Generating Engagement Behaviors in Human-Robot Interaction

By
Aaron Holroyd

A Thesis
Submitted to the faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Master of Science
in
Computer Science

April 2011

APPROVED:

Professor Charles Rich, Thesis Advisor

Professor David Brown, Thesis Reader

Professor Craig Wills, Head of Department

Abstract

Based on a study of the engagement process between humans, I have developed models for four types of connection events involving gesture and speech: directed gaze, mutual facial gaze, adjacency pairs and backchannels. I have developed and validated a reusable Robot Operating System (ROS) module that supports engagement between a human and a humanoid robot by generating appropriate connection events. The module implements policies for adding gaze and pointing gestures to referring phrases (including deictic and anaphoric references), performing end-of-turn gazes, responding to human-initiated connection events and maintaining engagement. The module also provides an abstract interface for receiving information from a collaboration manager using the Behavior Markup Language (BML) and exchanges information with a previously developed engagement recognition module. This thesis also describes a Behavior Markup Language (BML) realizer that has been developed for use in robotic applications. Instead of the existing fixed-timing algorithms used with virtual agents, this realizer uses an event-driven architecture, based on Petri nets, to ensure each behavior is synchronized in the presence of unpredictable variability in robot motor systems. The implementation is robot independent, open-source and uses the Robot Operating System (ROS).

Acknowledgments

I would first like to thank a variety of professors. To my main advisor Charles Rich, thanks for teaching me the research process, so many technical details, and a variety of English nuances. To Candy Sidner, who was essentially an advisor, thanks for helping scope my thesis and for your background in linguistics and psychological phenomenons.

My reader, David Brown, many thanks for fitting me into your busy schedule and for the questions you posed, which helped ensure that I have enough background within this document. Also, to Joseph Beck, thanks for your expertise in statistics, without whom, I may have gone on believing that I did not have many statistically significant results.

Brett, I am glad that I worked with you throughout this thesis. I have learned a lot about vision, specifically OpenCV. Have fun at iRobot, and perhaps we will work together again.

To the other students in the Interaction Lab, thanks for your help in debugging our study. Sometimes, just having someone else to talk something out to is a help and othertimes you helped find bugs, especially during pilot studies. I would also like to thank all of the participants who participated in the tangram game.

Finally, thanks to all of the other professors, secretaries, and staff for all that you have taught and done for me.

This work is supported in part by the National Science Foundation under awards IIS-0811942 and IIS-1012083.

Table of Contents

List of Figures

List of Tables

1 Introduction

Engagement is "the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake" [1]. To elaborate,

> ...when people talk, they maintain conscientious psychological connection with each other and each will not let the other person go. When one is finished speaking, there is an acceptable pause and then the other *must* return something. We have this set of unspoken rules that we all know unconsciously but we all use in every interaction. If there is an unacceptable pause, an unacceptable gaze into space, an unacceptable gesture, the cooperating person will change strategy and try to re-establish contact. Machines do none of the above, and it will be a whole research area when people get around to working on it. (Biermann, invited talk at User Modeling Conference, 1999)

This thesis is done in coordination with Ponsler [2] where the joint work and my work are separated by "we" and "I" throughout this thesis. This thesis begins with an analysis of a study completed during my Major Qualifying Project (MQP) [3] along with a theory of Connection events. This theory is followed by a policy driven model of maintaining engagement. I then describe the BML realizer that guarantees that the timing constraints provided by the generation policies are satisfied. Next I describe how we validated the work in both Ponsler's and my theses. Finally I describe future directions of this work.

1.1 Motivation

We believe that engagement is a fundamental process that underlies all human interaction and has common features across a very wide range of interaction circumstances. At least for humanoid robots, this implies that modeling engagement is crucial for constructing robots that can interact effectively with humans without special training.

This argument motivates the main goal of our research, which is to develop an engagement module that can be *reused* across different robots and applications. There is no reason that every project should need to re-implement the engagement process. Along with the creators of ROS [4] and others, we share the vision of increasing code reuse in the robotics research and development community.

1.2 Problem Statement

The primary goal of this work is to improve our understanding of engagement between humans and robots during collaborative tasks. In the current state of the art, robots are either able to interact naturally with humans, or complete tasks with humans. There are few robots that are able to both complete tasks effectively and interact naturally with humans at the same time. I developed an open-source module to ensure that the robot stays engaged during a task with a human.

1.3 Contributions

The main scientific contribution of this work is contained in the concept of connection events and the engagement generation *policies*, described in Section 3.3, which define the conditions under which the robot initiates specific behaviors that contribute to engagement. These policies are based on our own [3, 5] and others' observational study of human engagement. In particular, Section 3.3.2 proposes a novel analysis of optimal gesture-speech pairs for object reference. Chapter 5 describes an experimental validation of these policies.

The main practical contribution of this work is that, by careful attention to architecture (see Section 5.1), we have developed modules that can be *reused* across different robots and applications. I have also implemented the first event-driven Behavior Markup Language (BML) realizer (see Chapter 4), which is required for robotic applications of BML.

1.4 Related Work

In the area of artificially intelligent agents that interact with humans, there are two common approaches to embodying the AI agent. The first is to use a physical object, e.g., a humanoid robot. The second is to use a virtual representation of a person, or other animate beings. This thesis focuses on a robotic agent; however, the results concluded from this thesis may also be relevant to virtual agents.

Sidner *et al.* [6] laid the foundation for engagement in human-robot interaction by researching the effect of tracking faces during an interaction. They describe a study conducted which granted insight into human gaze behavior. This study shows that the participants of the study direct their gaze at the robot 70% or more of the time. This study also showed that the participants nodded, even though the robot did not know how to recognize nods. From these and other results, they conclude that there are different actions that people take in order to ensure that all the participants in an interaction understand what is happening.

1.4.1 Virtual Agents

Virtual agents are realistically rendered images of animate beings, and can vary from just a face to a full body. Bohus and Horvitz [7] use a realistically rendered avatar head that can interact via natural language to learn how people interact. Since this system does not have hands or other actuators, the avatar cannot point in the way that humans would. Instead, the avatar uses gaze to direct the attention of the humans. A humanoid robot may use gaze to direct another persons gaze when both arms are

being used by a more important task and the robot still needs to direct the humans attention.

Bohus and Horvitz [7] and I have a common starting point in the work by Sidner *et al.* [1]. One of the key differences is that the model I have developed has a deeper model of engagement, as opposed to their "is engaged" or "not engaged" model. Another difference is that their work is meant for multi-party engagement, i.e., more than two agents interacting. These differences do not exclude our models from working together.

Open-World vs. Uncontrolled Environment

It is also important to note that many virtual agents are placed in uncontrolled environments, whereas our system is in a controlled environment. For example, Bohus and Horvitz [7, 8] have placed their agent in a lobby as a receptionist to test their theory of engagement. We have simplified many of the vision problems, since we are not working on that problem and thus have controlled our environment, but have kept our system as open-world as possible for generalizability.

In terms of computational applications, the most closely related work is that of Peters [9], which involves agents in virtual environments, and Bohus and Horvitz [7, 8]. We share a similar theoretical framework with both of these efforts, but differ in dealing with a humanoid robot and in our focus on building a reusable engagement module.

1.4.2 Verbal Communication

There are two subfields of verbal communication that I draw from for this thesis including turn-taking and distractor sets. The definition of a turn varies slightly, but for this thesis I will use it as the time from when an agent, human or robot, is allowed to speak, begins speaking, and finally finishes speaking, similar to Sacks *et al.* [10]. A distractor set is the set of all objects that can be confused with a target object, given a description of the target object [11, 12]. For example, if a target object is a green square, someone asks for the green object, and there are more green objects, the distractor set contains all of the green objects that are not the square.

1.4.3 Non-verbal Communication

Mutlu *et al.* [13] have studied the interaction of gaze and turn-taking [14] using a humanoid robot. Flippo *et al.* [15] have developed a similar architecture to ours (see Section 5.1) with similar concerns of modularity and the fusion of verbal and nonverbal behaviors. However, their model focuses on multi-modal interfaces rather than robots. Neither of these efforts, however, use concepts from engagement.

1.4.4 Gaze Attention

In the area of human studies, Argyle and Cook [16] documented that failure to attend to another person via gaze is evidence of lack of interest and attention. Other researchers have offered evidence of the role of gaze in coordinating talk between speakers and listeners, in particular, how gestures direct gaze to the face and why gestures might direct gaze away from the face [17, 18, 19]. Nakano *et al.* [20]) reported on the use of the listener's gaze and the lack of negative feedback to determine whether the listener has understook [21] the speaker. We rely upon the background of all of this work in the analysis of our own empirical studies.

1.4.5 Behavior Markup Language

BML was created by Kopp *et al.* [22] who identified the need for a common behavior specification framework for embodied conversational agents. BML builds upon previous frameworks, including BEAT [23], MURML [24], APML [25], and RRL [26]. Each of these other languages for describing behaviors were analyzed and the best parts of each were taken to develop BML.

Vilhjalmsson *et al.*[27] revised BML to fix problems found in constraining behaviors and behaviors that were not fully defined in the initial version. They define new types of constraints and add attributes or sub-elements to behaviors that were partially defined, such as how long to gaze at a specified target. They also show how to interchange behavior planners and executors (see Section 4.1.3), for more flexibility.

All of the previous referenced work uses BML for virtual agents with a fixed-time schedule and thus are not effective for robotic applications (See Section 4.2.2). Anh and Pelachaud [28] developed a fixed-timing realizer for a Nao robot based on the Gretta animation engine. This system uses a database of motions and synchronizes them using timing constraints. They are aware of the issues surrounding a fixed-timing realizer for robots, but do not solve them. I will discuss these limitations in Chapter 4

2 Engagement Theory

Based on the human study conducted during my Major Qualifying Project (MQP), approximately a Bachelor's thesis at WPI, and on the previous research of engagement, we have developed the concept of connection events and four types. After determining the types, we analyzed the videos of interactions during the human study for statistics about the events.

2.1 Canapé Study

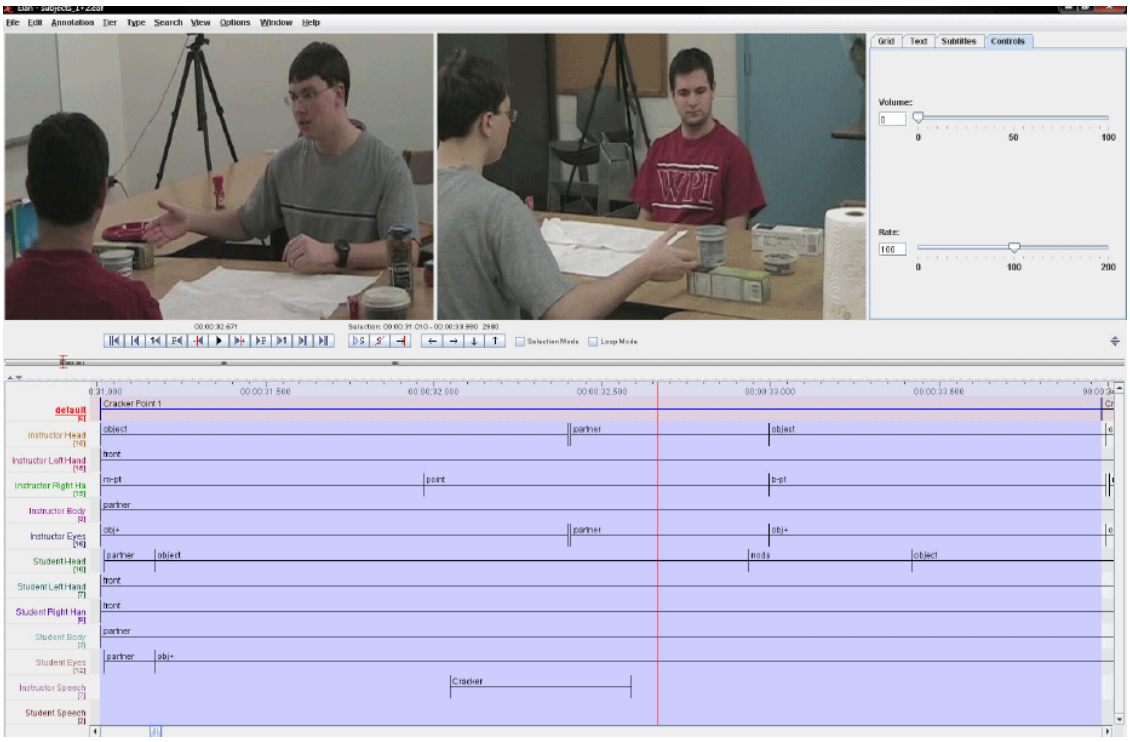


Figure 2.1: Coding of a directed gaze event during the canapé study

In the MQP, I conducted a study in which pairs of humans sat across an L-shaped table from each other and prepared canapés together (see Figure 2.1) [3]. Each of the four sessions involved an experimenter and two study participants and lasted about 15–

20 minutes. In the first half of each session, the experimenter instructed the participant how to make several different kinds of canapés using the different kinds of crackers, spreads and toppings arrayed on the table. The experimenter then left the room and was replaced by a second participant, who was then taught to make canapés by the first participant. The eight participants, six males and two females, were all college students at Worcester Polytechnic Institute (WPI), and the sessions were videotaped.

In our current analysis of the videotapes, we only looked at the engagement maintenance process. We did not analyze the participants' behaviors for initiating engagement (meeting, greeting, sitting down, etc.) or terminating engagement (ending the conversation, getting up from the table, leaving the room, etc.) These portions of the videotapes will be fruitful for future study.

