non-accommodation condition is not significantly different from time in the accommodation condition for the first two groups of three problems ($t(43)=1.01$; $t(39)=0.96$), but is significantly slower for the last two groups of three problems ($t(24)=2.74$, $p<.01$; $t(18)=2.56$, $p<.05$).

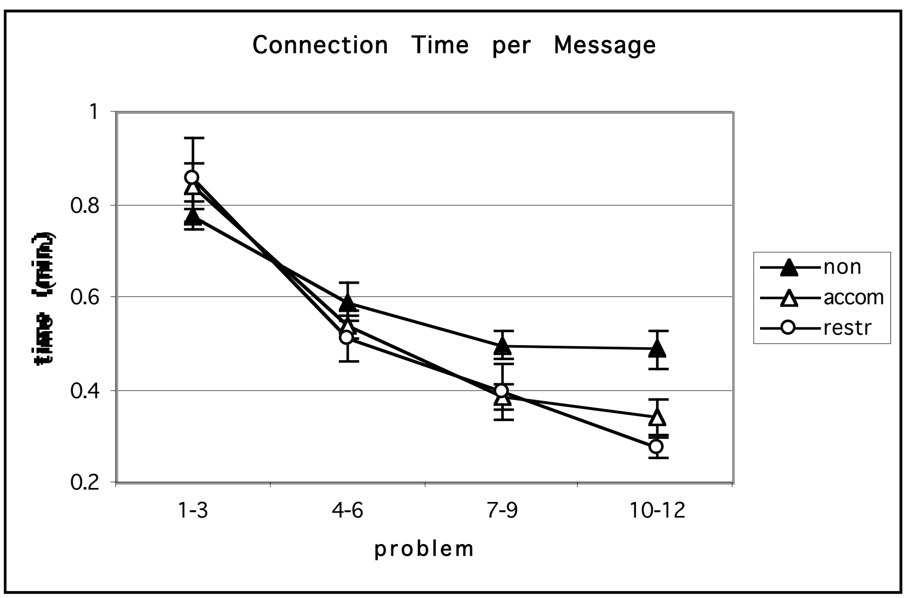


Figure 3.9: Time per message concerning connections of objects

The time spent typing a single message depends on two factors: the time spent typing each word in the message and how many words are typed. Figure 3.10 shows that there is no difference in the time spent per word in messages typed on the topic of connections of objects between the accommodation and non-accommodation conditions (t(43)=1.01; t(39)=0.37; t(24)=0.12; t(18)=0.35). For the restricted and accommodation conditions, there is no statistical difference in the time spent per word for the first three groups of problems (t(21)=0.94; t(16)=1.15; t(12)=1.25), but there is a difference the last group of problems (t(10)=2.28, p<.05).

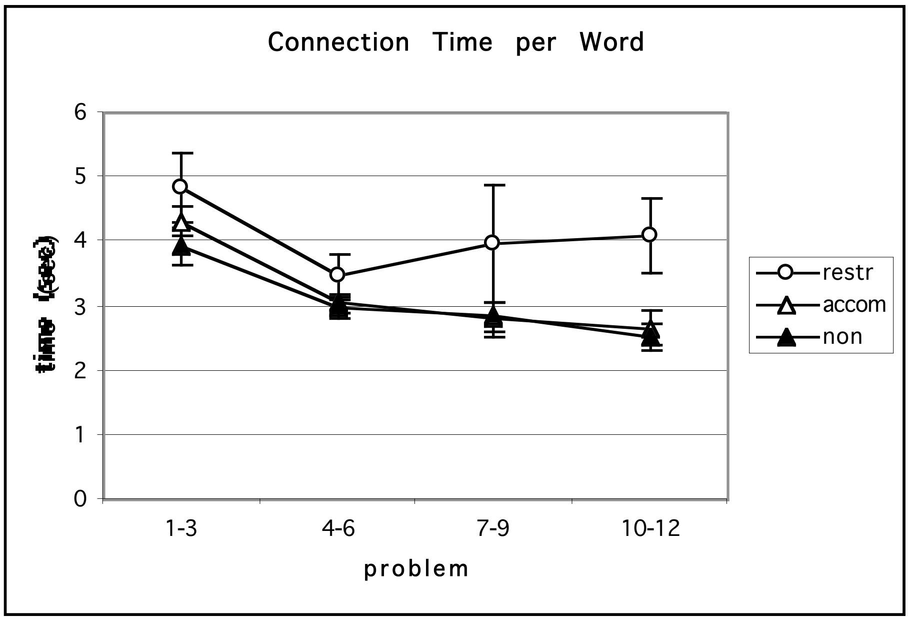


Figure 3.10: Time per word in messages concerning connections of objects

Figure 3.11 shows how many words were typed, or message length, for sentences concerning connections of objects. Message length tended to decrease with time, for example, a message such as "The small thin red object is above our large fat blue object" in the first problem could be reduced to messages such as "red above blue" by the twelfth problem.

Messages in the non-accommodation condition tended to be longer than those in the accommodation condition, not significantly in the first two groups of three problems (t(43)=0.55; t(39)=0.83) but significantly in the second two groups of three problems (t(24)=1.97, p<.05; t(18)=1.81, p<.05).

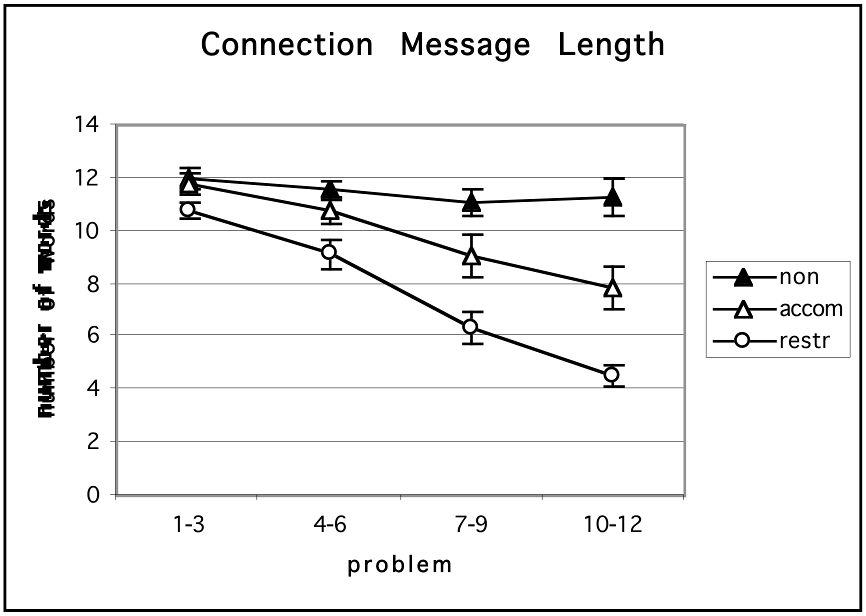


Figure 3.11: Message length for messages concerning connections of objects

Details of what words were skipped in messages pertaining to connection of objects can be seen in Figures 3.12a, b, and c. The fourteen values of the x-axis correspond to the position of words in the template used to create messages relating to connection of objects (see Figure 2.6). Each location represents a word category: determiner, global location, size, shape, color, object name, equivalence, connected direction, determiner, global location, size, shape, color, and object name. For example, one fourteen-word message could be "The small thin red object is above our large fat blue object". The y-axis is the percent use of that word category. Lines represent use for groups of three problems.

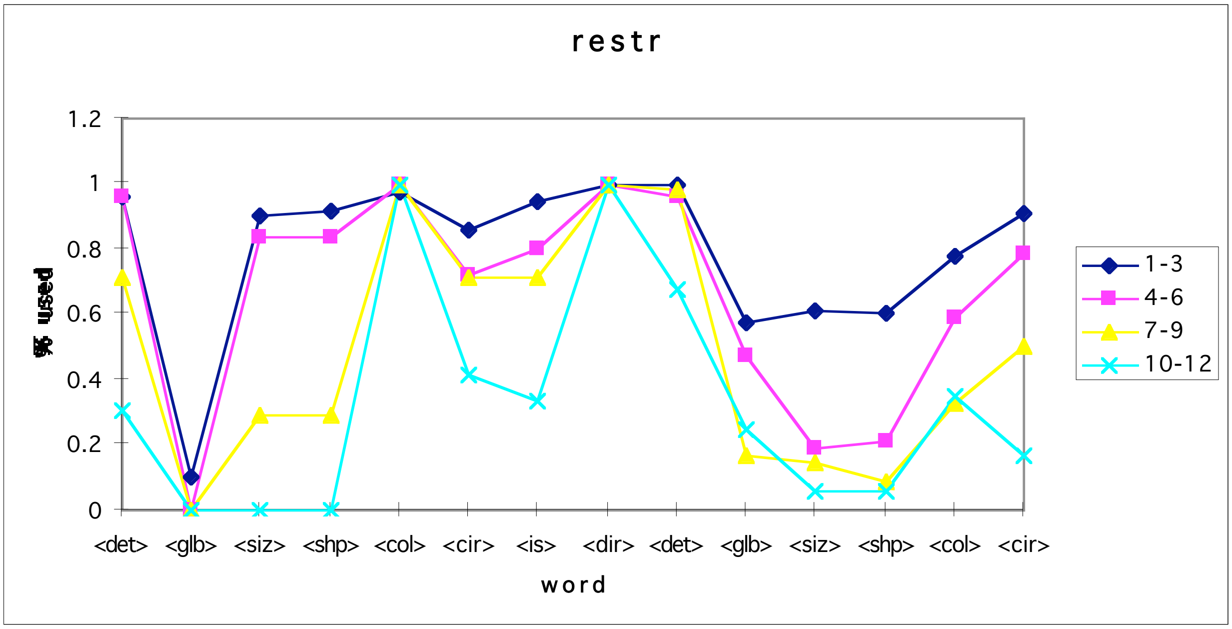


Figure 3.12a: Percent use of word categories in restricted interface condition

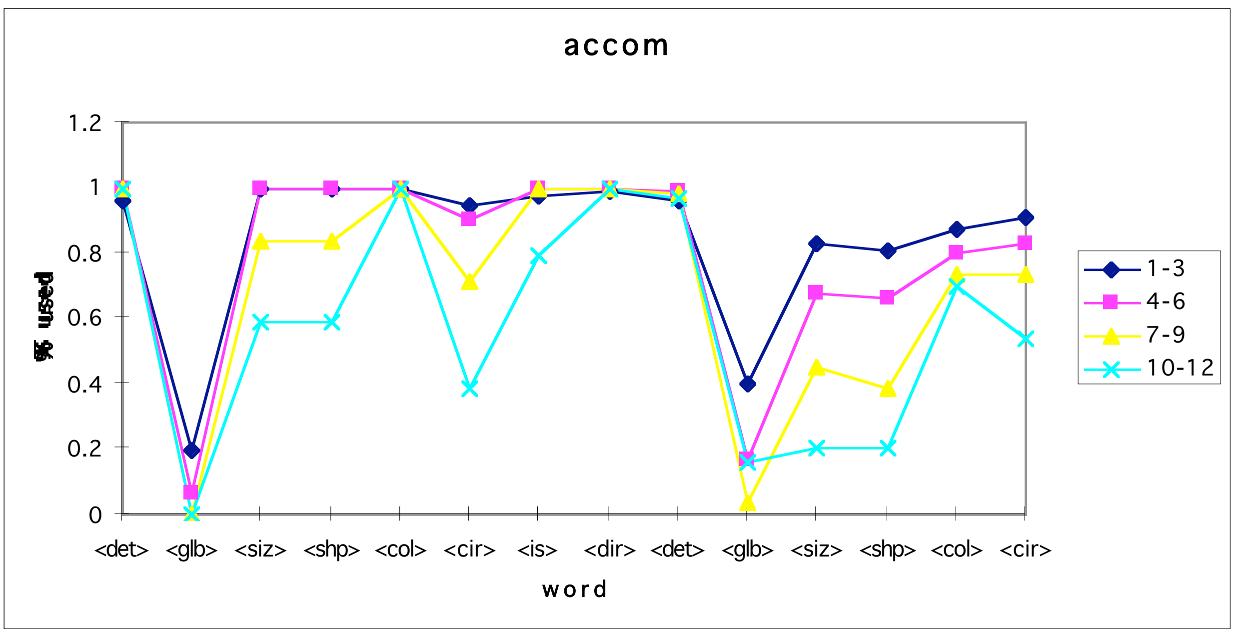


Figure 3.12b: Percent use of word categories in accommodation condition

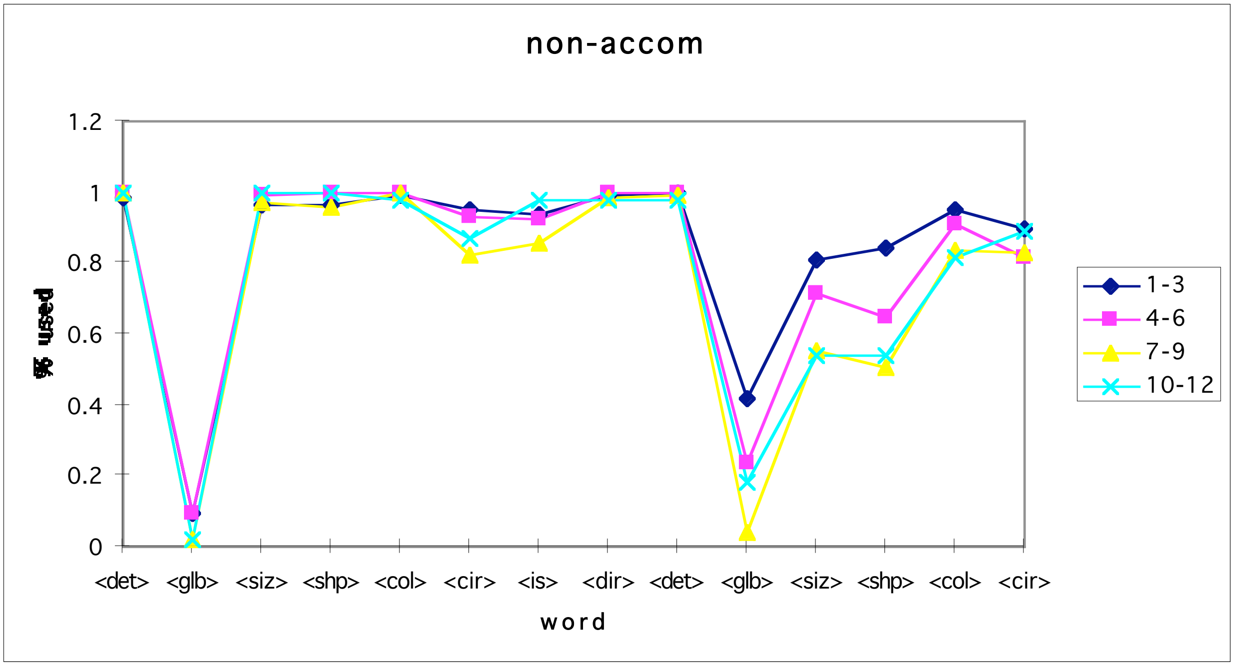


Figure 3.12c: Percent use of word categories in non-accommodation condition

Figure 3.13 shows a clearer comparison in word use for final performance (problems ten, eleven, and twelve) in the accommodation and non-accommodation conditions. The size, shape, and object name words were more likely to be skipped in the accommodation condition than in the non-accommodation condition. These are the same words that the models can accommodate to in the accommodating condition and against in the non-accommodation condition. Color is also a word that can be accommodated to and against, but it is preferred by the subjects to use to describe the new object and so is used more often than the size and shape words. Most messages describe a new object in the first half of the sentence in relation to the shared common object in the second half of the sentence, so words describing the common object can actually be skipped since the details of the common object are already a part of common knowledge