For each session, we coded throughout: where each person was looking (at the other person's face, at a specific object or group of objects on the table, or "away"), when they pointed at a specific object or objects on the table, and the beginning and end of each person's speaking turn. Based on this analysis and the literature on engagement cited above, we have identified four types of engagement maintenance actions, what we call *connection events*, namely directed gaze (DG), mutual facial gaze (MFG), adjacency pairs (AP) and backchannels (BC). Our hypothesis is that these events, occurring at some minimum frequency, are the process mechanism for maintaining engagement.

2.2 Connection Events

Figures 2.2–2.5 show timelines (with numbers for reference in text) for the four types of connection events we have analyzed and Table 2.1 shows some summary statistics. In the discussion below, we describe the objectively observable behavioral components of each event type and hypothesize regarding the accompanying intentions of the participants. Dotted lines indicate optional behaviors. Also, gesture and speech events often overlap. This notion of connection events and the following four types of connection events is the first half of the theoretical portion of this thesis.

2.2.1 Directed Gaze

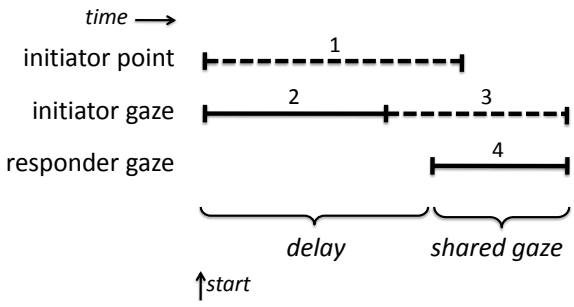


Figure 2.2: Time line for directed gaze

In directed gaze [17], one person (the *initiator*) looks and optionally points at some object or group of objects in the immediate environment, following which the other person (the *responder*) looks at the same object(s). We hypothesize that the initiator intends to bring the indicated object(s) to the responder's attention, i.e., to make the object(s) more salient in the interaction. This event is often synchronized with the initiator referring to the object(s) in speech, as in "now spread the *cream cheese* on the cracker." By turning his gaze where directed, the responder intends to be cooperative and thereby signals his desire to continue the interaction (maintain engagement).

In more detail (see Figure 2.2), notice first that the pointing behavior (the act of pointing) (1), if it is present, begins after the initiator starts to look (2) at the indicated object(s). This is likely because it is hard to accurately point at something without looking to see where it is located. It is usually possible to creatively imagine an exception to almost any rule such as this. For example, if a person is standing with his back to a mountain range, he might point over his shoulder to "the mountains" without turning around to look at them. I will not bother continuing to point out the possibility of such exceptions below. Furthermore, we observed several different configurations of the hand in pointing, such as extended first finger, open hand (palm up or palm down—see Figure 2.1), and a circular waving motion (typically over a group of objects). An interesting topic for future study (that will contribute to robot generation of these behaviors) is to determine which of these configurations are individual differences and which serve different communicative functions.

After some delay, the responder looks at the indicated object(s) (4). The initiator usually maintains the pointing (1), if it is present, at least until the responder starts looking at the indicated object(s). However, the initiator may stop looking at the indicated object(s) (2) before the responder starts looking (4), especially when there is pointing. This is often because the initiator looks at the responder's face, assumedly to check whether the responder has directed his gaze yet. (Such a moment is captured in Figure 2.1.)

Finally, there may be a period of shared gaze, i.e., a period when both the initiator (3) and responder (4) are looking at the same object(s). Shared gaze has been documented [29] as an important component of human interaction.

2.2.2 Mutual Facial Gaze

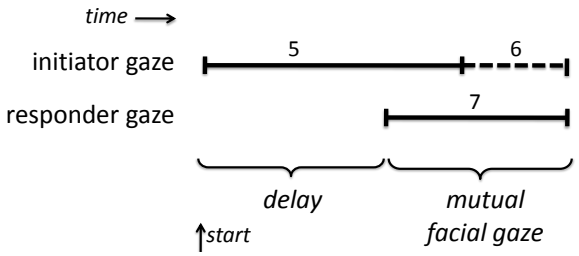


Figure 2.3: Time line for mutual facial gaze

Mutual facial gaze [16] has a time line (see Figure 2.3) similar to directed gaze, but simpler, since it does not involve pointing. The event starts when the initiator looks at the responder's face (5). After a delay, the responder looks at the initiator's face, which starts the period of mutual facial gaze (6,7). Notice that the delay can be zero, which occurs when both parties simultaneously look at each other.

The intentions underlying mutual facial gaze are less clear than those for directed gaze. We hypothesize that both the initiator and responder in mutual facial gaze engage in this behavior because they intend to maintain the engagement process. Mutual facial gaze does however have other interaction functions. For example, it is typical to establish mutual facial gaze at the end of a speaking turn.

Finally, what we are calling mutual facial gaze is often referred to informally as "making eye contact." This latter term is a bit misleading since people do not normally stare continuously into each other's eyes, but rather their gaze roams around the other person's face, coming back to the eyes from time to time [16].

2.2.3 Adjacency Pair

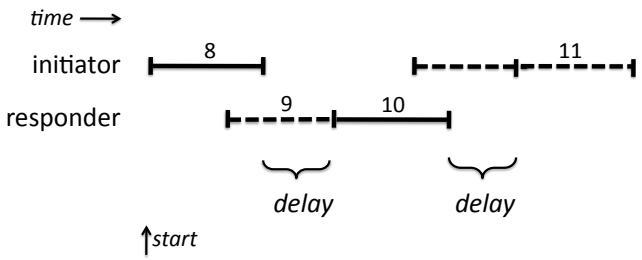


Figure 2.4: Time line for adjacency pair

In linguistics, an adjacency pair [14] consists of two utterances by two speakers, with minimal overlap or gap between them, such that the first utterance provokes the second utterance. A question-answer pair is a classic example of an adjacency pair. We generalize this concept slightly to include both verbal (utterances) and non-verbal communication acts. So for example, a nod could be the answer to a question, instead of a spoken "yes." Adjacency pairs, of course, often overlap with the gestural connection events, directed gaze and mutual facial gaze.

The simple time line for an adjacency pair is shown in Figure 2.4. First the initiator communicates what is called the *first turn* (8). Then there is a delay, which could be zero, or negative for an interruption) if the responder starts talking before the initiator finishes (9). Then the responder communicates what is called the *second turn* (9,10). In some conversational circumstances, this could also be followed by a *third turn* (11) in which the initiator, for example, repairs the responder's misunderstanding of his original communication.

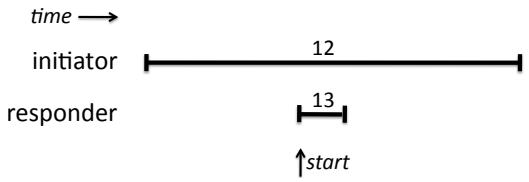


Figure 2.5: Time line for backchannel

2.2.4 Backchannel

A backchannel [14] is an event (see Figure 2.5) in which one party (the responder) directs a brief verbal or gestural communication (13) back to the initiator *during* the primary communication (12) from the initiator to the responder. Typical examples of backchannels are nods and/or saying "uh, huh." Backchannels are typically used to communicate the responder's comprehension of the initiator's communication (or lack thereof, e.g., a quizzical facial expression) and/or desire for the initiator to continue. Unlike the other three connection event types, the start of a backchannel event is defined as the start of the responder's behavior and this event has no concept of delay.

2.3 Canapé Study Results

Table 2.1: Summary statistics for human engagement study

		count	delay (sec)		
			min	mean	max
directed gaze	succeed	13	0	0.3	2.0
	fail	1	1.5	1.5	1.5
mutual facial gaze	succeed	11	0	0.7	1.5
	fail	13	0.3	0.6	1.8
adjacency pair	succeed	30	0	0.4	1.1
	fail	14	0.1	1.2	7.4
backchannel		15	n/a	n/a	n/a
mean time between connection events (MTBCE) = 5.7 sec max time between connection events = 70 sec					

Summary statistics from a detailed quantitive analysis of approximately nine minutes of engagement maintenance time are shown in Table 2.1. A connection event succeeds if the responder correctly completes the connection event, e.g. looking at the

initiator in a MFG, and fails if the responder does not complete the connection event. The time between connection events is defined as the time between the *start* of successive events, which properly models overlapping events. We hypothesize that the mean time between connection events (MTBCE) captures something of what is informally called the "pace" of an interaction [30]:

$$\text{pace} \quad \propto \quad \frac{1}{\text{MTBCE}}$$

In other words, the faster the pace, the less the time between connection events. Furthermore, Ponsler's implementation of an engagement recognition module [2] calculates the MTBCE on a sliding window and considers an increase as evidence for the weakening of engagement.

Two surprising observations in Table 2.1 are the relatively large proportion of failed mutual facial gaze (13/24) and adjacency pair (15/45) events and the 70 second maximum time between connection events. Since we do not believe that engagement was seriously breaking down anywhere during the middle of our sessions, we take these observations as an indication of missing factors in our model of engagement. In fact, reviewing the specific time intervals involved, what we found was that in each case the (non-)responder was busy with a detailed task on the table in front of him.

3 Engagement Generation

The second half of the theoretical portion of this thesis is generating each of the connection events described above. The generation module receives a *turn fragment* and decides which connection events to begin. It also responds to human initiated mutual facial gaze and directed gaze events. Adjacency pairs must be responded to by the collaboration manager since they require task knowledge. Current research does not provide a theory of how to automatically initiate backchannels (see Section 6.2).

3.1 Basic Engagement Architecture

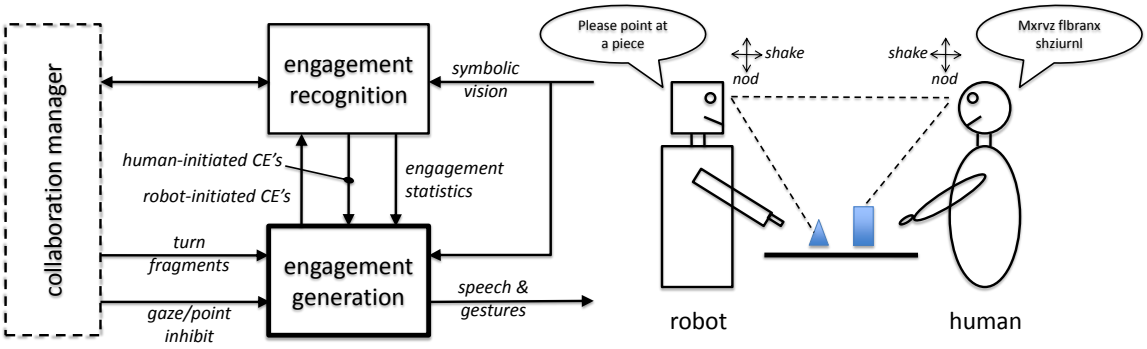


Figure 3.1: Basic engagement architecture

Figure 3.1 shows how engagement fits into a system architecture. At the highest level, the collaboration manager makes task decisions and decides what needs to be said. These decisions are passed, as *turn fragments*, to the engagement generation module that decides which connection events to initiate. The engagement recognition module determines when the connection events are completed or when a human initiated a connection event as described in [2]. The engagement recognition module then informs the engagement generation module to respond to the human initiated connection events.

3.2 Turn Fragment

Turn fragments are the fundamental representation by which the engagement generation module provides its services to the collaboration manager. A *turn fragment* comprises:

- a turn index (first, second or third turn of an adjacency pair–see Section 2.2.3),
- a turn status (beginning/middle/end of turn, full turn or backchannel), and
- a set of behaviors with associated synchronization constraints, where each behavior is one of either:
  - a gesture (gaze, point, nod, shake, etc.),
  - a phrase to be spoken, or
  - an object reference.

Turn fragments are encoded in an extension (EBML) to the Behavior Markup Language (BML) [27] (see Section 4.1 for a brief description of BML).

3.2.1 EBML Input from Collaboration Manager

Figure 3.2 is an example of a turn fragment in EBML, showing the new attributes added to BML to encode the turn index and turn status, and the new element type (ebml:reference), discussed further in Section 3.3.2, to encode object references. Note that the turn fragment representation supports incremental processing of communication from the robot to the human. The collaboration manager is not required to produce an entire turn, or even a complete sentence, in a single call to the generation module. For example, the robot utterance above could be broken into two turn fragments, one for "Please place this piece," followed by one for "on the right side of the purple square."

```
<ebml:ebml id="ebml-example" xmlns="org.mindmakers.bml"
           xmlns:ebml="edu.wpi.ebml"
           turnIndex="1" turnStatus="full">
    <speech id="speech-1">
        <text>Please place</text>
    </speech>
    <ebml:reference id="ref-1" target="yellow-triangle"
                    minReliability="0.85">
        <speechOption text="the triangle" cost="2">
            <distractor object="pink-triangle" />
            <distractor object="purple-triangle" />
        </speechOption>
        <speechOption text="the yellow triangle"
                      cost="3" />
        <speechOption text="it" cost="1" />
        <speechOption text="this piece" cost="2"
                      deictic="true">
            <distractor object="pink-triangle" />
            <distractor object="purple-triangle" />
            <distractor object="red-square" />
            <distractor object="blue-square" />
            <!--- and the five other pieces - -->
        </speechOption>
    </ebml:reference>
    <speech id="speech-3">
        <text>on the right side of</text>
    </speech>
    <ebml:reference id="ref-2" target="purple-square"
                    minReliability="0.85">
        <speechOption text="the square" cost="2">
            <distractor object="red-square" />
            <distractor object="blue-square" />
            <distractor object="pink-square" />
            <distractor object="green-square" />
        </speechOption>
        <speechOption text="the purple square"
                      cost="3" />
        <speechOption text="this piece" cost="2"
                      deictic="true">
            <distractor object="pink-triangle" />
            <distractor object="purple-triangle" />
            <distractor object="yellow-triangle" />
            <distractor object="red-square" />
            <!--- and the five other pieces - -->
```

```
45          </speechOption>
46      </ebml:reference>
47      <constraint id="constraint-1">
48          <synchronize ref="speech-1:end">
49              <sync ref="ref-1:start" />
50          </synchronize>
51          <synchronize ref="ref-1:end">
52              <sync ref="speech-3:start" />
53          </synchronize>
54          <synchronize ref="speech-3:end">
55              <sync ref="ref-2:start" />
56          </synchronize>
57      </constraint>
58  </ebml:ebml>
```

Figure 3.2: EBML turn fragment example

3.2.2 BML Output to BML Realizer

BML was chosen because it provides a rich language for expressing timing constraints between gestures and speech. Figure 3.3 shows an example of a standard BML behavior block drawn from the tangram game described in Section 5.2.1. This block will cause the robot to utter the sentence, "Please place this piece on the right side of the purple square," synchronized with initiation of a directed gaze connection event, where the gaze and pointing are timed as shown in Figure 2.2 and synchronized with the phrase "this piece."

3.3 Generation Policies

The policies inside of the generation module are broken down based on the input they consume. Figure 3.4 shows the representation of the policies inside the engagement generation module. There are policies for the turn fragments (robot initiated connection events), completion of human initiated events, and for maintenance of engagement. The turn policy consumes part of a turn fragment, encoded in EBML, and possibly outputs a robot initiated adjacency pair, mutual facial gaze, or backchannel. The reference policy consumes the reference elements of the EBML input and optionally

```
<bml id="bml-example" xmlns="org.mindmakers.bml">
    <gaze id="gaze-1" target="yellow-triangle" />
    <speech id="speech-1">
        <text>Please place</text>
    </speech>
    <gesture id="point-1" type="POINT"
             target="yellow-triangle" />
    <speech id="speech-2">
        <text>this piece</text>
    </speech>
    <speech id="speech-3">
        <text>on the right side of</text>
    </speech>
    <speech id="speech-4">
        <text>the purple square</text>
    </speech>
    <constraint id="constraint-1">
        <synchronize ref="gaze-1:stroke">
            <sync ref="speech-1:start" />
        </synchronize>
        <synchronize ref="gaze-1:stroke + 2">
            <sync ref="point-1:stroke_start" />
        </synchronize>
        <synchronize ref="point-1:stroke_end">
            <sync ref="speech-2:start" />
        </synchronize>
        <synchronize ref="speech-2:end">
            <sync ref="speech-3:start" />
        </synchronize>
        <synchronize ref="speech-3:end">
            <sync ref="speech-4:start" />
        </synchronize>
    </constraint>
</bml>
```

Figure 3.3: BML behavior block example

adds a speech, gaze, pointing gesture or robot initiated directed gaze. To respond to human initiated connection events, the response policy consumes messages from the engagement recognition module and possibly outputs gaze behaviors. The maintenance policy attempts to keep the mean time between connection events (MTBCE) constant by changing the robot's gaze.

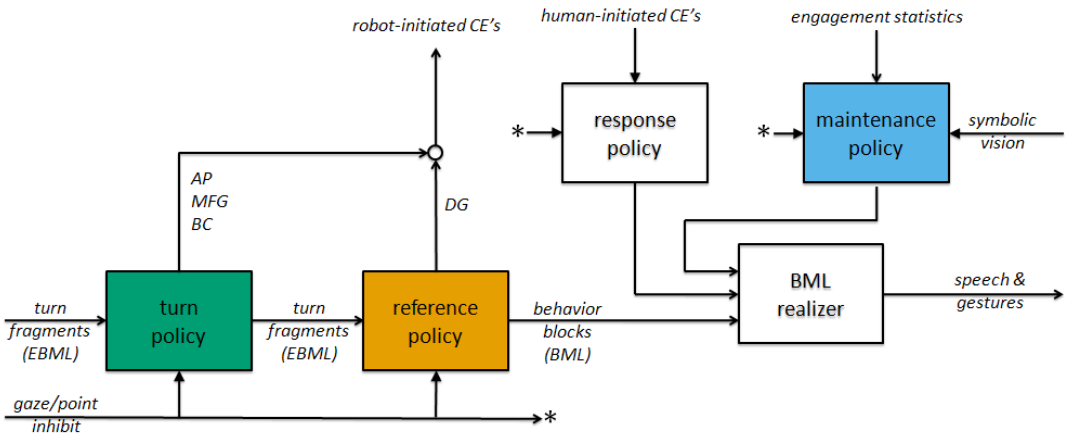


Figure 3.4: Diagram of generation policies

3.3.1 Turn Policy

The turn policy component (see Figure 3.4) consumes turn fragments from the collaboration manager and produces turn fragments that are consumed by the reference policy, optionally adding end-of-turn facial gaze gestures. It also notifies the recognition module of the start of robot-initiated adjacency pair, mutual facial gaze and backchannel events, as appropriate. Table 3.1 details the actions of the turn policy, which depend on the *turn index* and *turn status* values. These attributes are removed in the turn policy's output.

Turn Status	*Policy*
beginning	if first turn, initiate AP event
middle	
end	add facial gaze as last behavior initiate MFG event
full turn	if first turn, initiate AP event add facial gaze as last behavior initiate MFG event
backchannel	initiate BC event

Table 3.1: Turn Policy

In human conversation, the end of a speaker's turn is typically signaled simultaneously in three ways: first, the speaker stops speaking; second, the speaker looks at

the hearer (mutual facial gaze); and third, the speaker ends the final utterance of the turn at the lowest fundamental frequency of the entire turn (a *prosodic* signal). Our turn policy currently takes care of the first two signals, but expects the incoming turn fragments to already contain the appropriate prosodics. In the future, with continuing improvements in text-to-speech technology, it may be possible for the turn policy component to automatically adjust the prosodics of turn fragments.

3.3.2 Reference Policy

The reference policy decides on the best combination of gaze, pointing, and speech to use, in order to reference an object in the world. The reference policy component (see Figure 3.4) consumes turn fragments, possibly containing object references (EBML), from the turn policy and produces standard BML behavior blocks, which are then consumed by the BML realizer. The reference policy also notifies the recognition module of the start of robot-initiated directed gaze events, as appropriate.

An *object reference* comprises:

- an object identifier (uniquely identifying an object visible in the shared space) for the target object,
- the minimum required reliability ($0 < R_{min} < 1$), and
- a non-empty set of referring phrase options, each of which specifies:
  - a phrase to be spoken (may be empty),
  - the distractor set after speaking the phrase,
  - the cost of speaking the phrase, and
  - whether the phrase is deictic.

The reference policy processes turn fragments by "passing through" the standard BML elements, removing the object reference elements, and replacing them with a choice of one of the phrase options, possibly with the addition of a gaze and/or pointing gesture.

Figure 3.2 is an example of a turn fragment input to the reference policy which, under certain conditions, produces the BML output shown in Figure 3.3. Notice that the reference policy has replaced each object reference (`ref-1` and `ref-2`) by a speech element chosen from its speech options (`speech-2` and `speech-4`, respectively), and added a gaze (`gaze-1`), a pointing gesture (`point-1`) and two synchronization constraints.

Distractor Sets

The concept of a *distractor set*, which is central to the reference policy, originates in natural language generation, specifically the generation of referring phrases [31]. Given some universe of objects, such as the tangram pieces on a table, a referring phrase and a target object, the distractor set is the set of objects that are consistent with the referring phrase, but are *not* the target object.

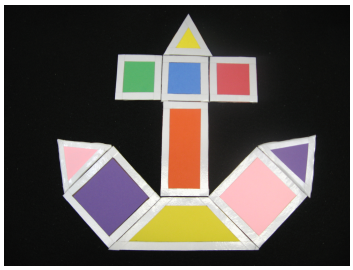


Figure 3.5: Tangrams.

For example, for the universe of ten tangram pieces in Figure 3.5, which includes a pink, a purple and a yellow triangle, the referring phrase "the triangle" (first speech option for `ref-1` in Figure 3.3) has two distractors, whereas the distractor set for "the yellow triangle" (the second speech option) is empty. As we will see below, I generalize the concept of distractor sets to gestures, such as gaze and pointing. The distractor set for pointing is often empty, but if the pointer is large and/or far away, even pointing can be ambiguous.

Reliability

The reliability of an object reference is intuitively the likelihood that it will be correctly understood, which is represented as a number between 0 and 1. The *minimum required reliability* may differ between object references due, for example, to the difference in importance for the collaborative task of correctly identifying the object in question.

In a world of perfect communication channels, the reliability of a referring phrase would depend only on the size of the distractor set—if it is zero, then the reliability is 1. However, in the real world, communication channels have noise, which means that the net communication reliability is always less than 1. The same analysis applies to the "channel" reliability of gestures—the lighting could be poor, the view could be partially obscured, etc., all of which could contribute to misunderstanding an otherwise unambiguous gaze or pointing.

Formally, the reliability of a gesture-speech pair is

$$R(j,s) = \frac{1}{|D(j,s)| + \frac{1}{r(j,s)}}$$

where

- $D(j,s)$ is the distractor set after performing (j,s), and
- $0 < r(j,s) < 1$ is the channel reliability of (j,s).

To understand the logical structure of this formula, consider first the hypothetical case where the channel reliability is 1. The overall reliability is then the likelihood that the correct object will be chosen by random chance out of the set which includes the target object and the distractors. On the other hand, suppose the distractor set is empty; the overall reliability is then just the channel reliability.

To simplify implementation of this policy, I assume that the channel reliability of gestures and speech are independent and can be characterized by simple parameters, i.e.,

$$r(j,s) = r(j) + r(s)$$

where

$$r(s) = \begin{cases} \lambda & : 0 \\ otherwise & : r_s \end{cases}$$

and

$$r(j) = \begin{cases} \lambda & : 0 \\ G & : r_G \\ DG & : r_{DG} \\ DGP & : r_{DGP} \end{cases}$$

The subscripted r parameters can be tuned depending on the robot's configuration and relation to the human. Furthermore, I expect in general that $r_G < r_{DG} < r_{DGP}$.

The distractor set after performing a gesture-speech pair can be decomposed as

$$D(j, s) = D_g(j) \cap D_p(j) \cap D_s(s)$$

where

- $D_g(j)$ is the distractor set after executing the gaze portion, if any, of j
- $D_p(j)$ is the distractor set after executing the pointing portion, if any, of j
- $D_s(s)$ is the distractor set after uttering the possibly-empty referring phrase s

The distractor sets after executing a gesture also depend on the robot's configuration and relation to the human. Currently, I compute $D_g(j)$ by intersecting the gaze cone of the robot with the table top and including all objects that fall within the resulting ellipse (minus the target object). I compute $D_p(j)$ by similarly intersecting a wide cylinder aligned with the pointing forearm of the robot with the table top. If there is no gaze or no pointing in j, or s is empty, then the respective distractor sets are the universe minus the target object.

As we have seen above, the distractor sets associated with each non-empty speech option, $D_s(s)$, are provided as input to the generation module by the collaboration manager as part of the object reference. It is up to the discourse and natural language facilities of the collaboration manager to decide how many different speech options to

provide and to compute the distractor sets for each, based on the semantics of the natural language and the application domain.

Two particular types of referring phrases, both illustrated in Figure 3.2, are particularly useful. The last speech option in both object references in this figure are the *deictic* phrase, "this piece." The `deictic="true"` attribute of this speech option sets the distractors to be the universe minus the target object, and constrains every gesture-speech pair which includes this referring phrase to also include a pointing gesture. Since deictic phrases are often much shorter (cost less) than more descriptive referring phrases, especially in complex domains, they will often be chosen as part of an optimal gesture-speech pair.

The third speech option for `ref-1` in Figure 3.3 is the *anaphoric* reference, "it." The collaboration manager would only provide this option when the discourse context uniquely resolves this pronoun to the target object. The distractor set for this option is thus empty, which makes it an attractive choice. However, if $r_s < R_{min}$, the reference policy will still need to add a gesture.

Cost

The cost of a gesture-speech pair is approximatley the time it takes to complete the behaivors. I decompose the cost of a gesture-speech pair as

$$C(j, s) = k_g C_g(j) + k_p C_p(j) + k_s C_s(s)$$

where

- $C_g(j) \geq 0$ is the cost of moving the robot's head from its current position to gaze at the given object,
- $C_p(j) \geq 0$ is the cost of moving the robot's appropriate hand from its current position to point at the given object,
- $C_s(s) \geq 0$ is the cost of uttering the given referring phrase, and
- $C_g(\lambda) = C_p(\lambda) = C_s(\lambda) = 0$

Currently, I compute $C_g(j)$ and $C_p(j)$ by adding together the number of degrees of rotation required in each of the joints to achieve the required movement. The k_g, k_p and k_s coefficients in the cost formula above are parameters of the engagement generation module, which can be used to tune the policy for different robots and configurations. $C_s(j)$ is provided as input to the generation module as part of the object reference.

Optimal Gesture-Speech Pair

The reference policy chooses the minimum cost combination of gesture and speech which satisfies the minimum reliability required for the given object reference. Formally, this is the gesture-speech pair $(j, s) \in J \times S$, which is

$$\underset{R(j,s) \geq R_{min}}{\operatorname{argmin}} \; C(j, s)$$

where

- $R(j, s)$ is the reliability function
- $C(j, s)$ is the cost function
- S is the set of speech options (λ is the empty phrase) ($s \in S$),
- $J = \{\lambda, G, DG, DGP\}$ ($j \in J$), and
  - λ means no gesture,
  - G means a gaze gesture,
  - DG means a directed gaze,
  - DGP means a directed gaze with pointing.

If no gesture-speech pair satisfies the minimum required reliability, then a most reliable pair is chosen and a warning message is sent to collaboration.

Notice that the first two cost functions above refer to the "current position" of the robot's head and hands. This means that the optimal sequence of gesture-speech pairs for a sequence of object references may be different from the sequence of pairs chosen by

optimizing the references one at a time. For example, if a single turn fragment contains two object references, such as the example in Figure 3.2, the sequence optimization should consider whether or not pointing at the first object leaves the hand in a good position for pointing at the second object.