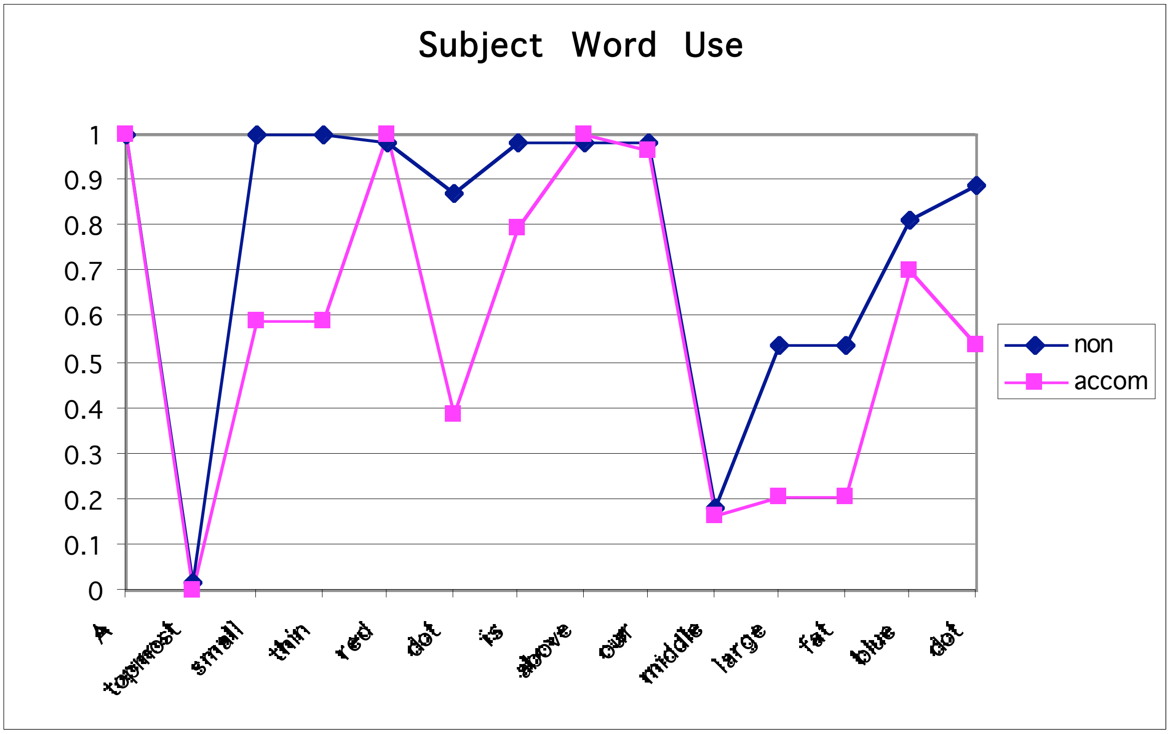


Figure 3.13: Word category use in problems 10-12 in accommodation and non-accommodation conditions

In order to see if subjects are following the output/input coordination principle by matching the message length of their partners, the within-pair difference in message length is compared to the between-pair difference in Figures 3.14a, b, and c. The lower within-pair difference shows that subjects are using the same message length as their partner. The ACT-R model used syntactic information contained in previous goals to accept input to create templates for producing output, therefore following the output/input coordination principle. The result of this coordination is shown as decreased message length in a previous graph and as a lower within-pair difference in message length in the second graph.

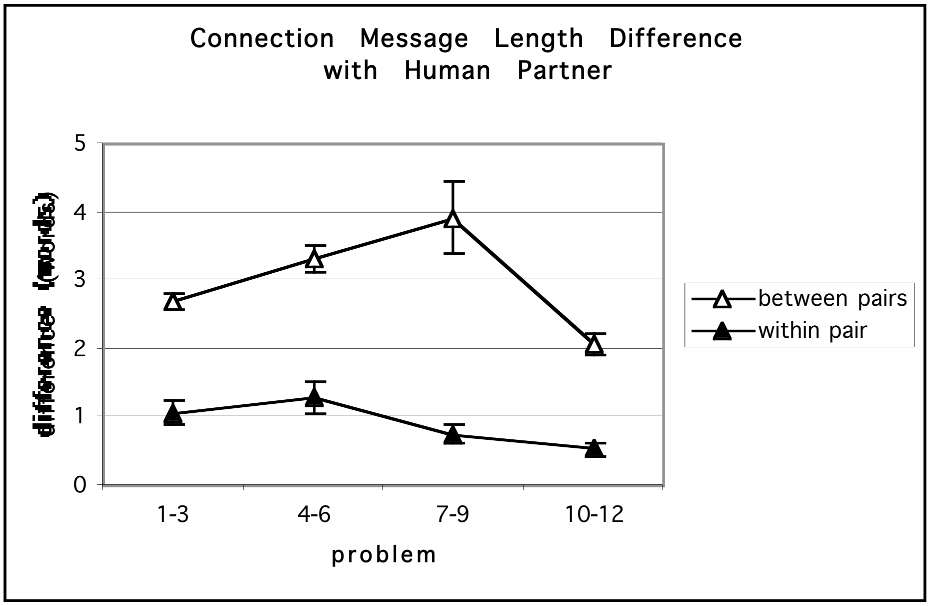


Figure 3.14a: Message length difference in restricted interface condition

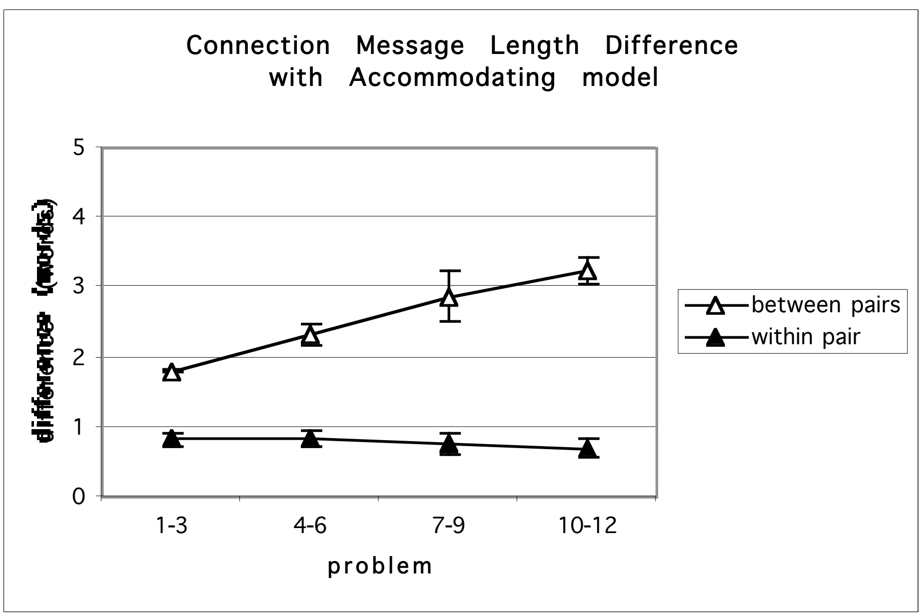


Figure 3.14b: Message length difference in accommodation condition

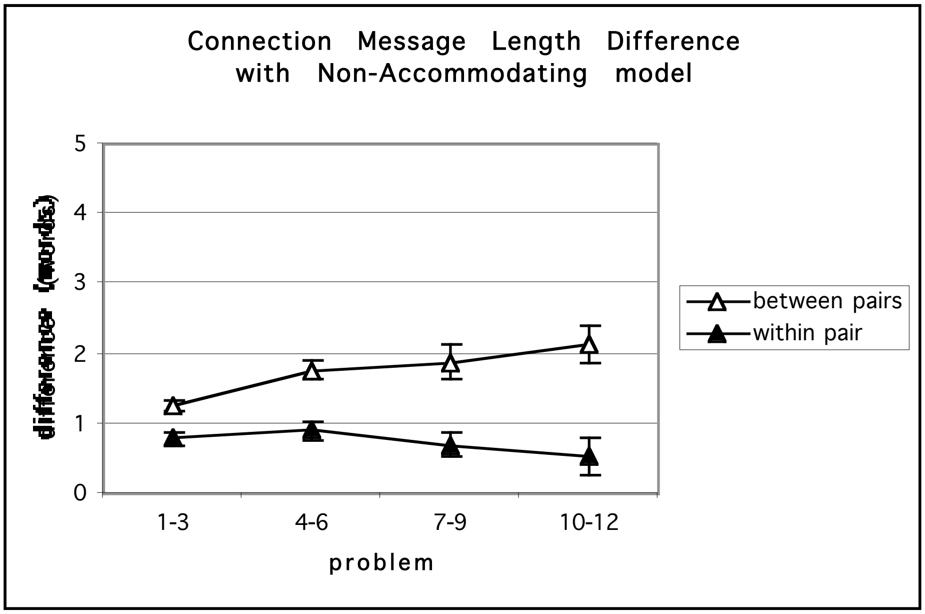


Figure 3.14c: Message length difference in non-accommodation condition

To make the differences between conditions explicit, the following are example transcripts from problem twelve from each of the conditions:

Restricted Subjects

A: "red left of our. green above that red. green right of our. see four"
B: "green above our. green below our. six"

Subject with Accommodating Model

A: "a blue above our"
B: "a green left of our"
A: "a blue right of our "
B: "a red below our"
A: "ok "
B: "a blue above the leftmost green"

Subject with Non-Accommodating Model

A: "a medium round green object is right of our object "
B: "a small thin red shape is west of our shape"
A: "a medium round green object is above the leftmost small thin red object "
B: "a medium round green shape is north of our shape"
A: "are you done "
B: "a medium round green shape is south of our shape"
A: "ok "
B: "i'm done with my objects"

Confirmation Typing Time

In addition to the connection of objects, significant time was also spent typing messages discussing confirmation actions and number of objects. Figure 3.15 shows that subjects in the non-accommodation condition spent significantly more time typing messages concerning confirmation actions than subjects in the accommodation condition ($t(43)=2.87$, $p<.002$; $t(39)=3.36$, $p<.001$; $t(24)=3.27$, $p<.001$; $t(18)=2.462$, $p<.05$).

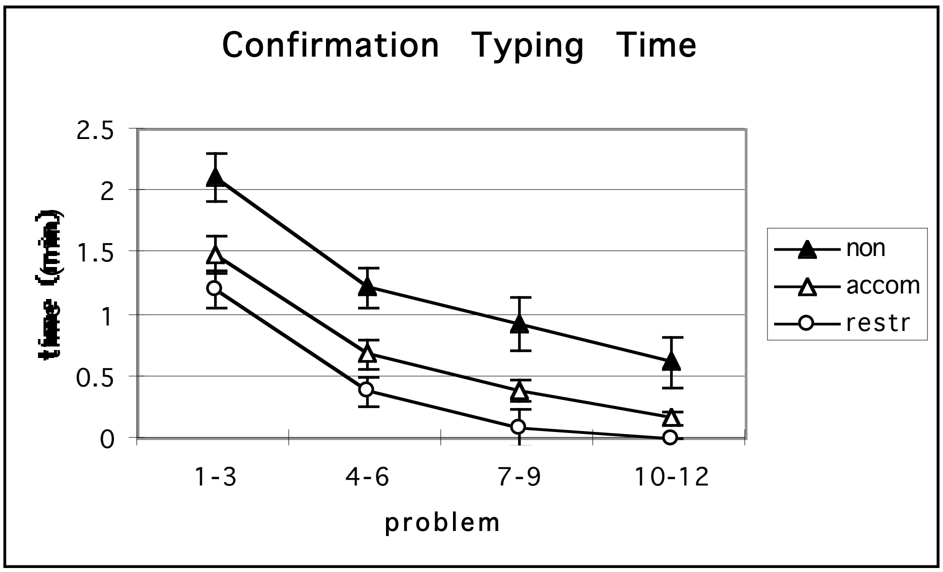


Figure 3.15: Time spent typing messages concerning confirmation of objects

The total time spent typing messages on a topic depends on two factors: how many messages are typed and the time spent typing each message. Figure 3.16 shows that subjects typed more messages concerning confirmation actions in the non-accommodation condition than in the accommodation condition ($t(43)=3.10$, $p<.005$; $t(39)=2.31$, $p<.05$; $t(24)=2.10$, $p<.05$; $t(18)=2.29$, $p<.05$). Since there are six objects to confirm in each problem, one might expect that six messages concerning confirmation actions are needed to solve the problem, but this is not the case. Once patterns of confirmation are established (for example, left to right starting with the top row), some pairs did not need to send messages about confirmations but simply confirmed them in a previously established order. The ACT-R models also had the ability to confirm without messages by referring to previous patterns, but only followed the lead of their human partner and did not skip a confirming message until their partner skipped a message.