To generalize this policy to optimize sequences of object references, I add the robot state as an explicit third argument to the cost functions, so that the optimal sequence of gesture-speech pairs (j_i, s_i) for a sequence of object references is

$$\underset{R(j_i,s_i)\geq R_{min_i}}{\operatorname{argmin}\ (j_i, s_i)} \sum_{i=1..n} C(j_i, s_i, T(j_{i-1}))$$

where $T(j_i)$ is the robot state after executing gesture j_i and $T(j_0)$ is the starting state of the sequence.

3.3.3 Response Policy

Both the reference and turn policies, discussed above, concern robot-initiated behavior (mostly at the behest of the collaboration manager). In contrast, the response policy component (see Figure 3.4) deals with the robot's response to human behaviors, specifically to the start of human-initiated directed gaze and mutual facial gaze events. The current policy for how the robot should respond to these behaviors is simple: the robot always looks where the human directs and meets the human's facial gaze, except when gaze/point inhibition is active.

Notice that the response policy does not include either providing the second turn of a human-initiated adjacency pair or responding with a backchannel to a human utterance. Both of these cases require the higher cognitive abilities (collaboration manager) of the robot because they concern application content, such as the correct answer to the human's question or whether or not the robot understands or agrees with what the human is saying.

3.3.4 Maintenance Policy

Because engagement is about how people "*maintain* ... their perceived connection," [1] a key component of the generation module is the engagement maintenance policy. Currently, this policy is only concerned with the robot's gaze. In the future (see Section 6.2), I plan to add nodding and other nonverbal behaviors to the robot's engagement maintenance repertoire.

The maintenance policy (see Figure 3.4) is organized as the following set of priorities regarding where the robot looks (starting with the highest priority):

1. obey gaze inhibit input
2. obey gaze behaviors of reference and turn policies
3. a quick glance at the human's face if the elapsed time since the last connection event exceeds the mean time between connection events (MTBCE)
4. obey gaze behaviors of response policy
5. track the human's hands if they are moving
6. track the human's face

These priorities are based on our own observational study of human engagement [5] and other human gaze interaction research cited above. The quick glance (3) is an optional MFG to determine if the person is still paying attention to the same object or the task at hand.

The two external inputs to the maintenance policy (see Figure 3.4) are symbolic vision, which supports the hand tracking, and statistics from the engagement recognition module [2], which provide the MTBCE.

4 BML Realizer

In this chapter, I build upon research in the virtual agent community by using the Behavior Markup Language (BML) [22] to control robots. Virtual agents such as [32, 33] were controlled using BML. Using BML for robots is as powerful as for virtual agents, since many robots and virtual agents interact with people similarly.

This chapter describes the benefits of applying BML to robots and the problems that needed to be solved. First, I present a motivating example for using an event-driven approach, based on Petri nets [34]. Second, I describe why a fixed-timing algorithm won't work for robots, based on the fundamental difference between animation being controlled open-loop and robots being controlled closed-loop. I then review the basic specifications of BML and BML realizers and explain our algorithm in detail. Finally, I discuss limitations and future improvements of this work.

4.1 BML Basics

A BML block is a set of behaviors and synchronization constraints on those behaviors. BML is represented using XML, where each behavior is an element. Each behavior has 7 synchronization points (*sync points*) during its execution. A BML realizer consumes a BML block and ensures that the agent performs the given behaviors, while satisfying the given constraints. A BML realizer has two phases: planning and execution.

4.1.1 Behaviors

A *behavior* is an atomic action that the agent can perform. Examples include gazing at an object, pointing, speaking, and changing the robot's facial expression. Each behavior is represented as an element with an identifier, attributes and sub-elements specifically for that behavior. For example, the gaze behavior has a target to look at, an offset angle, and which parts of the body to use (some attributes are optional).

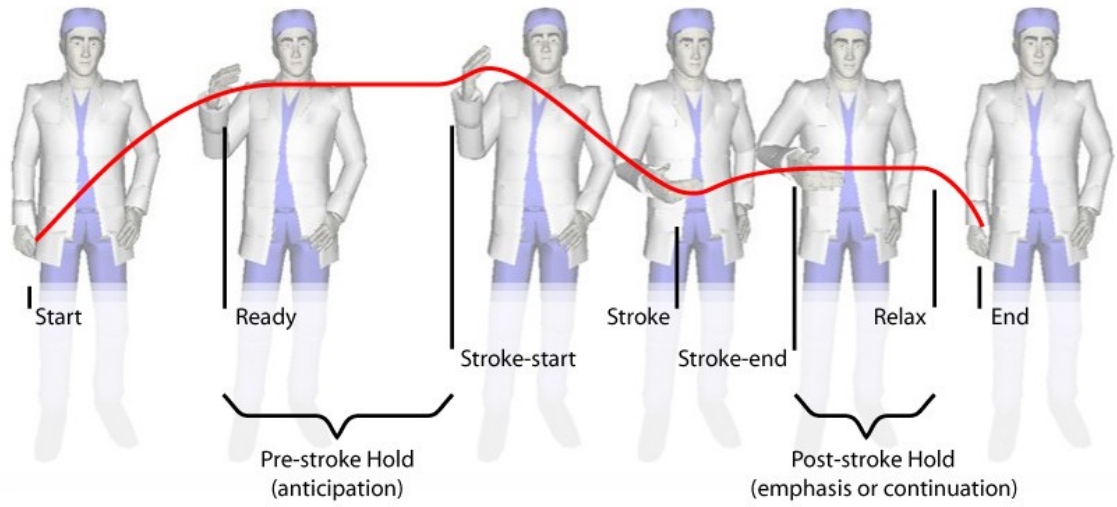


Figure 4.1: Sync points in a beat behavior (taken from [22])

4.1.2 Sync Points

The execution of a behavior has 7 sync points: start, ready, stroke_start, stroke, stroke_end, relax, and end. The constraint element uses these sync points to schedule the behaviors relative to one another. Not every behavior changes the motion of a joint during each sync point (e.g., the gaze behavior's start, ready, and stroke_start happen at the same time).

To explain what each sync point means, consider the beat behavior example in Figure 4.1. During this behavior, the *start* sync point is when the arm begins to move from its resting position. The *ready* sync point occurs when the arm is almost at its highest point in the air. The *stroke_start* sync point begins as the arm begins falling (to emphasize what is being said). The *stroke* sync point is when the arm is at a local minimum, defining the beat. The *stroke_end* sync point occurs when the arm is raised after the beat gesture. The *relax* sync point is when the arm is moving back toward a "home position." Finally, the *end* sync point occurs when the arm reaches its home position and stops moving.

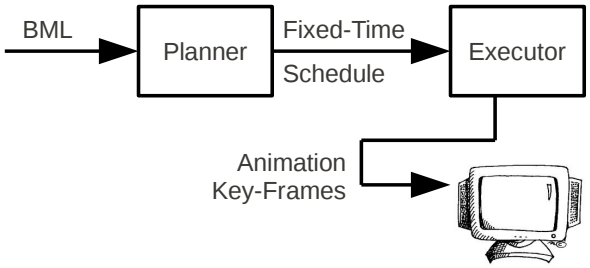


Figure 4.2: Animation BML realizer architecture

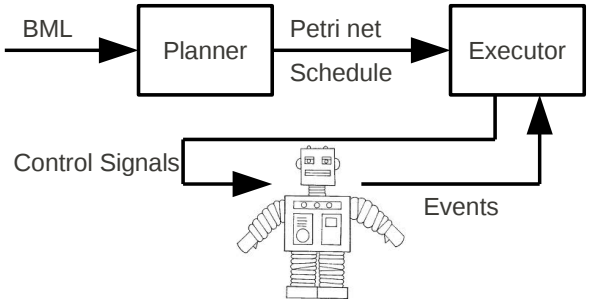


Figure 4.3: Robot BML realizer architecture

4.1.3 Planning and Execution

All BML realizers have two phases: planning and execution (see Figure 4.2 and Figure 4.3). In the planning phase, all the behaviors are interpreted and scheduled. The planning phase creates the shortest possible schedule which satisfies the constraints, and passes this schedule to the execution phase. The execution phase then uses this schedule to perform the behaviors in real time.

4.2 Motivating Example

Imagine a person ambiguously asks a robot to pick up a soda can when there are two cans on the table. The robot might respond by saying "this can?," while simultaneously looking and pointing toward one of the cans. The looking and pointing behaviors need to be synchronized correctly in relation to the speech for a natural, human-like performance.

```
<bml id="bml-example-1"
     xmlns="org.mindmakers.bml">
  <gaze id="gaze-1"
        target="can-1"/>
  <gesture id="point-1" type="POINT"
           target="can-1"/>
  <speech id="speech-1">
    <text>This can?</text>
  </speech>
  <constraint id="constraint-1">
    <synchronize ref="gaze-1:stroke">
      <sync ref="point-1:start" />
    </synchronize>
    <synchronize
        ref="point-1:stroke_start">
      <sync ref="speech-1:start" />
    </synchronize>
  </constraint>
</bml>
```

Figure 4.4: Example BML block

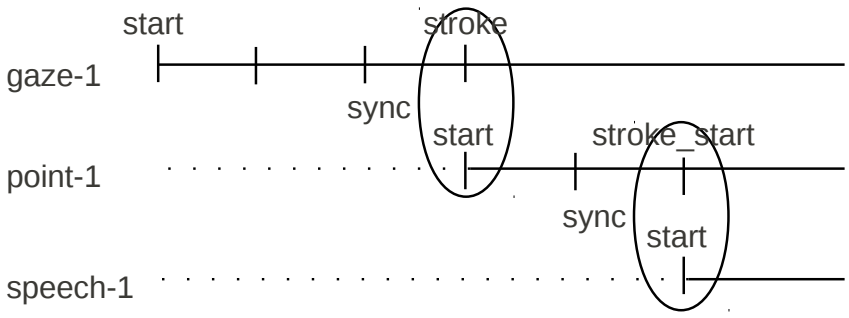


Figure 4.5: The desired timeline of sync points specified by Figure 4.1

The example BML in Figure 4.4 shows the behaviors involved in this example and the constraints between them. The example contains a single gaze behavior to look at one of the cans, along with a point gesture behavior. The speech behavior contains the text "this can?" to be utterred during the other behaviors. The constraints ensure that the point gesture does not start until the robot is looking at the object, and that the speech starts when the robot is beginning to point toward the object.

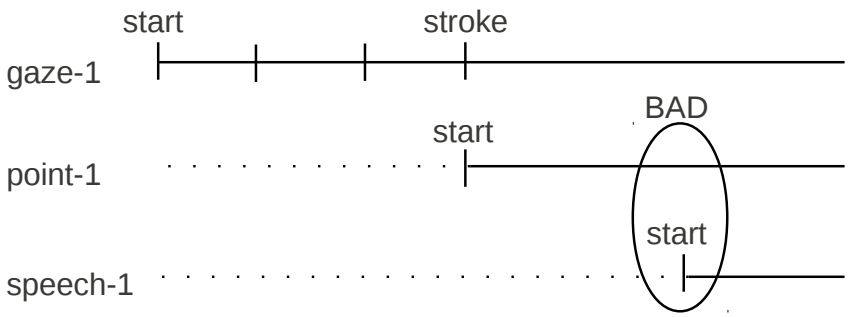


Figure 4.6: Arm is slowed after start of point behavior

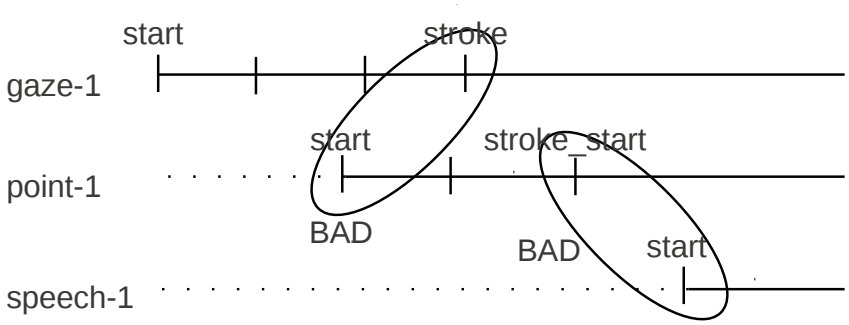


Figure 4.7: Arm is closer to target than expected

4.2.1 Desired Timeline

Based on the constraints in the above example, the timeline in Figure 4.5 is created by current BML realizers (see Section 1.4). This timeline satisfies the constraints that the `point-1:start` and `gaze-1:stroke` occur at the same time and that the `point-1:stroke_start` and `speech-1:start` occur together.

In an animation, since the time between each of the sync points can be *chosen*, fixed-timing algorithms are easy to implement. Also, in the animation setting, the timeline is guaranteed[1] to execute properly. In robotics there are several problems that can occur, which cause fixed-timing approaches not to work.

[1]Animation engines can skip frames if necessary to guarantee timing in the presence, for example, of varying CPU load.

4.2.2 Problematic Outcomes in Robots

The first, and most common, problem is when the motion of a joint motor is slower than expected (see Figure 4.6), perhaps because it is stalled or the battery is partially drained. In this case the algorithm needs to delay the other motions dynamically and wait. For example, the difference between Figure 4.4 and Figure 4.6 requires that the realizer wait for the arm to reach the correct position before starting to speak. A second example problem (see Figure 4.7), occurs when the robot's arm reaches its target position *earlier* than expected, perhaps because the arm was closer to the target than the robot believed.

4.3 Solution

Considering the example above and the possible problems presented, I have developed an event-driven BML realizer for robots. An alternative real-time solution would be to completely re-plan the fixed-time schedule as new data is acquired, but this requires a lot of processing power and time (which is usually limited in robotic applications).