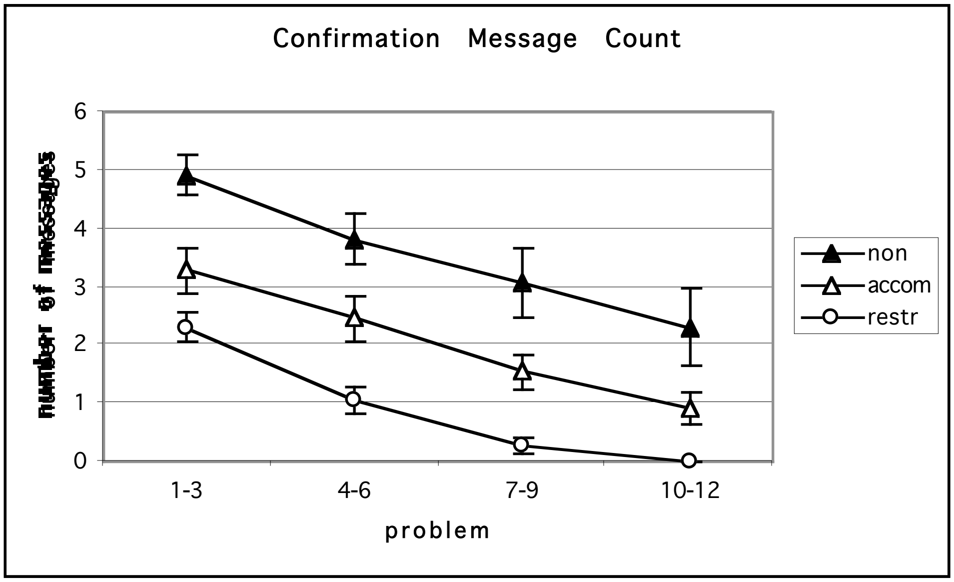


Figure 3.16: Number of messages concerning confirmation actions

Figure 3.17 shows that there is no significant difference in the time spent typing messages concerning confirmation actions between the accommodation and non-accommodation conditions condition (t(43)=0.62; t(39)=0.21; t(24)=1.32,; t(18)=1.23).

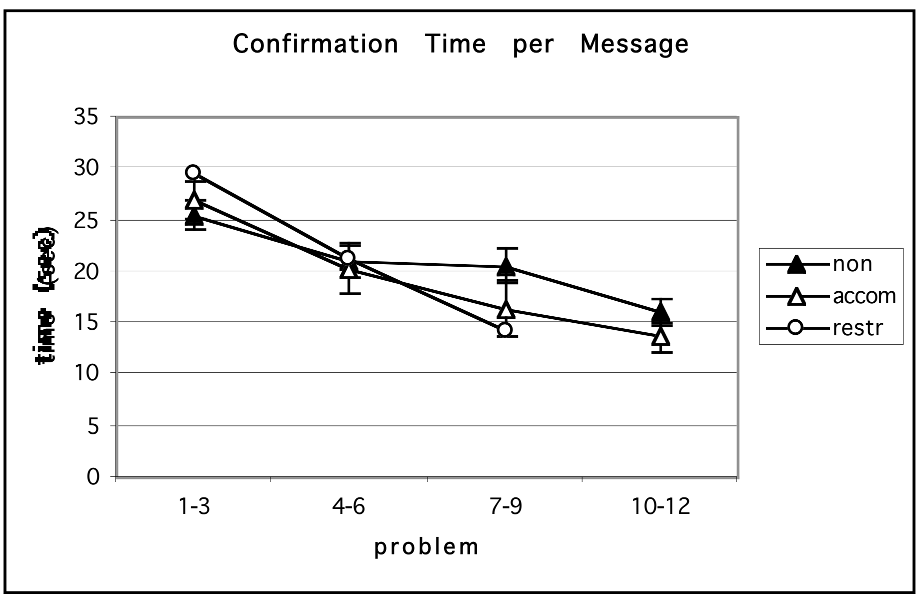


Figure 3.17: Time per message concerning confirmation actions

Number Typing Time

Figure 3.18 shows that there was no significant difference in the time spent on messages concerning the number of objects between the accommodation and non-accommodation conditions (t(43)=0.61; t(39)=0.20; t(24)=1.53; t(18)=0.13).

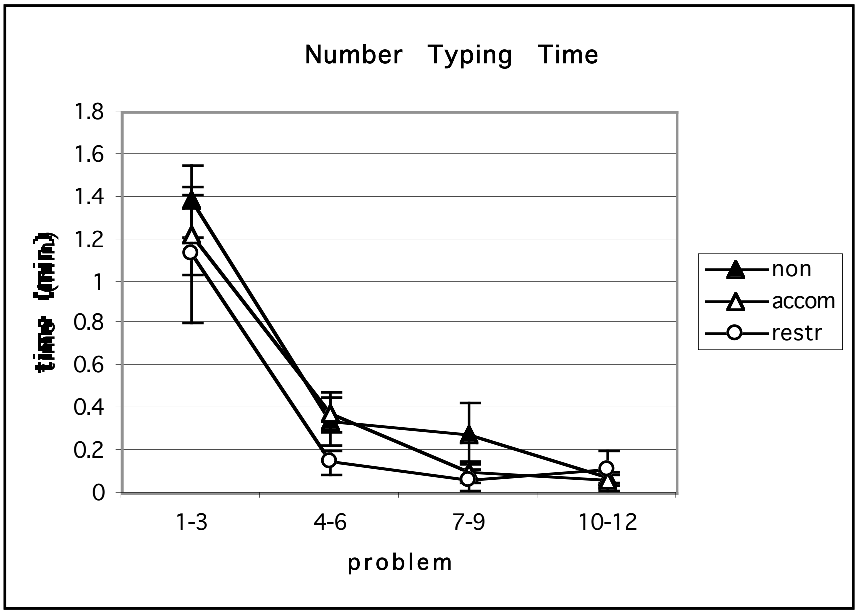


Figure 3.18: Time spent typing messages concerning numbers of objects

After the experiment, subjects were asked to rate their partner and themselves in terms of cooperativeness and ability to do the task on a Likert-type seven-point scale. Figure 3.19 shows that subjects rated their partner (actually an ACT-R model) as less cooperative (t(43)=1.74, p<.05) and as having less ability (t(43)=2.08, p<.05) in the non-accommodation condition than in the accommodation condition. Ratings of self-coopertiveness and self-ability were lower in the non-accommodation condition than in the accommodation condition, but not significantly so.

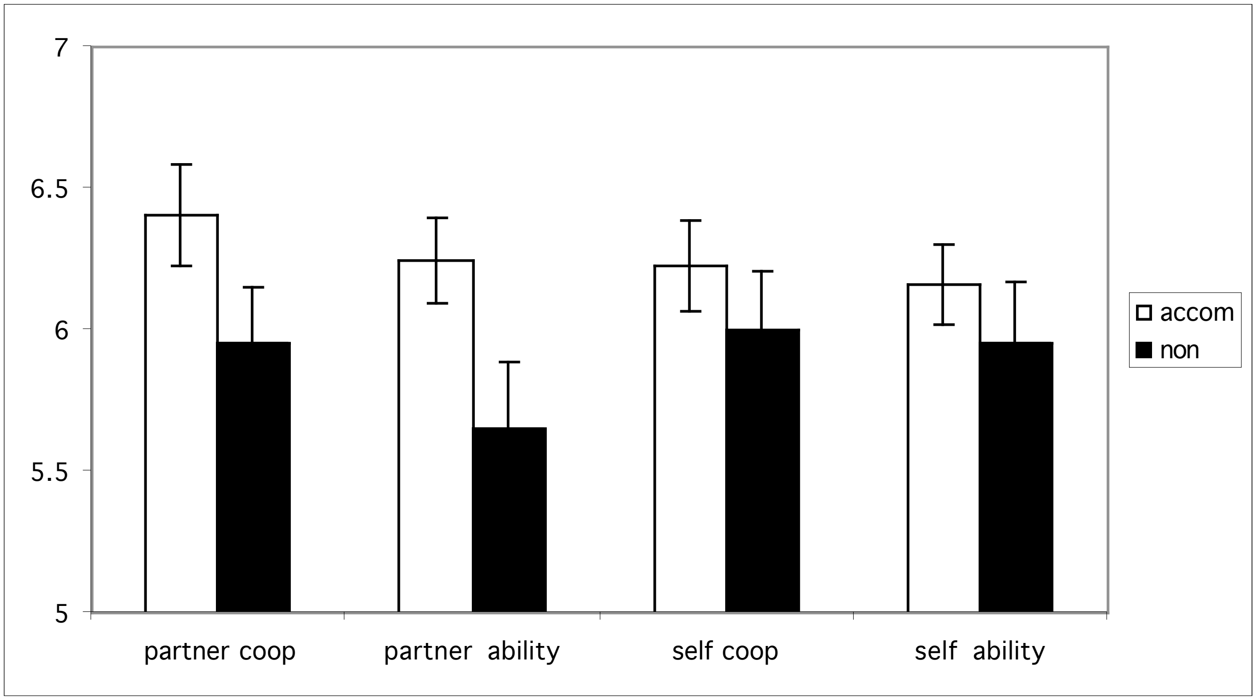


Figure 3.19: Survey ratings of cooperativeness and ability for partner and self

The ratings of partner cooperativeness were used to split subjects into those rating ACT-R models high in cooperativeness and those rating the models low in cooperativeness. Since the median of these ratings was 6, the high category consisted of ratings of 7 and the low category consisted of ratings 5 or below. Since about one third of the subjects used a rating of 6, the power of any statistical analyses was reduced, and so only qualitative results are reported. Figure 3.20 shows the time to solve a problem for subjects using high and low cooperativeness ratings interacting with accommodating and non-accommodating models. Subjects who had low ratings for accommodating models did not finish more than six problems. For the first nine problems solved, subjects who had low ratings of the models took longer to solve problems than subjects who had high ratings. For problems ten through twelve, no subject interacting with an accommodating model used a low rating and only one subject with a non-accommodating model used a low rating.

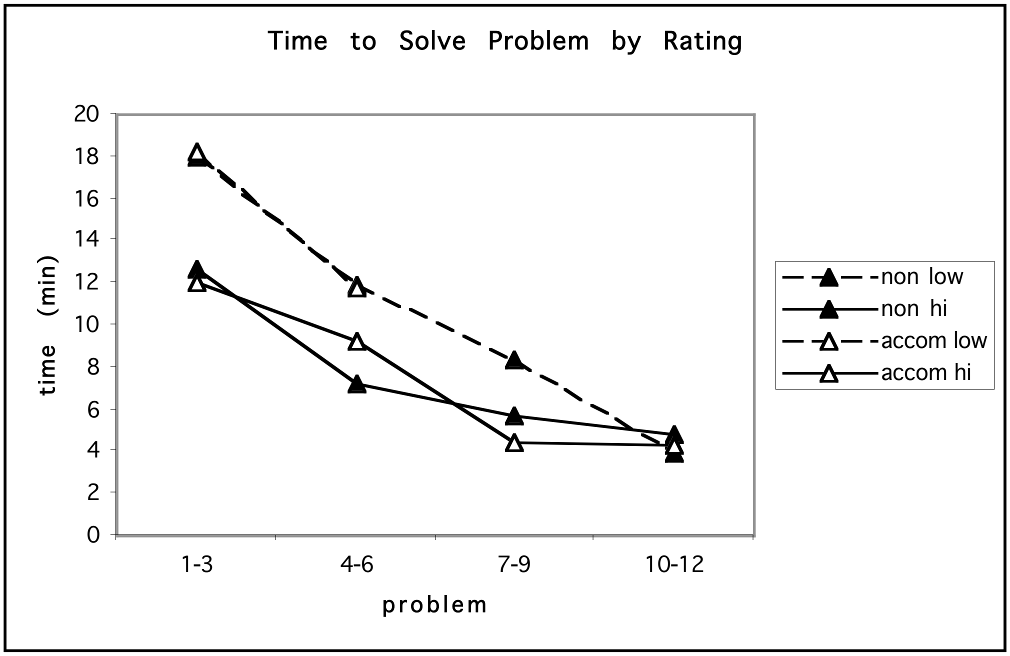


Figure 3.20: Time to solve problem for low and high cooperativeness ratings

Summary

As suggested by the pilot study, the restricted interface does not appear to unduly hinder performance in the task. After an adaptation period, errors and the time to solve problems not significantly differ between the restricted and unrestricted interface (Figures 3.1 and 3.2). Also, subject's performance with the accommodating model does not greatly differ from performance with other humans. Except for higher errors in the second group of three problems, errors and time to solve problems did not significantly differ between subjects interacting with an accommodating model and subjects interacting with other humans (Figures 3.1 and 3.2). The strategy of both the accommodating and non-accommodating models to accommodate to syntax (measured as a low difference in message length) was also shown to occur in subjects interacting with other humans (Figures 3.14a, b, and c).

The non-accommodating model behaved exactly the same as the accommodating model except for choosing not to match certain words that its partner used. Therefore, any differences in the accommodation and non-accommodation conditions are due solely to the subject's reaction to non-accommodating behavior. No differences in error were found between the non-accommodation and accommodation conditions. This is not surprising, since the non-accommodating words chosen were functionally equivalent to the partner's words. However, subjects interacting with non-accommodating models for more than six problems took significantly longer to solve problems than subjects interacting with accommodating models. This was shown to be the result of subjects interacting with accommodating models choosing to use shorter messages to describe their objects and choosing not to type messages describing confirmation actions, but instead just using previously-established confirmation patterns.

Since these two behaviors of skipping words describing objects and skipping messages describing confirmation actions were responsible for the differences found in the accommodation and non-accommodation conditions, they will be the focus of a third "human" model representing human performance. The difference in word and message skipping between the conditions will be explained with rules for efficient problem-solving (word and message skipping) that are sensitive to cooperative behavior of partners (word accommodation). The next chapter will describe this model along with the accommodating and non-accommodating models.

Chapter 4: ACT-R & Communication

Any interactive model of communication must be able to establish mutual knowledge, interpret the communicative intent of a partner, follow basic communicative obligations, and use communication to further some goal. These abilities have been the focus of a number of lines of research in the communication literature (Clark & Schaefer, 1989; Core & Allen, 1997; Poesio & Traum, 1998; Traum & Allen, 1994) and the ACT-R model of communication presented in this dissertation is guided by theories in this literature. ACT-R itself is a method for describing human cognition in terms of facts and rules, but the content of the facts and rules used in communication must be guided by current theories of communication. This model of communication was used to test the effect of accommodation (the matching of partner vocabulary) on communication efficiency by having two ACT-R models created from the basic communication model, one accommodating to word use and one non-accommodating[1].

Common Ground

One goal of communication is to establish mutual knowledge. Clark and Schaefer (1989) proposed that a speaker cannot believe their contribution is part of mutual knowledge until the listener gives evidence of understanding the contribution. The ACT-R model of communication is organized around this principle, which can be naturally implemented in terms of ACT-R's fact-learning theory. The ACT-R theory states that new declarative facts are created from successfully completed goals; for example, the goal to find the sum of three and four can be successfully completed by using fingers to count to seven, and later this goal can be retrieved as a fact. So given a speaker's goal to present information, the successful completion of the goal (as shown by acknowledgement from the listener) allows the goal to become a fact in the speaker's knowledge base that both the speaker and the listener understand and accept the information. From the listener's point of view, a goal to accept information from the speaker can be successfully

[1] The ACT-R code for these models can be found by following the “published models” link from the ACT-R home page http://act.psy.cmu.edu.

completed by acknowledging the speaker, which allows the goal to become a fact in the listener's knowledge base that both the listener and the speaker understand and accept the information.

The theory of Clark and Schaefer allows for this process to be recursive, since acknowledgement from the listener can be considered to be a new presentation, which then requires another acknowledgement. An example given by Clark and Schaefer is the following:

> A: Well what shall we do about *this* boy then?
>   B: Duveen?
>   A: Mm.
> B: Well I propose to *write* saying I'm very sorry...

The first utterance presents information to be evaluated by B, but the second utterance delays acknowledgement with another utterance to be evaluated by A. A acknowledges the utterance of B with the third utterance, and B acknowledges A's first utterance with the fourth utterance. The ACT-R model of communication is not designed to process recursive conversations but instead uses a fixed subgoal structure (this is because the restricted interface in the experiment is designed to encourage subjects to solve problems with simple messages and not engage in extensive subgoaling behavior). A goal to present information contains a subgoal to wait for a response to see if that information was accepted by the model's partner (any interpretable response besides "No" or "I don't know"). This response is not considered recursively as a new presentation, but evaluated as either a positive or negative acceptance by the partner. A goal to accept information contains an internal decision process to see if the information can be accepted (any interpretable response). The ACT-R model does not start a new subdialog with the partner to get further information for the decision to accept. The following subgoal representations show the difference between the recursive subgoaling allowed by Clark and Schaefer and the fixed subgoaling used by the model:

```
Recursive                        Fixed

present                          present
    understand                       understand
        present                          (check-acceptance-by-self)
            understand ...           (check-acceptance-by-other)
                                 present...
```

Dialogue Acts

Clark and Schaefer (1989) also claimed that a contribution to common ground is done with an illocutionary act such as making an assertion, asking a question, etc. A set of widely accepted acts comes from the Discourse Resource Initiative (DRI) (Core & Allen, 1997), developed by an international team of dialogue researchers. These dialogue acts represent ways to introduce new information (forward-looking acts) and ways to respond to previous dialogue acts (backward-looking acts) and can be seen in Table 4.1. For example, asking a question would be an example of an Info Request which is a Directive Forward-looking act because the speaker is introducing a new topic which is asking the listener to do something (give information). If the listener couldn't hear the question, the reply "What?" would be an example of a Signal Non-Understanding which is a Understanding Backward-looking act because the listener is responding to a previous act and is signaling that the act was not understood.

Forward-looking Acts

- Statement
    - Assert
    - Reassert
    - Other Statement
- Influencing Addressee Future Action
    - Open option
- Directive
    - Info Request
    - Action Directive
- Committing Speaker Future Action
    - Offer
    - Commit
- Performative

Backward-looking Acts

- Agreement
    - Accept
    - Accept Part
        - Maybe
        - Reject Part
        - Reject
    - Hold
- Understanding
    - Signal Non-Understanding
    - Signal Understanding
        - Acknowledge
        - Repeat-Rephrase
        - Completion
    - Correct Misspeaking
- Answer

Table 4.1: DRI dialogue acts

The way these dialogue acts relate to the beliefs and intentions of individuals involved in communication is given by Poesio and Traum (1998) as an axiomatisation of the DRI dialogue acts in terms of mental attitudes of individuals, where reactions to certain dialogue acts can make certain changes in beliefs and intentions in common ground. Some of these effects can be seen in Table 4.2.

A's Dialogue Act	B's Action	Change to common ground
Any	recognize	B is obligated to produce an Understanding-act
Directive(A,B,x)	recognize	B is obligated to Address the directive
	accept	B is obligated to perform x
Statement (A,B,K)	acknowledge	A is socially committed to B to K being true
	accept	B is socially committed to A to K being true

Table 4.2: Dialogue Act effects on common ground (Poesio & Traum, 1998)

Since common ground is declarative knowledge, since productions are the only way to change declarative knowledge in the ACT-R theory, and since Poesio and Traum suggest reactions to dialogue acts that change common ground, the reactions to dialogue acts are represented as productions in the ACT-R communication model. The goals of these productions contain knowledge of dialogue acts and public beliefs, intentions, and social commitments, and when completed these goals become part of declarative memory and can become part of common ground.

When trying to understanding a message from a partner, the model parses the message one word at a time and tries to build up the semantic meaning of the message, in much the same way as Lewis (1999). This semantic meaning includes the dialogue action represented by the message and any semantic relationships that may be present. For example, after seeing the phrase "the red object is above..." the model has the following semantic representation:

```
UND32                              CIRC23
     isa UNDERSTAND                isa CIRCLE
     action STATE                       color RED
     topic PAIRS                        size NIL
     arg1 CIRC23                        shape NIL
     relation ABOVE-REL
     arg2 NIL
```

The restricted interface was designed to help this process by having unique words corresponding to dialogue acts and a simple phrase structure to parse. While a semantic meaning is being determined, the syntactic list structure (similar to those used in Lewis (1999) and Anderson and Matessa (1997)) is also being built. Since all message templates only allow the construction of simple sentences, a flat list structure is used without hierarchy or recursion. For example, after the phrase "the red object..." is seen by the model, the following representation is built:

```
                        PHRASE10
                          isa PHRASE
                          part-of NIL
                          head NIL
                          position NIL

PHRASE11                PHRASE12                PHRASE13
  isa PHRASE              isa PHRASE              isa PHRASE
  part-of PHRASE10        part-of PHRASE10        part-of PHRASE10
  head THE                head RED                head OBJECT
  position ONE            position TWO            position THREE
```

Communicative Obligations

When relating dialogue acts to beliefs and intentions, Poesio and Traum (1998) use the term "obligation" as defined in Traum and Allen (1994) to mean a social pressure to do what a cooperative communicative agent should do according to some set of norms. This is not as strong a cooperation as other theories such as Joint Intentions (Cohen & Levesque, 1991) or Shared Plans (Grosz & Sidner, 1990) would suggest. As Traum and Allen point out:

> Consider a stranger approaching an agent and asking, "Do you have the time?" It is unlikely that there is a joint intention or shared plan, as they have never met before. From a purely strategic point of view, the agent may have no interest in whether the stranger's goals are met. Yet, typically agents will still respond in such situations.

Communicative obligations in the ACT-R model consist of addressing queries for information and requests for actions, stating when a previous message is unclear, and waiting for message turns.

In a similar manner, there is a social pressure to solve the task assigned by the experimenter. Task obligations are then the appropriate actions needed to solve the task. In the current experiment they consist of stating and obtaining object information, confirming objects, and pressing the "done" button.

Goal Structure

In addition to goals to present and understand messages and to check the acceptance by self and by partner, the model also has goals to solve a problem, decide on actions, see if certain conditions of the problem are true, count the number of objects, ground information to be presented globally, phrase a received message, determine communicative obligations, ground received references to objects and locations on the screen, determine left to right order, and stall for a specified amount of time. These goals and their subgoal relations, along with the number of productions using each goal with examples, can be seen in Table 4.3.

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, ...)

Table 4.3: Goals of ACT-R communication model

The solve goal involves successfully completing the problems in the experiment. The decide goal involves a choice of action given a problem state. The seeif goal involves the determination of particular states of the problem. The count goal involves determining the number of objects on the screen. The global-ground goal involves determining the relative location of a particular object. The present goal involves the passing of a message to a partner. The understand goal involves the classification of a partner's message and the grounding of its references. The phrase goal involves the building of syntactic information from a message. The obligation goal involves the determination of social pressures in response to a message. The ground goal involves the mapping of references in a message to objects on the screen. The l2r goal involves the determination of left to right ordering of objects. The stall goal involves letting time pass before sending a message in order to imitate human timing performance.

For example, the productions used with the decide goal include decisions to request that the partner confirm an object (decide-request-confirm), to let the partner know their message was not understood (decide-assess-huh), and to answer a question concerning how many objects are on the screen (decide-find-count).

The basic communication model has 338 productions, with an additional 62 productions for non-accommodating behavior. Table 4.4 presents a trace of the non-accommodating model as it works with a human to solve a problem. Information about the location of objects is discussed in sections A1-A10. In section A11, the model lets its partner know that has no new objects to discuss. In section A12 the human partner asks the model to confirm a particular object. In sections A13-A15, the model confirms the object and lets its partner know that it has done so. In section A16 the partner also confirms the object. In sections A17-A19 the model decides to wait for its partner before confirming any more objects because no pattern of confirming has yet been established. In section A20 the partner suggests a pattern of confirming by rows. In sections A21-A23 the model lets its partner know that it accepts the suggestion. In section A24, the partner confirms a second object without typing a message. In sections A25-A27 the model confirms the next object as determined by the row pattern. In sections A28-A43 the model and its partner continue to confirm objects without typing messages. In section A44 the partner states he is done with the problem. In sections A45-A47 the model lets its partner know that it is also done. The model sees that the partner has pressed the DONE button in section A48.

A1) Wait for first message and understand it:

```
 Cycle 0  Start-Wait
 Cycle 1  Wait
 Cycle 2  Wait
 Cycle 3  Wait
 Cycle 56  Wait
(A MEDIUM ROUND GREEN OBJECT IS ABOVE
OUR MEDIUM ROUND GREEN OBJECT)
 Cycle 57  See-First-Message
 (You said something.  I'll try to
understand.)
    Cycle 58  Parse-Current
       Cycle 59  Parse-Lex
*A* PREDOT-ABSTR
    Cycle 60  Read-Next
    Cycle 61  Find-Size-1st
    Cycle 62  Parse-Current
       Cycle 63  Parse-Lex
*MEDIUM* SIZE
    Cycle 64  Read-Next
    Cycle 65  Find-Shape
    Cycle 66  Parse-Current
       Cycle 67  Parse-Lex
*ROUND* SHAPE-ABSTR
    Cycle 68  Read-Next
    Cycle 69  Find-Color
    Cycle 70  Parse-Current
       Cycle 71  Parse-Lex
*GREEN* COLOR
    Cycle 72  Read-Next
    Cycle 73  Find-Dotname
    Cycle 74  Parse-Current
       Cycle 75  Parse-Lex
*OBJECT* DOT-NAME
    Cycle 76  Read-Next
    Cycle 77  Parse-Current
       Cycle 78  Parse-Lex
*IS* IS-ABSTR
    Cycle 79  Read-Next
    Cycle 80  Find-Direction
    Cycle 81  Parse-Current
       Cycle 82  Parse-Lex
*ABOVE* DIRECTION
    Cycle 83  Read-Next
    Cycle 84  Find-2nd-Common-1st
```

```
    Cycle 85  Find-Common-Phrase2
    Cycle 86  Parse-Current
       Cycle 87  Parse-Lex
*OUR* PREDOT-ABSTR
    Cycle 88  Read-Next
    Cycle 89  Find-2nd-Size
    Cycle 90  Parse-Current
       Cycle 91  Parse-Lex
*MEDIUM* SIZE
    Cycle 92  Read-Next
    Cycle 93  Find-2nd-Shape
    Cycle 94  Parse-Current
       Cycle 95  Parse-Lex
*ROUND* SHAPE-ABSTR
    Cycle 96  Read-Next
    Cycle 97  Find-2nd-Color
    Cycle 98  Parse-Current
       Cycle 99  Parse-Lex
*GREEN* COLOR
    Cycle 100  Read-Next
    Cycle 101  Parse-Current
       Cycle 102  Parse-Lex
*OBJECT* DOT-NAME
    Cycle 103  Read-Done-No-Action
    Cycle 104  Read-Done-No-Topic
    Cycle 105  Read-Done
    Cycle 107  Ground-Common2
    Cycle 108  Ground-Above
       Cycle 109  Get-Pos-Old
    Cycle 110  Ground-Done
    Cycle 111  Understand-Accept
```

A2) Update state of problem:

```
 Cycle 112  Check-Accept
 Cycle 113  Done-Update-New
 Cycle 114  Find-Comm-Obligation
    Cycle 115  Understand-Statement
    (You are trying to make me
believe Prop11.  I'll try.)
 Cycle 116  Task-Start-Discussing
 Cycle 117  Task-Start-Other-
Discussing
 Cycle 118  Done-Task-Status
 Cycle 119  Task-State-Oblig
 Cycle 120  Update-Null-Other-Conf-
Status
 Cycle 121  Update-Null-Self-Conf-
Status
 Cycle 122  Update-Null-Conf-Plan
 Cycle 123  Update-Null-Old-Present
```