A robotic realizer is fundamentally different from an animation realizer in that it must be controlled closed-loop. Figure 4.3 shows how the events must be returned from the control specific module. The output of the planning stage in this approach is a Petri net [34] representing a minimal time event-based schedule.

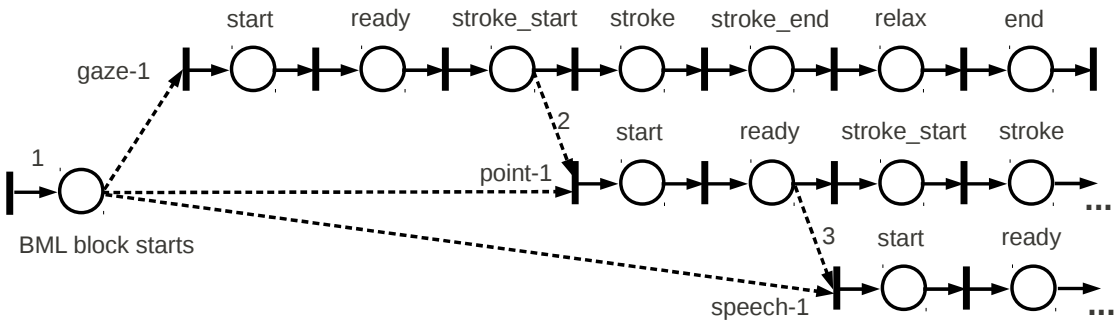


Figure 4.8: The Petri net for Figure 4.1

4.3.1 Petri Net Planning Example

Figure 4.8 shows the Petri net schedule resulting from Figure 4.4. A Petri net consists of places (represented as circles), transitions (represented as vertical bars), and tokens which are transmitted between places and transitions. Each transition and place waits for all incoming tokens and sends one token out on each arrow. Note the "BML block starts" transition 1 that starts the entire execution.

Individual Behaviors

Our planning algorithm starts by creating a separate Petri net for each behavior that is given. Ignoring the dotted lines, Figure 4.8 contains three such sub-nets, (only the `gaze-1` behavior is completely shown due to limits on space). If arrow 2 between `gaze-1:stroke_start` and `point-1:start` is not present, the gaze behavior continues to execute one sync point after another until the end.

Synchronizing Behaviors

Adding a *synchronize* constraint between behaviors corresponds to adding a single transition to the Petri net. The constraint synchronizing the `gaze-1:stroke` and `point-1:start` sync points is achieved by adding arrow 2. Note that this arrow is after `gaze-1:stroke_start` because the following two sync points must begin together.

Before and After Behaviors

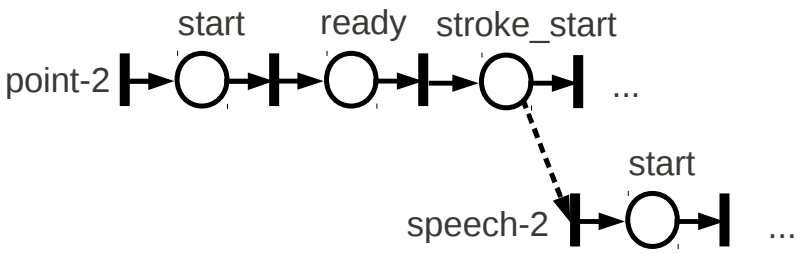
Adding a *before* or *after* constraint is done in approximately the same way as synchronize constraints. Modifying the second constraint in Figure 4.4 to be an *after* constraint would result in the BML and Petri net of Figure 4.9. Note that the *after* constraint is the inverse of a *before*, thus a constraint of

```
1 <before ref="speech-2:start">
2     <sync ref="point-2:stroke_start"/>
3 </before>
```

is equivalent to the constraint in Figure 4.9(a).

```
1 <bml id="bml-example-2"
2      xmlns="org.mindmakers.bml">
3   <gesture id="point-2" type="POINT"
4            target="can-2"/>
5   <speech id="speech-2">
6     <text>This can?</text>
7   </speech>
8   <constraint id="constraint-2">
9     <after ref="point-2:stroke_start">
10      <sync ref="speech-2:start" />
11    </after>
12  </constraint>
13 </bml>
```

(a) Synchronization using "after"



(b) Partial Petri net of (a)

Figure 4.9: Adding "after" constraint to a Petri net

Timed Behaviors

Adding timing offsets can be accomplished by adding one place and one transition. Modifying the example in Figure 4.9, if I constrained the speech to start 2 seconds after the `point-1:stroke_start` (see Figure 4.10 the planning algorithm would add a place and a "sleep" transition[2] for between the `point-1:ready` and `speech-1:start`. The new Petri net is shown in Figure 4.10(b).

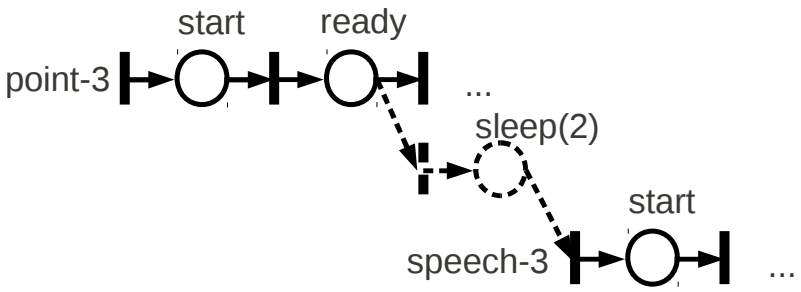
4.3.2 Execution of Petri Nets

Our BML executor is implemented in Java using the ROS framework. Compared to planning, the execution phase in our approach is fairly simple. For example, starting

[2]A transition that delays the token being passed for a specified amount of time.

```
1 <bml id="bml-example-3"
2      xmlns="org.mindmakers.bml">
3   <gesture id="point-3" type="POINT"
4            target="can-3"/>
5   <speech id="speech-3">
6     <text>This can?</text>
7   </speech>
8   <constraint id="constraint-3">
9     <synchronize
10         ref="point-3:stroke_start + 2">
11       <sync ref="speech-3:start" />
12    </synchronize>
13  </constraint>
14 </bml>
```

(a) Synchronization using timing offset



(b) Partial Petri net of (a)

Figure 4.10: Adding timing delay to a Petri net

with 1 in Figure 4.8, the executor calls an ROS service provided by a robot-specific control ROS module for each sync point of each behavior. This control module sends joint commands to the robot and returns events when the given sync point is reached. Each fan-out after a place creates a new thread of execution. Similarly, each transition joins its input threads. While fanning out can create a large number of threads, this has not been a problem in practice because Java threads are lightweight.

4.4 Implementation Limitations

I imposed a few limitations on the input BML block for easier initial implementation, none of which restrict the power of the approach. Non-conforming BML blocks can always be rewritten to conform. The limitations are: negative time, circular synchronizations, and multiple synchronizations for a single behavior. Each of these limitations, discussed in the following sections, can be removed in a given BML block by transforming the BML either manually or computationally. I will also point out why these limitations are not an issue for fixed-time realizers.

4.4.1 Negative Time

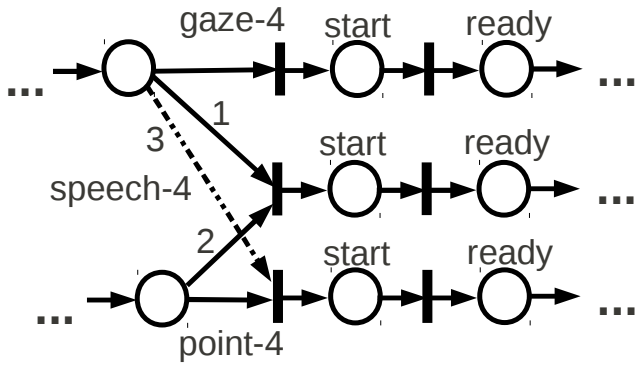
In the event-driven planning algorithm sketched above, sync points cannot be synchronized by *subtracting time*. For example, if a gaze behavior needs to start 2 seconds before a point behavior starts, the constraint must be rewritten such that the point behavior starts 2 seconds after the gaze behavior instead. This constraint is automatically repaired in the current implementation, and therefore allows all timing offset constraints. Fixed-time planners solve this by backtracking in the schedule.

4.4.2 Circular Synchronizations

The Petri net representing the sync point constraints must be acyclic in our current implementation. One could imagine a legal cycle in a realizer where, for example, a gaze must start with a point, the point must start with a speech, and the speech must start with the original gaze. In this case, the cycle must be eliminated by rewriting the constraints such that the point and speech both rely on the start of the gaze, otherwise a deadlock may be created. Fixed-time planners simply start multiple behaviors at the same time.

```
1 <bml id="bml-example-4"
2       xmlns="org.mindmakers.bml">
3   <gaze id="gaze-4"
4         target="can-4"/>
5   <gesture id="point-4" type="POINT"
6            target="can-4"/>
7   <speech id="speech-4">
8     <text>This can?</text>
9   </speech>
10  ...
11  <constraint id="constraint-4">
12    <synchronize ref="gaze-4:start">
13      <sync ref="speech-4:start" />
14    </synchronize>
15    <synchronize ref="point-4:start">
16      <sync ref="speech-4:start" />
17    </synchronize>
18    ...
19  </constraint>
20 </bml>
```

(a) Multiple synchronizations to same behavior



(b) Partial Petri net of (a)

Figure 4.11: Adding multiple synchronizations of a single behavior

4.4.3 Multiple Synchronizations

Another type of problem occurs when a sync point has a large number of synchronization constraints. To temporarily solve this, each transition is allowed only two inputs: the previous sync point and one other.[3] Figure 4.11 shows three behaviors that are

[3] *Start* sync points rely on the BML block start

constrained to start together, where the third is synchronized to the other two (see arrows 1 and 2). In this case, there is no cycle, but the first two behaviors also need to be synchronized to each other to ensure that none of the three starts ahead of another. The second synchronize (in bold in Figure 4.11(a)) would be rewritten as

```
1    <synchronize ref="gaze-4:start">
2        <sync ref="point-4:start" />
3    </synchronize>
```

to replace arrow 2 with arrow 3. Fixed-time realizers can simply start all the behaviors at the same time.

4.5 Discussion

I now discuss how animation realizers and our robotic realizer deal with the structure of the humanoid body, conflicting behaviors, and the possible errors that can occur during execution. All of these issues bear on how to separate the generic realizer implementation from the robot-specific control implementation.

4.5.1 Body Structure Dependence

The output of the planning phase in an animation realizer is a sequence of key-frames with the sync points being executed during each key-frame. Fixed-timing executors rely on a database of possible motions to display the behaviors. Animation realizers thus assume a fixed skeleton that is used for all realizers. Every robot is constructed with a different set of joints and thus there cannot be a single database of motions. Therefore, this implementation calls a separate robot-specific control module.

4.5.2 Conflicting Behaviors

Regardless of whether a realizer is for animation or for robots, a single joint cannot be given two different angles at the same time. Current animation realizers "blend," (in-

terpolate between) conflicting behaviors, e.g. a gaze and a nod. In our algorithm, since each robot is different, the joint conflicts must be detected in a robot-specific module. Since two of the same type of behavior are guaranteed to conflict, our implementation initially schedules behaviors of the same type in series in the order given in the BML block. Doing so reduces the number of behaviors that conflict, but must ignore the minimum time constraint. The constraints in the BML block take precedence over this initial scheduling.

4.5.3 Errors During Execution

The execution phase in animation does not have errors since they should have been caught in the planning phase, but with robotics this may not be the case. For example, if a network connection goes down and the robot can no longer be controlled, or if a motor overheats and the system shuts the motor down to cool, the realizer may have to abort a behavior and propagate the error to the calling module. Also, safe error handling during behavior execution needs to be addressed to ensure that the robot is not left in an undesirable state. For example, while pilot testing for the task presented in [35], the realizer threw an error and our robot's arms were left hanging in the air. Expanding upon the conflicting behaviors issue, if there is a conflict which cannot be solved by the robot-specific control module, then those errors are propagated as well. For example, a gaze and a head behavior both require the neck motors.

5 Validation

In order to validate the model of engagement generation, a human-robot study was developed where the robot instructed the human to create a tangram. In the operational condition the robot's behaviors are controlled through the generation module described above. In the degraded generation condition, the generation module does not generate any behaviors, but instead the system acts solely on the input from the task model. In the same study as these two conditions, a third degraded recognition condition was run to validate the recognition module developed by Ponsler [2].

5.1 System Architecture

Figure 5.1 shows the setting of our current architecture and implementation, which mirrors the setting of the human engagement study, namely a human and a humanoid robot with a table of objects between them. Either the robot or the human can be the initiator (or responder) in the connection event timelines shown in the previous section.

Like the engagement maintenance part of the human study, mobility is not part of this setting. Unlike the human study, we are not dealing here with manipulation of the objects or changes in the human's or the robot's stance (e.g., turning the body to point to or manipulate objects on the side part of the L-shaped table, see Figure 5.3).