A3) Decide on action and present message:

```
 Cycle 124  Decide-Action
    Cycle 125  Decide-Left-Common
    Cycle 126  Decide-Done
 Cycle 127  Topic-Circle
 Cycle 128  Topic-End
 Cycle 129  Do-State-Pairs
    Cycle 130  Pick-Common-Template2
    Cycle 131  Get-Next
    Cycle 132  Fill-Any
    Cycle 133  Get-Next
    Cycle 134  Fill-Size
    Cycle 135  Get-Next
    Cycle 136  Fill-Shape
    Cycle 137  Get-Next
    Cycle 138  Fill-Color
    Cycle 139  Get-Next
    Cycle 140  Fill-Dot
    Cycle 141  Get-Next
    Cycle 142  Fill-Is
    Cycle 143  Get-Next
    Cycle 144  Fill-Direction
    Cycle 145  Get-Next
    Cycle 146  Fill-Predot-Common2
    Cycle 147  Get-Next
    Cycle 148  Fill-Size2
    Cycle 149  Get-Next
    Cycle 150  Fill-Shape2
    Cycle 151  Get-Next
    Cycle 152  Fill-Color2
    Cycle 153  Get-Next
    Cycle 154  Fill-Dot
    Cycle 155  Get-Done
    Cycle 156  Type-Next
** A **
    Cycle 157  Type-Next
** SMALL **
    Cycle 158  Type-Next
** THIN **
    Cycle 159  Type-Next
** RED **
    Cycle 160  Type-Next
** SHAPE **
    Cycle 161  Type-Next
** IS **
    Cycle 162  Type-Next
** WEST OF **
    Cycle 163  Type-Next
** OUR **
    Cycle 164  Type-Next
** MEDIUM **
    Cycle 165  Type-Next
** ROUND **
    Cycle 166  Type-Next
** GREEN **
    Cycle 167  Type-Next
```

```
** SHAPE **
    Cycle 168  Type-Stall
       Cycle 169  Stall-First
       Cycle 170  Stall
       Cycle 171  Stall
       Cycle 172  Stall
       Cycle 218  Stall
       Cycle 219  Stall-Done
    Cycle 220  Type-Done
```

A4) Inside presentation goal, wait for next message and understand it:

```
    Cycle 221  Present-Start-Wait
    Cycle 222  Present-Wait
    Cycle 223  Present-Wait
    Cycle 245  Present-Wait
(A SMALL THIN RED SHAPE IS LEFT OF
OUR TOPMOST MEDIUM ROUND GREEN
OBJECT)
    Cycle 246  Present-See-Message
    (You said something.  I'll try to
understand.)
       Cycle 247  Parse-Current
          Cycle 248  Parse-Lex
*A* PREDOT-ABSTR
       Cycle 249  Read-Next
       Cycle 250  Find-Size-1st
       Cycle 251  Parse-Current
          Cycle 252  Parse-Lex
*SMALL* SIZE
       Cycle 253  Read-Next
       Cycle 254  Find-Shape
       Cycle 255  Parse-Current
          Cycle 256  Parse-Lex
*THIN* SHAPE-ABSTR
       Cycle 257  Read-Next
       Cycle 258  Find-Color
       Cycle 259  Parse-Current
          Cycle 260  Parse-Lex
*RED* COLOR
       Cycle 261  Read-Next
       Cycle 262  Find-Dotname
       Cycle 263  Parse-Current
          Cycle 264  Parse-Lex
*SHAPE* DOT-NAME
       Cycle 265  Read-Next
       Cycle 266  Parse-Current
          Cycle 267  Parse-Lex
*IS* IS-ABSTR
       Cycle 268  Read-Next
       Cycle 269  Find-Direction
       Cycle 270  Parse-Current
          Cycle 271  Parse-Lex
*LEFT OF* DIRECTION
       Cycle 272  Read-Next
       Cycle 273  Find-2nd-Common-1st
       Cycle 274  Find-Common-Phrase2
       Cycle 275  Parse-Current
          Cycle 276  Parse-Lex
*OUR* PREDOT-ABSTR
       Cycle 277  Read-Next
       Cycle 278  Find-Y2
       Cycle 279  Delete-2nd-Common
       Cycle 280  Find-Global-Y-
Phrase2
       Cycle 281  Parse-Current
          Cycle 282  Parse-Lex
*TOPMOST* TOPMOST-REL
       Cycle 283  Read-Next
       Cycle 284  Find-2nd-Size
       Cycle 285  Parse-Current
          Cycle 286  Parse-Lex
*MEDIUM* SIZE
       Cycle 287  Read-Next
       Cycle 288  Find-2nd-Shape
       Cycle 289  Parse-Current
          Cycle 290  Parse-Lex
*ROUND* SHAPE-ABSTR
       Cycle 291  Read-Next
       Cycle 292  Find-2nd-Color
       Cycle 293  Parse-Current
          Cycle 294  Parse-Lex
*GREEN* COLOR
       Cycle 295  Read-Next
       Cycle 296  Find-Dotname
       Cycle 297  Parse-Current
          Cycle 298  Parse-Lex
*OBJECT* DOT-NAME
       Cycle 299  Read-Done-No-Action
       Cycle 300  Read-Done-No-Topic
       Cycle 301  Read-Done
       Cycle 303  Ground-Topmost-
Color2
          Cycle 304  Get-Topmost-
Color
       Cycle 305  Ground-Val2
       Cycle 306  Ground-Left
          Cycle 307  Get-Pos-New
       Cycle 308  Ground-Link-Left
       Cycle 309  Ground-Done
       Cycle 310  Understand-Accept
```

A5) Check if presentation was accepted and complete presentation goal:

```
    Cycle 311  Present-Accept
```

A6) Update state of problem:

```
 Cycle 312  Update-Old-State-Circles
 Cycle 313  Update-Old-Circle
 Cycle 314  Done-Update-Old-Circle
 Cycle 315  Check-Accept
 Cycle 316  Done-Update-New
 Cycle 317  Find-Comm-Obligation
    Cycle 318  Understand-Statement
    (You are trying to make me
believe Prop36.  I'll try.)
 Cycle 319  Done-Task-Status
 Cycle 320  Task-State-Oblig
```

A7) Decide on action and present message:

```
 Cycle 321  Decide-Action
    Cycle 322  Decide-Below-Common
    Cycle 323  Decide-Done
 Cycle 324  Topic-Circle
 Cycle 325  Topic-End
 Cycle 326  Do-State-Pairs
    Cycle 327  Pick-Common-Template2
    Cycle 328  Get-Next
    Cycle 329  Fill-Any
    Cycle 330  Get-Next
    Cycle 331  Fill-Size
    Cycle 332  Get-Next
    Cycle 333  Fill-Shape
    Cycle 334  Get-Next
    Cycle 335  Fill-Color
    Cycle 336  Get-Next
    Cycle 337  Fill-Dot
    Cycle 338  Get-Next
    Cycle 339  Fill-Is
    Cycle 340  Get-Next
    Cycle 341  Fill-Direction
    Cycle 342  Get-Next
    Cycle 343  Fill-Predot-Common2
    Cycle 344  Get-Next
    Cycle 345  Fill-Size2
    Cycle 346  Get-Next
    Cycle 347  Fill-Shape2
    Cycle 348  Get-Next
    Cycle 349  Fill-Color2
    Cycle 350  Get-Next
    Cycle 351  Fill-Dot
    Cycle 352  Get-Done
    Cycle 353  Type-Next
** A **
    Cycle 354  Type-Next
** LARGE **
    Cycle 355  Type-Next
** FAT **
    Cycle 356  Type-Next
** BLUE **
    Cycle 357  Type-Next
** SHAPE **
    Cycle 358  Type-Next
** IS **
    Cycle 359  Type-Next
** SOUTH OF **
    Cycle 360  Type-Next
** OUR **
    Cycle 361  Type-Next
** MEDIUM **
    Cycle 362  Type-Next
** ROUND **
    Cycle 363  Type-Next
** GREEN **
    Cycle 364  Type-Next
** SHAPE **
    Cycle 365  Type-Stall
       Cycle 366  Stall-First
       Cycle 367  Stall
       Cycle 368  Stall
       Cycle 369  Stall
       Cycle 415  Stall
       Cycle 416  Stall-Done
    Cycle 417  Type-Done
```

A8) Inside presentation goal, wait for next message and understand it:

```
    Cycle 418  Present-Start-Wait
    Cycle 419  Present-Wait
    Cycle 420  Present-Wait
    Cycle 421  Present-Wait
    Cycle 452  Present-Wait
(A LARGE FAT BLUE OBJECT IS RIGHT OF
OUR MEDIUM ROUND GREEN OBJECT)
    Cycle 453  Present-See-Message
    (You said something.  I'll try to
understand.)
       Cycle 454  Parse-Current
          Cycle 455  Parse-Lex
*A* PREDOT-ABSTR
       Cycle 456  Read-Next
       Cycle 457  Find-Size-1st
       Cycle 458  Parse-Current
          Cycle 459  Parse-Lex
*LARGE* SIZE
       Cycle 460  Read-Next
       Cycle 461  Find-Shape
       Cycle 462  Parse-Current
          Cycle 463  Parse-Lex
*FAT* SHAPE-ABSTR
       Cycle 464  Read-Next
       Cycle 465  Find-Color
       Cycle 466  Parse-Current
```

```
        Cycle 467  Parse-Lex
*BLUE* COLOR
     Cycle 468  Read-Next
     Cycle 469  Parse-Current
        Cycle 470  Parse-Lex
*OBJECT* DOT-NAME
     Cycle 471  Read-Next
     Cycle 472  Parse-Current
        Cycle 473  Parse-Lex
*IS* IS-ABSTR
     Cycle 474  Read-Next
     Cycle 475  Find-Direction
     Cycle 476  Parse-Current
        Cycle 477  Parse-Lex
*RIGHT OF* DIRECTION
     Cycle 478  Read-Next
     Cycle 479  Find-2nd-Common-1st
     Cycle 480  Find-Common-Phrase2
     Cycle 481  Parse-Current
        Cycle 482  Parse-Lex
*OUR* PREDOT-ABSTR
     Cycle 483  Read-Next
     Cycle 484  Find-2nd-Size
     Cycle 485  Parse-Current
        Cycle 486  Parse-Lex
*MEDIUM* SIZE
     Cycle 487  Read-Next
     Cycle 488  Find-2nd-Shape
     Cycle 489  Parse-Current
        Cycle 490  Parse-Lex
*ROUND* SHAPE-ABSTR
     Cycle 491  Read-Next
     Cycle 492  Find-2nd-Color
     Cycle 493  Parse-Current
        Cycle 494  Parse-Lex
*GREEN* COLOR
     Cycle 495  Read-Next
     Cycle 496  Parse-Current
        Cycle 497  Parse-Lex
*OBJECT* DOT-NAME
     Cycle 498  Read-Done-No-Action
     Cycle 499  Read-Done-No-Topic
     Cycle 500  Read-Done
     Cycle 502  Ground-Common2
     Cycle 503  Ground-Right
        Cycle 504  Get-Pos-New
     Cycle 505  Ground-Link-Right
     Cycle 506  Ground-Done
     Cycle 507  Understand-Accept
```

A9) Check if presentation was accepted and complete presentation goal:

```
    Cycle 508  Present-Accept
```

A10) Update state of problem:

```
 Cycle 509  Update-Old-State-Circles
 Cycle 510  Update-Old-Circle
 Cycle 511  Done-Update-Old-Circle
 Cycle 512  Check-Accept
 Cycle 513  Done-Update-New
 Cycle 514  Find-Comm-Obligation
    Cycle 515  Understand-Statement
    (You are trying to make me
believe Prop61.  I'll try.)
 Cycle 516  Done-Task-Status
 Cycle 517  Task-State-Oblig
```

A11) Decide on action and present message:

```
 Cycle 518  Decide-Action
    Cycle 519  Decide-Tell-Done-Dots
    Cycle 520  Decide-Done
 Cycle 521  Topic-End
 Cycle 522  Do-State-Done-Dots
    Cycle 523  Pick-Sample-Template-
Start
    Cycle 524  Read-Sample-First
    Cycle 525  Parse-Lex-Sample
       Cycle 526  Parse-Lex
*I'M* APOS-ABSTR
    Cycle 527  Read-Sample-Next
    Cycle 528  Parse-Lex-Sample
       Cycle 529  Parse-Lex
*DONE* DONE-ABSTR
    Cycle 530  Read-Sample-Next
    Cycle 531  Parse-Lex-Sample
       Cycle 532  Parse-No-Lex
*WITH* MISC
    Cycle 533  Read-Sample-Next
    Cycle 534  Parse-Lex-Sample
       Cycle 535  Parse-Lex
*MY* PREDOT-ABSTR
    Cycle 536  Read-Sample-Next
    Cycle 537  Parse-Lex-Sample
       Cycle 538  Parse-Lex
*PUZZLE* PHASE-ABSTR
    Cycle 539  Read-Sample-Done
    Cycle 540  Pick-Sample-Template-
End
    Cycle 541  Get-Next
    Cycle 542  Fill-Apos
    Cycle 543  Get-Next
    Cycle 544  Fill-Done
    Cycle 545  Get-Next
```

```
    Cycle 546  Fill-Misc
    Cycle 547  Get-Next
    Cycle 548  Fill-My-Dots
    Cycle 549  Get-Next
    Cycle 550  Fill-Phase
    Cycle 551  Get-Done
    Cycle 552  Type-Next
** I'M **
    Cycle 553  Type-Next
** COMPLETED **
    Cycle 554  Type-Next
** WITH **
    Cycle 555  Type-Next
** MY **
    Cycle 556  Type-Next
** DOTS **
    Cycle 557  Type-Stall
       Cycle 558  Stall-First
       Cycle 559  Stall
       Cycle 560  Stall
       Cycle 561  Stall
       Cycle 622  Stall
       Cycle 623  Stall-Done
    Cycle 624  Type-Done
```

A12) Inside presentation goal, wait for next message and understand it:

```
    Cycle 625  Present-Start-Wait
    Cycle 626  Present-Wait
    Cycle 627  Present-Wait
    Cycle 662  Present-Wait
(PLEASE CONFIRM THE TOPMOST SMALL
THIN RED OBJECT)
    Cycle 663  Present-See-Message
    (You said something.  I'll try to
understand.)

       Cycle 664  Parse-Current
          Cycle 665  Parse-No-Lex
*PLEASE* MISC
       Cycle 666  Read-Next
       Cycle 667  Find-Confirm-No-
Action
       Cycle 668  Parse-Current
          Cycle 669  Parse-Lex
*CONFIRM* CONF-ABSTR
       Cycle 670  Read-Next
       Cycle 671  Parse-Current
          Cycle 672  Parse-Lex
*THE* PREDOT-ABSTR
       Cycle 673  Read-Next
       Cycle 674  Find-Y-1st2
       Cycle 675  Find-Global-Y-
Phrase2
       Cycle 676  Parse-Current
          Cycle 677  Parse-Lex
*TOPMOST* TOPMOST-REL
       Cycle 678  Read-Next
       Cycle 679  Find-2nd-Size
       Cycle 680  Parse-Current
          Cycle 681  Parse-Lex
*SMALL* SIZE
       Cycle 682  Read-Next
       Cycle 683  Find-2nd-Shape
       Cycle 684  Parse-Current
          Cycle 685  Parse-Lex
*THIN* SHAPE-ABSTR
       Cycle 686  Read-Next
       Cycle 687  Find-2nd-Color
       Cycle 688  Parse-Current
          Cycle 689  Parse-Lex
*RED* COLOR
       Cycle 690  Read-Next
       Cycle 691  Parse-Current
          Cycle 692  Parse-Lex
*OBJECT* DOT-NAME
       Cycle 693  Read-Done
       Cycle 695  Ground-Topmost-
Color2
          Cycle 696  Get-Topmost-
Color
       Cycle 697  Ground-Val2
       Cycle 698  Ground-Done
       Cycle 699  Understand-Accept
```

A13) Check if presentation was accepted and complete presentation goal:

```
    Cycle 700  Present-Accept
```

A14) Update state of problem:

```
 Cycle 701  Done-Update-Old
 Cycle 702  Check-Accept
 Cycle 703  Update-New-Conf-Request
 Cycle 704  Find-Comm-Obligation
    Cycle 705  Understand-Request
    (You would like me to Prop78.
I'll let you know.)
 Cycle 706  Done-Task-Status
 Cycle 707  Task-Confirm-Oblig
```

A15) Decide on action and present message:

```
 Cycle 708  Decide-Action
    Cycle 709  Decide-Confirm
    Cycle 710  Decide-Done
 Cycle 711  Topic-End
 Cycle 712  Do-Confirm-Requested-
First
** <CONFIRM> **
 Cycle 713  Find-Comm-Obligation
    Cycle 714  Understand-Request
    (You would like me to Prop78.
I'll let you know.)
 Cycle 715  Done-Task-Status
 Cycle 716  Task-Confirm-Oblig
 Cycle 717  Decide-Action
    Cycle 718  Decide-State-Confirm
    Cycle 719  Decide-Done
 Cycle 720  Topic-End
 Cycle 721  Do-State-Confirming
    Cycle 722  Pick-Sample-Template-
Start
    Cycle 723  Read-Sample-First
    Cycle 724  Parse-Lex-Sample
       Cycle 725  Parse-Lex
*I'M* APOS-ABSTR
    Cycle 726  Read-Sample-Next
    Cycle 727  Parse-Lex-Sample
       Cycle 728  Parse-Lex
*CONFIRMING* PHASE-ABSTR
    Cycle 729  Read-Sample-Next
    Cycle 730  Parse-Lex-Sample
       Cycle 731  Parse-Lex
*ANOTHER* PREDOT-ABSTR
    Cycle 732  Read-Sample-Next
    Cycle 733  Parse-Lex-Sample
       Cycle 734  Parse-Lex
*MIDDLE* GLOBAL-X-ABSTR
    Cycle 735  Read-Sample-Next
    Cycle 736  Parse-Lex-Sample
       Cycle 737  Parse-Lex
*SMALL* SIZE
    Cycle 738  Read-Sample-Next
    Cycle 739  Parse-Lex-Sample
       Cycle 740  Parse-Lex
*FAT* SHAPE-ABSTR
    Cycle 741  Read-Sample-Next
    Cycle 742  Parse-Lex-Sample
       Cycle 743  Parse-Lex
*BLUE* COLOR
    Cycle 744  Read-Sample-Next
    Cycle 745  Parse-Lex-Sample
       Cycle 746  Parse-Lex
*OBJECT* DOT-NAME
    Cycle 747  Read-Sample-Done
    Cycle 748  Pick-Sample-Template-
End
    Cycle 749  Get-Next
    Cycle 750  Fill-Any
    Cycle 751  Get-Next
    Cycle 752  Fill-Confirm
    Cycle 753  Get-Next
    Cycle 754  Fill-Another
    Cycle 755  Get-Next
    Cycle 756  Skip-Global2
    Cycle 757  Get-Next
    Cycle 758  Fill-Size2-Any
    Cycle 759  Get-Next
    Cycle 760  Fill-Shape2-Any
    Cycle 761  Get-Next
    Cycle 762  Fill-Color2-Any
    Cycle 763  Get-Next
    Cycle 764  Fill-Dot
    Cycle 765  Get-Done
    Cycle 766  Type-Next
** I'M **
    Cycle 767  Type-Next
** CONFIRMING **
    Cycle 768  Type-Next
** A **
    Cycle 769  Type-Next
** SMALL **
    Cycle 770  Type-Next
** THIN **
    Cycle 771  Type-Next
** RED **
    Cycle 772  Type-Next
** SHAPE **
    Cycle 773  Type-Stall
    Cycle 775  Type-Done
```

A16) Inside presentation goal, wait for next message and understand it:

```
    Cycle 776  Present-Start-Wait
    Cycle 777  Present-Wait
    Cycle 778  Present-Wait
<CONFIRM>
    Cycle 786  Present-See-Confirm
       Cycle 787  Ground-Done
       Cycle 788  Understand-Accept
```

A17) Check if presentation was accepted and complete presentation goal:

```
    Cycle 789  Present-Accept
```

A18) Update state of problem:

Cycle 790 Done-Update-Old
Cycle 791 Check-Accept
Cycle 792 Update-New-See-Conf-End
Cycle 793 Turn-Wait-Obligation
Cycle 794 Done-Task-Status
Cycle 795 Task-Confirm-Oblig

A19) Decide on action and present message:

Cycle 796 Decide-Action
Cycle 797 Decide-Wait-For-Turn
Cycle 798 Decide-Done
Cycle 799 Topic-End
Cycle 800 Do-Wait-Turn
Cycle 801 Present-Silence

A20) Inside presentation goal, wait for next message and understand it:

Cycle 802 Present-Start-Wait
Cycle 803 Present-Wait
Cycle 804 Present-Wait
Cycle 822 Present-Wait
(LET'S CONFIRM BY ROWS LEFT TO RIGHT)
Cycle 823 Present-See-Message
(You said something. I'll try to understand.)
Cycle 824 Parse-Current
Cycle 825 Parse-No-Lex
LET'S MISC
Cycle 826 Read-Next
Cycle 827 Find-Confirm-No-Action
Cycle 828 Parse-Current
Cycle 829 Parse-Lex
CONFIRM CONF-ABSTR
Cycle 830 Read-Next
Cycle 831 Parse-Current
Cycle 832 Parse-No-Lex
BY MISC
Cycle 833 Read-Next
Cycle 834 Find-Conf-Plan
Cycle 835 Parse-Current
Cycle 836 Parse-Lex
ROWS CONF-PLAN-ABSTR
Cycle 837 Read-Next
Cycle 838 Parse-Current
Cycle 839 Parse-Lex
LEFT DIRECTION
Cycle 840 Read-Next
Cycle 841 Parse-Current
Cycle 842 Parse-No-Lex
TO MISC
Cycle 843 Read-Next
Cycle 844 Parse-Current
Cycle 845 Parse-Lex
RIGHT DIRECTION
Cycle 846 Read-Done
Cycle 848 Ground-Done
Cycle 849 Understand-Accept

A21) Check if presentation was accepted and complete presentation goal:

Cycle 850 Present-Accept

A22) Update state of problem:

Cycle 851 Done-Update-Old
Cycle 852 Check-Accept
Cycle 853 Update-New-Conf-Plan
Cycle 854 Find-Comm-Obligation
Cycle 855 Understand-Request
(You would like me to Prop101. I'll let you know.)
Cycle 856 Done-Task-Status
Cycle 857 Task-Confirm-Oblig

A23) Decide on action and present message:

Cycle 858 Decide-Action
Cycle 859 Decide-Assess-Plan-Ok
Cycle 860 Decide-Done
Cycle 861 Topic-End
Cycle 862 Do-Assess
Cycle 863 Pick-Sample-Template-Start
Cycle 864 Read-Sample-First
Cycle 865 Parse-Lex-Sample
Cycle 866 Parse-Lex
I DON'T THINK SO ASSESS-ABSTR
Cycle 867 Read-Sample-Done
Cycle 868 Pick-Sample-Template-End
Cycle 869 Get-Next
Cycle 870 Fill-Assess
Cycle 871 Get-Done
Cycle 872 Type-Next
** OK **
Cycle 873 Type-Done-Ok

A24) Inside presentation goal, wait for next message and understand it:

Cycle 874 Present-Start-Wait
Cycle 875 Present-Wait
Cycle 886 Present-Wait
<CONFIRM>
Cycle 887 Present-See-Confirm
Cycle 888 Ground-Done
Cycle 889 Understand-Accept

A25) Check if presentation was accepted and complete presentation goal:

Cycle 890 Present-Accept

A26) Update state of problem:

Cycle 891 Done-Update-Old
Cycle 892 Check-Accept
Cycle 893 Update-New-See-Conf-Start
Cycle 894 Turn-Wait-Obligation
Cycle 895 Done-Task-Status
Cycle 896 Task-Confirm-Oblig

A27) Decide on action and present message:

Cycle 897 Decide-Action
Cycle 898 Decide-Confirm-L2r
Cycle 899 Start-L2rCIRCLE50
Cycle 900 Lower-Y-L2r
Cycle 901 Y-Done-L2r
Cycle 902 X-Done-L2r
Cycle 903 Decide-Ground-Confirm-L2r
Cycle 904 Gg-Start
Cycle 905 Gg-Topmost
Cycle 906 Gg-Done
Cycle 907 Decide-Done
Cycle 908 Topic-End
Cycle 909 Do-Confirm-Requested-Match
** <CONFIRM> **
Cycle 910 Done-Find-Oblig
Cycle 911 Done-Task-Status
Cycle 912 Task-Confirm-Oblig
Cycle 913 Decide-Action
Cycle 914 Decide-Wait-For-Turn
Cycle 915 Decide-Done
Cycle 916 Topic-End
Cycle 917 Do-Wait-Turn
Cycle 918 Present-Silence

A28) Inside presentation goal, wait for next message and understand it:

Cycle 919 Present-Start-Wait
Cycle 920 Present-Wait
Cycle 923 Present-Wait
<CONFIRM>
Cycle 924 Present-See-Confirm
Cycle 925 Ground-Done
Cycle 926 Understand-Accept

A29) Check if presentation was accepted and complete presentation goal:

Cycle 927 Present-Accept

A30) Update state of problem:

Cycle 928 Done-Update-Old
Cycle 929 Check-Accept
Cycle 930 Update-New-See-Conf-Start
Cycle 931 Done-Find-Oblig
Cycle 932 Done-Task-Status
Cycle 933 Task-Confirm-Oblig

A31) Decide on action and present message:

Cycle 934 Decide-Action
Cycle 935 Decide-Confirm-L2r
Cycle 936 Start-L2rCIRCLE50
Cycle 937 Y-Done-L2r
Cycle 938 Lower-X-L2r
Cycle 939 X-Done-L2r
Cycle 940 Decide-Ground-Confirm-L2r
Cycle 941 Gg-Start
Cycle 942 Gg-Bottommost

```
      Cycle 943  Gg-Done
   Cycle 944  Decide-Done
Cycle 945  Topic-End
Cycle 946  Do-Confirm-Requested-
Match
** <CONFIRM> **
Cycle 947  Done-Find-Oblig
Cycle 948  Done-Task-Status
Cycle 949  Task-Confirm-Oblig
Cycle 950  Decide-Action
   Cycle 951  Decide-Wait-For-Turn
   Cycle 952  Decide-Done
Cycle 953  Topic-End
Cycle 954  Do-Wait-Turn
   Cycle 955  Present-Silence
```

A32) Inside presentation goal, wait for next message and understand it:

```
   Cycle 956  Present-Start-Wait
   Cycle 957  Present-Wait
   Cycle 958  Present-Wait
<CONFIRM>
   Cycle 959  Present-See-Confirm
      Cycle 960  Ground-Done
      Cycle 961  Understand-No-
Accept
```

A33) Check if presentation was accepted and complete presentation goal:

```
   Cycle 962  Present-Accept
```

A34) Update state of problem:

```
Cycle 963  Done-Update-Old
Cycle 964  Check-Accept
Cycle 965  Update-New-See-Conf-Start
Cycle 966  Done-Find-Oblig
Cycle 967  Done-Task-Status
Cycle 968  Task-Confirm-Oblig
```

A35) Decide on action and present message:

```
Cycle 969  Decide-Action
   Cycle 970  Decide-Confirm-L2r
      Cycle 971  Start-L2rCIRCLE50
      Cycle 972  Y-Done-L2r
      Cycle 973  Lower-X-L2r
      Cycle 974  X-Done-L2r
   Cycle 975  Decide-Confirm-Common-
L2r
   Cycle 976  Decide-Done
Cycle 977  Topic-End
Cycle 978  Do-Confirm-Requested-
Match
** <CONFIRM> **
Cycle 979  Done-Find-Oblig
Cycle 980  Done-Task-Status
Cycle 981  Task-Confirm-Oblig
Cycle 982  Decide-Action
   Cycle 983  Decide-Wait-For-Turn
   Cycle 984  Decide-Done
Cycle 985  Topic-End
Cycle 986  Do-Wait-Turn
   Cycle 987  Present-Silence
```