Both the human and the robot can perform the following behaviors and observe them in the other:

- look at the other's face, objects on the table or "away"
- point at objects on the table
- nod the head (up and down)
- shake the head (side to side)

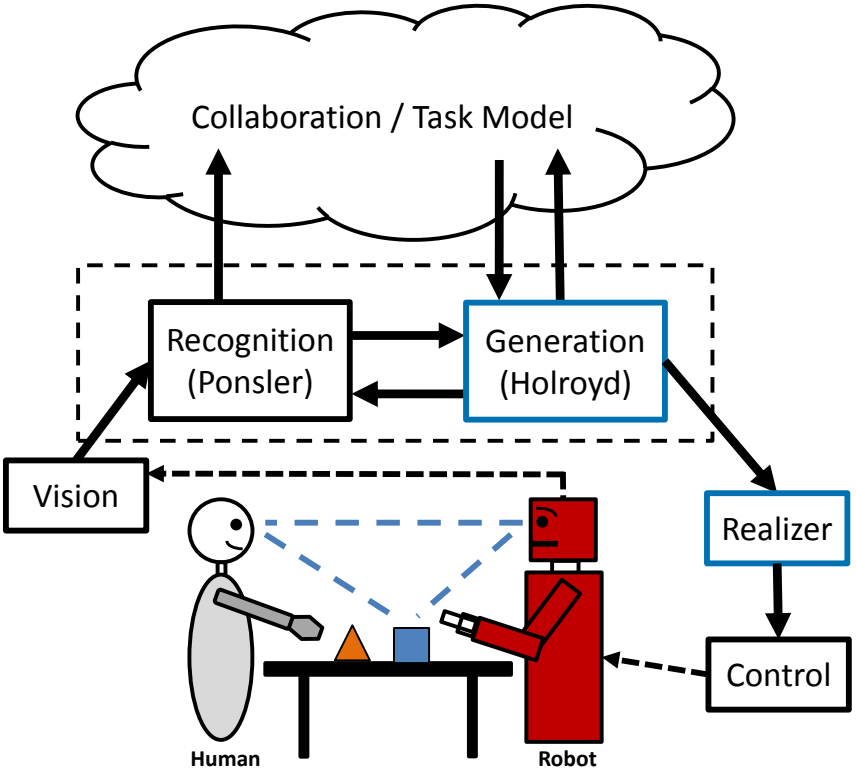


Figure 5.1: System Architecture Diagram

The robot can generate speech that is understood by the human. However, our current system does not include natural language understanding, so the robot can only detect the beginning and end of the human's speech.

5.1.1 ROS architecture

The architecture is built using the Robot Operation System (ROS) [4] which is a multi-language library for message passing between multiple processes on multiple machines. Each message that can be passed is created separately and can be used in new messages. ROS also has the notion of a service, which is where one node (a process) sends a message to another node (which could be itself) and then receives a response message.

```
1 # Unique string identifier for the actor
2 engagement_msgs/ActorID actor
3
4 # A flag which determines between DONE and NOT_DONE
5 engagement_msgs/Flag done
6
7 # The sequence of entities
8 engagement_msgs/Entity[] objects
9 ---
10 # Use a Flag message as a response
11 engagement_msgs/Flag result
```

Figure 5.2: Robot initiated directed gaze ROS service example

Figure 5.2 shows a sample ROS service file for the robot initiated directed gaze service provided by the recognition module and used by the generation module. Lines 1-8 show the request sent from the generation module to the recognition module, and lines 10-11 are the response back to the generation module. In this service, the generation module sends the actor (the human) who should be responding to the directed gaze, the flag NOT_DONE to signify that the event is about to start, and the objects which the actor should look at. Upon completion of the directed gaze, recognition will return whether the connection event succeeded, failed, or invalid data was given, e.g. the actor does not exist.

5.1.2 Control

The control module calculates joint positions for the robot based on the behaviors being executed. The control module in this architecture is based on the control module described in [3]. I modified this module in order to incorporate the ROS architecture and implement the BML behaviors described in Chapter 4.

5.1.3 Vision

The vision module detects faces and task specific objects. Faces are detected using Watson [36] which gives location and orientation in 3-space. A second module detects

the tangram pieces in this situation. All of this information is then combined to give a model of the world.

5.1.4 Recognition

The main job of the engagement recognition module, from the standpoint of engagement generation, is to notify the generation module of human-initiated connection events, so that the generation module can respond appropriately. For example, when the human points to an object, engagement recognition recognizes the start of a human-initiated directed gaze event (see Section 2.2), which the generation module completes by generating a robot gaze at the object. Symmetrically, generation also notifies recognition when the robot initiates a connection event, such as directed gaze. For more details on engagement recognition, including the information flow between recognition and collaboration, see [2].

5.1.5 Task Model

The collaboration manager and task model are the same module in this architecture. This module implements the algorithms and task description described in Section 5.2.1.

5.1.6 Generation

The main job of the engagement generation module, the first main implementation contribution of this thesis, is to generate nonverbal behavior that contributes to engagement between the human and the robot. The main input to generation is a real-time stream of *turn fragments* from the collaboration manager. The details of turn fragments are described in Section 3.2; basically, these are the speech and gestures which the collaboration manager has decided are appropriate based on the current task state. The main output of generation is a real-time synchronized stream of speech and gestures to be performed by the robot, most of which come from the turn fragment inputs, but which also include gestures added to enhance engagement.

5.1.7 Realizer

The second major implementation contribution of this thesis is an implementation of a BML realizer. The BML realizer in Figure 3.4 was specially implemented for this project and is, to our knowledge, the second BML realizer (after [28]) to control a physical robot rather than an animated virtual character and the first event-driven realizer. This module takes, as input, a BML block and ensures that the robot performs the behaviors specified with the given constraints.

5.2 Study Design

In this study we ran a between participant study in which we compare the control condition of degraded generation to the operational condition. We developed a game using tangrams in which ten pieces are assembled to create an anchor. The participants are given an instruction sheet (see Appendix A), are brought into the room to interact with Melvin, are given a questionnaire (see Appendix C), and then are debriefed about their experience. During the interaction each participant was filmed from two angles (one focused on Melvin and one focused on the participant) and the system recorded the connection events as they occurred.

5.2.1 Tangram Game

Tangrams are puzzles that, most often, young children put together to create simple pictures out of simple shapes. We call each shape a piece, though some people refer to the shapes as tans, and each piece is colored. To make the vision problem a little easier, we added retro-reflective tape to the edge of the pieces. Doing this caused the edge of the pieces to always look white, since the lights were just behind the camera.

We added some rules which are not normally used in creating tangrams to also make the vision problem easier. First, touching edges must be parallel and of the same length. This simplification removed the possibility that corners were touching edges.

The second simplification is that each piece can be disambiguated through speech. For example, there are no two pieces that are both purple and a square.

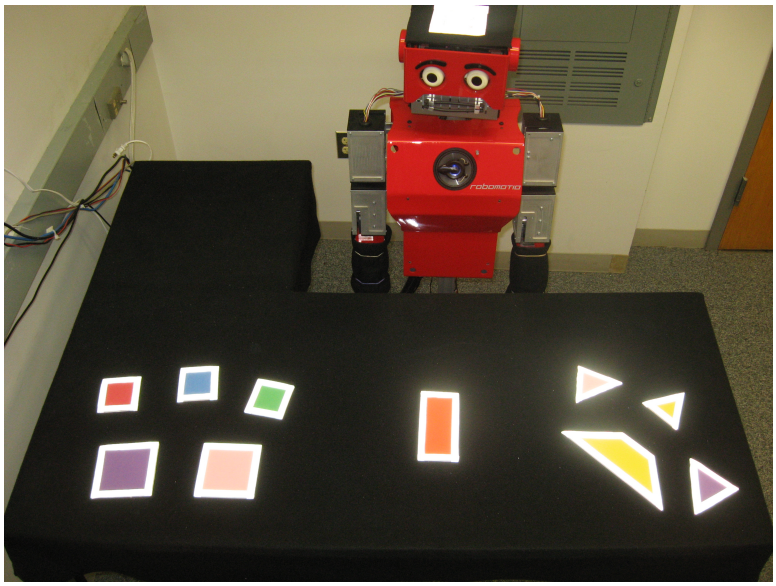


Figure 5.3: Disassembled Anchor Puzzle

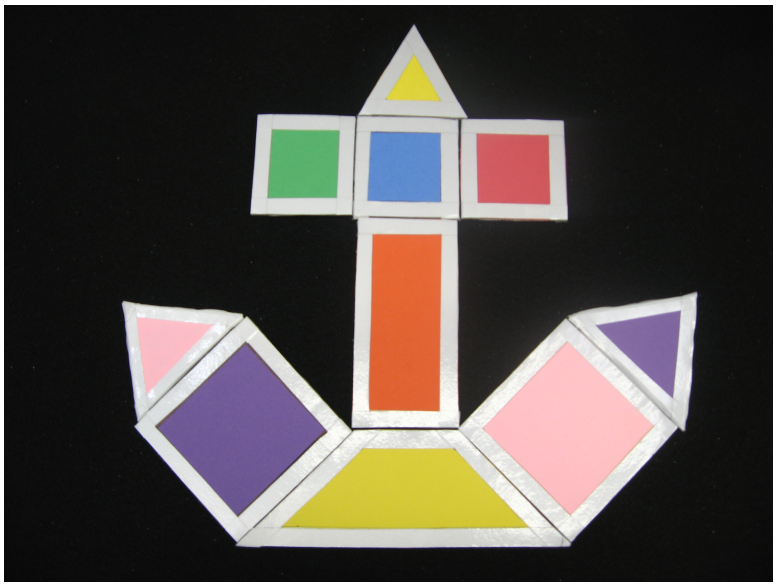


Figure 5.4: Completed Anchor Puzzle

Anchor Puzzle

We chose to instruct the human to create an anchor because it is difficult to figure out and has many pieces. We did not want people to be able to figure out the puzzle because they would then complete the puzzle without Melvin's help. We also needed many pieces in order to have a sufficient number of connection events per interaction. Figure 5.3 shows the puzzle as it is when the human first walks in and Figure 5.4 shows the completed puzzle.

Sample Interaction

Appendix B shows a sample interaction transcript from the study. In this example, the participant and the system make no mistakes during the interaction. The participant enters the room just before this transcript begins, and leaves to take the questionnaire right after. Note that the interaction has a large number of connection events because the tangram game itself requires them.

5.2.2 Hypotheses

During the study design we developed a series of hypotheses. We believe that in the operational condition as compared to the degraded generation condition the following will hold:

1. (a) Participants will report that Melvin is more human-like.
   (b) Participants will report that the interaction with Melvin is more fluent and natural.
2. Participants will report that Melvin understands what they did more often.
3. Participants will make more eye contact.
4. Participants will perform more backchannel events.
5. Participants will produce more utterances.
6. The mean time between adjacency pairs will be less.

7. The average adjacency pair delay will be less.

8. The puzzle completion time will be less.

9. Participants will make fewer mistakes.

10. Participants will report that it is easier to identify objects Melvin referred to.

We also believe that in the degraded case:

11. Participants will make progressively less eye contact during the interaction.

Questions 1a, 1b, 2, 3, and 10 will be validated using the questionnaire. Questions 3, 6, 7, 8, and 11 will be validated using the system statistics. Questions 4, 5, and 9 will be validated by observing the participants and video analysis.

5.2.3 Questionnaire

We constructed a questionnaire based on the original Interactive Experiences Questionnaire used by Sidner *et al.* [1] and Lombard *et al.* [37]. The questions were compared to the hypotheses, and we used the questions which should support at least one hypothesis. The selected questions were randomized and given to each participant with some personal information questions. The full questionnaire can be found in Appendix C.

5.3 Pilot Study

Before we ran the full study described above, we first ran a pilot study. In this study, during the degraded generation case we kept Melvin's arms above the table and his head pointed down at the table. When we gave a preliminary questionnaire to the participants, there were a large number of participants who indicated that Melvin had pointed at objects. Since, in the degraded generation case Melvin does not point at objects, we modified the condition and the questionnaire by keeping his arms below the table, having him stare to the upper right, and rewording questions in the hopes that they are less ambiguous.

5.4 Results

We have not had time to process all of the results, but the questionnaire has given us statistically significant results on many questions. The full results for each question is provided in Table D.1 and Table D.2 after using a two-tailed p-test. We conducted a between-subjects study with 29 participants, ages 18 to 26, in the two conditions.

	Operational	Degraded
male	10	12
female	4	3

Table 5.1: Participant distribution

The results of the questionnaire are very supportive of our hypotheses, but we have not had time to go over the system data or videos collected. However, we believe that there was either confusion with the questions themselves, or confusion during the interaction, as evidenced by the confound of vision errors in question 19.

5.4.1 Hypothesis Support

This experiment shows that the generation policies made a significant improvement in the robot's performance as a collaborator. Questions 11, 12, and 21 attest to the robot being perceived as more human-like (hypothesis 1a). Questions 1, 2, 9, 10, 15, 17, 18, 21, and 24 show proof of Melvin performing the correct actions for a fluent and natural interaction (hypothesis 1b). The results from questions 7 and 26 show that participants felt that they made more eye contact with the robot (hypothesis 3). Finally, the results of 6, 8, 14, 20, and 22 show that participants felt that the actions Melvin took made it easier to identify objects (hypothesis 10), but that the participants in each case were still able to identify the objects, as evidenced by questions 14, 16, and 18.

These results support our belief that to interact with people, robots should have fluent and natural behaviors that are similar to, if not exactly like, human behaviors. The results also show that people will reciprocate to connection events that the robot initiates.

The following quote was written by a participant in the comments section of the questionnaire:

> *"Melvin was great to interact with. The one moment that kind of took me out of the feeling of interaction was when I put the piece in what I was sure was the right spot, but Melvin didn't think so. He asked me to fix the piece so I rotated it 90 degrees (which did nothing in effect because it was a square) and put it back where I had placed it before. He then accepted this. Other than that one moment, I felt engaged with Melvin, even grinning and almost laughing at times out of disbelief that I was actually interacting with a robot in this way so naturally."*

5.4.2 Vision Errors

There were a large number of different errors that caused the entire interaction to be slightly degraded from what we would consider to be the proper interaction. First, we had imperfect edge and corner detection of the tangram pieces in the vision software because of imperfect vision conditions, e.g. lighting conditions, color detection, etc. Second, people sometimes covered pieces when we were not expecting it, interfering with vision updates. Third, the calculated relations of pieces relative to one another were sometimes not the correct relations. For example, the vision system would sometimes detect the blue square as "touching the top left" of the rectangle instead of just "touching the top."