A36) Inside presentation goal, wait for next message and understand it:

```
   Cycle 988  Present-Start-Wait
   Cycle 989  Present-Wait
   Cycle 990  Present-Wait
<CONFIRM>
   Cycle 991  Present-See-Confirm
      Cycle 992  Ground-Done
      Cycle 993  Understand-Accept
```

A37) Check if presentation was accepted and complete presentation goal:

```
   Cycle 994  Present-Accept
```

A38) Update state of problem:

```
Cycle 995  Done-Update-Old
Cycle 996  Check-Accept
Cycle 997  Update-New-See-Conf-Start
Cycle 998  Done-Find-Oblig
Cycle 999  Done-Task-Status
Cycle 1000  Task-Confirm-Oblig
```

A39) Decide on action and present message:

```
 Cycle 1001  Decide-Action
    Cycle 1002  Decide-Confirm-L2r
       Cycle 1003  Start-L2rCIRCLE50
       Cycle 1004  Y-Done-L2r
       Cycle 1005  X-Done-L2r
    Cycle 1006  Decide-Ground-
Confirm-L2r
       Cycle 1007  Gg-Start
       Cycle 1008  Gg-Rightmost
       Cycle 1009  Gg-Done
    Cycle 1010  Decide-Done
 Cycle 1011  Topic-End
 Cycle 1012  Do-Confirm-Requested-
Match
** <CONFIRM> **
 Cycle 1013  Done-Find-Oblig
 Cycle 1014  Done-Task-Status
 Cycle 1015  Task-Confirm-Oblig
 Cycle 1016  Decide-Action
    Cycle 1017  Decide-Wait-For-Turn
    Cycle 1018  Decide-Done
 Cycle 1019  Topic-End
 Cycle 1020  Do-Wait-Turn
    Cycle 1021  Present-Silence
```

A40) Inside presentation goal, wait for next message and understand it:

```
    Cycle 1022  Present-Start-Wait
    Cycle 1023  Present-Wait
<CONFIRM>
    Cycle 1024  Present-See-Confirm
       Cycle 1025  Ground-Done
       Cycle 1026  Understand-Accept
```

A41) Check if presentation was accepted and complete presentation goal:

```
    Cycle 1027  Present-Accept
```

A42) Update state of problem:

```
 Cycle 1028  Done-Update-Old
 Cycle 1029  Check-Accept
 Cycle 1030  Update-New-See-Conf-
Start
 Cycle 1031  Done-Find-Oblig
 Cycle 1032  Done-Task-Status
 Cycle 1033  Task-Confirm-Oblig
```

A43) Decide on action and present message:

```
 Cycle 1034  Decide-Action
    Cycle 1035  Decide-Confirm-L2r
       Cycle 1036  Start-L2rC573
       Cycle 1037  Y-Done-L2r
       Cycle 1038  X-Done-L2r
    Cycle 1039  Decide-Ground-
Confirm-L2r
       Cycle 1040  Gg-Start
       Cycle 1041  Gg-Leftmost
       Cycle 1042  Gg-Done
    Cycle 1043  Decide-Done
 Cycle 1044  Topic-End
 Cycle 1045  Do-Confirm-Requested-
Match
** <CONFIRM> **
 Cycle 1046  Done-Find-Oblig
 Cycle 1047  Done-Task-Status
 Cycle 1048  Task-Confirm-Oblig
 Cycle 1049  Decide-Action
    Cycle 1050  Decide-Wait-For-Turn
    Cycle 1051  Decide-Done
 Cycle 1052  Topic-End
 Cycle 1053  Do-Wait-Turn
    Cycle 1054  Present-Silence
```

A44) Inside presentation goal, wait for next message and understand it:

```
    Cycle 1055  Present-Start-Wait
    Cycle 1056  Present-Wait
    Cycle 1057  Present-Wait
    Cycle 1071  Present-Wait
(I'M DONE WITH THE PROBLEM)
    Cycle 1072  Present-See-Message
    (You said something.  I'll try to
understand.)
       Cycle 1073  Parse-Current
          Cycle 1074  Parse-Lex
*I'M* APOS-ABSTR
       Cycle 1075  Read-Next
       Cycle 1076  Find-Topic-Phases
       Cycle 1077  Parse-Current
          Cycle 1078  Parse-Lex
*DONE* DONE-ABSTR
```

```
Cycle 1079  Read-Next
Cycle 1080  Parse-Current
   Cycle 1081  Parse-No-Lex
*WITH* MISC
Cycle 1082  Read-Next
Cycle 1083  Parse-Current
   Cycle 1084  Parse-Lex
*THE* PREDOT-ABSTR
Cycle 1085  Read-Next
Cycle 1086  Parse-Current
   Cycle 1087  Parse-No-Lex
*PROBLEM* MISC
Cycle 1088  Read-Done-No-Action
Cycle 1089  Read-Done-No-Arg1
Cycle 1090  Read-Done-No-Arg2
Cycle 1091  Read-Done
Cycle 1093  Ground-Done
Cycle 1094  Understand-Accept
```

A45) Check if presentation was accepted and complete presentation goal:

```
Cycle 1095  Present-Accept
```

A46) Update state of problem:

```
Cycle 1096  Done-Update-Old
Cycle 1097  Check-Accept
Cycle 1098  Update-New-Task-Done-Unknown
Cycle 1099  Find-Comm-Obligation
Cycle 1100  Understand-Statement
(You are trying to make me believe Prop116.  I'll try.)
Cycle 1101  Done-Task-Status
Cycle 1102  Task-Confirm-Oblig
```

A47) Decide on action and present message:

```
Cycle 1103  Decide-Action
   Cycle 1104  Decide-Tell-Done-Puzzle
   Cycle 1105  Decide-Done
Cycle 1106  Topic-End
Cycle 1107  Do-State-Done-Puzzle
   Cycle 1108  Pick-Matching-Template
   Cycle 1109  Get-Next
   Cycle 1110  Fill-Apos
   Cycle 1111  Get-Next
   Cycle 1112  Fill-Done
   Cycle 1113  Get-Next
   Cycle 1114  Fill-Misc
   Cycle 1115  Get-Next
   Cycle 1116  Fill-Any
   Cycle 1117  Get-Next
   Cycle 1118  Fill-Misc
   Cycle 1119  Get-Done
   Cycle 1120  Type-Next
** I'M **
   Cycle 1121  Type-Next
** DONE **
   Cycle 1122  Type-Next
** WITH **
   Cycle 1123  Type-Next
** THE **
   Cycle 1124  Type-Next
** PROBLEM **
   Cycle 1125  Type-Stall
      Cycle 1126  Stall-First
      Cycle 1127  Stall
      Cycle 1128  Stall
      Cycle 1129  Stall
      Cycle 1159  Stall
      Cycle 1160  Stall-Done
   Cycle 1161  Type-Done
```

A48) Inside presentation goal, wait for next message and understand it:

```
Cycle 1162  Present-Start-Wait
Cycle 1163  Present-Wait
Cycle 1164  Present-Wait
Cycle 1165  Present-Wait
Cycle 1167  Present-Wait
<DONE>
Cycle 1168  Present-See-Done
Cycle 1169  See-Score-Done
```

Table 4.4: A trace of the ACT-R model solving a problem with a human partner

Two Channels of Communication

The interface for this experiment actually allows two channels of communication, one through the chat window and one through a status window. This window gives feedback on the success of an object confirmation, but also lets the subject know if their partner is attempting to confirm an object. Advanced partners can give information about objects through the chat window and then use a standard pattern (left to right) to confirm all of the objects by using the status window to coordinate confirmations without using the chat window. This means a model of this behavior must be able to react to messages from the chat window or status window independently. This was done by giving the model two input buffers, one for typed text from the partner and one for messages from the status window. These buffers represent information that could be accessed by moving attention to different parts of the experiment screen. This attention was not modeled, and rules requiring input information had direct access to the buffers.

Accommodating/Non-Accommodating Models

The effect of accommodation can be tested by using two different models, one accommodating and one non-accommodating. The accommodating model matches the object description word (object, shape, dot, or blob), directional mode (compass (north, south, east, or west) or relational (above, below, right, or left)), and object dimensions (size, shape, or color). The non-accommodating model uses a different object description word from the partner, uses a different directional mode, and different object dimensions (if less than three dimensions are used). Both the accommodating and non-accommodating models accommodate to the message length used by the partner. This allows message length to be used as a dependent variable, since both models treat it equally and the only way it changes is by a decision of the partner. The way the models accommodate to message length is to take a previous message of the partner and use it as a template to create a new message. Since the message was created by the partner to satisfy the goal of presenting information, this is another example of the models retrieving successful goals as memories to solve a new goal. This re-use of a partner's message form has previously been successfully used in AI systems of natural language processing (Green & Lehman, 1998), but an explicit link to the accommodation literature was not made.

The following example gives an initial message with an accommodating response and a non-accommodating response. Note that a new medium round green object is introduced, and the accommodating model matches the relational directions "topmost" and "above" with relational directions "leftmost" and "left of" and matches the object description "dot". The non-accommodating model mismatches the relational directions "topmost" and "above" with the compass directions "western" and "west of" and mismatches the object description "dot" with "object".

```
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
```

The following examples show the non-accommodating model responding first to a message with two dimensions used to describe an object, then with one dimension used. Note that the length of the message is accommodated to even when the dimensions used to the object are not accommodated to.

```
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
```

The "Human" Model

The accommodating and non-accommodating models are able to solve the communication task, but cannot by themselves explain the effect of accommodation. This is because they are "passive" in that they are not the first to decide to skip words in messages descriptions or to skip messages describing confirmation actions. Instead, they follow the lead of their partner and skip words when their partner skips words and skip messages when their partner skips messages. What is needed is an "assertive" "human" model that can decide to skip words and messages first. This model should also be able to account for differences found when subjects interact with accommodating and non-accommodating models.

This "human" model was created by extending the accommodation model with extra rules for actively skipping words and confirmation messages. Since time is saved by not typing, these rules make solving the problem more efficient. Research has shown that efficiency is increased when partners are perceived as cooperative (Brichcin, Janousek, Uhlar, & Hnilica, 1994; Deutsch, 1980; Tjosvold, 1998). This effect is achieved in the current situation by having the efficiency rules be sensitive to cooperative actions of the partner (with accommodative word matching signaling cooperative behavior).

The efficiency rules added to the accommodation model were:

Rule	Reliability	Description
skip-word-match-eff	.735	skips a nonessential word if partner matches word
skip-word-nomatch-eff	.730	skips a nonessential word
skip-confirm-match-eff	.745	skips a confirm message if partner matches word
skip-confirm-nomatch-eff	.735	skips a confirm message
skip-confirm-continue-eff	.765	continues to skip if skipped before

These rules have a subsymbolic value, reliability, associated with them that affects the probability with which they will be used -- rules with higher reliabilities have a higher probability of being used. Other ACT-R parameters used were expected gain noise, set at 0.01, and retrieval threshold, set at -5.0.

Two of the rules, skip-word-match-eff and skip-confirm-match-eff, attempt to retrieve memories of their partner matching their own word use. This gives these rules a sensitivity to whether their partner is accommodating or non-accommodating. The other rules do not attempt to retrieve matching memories. The rationale behind these rules is that the decision to skip a word or confirmation message will more likely lead to success if the partner has been cooperative in their behavior, and memories of word matching by the partner give evidence of this cooperation. Rules that find this evidence have a higher reliability because the evidence increases the probability that skipping will lead to success. The rule to continue confirmation message skipping has the highest reliability because of prior experience with message skipping being accepted. The analogous rule to continue skipping words is not needed because the continuation of skipping words comes as a consequence of using the partner's messages as templates to create new messages. If the proposal to skip a word is accepted, the word will no longer be in the partner' s message.

This method of having a rule to perform an action that is sensitive to a certain context have a higher reliability than another rule to perform the same action without context at a lower reliability has been used before in ACT-R modeling. For example, Lovett (1998) uses this method in her building sticks task where subjects are asked to create a desired stick by adding and removing lengths from a current stick. Lovett shows that subject behavior can be explained with rules for adding and removing sticks, with some rules incorporating the context of how close the current stick is to the desired stick and some rules not utilizing this context. As with the modeling in this dissertation, the method she used to model behavior was to have the rules sensitive to length difference retrieve memories of the lengths and have a higher reliability than rules that were not sensitive to length.

Looking at message length, Figure 4.1 shows results of twenty runs of the "human" model (shown as a dashed line) interacting with the accommodating model and non-accommodating model compared to the results of subjects interacting with the two models. The parameters discussed above produce a close fit to human performance. The “human” model was then run with another copy of the”human” model with those parameters to produce a zero-parameter prediction of human/human performance in the restricted interface condition. Again, Figure 4.1 shows a close fit to human performance.