5.4.3 Control Errors

Besides the vision issues, there were a few control issues which were less than perfect. The joints were not calibrated as well as they could be, which lead to the participant believing that Melvin was pointing slightly off center of pieces. The motions were also not quite as fluid as they could be. For example, when the arm was returning to Melvin's side, the arm would sometimes run into the table. The neck movements

were also not fluid. The last noticeable difference was that our speech synthesis system would sometimes not produce sound, and instead Melvin's lips would just move without speech.

6 Conclusion

To summarize, in this thesis I have constructed a theory of engagement using connection events, shown how a humanoid robot can generate the connection events, created the first event-driven BML realizer, and validated the implementations of these theories through a human study. The implementations of the engagement generation and BML realizer modules are open-source and provided at `http://sourceforge.net/projects/ros-engagement/` as the "ROS engagement stack." The "engagement_recognition" package is provided by Ponsler [2], the "engagement_generation" package and "engagement_realizer" packages are provided by me.

6.1 Improvements

Besides the errors in Section 5.4.2 and Section 5.4.3 being fixed, we also believe that there are a few changes that we could make. These improvements include modifying the design of the study, the questionnaire, and the system.

6.1.1 Within Participant Design

When we first developed the above study, we only considered using a between subject study instead of including the possibility of within subject. While the between subject design is often used because it produces cleaner results, we believe that a within subject study may yield better results. If we choose to use this type of study, we will either have to (a) create three puzzles for the participant to complete or (b) have the participant complete only a part of the puzzle in each condition. In option a, the new study would have to ensure that each order of the puzzles and each order of the conditions were included, making a total of 36 options which would need to be run. In option b, the new study would require modification to our current collaboration module, but would

only require 6 options of the different orderings of the conditions. To find statistical results in condition b, we would most likely need to have a total of 18 participants, half of the total for option a.

6.1.2 Inapplicable Questions

The second issue, which we did not foresee, was that many of the questions were confusing to the participants because they did not apply to the condition they were in. For example, Question 3 assumes that Melvin looked at objects, but in the degraded generation case Melvin does not look at anything. In this case the participant could answer the question with any answer and still provide a logical rational for choosing that answer. We believe that this can be solved by either using a within subject design, or by rewording the questions.

6.1.3 Processing Speed

Another issue we we found, but believed was not an issue, was that our algorithm for determining the next piece to move delayed the interaction. We found that this was an annoyance to the participants in the pilot study, but they did not indicate that it was important. Instead of having Melvin stay quiet while the next piece algorithm was running, about two seconds per move, we instructed Melvin to say "umm" like people do when they are thinking. Most of our participants considered this a nice addition, and said that it made Melvin feel more human, though they did not say this in the questionnaire but after in the debriefing session. Regardless of the participants comments, we would attempt to increase the efficiency of the next piece algorithm to speed up the interaction, before running another study. We could also use a Wizard of Oz approach, such as manually specifying which pieces are touching where, to improve performance instead of using the next piece algorithm.

6.2 Future Work

Our first order of business is to conduct some additional pilot studies to tune the various parameters in the policies, such as the cost coefficients in Section 3.3.2. Stepping back, there are a number of small improvements in architecture and policies we would like to explore. For example, as mentioned in Section 3.3.3, we currently require the collaboration manager to handle the generation of backchannels, such as confirmatory nods. We would like to find a way to off-load this behavior to the generation module. One theory we have is that backchannels can also be invited, perhaps by pauses or other actions the speaker is taking. We would also like to explore using our robot's—albeit limited—abilities for emotional expression (eyebrows and mouth) to contribute to engagement.

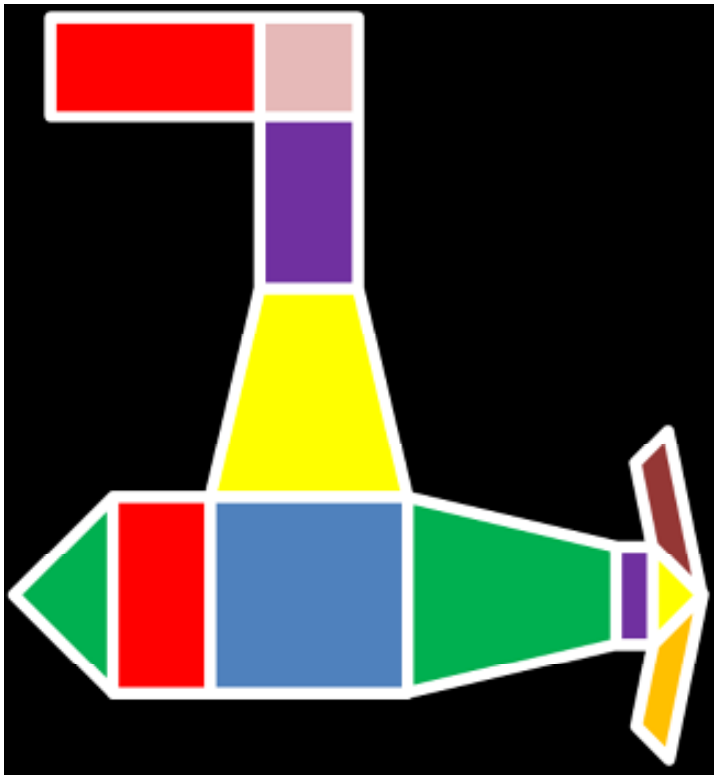
There is much more work to be done on strategies and policies for initiating and terminating engagement. A (very) challenging test case for this work would be for a robot to start up a conversation with a chosen person at a cocktail party, and then later gracefully end the conversation.

Finally, this current model of engagement, applies only to one human and one robot. Relaxing this restriction and exploring models for multi-party interactions would help the field, and may include work on stance based on Kuzuoka's research [38]. I also believe that the models we have developed also have something to say about human interaction with non-humanoid robots. For example, applying these engagement models to an iRobot Create or Packbot would be interesting test cases.

A Tangram Explanation

Tangrams

You will interact with a humanoid robot, Melvin, to construct a **tangram**. A tangram is a puzzle containing several flat, colored pieces arranged to form a specific picture. For example, the twelve pieces below make a submarine picture (this is **not** what you will be making!).

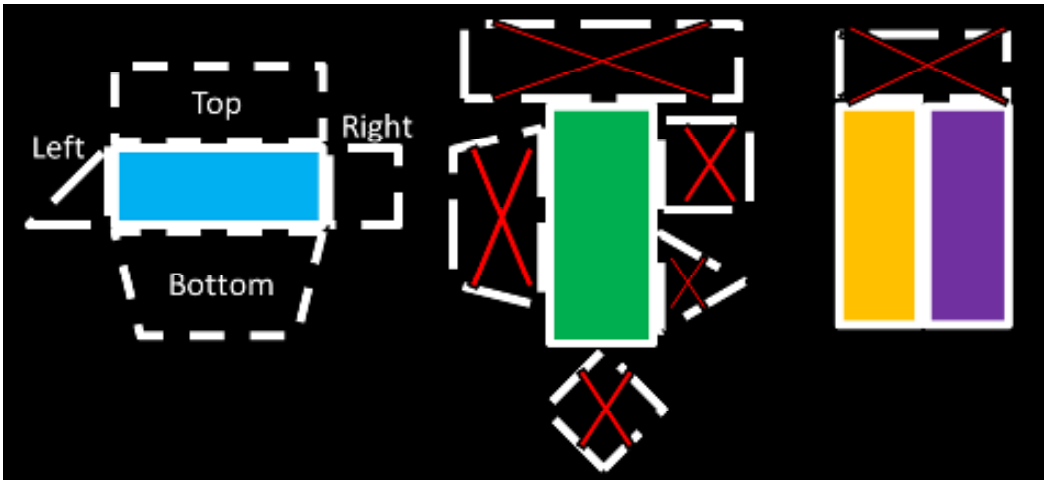


During the interaction, Melvin will **direct you where to place pieces** one-by-one, but will **not** tell you the name of the final picture until you have finished.

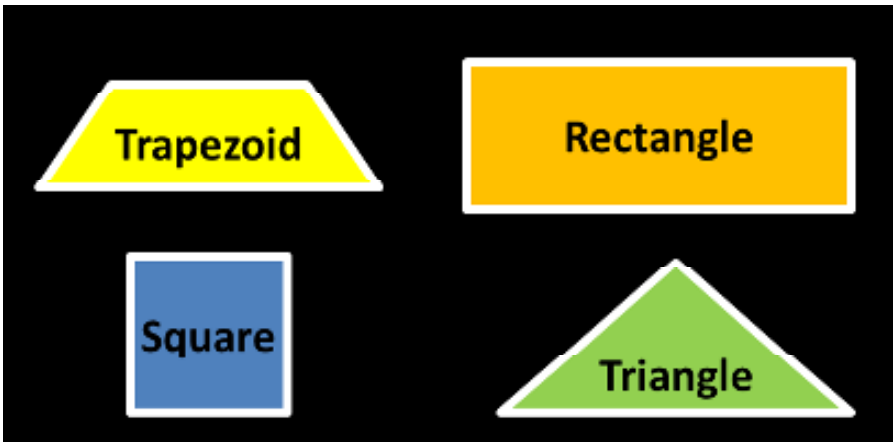
- Melvin can hear you speak, but **cannot understand** what you say.
- Please keep your hands in your **lap** while you are not moving the pieces.

There are some constraints on how pieces may be placed.

- Pieces **must remain flat** on the surface of the table and may not overlap.
- **All** the pieces will be used to construct the final picture.
- Properly aligned pieces should have **exactly one edge** of one piece touching **exactly one edge** of another piece. The edges that touch **must be equal** length. The pieces shown with red X's are i**ncorrectly** placed.



Note: Melvin will use the terms 'top', 'bottom', 'left', and 'right' as they pertain to **your** point of view.



Finally, prior to beginning:

1) Please remove any watches, bracelets, or anything else that is on **either** of your wrists.
2) Please roll up the sleeves of your shirt to your elbow.

B Sample Interaction Transcript

- <*Action taken by participant*>, [**action taken by Melvin**], "Melvin's synthesized speech"
- [**DG at ***], [**MFG**], and [**AP**] indicate the start of the respective connection events, directed gaze, mutual facial gaze, and adjacency pair
- ***** Indicates a failed connection event, + Indicates an optional connection event
- **Note**: adjacency pairs were initiated every time Melvin spoke, even if a response was not required.

1. <*Looks at the table*> <*sits on stool*> <*facial gaze*> <*looks at the table*> <*facial gaze*>
2. [**AP**][**facial gaze**] "Hello, let's make a tangram." [**MFG+**]
3. [**AP**]"We are going to use the pieces on the table <*scans the table*> to make a picture together. As you can see, <*facial gaze*> on the table we have: squares, triangles, <*scans the table*> [**DG at rectangle**] a <*facial gaze*> rectangle, and <*looks at the rectangle*> [**DG with pointing at trapezoid**] a <*looks at the trapezoid*> trapezoid of <*facial gaze*> different colors [**MFG+**]." <*looks at the table*>
4. [**AP**][**facial gaze**] "Okay, <*scans the table*> let's start." [**MFG+**]
5. [**AP**][**facial gaze**] "Please move [**DG with pointing at blue piece**] the <*looks at the blue piece*> blue piece [**facial gaze**] so <*grabs the blue piece*> that it is touching the top of [**DG at rectangle**] the rectangle." <*looks at the rectangle*> [**MFG+**]
6. <*Moves the blue square to the top of the rectangle*> [**looks at the human's hand**]<*facial gaze*>

7. [**AP**]<*Looks at the table*> [**facial gaze**] "Ummmm, [**looks at the table**] good." <*facial gaze*> [**MFG+**]
8. [**AP**][**facial gaze**] "Please move [**DG with pointing at yellow triangle**] the <*looks at the yellow triangle*> yellow triangle [**facial gaze**] so that it is touching the top of [**DG at blue piece**] the blue piece <*looks at the blue piece*>." [**MFG+**] <*grabs the yellow triangle*>
9. <*Moves the yellow triangle to the top of the blue piece*> [**looks at the human's hand**] <*facial gaze*>
10. [**AP**]<*Looks at the table*> [**facial gaze**] "Ummmm, <*facial gaze*> [**looks at the table**] good." [**MFG+**]

⋮ ⋮ ⋮

11. [**AP**]<*Looks at the table*> [**facial gaze**] "Please point to the piece you would like to move next." [**MFG+**]
12. <*Points to the pink square*> <*facial gaze*> <*looks at the pink square*> <*facial gaze*> [**looks at the pink square**]
13. [**AP**][**facial gaze**] "Please move [**DG with pointing at pink square**] the <*looks at the pink square*> pink square [**facial gaze**] <*facial gaze*> so that it is <*looks at the table*> touching the top right of [**DG at trapezoid**] the trapezoid. Make sure <*grabs the pink square*> the square is touching the slanted edge of the trapezoid." [**MFG+**] <*looks at the table*>
14. <*Moves the pink square to the top right of the trapezoid*> [**looks at the human's hand**]
15. [**AP***][**facial gaze**] "Ummmm, [**looks at the table**] good." [**MFG+**]<*facial gaze*>
16. [**AP**][**facial gaze**] "Please point to the piece you would like to move next." [**MFG+**]
17. <*Looks at the table*> <*facial gaze*> <*looks at the purple triangle*> <*points to the purple triangle*> <*facial gaze*> [**looks at the human's hand**]

18. [**AP**][**facial gaze**] "Please move [**DG* with pointing at purple triangle**] that [**facial gaze**] <*facial gaze*> so that it is touching the top right of <*looks at the table*> [**DG at pink square**] the pink square." <*looks at the purple triangle*> [**MFG+**]

19. <*Grabs the purple triangle*> <*moves the purple triangle to the top right of the trapezoid*>

20. [**AP**]<*Looks at the table*> [**facial gaze**] "Ummmm, <*facial gaze*> [**looks at the table**] good." [**MFG+**]<*looks at the table*>

21. [**AP**][**facial gaze**] "Please <*facial gaze*> move [**DG with pointing at green square**] <*looks at the green square*> this [**facial gaze**] <*facial gaze*> so that it is touching the left side of <*looks at the table*> [**DG at blue piece**] the <*facial gaze*> blue piece <*looks at the table*> ." [**MFG+**]

22. <*Looks at the green square*> <*grabs the green square*> <*moves the green square to the left of the blue piece*> <*facial gaze*> [**looks at the human's hand**]

23. [**AP**][**facial gaze**] <*facial gaze*> "Ummmm, <*looks at the table*> [**looks at the table**] <*facial gaze*> good." [**MFG+**]

24. [**AP**]"Look, we made an <*looks at the table*> anchor." [**MFG+**]

25. [**AP**]"Thanks <*facial gaze*> for playing. That was fun, goodbye." [**MFG+**]

26. <*Looks at the table*> <*facial gaze*> <*looks at the table*> <*leaves the stool*>

C Tangram Questionnaire

C.1 Likert Scale Questions

1. Melvin looked at the table and the puzzle pieces at appropriate times.

Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree

2. The interaction felt natural all the time.
3. I always knew what object Melvin looked at.
4. Melvin was reliable.
5. Melvin pointed at objects during the interaction.
6. The puzzle piece descriptions were easy to understand.
7. I looked at Melvin's face often.
8. I always knew what object Melvin pointed at.
9. Melvin slowed me down during the interaction.
10. Melvin always looked at me in a natural way.
11. Melvin seemed more like a human than a robot.
12. I like Melvin.
13. Melvin responded appropriately to my actions.
14. I always knew what object Melvin talked about.
15. I spent a lot of time waiting for Melvin to tell me what to do.
16. I could easily identify the objects that Melvin referred to.
17. There were awkward paused during the interaction when I wasn't sure what was supposed to happen next.

18. I always understood Melvin's instructions.
19. I spoke to Melvin during the interaction.
20. I could easily tell which objects Melvin looked at.
21. Melvin makes me feel comfortable, as if I am with a friend.
22. I easily found the puzzle piece that Melvin described to me.
23. Melvin always understood what I was doing.
24. Melvin looked at me at appropriate times.
25. Melvin always knew what I was doing when I pointed to or moved a piece.
26. I made eye contact with Melvin frequently.
27. I could easily tell the object that Melvin pointed to.
28. Melvin looked at me during the interaction.

C.2 Personal Experience Questions

How old are you?

Please indicate your gender.

What is your level of education?

How much do you know about robots?

Additional comments about Melvin or the interaction.

D Tangram Results

Table D.1: Questionnaire Results Summary

	Mean		Standard Deviation		
Question	Operational	Degraded	Operational	Degraded	p-value
1	6.2	4.5	1.1	2.6	.026*
2	4.8	3.9	1.1	1.2	.047*
3	6.1	3.8	1.6	2.7	.011*
4	5.7	5.3	1.4	1.5	.483
5	7.0	1.6	0.0	1.6	.000***
6	6.9	6.9	0.3	0.5	.692
7	6.1	5.0	1.1	1.5	.038*
8	6.6	3.1	1.3	2.4	.000***
9	3.8	3.7	1.6	2.0	.939
10	4.6	2.7	1.2	1.8	.003**
11	3.4	2.3	1.4	1.4	.036*
12	6.4	6.3	0.85	0.98	.782
13	5.9	5.0	1.1	1.5	.074+
14	6.6	6.6	1.1	0.7	.945
15	3.8	4.0	1.4	1.3	.670
16	6.6	6.8	1.1	0.8	.655
17	4.4	4.2	1.1	1.7	.667
18	6.2	6.2	1.3	1.5	.978
19	1.8	1.6	1.8	1.3	.843
20	5.7	3.4	1.2	2.4	.003**
21	4.6	3.5	1.5	1.4	.047*
22	6.6	6.8	0.93	0.56	.583
23	5.6	5.3	1.2	1.6	.575
24	6.2	2.4	0.97	1.5	.000***
25	5.9	5.5	1.1	1.4	.415
26	5.4	4.0	1.8	1.6	.032*
27	6.4	3.1	1.2	2.3	.000***
28	6.2	2.6	1.1	1.8	.000***
Completion Time	5:45	5:17	1:36	2:26	0.550
Backchannels	1.1	1.1	1.4	2.6	0.995
Utterances	2.2	2.0	5.2	2.6	0.889
Vision Errors in Pointing Recognition	0.8	0.5	0.8	0.6	0.033*

+p-value < 0.1 (trending) *p-value < 0.05 (weakly significant)
**p-value < 0.01 (significant) ***p-value < 0.001 (highly significant)

Table D.2: Significant Factors

Question	Parameter	β	Std. Error	Sig
1	Operational?	1.745	.729	.024*
2	Operational?	0.902	.452	.057+
3	Operational?	2.165	.830	.015*
5	Operational?	5.438	.401	.000***
7	Operational?	1.076	.501	.042*
8	Operational?	3.583	.717	.000***
10	Operational?	1.860	.584	.004**
11	Operational?	1.123	.546	.050*
13	Operational?	0.954	.513	.075+
19	Vision Errors	-.300	.133	.033*
20	Operational?	2.331	.719	.003**
21	Operational?	1.079	.539	.056+
24	Operational?	3.722	.457	.000***
26	Operational?	1.511	.623	.023*
27	Operational?	3.449	.695	.000***
28	Operational?	3.571	.611	.000***

Table D.3: Support for Hypotheses

Hypothesis	Questions Averaged	β	Std. Error	Sig
1a	12, 21, 11	0.770	.335	.030*
1b	18, 10, 21, 17, 2, 15, 9, 24, 1	1.040	.291	.001***
2	23, 13, 25	0.553	.427	.207
3	7, 26	1.294	.519	.020*
10	8, 20, 14, 6, 22	1.418	.379	.001***

Bibliography

[1] C. L. Sidner, C. Lee, C. Kidd, N. Lesh, and C. Rich, "Explorations in engagement for humans and robots," *Artificial Intelligence*, vol. 166, no. 1-2, pp. 104–164, 2005.

[2] B. Ponsler, "Recognizing engagement behaviors in human-robot interaction," Master's thesis, Computer Science Department, Worcester Polytechnic Institute, Worcester, Mass., USA, December 2010.

[3] A. Holroyd, B. Ponsler, and P. Koakiettaveechai, "Hand-eye coordination in a humanoid robot," major qualifying project, Computer Science Department, Worcester Polytechnic Institute, Worcester, Mass., USA, 2009.

[4] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[5] C. Rich, B. Ponsler, A. Holroyd, and C. Sidner, "Recognizing engagement in human-robot interaction," in *Proc. ACM Conf. on Human-Robot Interaction*, (Osaka, Japan), Mar. 2010.

[6] C. L. Sidner, C. D. Kidd, C. H. Lee, and N. Lesh, "Where to look: a study of human-robot engagement," in *ACM International Conference on Intelligent User Interfaces (IUI)*, pp. 78–84, ACM, 2004.

[7] D. Bohus and E. Horvitz, "Learning to predict engagement with a spoken dialog system in open-world settings," in *Proceedings of the SIGDIAL 2009 Con-*

ference, (London, UK: Association for Computational Linguistics), pp. 244–252, Sept. 2009.

[8] D. Bohus and E. Horvitz, "Models for multiparty engagement in open-world dialog," in *Proc. SIGDIAL 2009 Conference*, (London, UK: Association for Computational Linguistics), pp. 225–234, Sept. 2009.

[9] C. Peters, "Direction of attention perception for conversation initiation in virtual environments," in *Proc. 5th Int. Conf. Intelligent Virtual Agents*, (Kros, Greece), pp. 215–218, 2005.

[10] H. Sacks, E. A. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, no. 4, pp. 696–735, 1974.

[11] D. Devault, C. Rich, and C. L. Sidner, "Natural language generation and discourse context: Computing distractor sets from the focus stack," in *17th Int. Florida Articial Intelligence Research Symp.*, (Miami, FL), pp. 887–892, May 2004.

[12] M. Stone, C. Doran, B. Webber, T. Bleam, and M. Palmer, "Microplanning with communicative intentions: The spud system," *Computational Intelligence*, vol. 19, no. 4, pp. 311–381, 2003.

[13] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita, "Footing in human-robot conversations: How robots might shape participant roles using gaze cues," in *Proc. ACM Conf. on Human-Robot Interaction*, (San Diego, CA), 2009.

[14] D. Crystal, *The Cambridge Encyclopedia of Language*. Cambridge, England: Cambridge University Press, 1997.

[15] F. Flippo, A. Krebs, and I. Marsic, "A framework for rapid development of multimodal interfaces," in *Proc. 5th Int. Conf. Multimodal Interfaces*, pp. 109–116, Nov. 2003.

[16] M. Argyle and M. Cook, *Gaze and Mutual Gaze*. Cambridge University Press, New York, 1976.

[17] A. Kendon, "Some functions of gaze direction in two person interaction," *Acta Psychologica*, vol. 26, pp. 22–63, 1967.

[18] S. Duncan, "Some signals and rules for taking speaking turns in conversations," *Journal of Personality and Social Psychology*, vol. 23, no. 2, pp. 283–292, 1972.

[19] C. Goodwin, "Gestures as a resource for the organization of mutual attention," *Semiotica*, vol. 62, no. 1/2, pp. 29–49, 1986.

[20] Y. Nakano, G. Reinstein, T. Stocky, and J. Cassell, "Towards a model of face-to-face grounding," in *Proc 41st Meeting of Assoc. for Computational Linguistics*, (Sapporo, Japan), pp. 553–561, 2003.

[21] H. H. Clark, *Using Language*. Cambridge: Cambridge Univ. Press, 1996.

[22] S. Kopp, B. Krenn, S. Marsella, A. Marshall, C. Pelachaud, H. Pirker, K. Thrisson, and H. Vilhjlmsson, "Towards a common framework for multimodal generation: The behavior markup language," in *Intelligent Virtual Agents* (J. Gratch, M. Young, R. Aylett, D. Ballin, and P. Olivier, eds.), vol. 4133 of *Lecture Notes in Computer Science*, pp. 205–217, Springer Berlin / Heidelberg, 2006.

[23] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore, "Beat: the behavior expression animation toolkit," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, (New York, NY, USA), pp. 477–486, ACM, 2001.

[24] A. Kranstedt, S. Kopp, and I. Wachsmuth, "Murml: A multimodal utterance representation markup language for conversational agents," in *Porc. of the AAMAS Workshop on Embodied conversational agents Lets specify and evaluate them!*, 2002.

[25] B. De Carolis, C. Pelachaud, I. Poggi, and M. Steedman, "APML, a Markup Language for Believable Behavior Generation," in *Life-Like Characters: Tools, Affective Functions, and Applications* (H. Prendinger and M. Ishizuka, eds.), pp. 65–86, Springer, January 2004.

[26] P. Piwek, B. Krenn, M. Schrder, M. Grice, S. Baumann, and H. Pirker, "Rrl: A rich representation language for the description of agent behaviour in neca," in *Porc. of the AAMAS Workshop on Embodied conversational agents Lets specify and evaluate them!*, 2002.

[27] H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. N. Marshall, C. Pelachaud, Z. Ruttkay, K. R. Thórisson, H. Welbergen, and R. J. Werf, "The behavior markup language: Recent developments and challenges," in *Proceedings of the 7th international conference on Intelligent Virtual Agents*, IVA '07, (Berlin, Heidelberg), pp. 99–111, Springer-Verlag, 2007.

[28] L. Q. Anh and C. Pelachaud, "Expressive gesture model for storytelling humanoid agent," in *The fourth workshop on Embodied Conversational Agents, WACA 2010*, (Lilles, France), pp. 25–26, November 2010.

[29] S. Brennan, "How conversation is shaped by visual and spoken evidence," *Approaches to Studying World-Situated Language Use*, pp. 95–129. Cambridge, MA: MIT Press, 1999.

[30] A. Dix, "Pace and interaction," in *Proceedings of the conference on People and computers VII*, (New York, NY, USA), pp. 193–207, Cambridge University Press, 1993.

[31] C. Gardent, "Generating minimal definite descriptions," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (Morristown, NJ, USA), pp. 96–103, Association for Computational Linguistics, 2002.

[32] M. Thiebaux and S. Marsella, "Smartbody: Behavior realization for embodied conversational agents," in *In 7th International Conference on Intelligent Virtual Agents (IVA)*, 2007.

[33] M. Kallmann and S. Marsella, "Hierarchical motion controllers for real-time autonomous virtual humans," in *Intelligent Virtual Agents* (T. Panayiotopoulos,

J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist, eds.), vol. 3661 of *Lecture Notes in Computer Science*, pp. 253–265, Springer Berlin / Heidelberg, 2005.

[34] C. Petri, *Communication with automata.* DTIC Research Report AD0630125, 1966.

[35] A. Holroyd, C. Rich, C. L. Sidner, and B. Ponsler, "Generating connection events for human-robot collaboration." Submitted to 20th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2011, Atlanta, GA, USA, Aug. 2011.

[36] L. P. Morency, A. Rahami, and T. Darrell, "Adaptive view-based appearance model," in *IEEE Conf. on Computer Vision and Pattern Recognition*, (Madison, WI), pp. 803–810, June 2003.

[37] M. Lombard, T. B. Ditton, D. Crane, B. Davis, G. Gil-Egui, K. Horvath, and J. Rossman, "Measuring presence: A literature-based approach to the development of a standardized paper-and-pencil instrument," *Presence 2000: The Third International Workshop on Presence, Netherlands*, 2000.

[38] H. Kuzuoka, Y. Suzuki, J. Yamashita, and K. Yamazaki, "Reconfiguring spatial formation arrangement by robot body orientation," in *Proc. ACM Conf. on Human-Robot Interaction*, (Osaka, Japan), Mar. 2010.