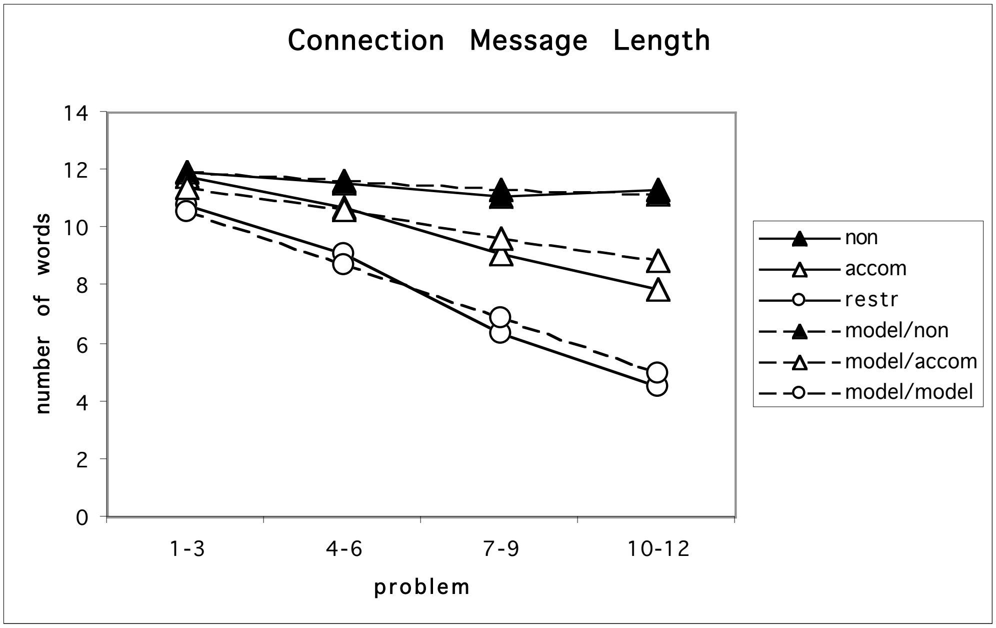


Figure 4.1: Message lengths for "human" model (dashed) and human subjects (solid)

Looking at the number of confirmation messages, Figure 4.2 shows results of twenty runs of the "human" model (shown as a dashed line) interacting with the accommodating model and non-accommodating model compared to the results of subjects interacting with the two models. Again, the parameters discussed above produce a close fit to human performance. The "human" model was then run with another copy of the"human" model with those parameters to produce a zero-parameter prediction of human/human performance in the restricted interface condition. Figure 4.2 shows that the models predict fewer confirmation messages in earlier problems than human subjects produce. One explanation may be an inhibition on the subjects' part for skipping messages before a number of problems have been solved.

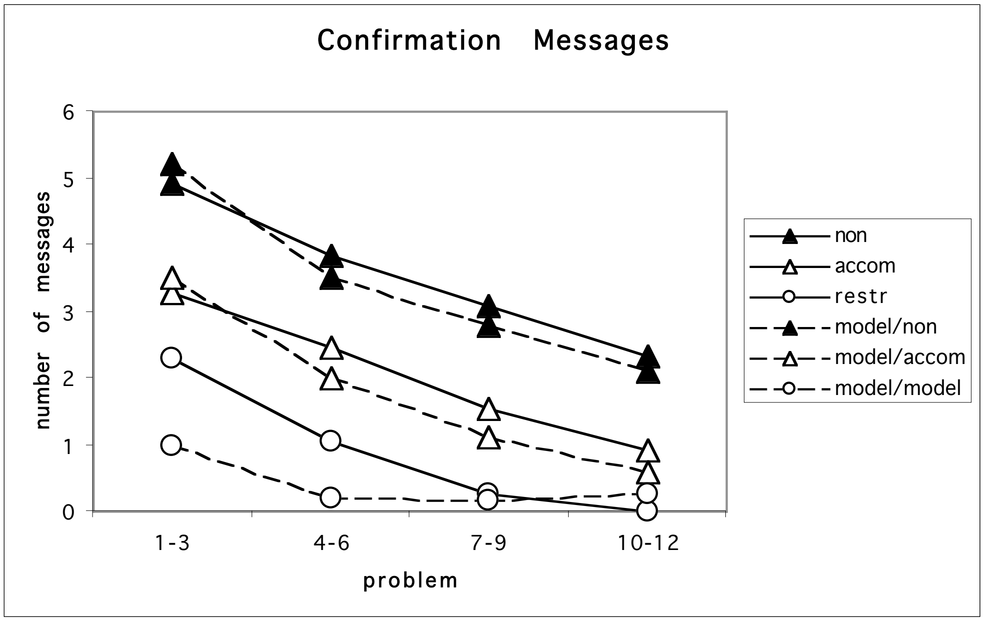


Figure 4.2: Number of confirmation messages for "human" model and human subjects

The "human" model proposes that word matching is seen as cooperative behavior. Evidence that subjects also see word matching as cooperative behavior can be found in the results of the post-experiment survey. Figure 3.19 showed that subjects rated their partner (actually an ACT-R model) as less cooperative ($t(43)=1.74$, $p<.05$) in the non-accommodation condition than in the accommodation condition.

The “human” model also proposes that the perception of cooperative behavior can influence the time to solve problems. Support for this can be seen in Figure 3.20, where subjects are grouped into categories of high partner cooperativeness rating and low cooperativeness rating. Although this grouping process decreases the number of subjects and therefore allows only qualitative analysis, subjects rating their partner as highly cooperative tend to solve problems faster than subjects using a low cooperativeness rating.

Chapter 5: Conclusions

Reflections on the Data

Data from the main experiment show that subjects interacting with accommodating models that match their word choice can solve problems faster than subjects interacting with non-accommodating partners. This result ties together results from the referential communication literature showing partner-based effects on efficiency with results from the accommodation literature showing accommodating behavior motivated by efficiency.

The careful reader will notice that the results also show that lexical accommodation (word matching) actually encouraged syntactic non-accommodation (shortening of message length). This counter-intuitive effect can be understood by looking at the processes involved. For one, subjects introduce changes to message length gradually. As seen in Figures 3.14a,b, and c, pairs do not normally differ in message length by more than one word. Another point is that gradual subject proposals to shorten the message length are usually met with acceptance on the human partner's part (and always on the model's part) by removing the same word as the subject. Rarely, a subject will suggest too drastic a change by dropping out too many words too early in the experiment, and their human partner will reject the proposal by continuing to use longer messages.

Reflections on the Models

Having a theory of communication in a computational form allowed testing of the theory by having it directly interact with subjects. In terms of errors and time to solve problems, subjects generally reacted to the accommodating model incorporating the theory much like any other human. In fact, in a post-experiment questionnaire subjects guessed they were interacting with a human 43% of the time they were interacting with the accommodating model (subjects guessed the non-accommodating model was human 48% of the time, but the difference is not significant). There is still room for improvement however, since only 10% of subjects interacting with human partners thought their partners were computers.

Contributions

This dissertation offers several contributions to the study of collaborative communication:

1. A referential communication task: Unlike the director/matcher paradigm found in the literature, partners in this task have equal abilities and responsibilities to communicate and act. This combination of communication and action was useful in showing the effects of word accommodation were not limited to just sentences involving the accommodated word, but also to actions not requiring any communication. Unlike tasks involving objects that are difficult to name, objects in this task have features with clearly defined names. Also, the difficulty of a problem in this task can be easily manipulated by changing the total number of objects or the number of similar objects.
2. A restricted interface: With some experience, subjects using the restricted interface show similar performance to subjects using an unrestricted chat interface. The restricted interface allows the control of the lexicon and syntax that subjects use to communicate. This permitted the introduction of functionally equivalent but different words that could be used for non-accommodation.
3. Empirical evidence of an accommodation effect: Subjects interacting with accommodating ACT-R models were shown to be significantly faster than those interacting with non-accommodating models. This speed-up was due to skipping words in descriptions of the connection of objects and to skipping entire messages involving object confirmations.

4. A computational theory of collaborative communication: The ACT-R models offer an explanation of the creation of common ground as the creation of memories of the successful completion of goals to present information. The effect of accommodation is explained as the use of rules to skip words and messages that are sensitive to memories of cooperative partner behavior.

References

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*: Mahwah, NJ: Erlbaum.

Anderson, J. R., & Matessa, M. (1997). A production system theory of serial memory. *Psychological Review, 104*, 728-748.

Austin, J. L. (1962). *How to do things with words*. Oxford: Oxford University Press.

Baker, M., & Lund, K. (1997). Promoting reflective interactions in a CSCL environment. *Journal of Computer Assisted Learning, 13(3)*, 175-193.

Bock, K., & Griffin, Z. M. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General, 129*(2), 177-192.

Bock, K. J. (1986). Syntactic persistence in language production. *Cognitive Psychology, 18*(3), 355-387.

Boyland, J. T., & Anderson, J. R. (1997). *Comprehension and production as avenues of syntactic priming.* Paper presented at the 19th Annual Conference of the Cognitive Science Society.

Brichcin, W., Janousek, J., Uhlar, P., & Hnilica, K. (1994). Joint problem-solving mediated by a computer network. *Journal of Russian & East European Psychology, 32*(1), 78-96.

Budiu, R., & Anderson, J. R. (2000). *Integration of Background Knowledge in Sentence Processing: A Unified Theory of Metaphor Understanding, Semantic Illusions and Text Memory.* Paper presented at the 3rd International Conference on Cognitive Modeling, Groningen, Netherlands.

Chick, J. K. (1985). The interactional accomplishment of discrimination in South Africa. *Language in Society, 14*, 299-326.

Clark, H. (1996). *Using Language*: Cambridge University Press.

Clark, H., & Schaefer, E. (1989). Contributing to Discourse. *Cognitive Science, 13*, 259-294.

Clark, H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition, 22*, 1-39.

Cohen, P. R., & Levesque, H. J. (1991). *Confirmations and Joint Action.* Paper presented at the IJCAI-91.

Connor-Linton, J. (1999). Competing communicative styles and crosstalk: A multi-feature analysis. *Language in Society, 28*(1), 25-56.

Core, M. G., & Allen, J. F. (1997). *Coding dialogs with the DAMSL scheme.* Paper presented at the AAAI Fall Symposium on Communicative Action in Humans and Machines, Boston, MA.

Deutsch, M. (1980). Fifty years of conflict. In L. Festinger (Ed.), *Retrospections on social psychology* (pp. 46-77). New York: Oxford University Press.

Deutsch, M., & Gerard, H. B. (1955). A study of normative and informational social influence upon individual judgment. *Journal of Abnormal and Social Psychology, 51*, 629-636.

Fais, L. (1994). Conversation as collaboration: Some syntactic evidence. *Speech Communication, 15*, 231-242.

Fais, L. (1998). Lexical accommodation in human- and machine-interpreted dialogues. *International Journal of Human-Computer Studies, 48*, 217-246.

Filer, L., & Scukanec, G. (1995). Collaborative referencing in elderly women. *Perceptual and Motor Skills, 81*, 995-1000.

Frazier, L. (1987). Sentence processing: A tutorial review. In M. Coltheart (Ed.), *Attention and performance 12: The psychology of reading* (pp. 559-586). London, UK: Lawrence Erlbaum Associates.

Garrod, S., & Anderson, A. (1987). Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition, 27*, 181-218.

Garrod, S., & Doherty, G. (1994). Conversation, co-ordination and convention: an empirical investigation of how groups establish linguistic conventions. *Cognition, 53*, 181-215.

Giles, H., Mulac, A., Bradac, J. J., & Johnson, P. (1987). Speech accommodation theory: The first decade and beyond. *Communication Yearbook, 10*.

Green, N., & Lehman, J. F. (1998). *An Application of Explanation-Based Learning to Discourse Generation and Interpretation.* Paper presented at the AAAI 1998 Spring Symposium on Applying Machine Learning to Discourse Processing.

Grice, H. P. (1957). Meaning. *Philosophical Review, 66*, 377-388.

Grosz, B., & Sidner, C. (1990). Plans for Discourse. In P. R. Cohen, J. Morgan, & M. E. Pollack (Eds.), *Intentions in Communication* : MIT Press.

Hupet, M., & Chantraine, Y. (1992). Changes in repeated references: Collaboration or repetition effects? *Journal of Psycholinguistic Research, 21*(6), 485-496.

Hupet, M., Chantraine, Y., & Nef, F. (1993). References in conversation between young and old normal adults. *Psychology and Aging, 8*(3), 339-346.

Krauss, R. M., & Fussell, S. R. (1991). Constructing shared communicative environments. In L. B. Resnick & J. M. Levine (Eds.), *Perspectives on socially shared cognition* (pp. 172-200).

Krauss, R. M., & Weinheimer, S. (1966). Concurrent feedback, confirmation and the encoding of referents in verbal communication. *Journal of Personality and Social Psychology, 4*, 343-346.

Lebiere, C., & Anderson, J. R. (1998). Cognitive Arithmetic. In J. R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought* (pp. 297-342): Hillsdale, Mawhaw, NJ: Erlbaum.

Lehman, J. F. (1989). *Adaptive Parsing: Self-extending Natural Language Interfaces.* , Carnegie Mellon University, Pittsburgh.

Levelt, W. J. M. (1982). Linearization in describing spatial networks. In S. Peters & E. Saarinen (Eds.), *Processes, beliefs, and questions* (pp. 199-220): Dordrecht: D. Reidel.

Lewis, R. (1999). A New Computational Model of Human Sentence Processing: Interference-Limited Working Memory .

Lovett, M. C. (1998). Choice. In J. R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought* (pp. 255-296): Mahwah, NJ: Erlbaum.

Matessa, M., & Anderson, J. R. (2000). Modelling Focused Learning in Role Assignment. *Language and Cognitive Processes, 15*(3), 263-292.

Novick, D., Hansen, B., & Lander, T. (1994). *Letter-sequence dialogues* (CSE 94-007): Department of Computer Science and Engineering, Oregon Graduate Institute of Science & Technology.

Poesio, M., & Traum, D. (1998). Towards an Axiomatization of Dialogue Acts. In J. Hulstijn & A. Nijholt (Eds.), *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology)* (pp. 207-222). Enshede, NL.

Rich, C., & Sidner, C. L. (1998). COLLAGEN: A Collaboration Manager for Software Interface Agents. *User Modeling and User-Adapted Interaction, 8(3/4)*, 315-350.

Rosenberg, S., & Cohen, B. D. (1966). Referential processes of speakers and listeners. *Psychological Review, 73*(3), 208-231.

Searle, J. (1969). *Speech Acts*. Cambridge: Cambridge University Press.

Slator, B. M., Anderson, M. P., & Conley, W. (1986). Pygmalion at the interface. *Communications of the ACM, 29*, 499-514.

Taatgen, N., & Anderson, J. R. (2000). *Why do children learn to say "broke"? A model of learning the past tense.* Paper presented at the Seventh Annual ACT-R Workshop, Pittsburgh, PA.

Tjosvold, D. (1998). Making employee involvement work: Cooperative goals and controversy to reduce costs. *Human Relations, 51*(2), 201-214.

Traum, D., & Allen, J. (1994). *Discourse Obligations in Dialogue Processing.* Paper presented at the 32nd Annual Meeting of the Association for Computational Linguistics.

Zoltan-Ford, E. (1991). How to get people to say and type what computers can understand. *International Journal of Man-Machine Studies, 34*, 527-547.