# Symmetric Multimodal Interaction in a Dynamic Dialogue

**Aaron Adler**
MIT CSAIL
32 Vassar St, 32-239
Cambridge, MA 02139 USA
cadlerun@csail.mit.edu

**Randall Davis**
MIT CSAIL
32 Vassar St, 32-237
Cambridge, MA 02139 USA
davis@csail.mit.edu

**ABSTRACT**
Two important themes in current work on interfaces are multimodal interaction and the use of dialogue. Human multimodal dialogues are symmetric, i.e., both participants communicate multimodally. We describe a proof of concept system that supports symmetric multimodal communication for speech and sketching in the domain of simple mechanical device design. We discuss three major aspects of the communication: multimodal input processing, multimodal output generation, and creating a dynamic dialogue. While previous systems have had some of these capabilities individually, their combination appears to be unique. We provide examples from our system that illustrate a variety of user inputs and system outputs.

**Author Keywords**
multimodal, dynamic dialogue, sketch recognition, sketch generation, speech

**ACM Classification Keywords**
H.5.2 Information Interfaces and Presentation (e.g., HCI): User Interfaces—*Input devices and strategies, Voice I/O, Natural language*

**INTRODUCTION**
Two important themes in current work on interfaces are multimodal interaction and the use of dialogue. Both of these aim to lower the cognitive load of communication. Multimodal interfaces lower cognitive effort by providing both familiar modalities and additional channels for communication. The benefits of dialogue in lowering cognitive effort are well established: In the absence of a dialogue, the speaker must anticipate and preemptively eliminate every ambiguity, and must ensure that the communication is both complete and unmistakably clear, an exhausting set of demands. Human conversation is (often) easy in part because we rely on the listener to ask when things are unclear.

In contrast to current computerized systems, human multimodal dialogue is *symmetric*: both participants communicate multimodally. We describe a proof of concept system that supports symmetric multimodal communication for speech and sketching, set initially in the domain of simple mechanical device design.

This paper focuses on three major aspects of the system: multimodal input processing, creating a dynamic dialogue, and multimodal output generation.

Our system is designed to interact with a user who is describing a simple mechanical device using sketching and speech. The system's task is to simulate the behavior of the device, using a qualitative physics simulator. The system asks the user for additional information whenever it determines that the current physical situation is unclear or ambiguous, or when the user's input has not been understood. The user's answers (delivered multimodally) update the physical model (or clarify a previous response), allowing the simulator to take the next step, which in turn affects which questions are asked next. The dialogue is thus driven from moment to moment by the physics, not by a prepared script.

The system asks its questions by generating multimodal output, for example, circling a spring and asking aloud "Which way does this spring move?" As we discuss below, there are several challenges in generating coherent simultaneous speech and pen output and timing them properly: Much like an orchestra score, both the correctness of the individual parts (sketching and speech) and their timing are vital to the composition.

We begin by discussing the motivation for our system, then briefly describe two user studies we performed and their key results that guide our research. The rest of the paper focuses on the components outlined above: multimodal input, multimodal output, and dynamic conversation.

**MOTIVATION**
Our group has developed many sketching systems that can be used in a variety of domains, such as mechanical systems [18], electric circuits [5], and chemistry diagrams [23]. These systems allow the user to communicate using a sketch, but some concepts of the designs remain difficult to communicate using only this medium. Newton's Cradle is a simple example that illustrates the limitations of sketching. This system of pendulums is designed to go through a series of collisions, and its successful functioning requires precise positioning of its component pendulums – they must be identical and touching (see Figure 1). This constraint is difficult to


IUI 2009 Workshop on Sketch Recognition
February 8, 2009, Sanibel Island, Florida
Chair: Tracy Hammond

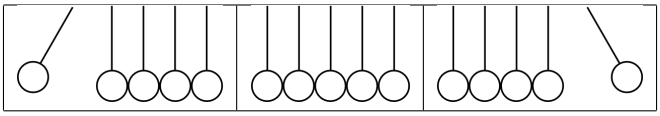


**Figure 1. A sequence of images showing Newton's Cradle when one of the pendulums is pulled back and released.**

sketch, but can easily be expressed verbally: "There are five identical and touching pendulums." Using this additional information, the sketch can be updated appropriately [2]. The multimodal combination of inputs expands the space of devices we can describe in our system.

While multimodality can reduce the frequency of error, it cannot of course eliminate it. The system may, for instance, interpret a spoken utterance to refer to four pendulums but if there are five drawn pendulums, there are two possible errors. The user could have said five and the speech recognizer could have made an error, or the user could actually be referring to a subset of the pendulums. In situations like this, a person would ask a question. We want our system to have a similar capability.

## USER STUDY 1: MULTIMODAL DEVICE DESCRIPTIONS

Two user studies have guided the development of our system. The first [2] was an empirical study of the informal speech and sketching of users describing a mechanical system. One participant was asked to draw a device on a whiteboard while talking about it to a silent listener. The study produced interesting observations about the language and timing people use when describing these systems. The observations most relevant to our current work are:

- Disfluencies ("ahh", "umm", etc.) and several key phrases indicated a new topic,
- A substantial pause in both modalities was likely an indication of a topic change,
- Participants never talked about one topic while sketching another.

This study led to an initial system [1] capable of handling sketching and speech, but it lacked the conversational capabilities to resolve uncertain inputs of the sort mentioned above.

## USER STUDY 2: HUMAN MULTIMODAL DIALOGUE

Our second study examined human-human dialogues with the goal of illuminating how two people sketch and talk when engaged in an ongoing conversation. The domain in this study was electrical circuit diagrams. The experimenter and participant each had a Tablet PC equipped with software that replicated what each of them drew on the other tablet, in effect giving them a single, shared sketching surface. They communicated with each other using only verbal communication and sketching on the Tablet PCs. The participant drew various digital circuits and described a class project from a digital circuit design class. Figure 2 is a sketch a participant

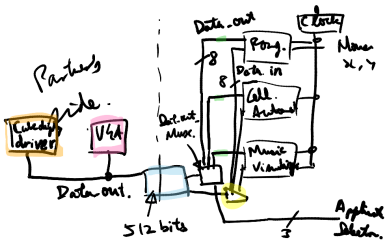



**Figure 2. A sketch of a participant's project from User Study 2.**

created when describing their class project. During these explanations, the experimenter asked simple questions about the device.

The software we created allowed the participant and experimenter to sketch using a pen or a highlighter of various colors. They could also use a pixel-based erase mode to erase parts of strokes. The software recorded the (x, y) position, time, and pressure for each drawn point. Two video cameras and headset microphones were used to record the study. The audio, video, and sketching were synchronized which allowed for playback and a detailed analysis of the results.

A subset of the audio data from the study was transcribed. We used a speech forced-alignment system to obtain precise timestamps in the audio track to complement the timestamps in the sketch data. The results of the study are described in detail in [3]; here we focus on three findings that guided the design of our current system: the use of color, the participants' speech, and the responses to the experimenter's questions.

The study revealed that pen color is important in interpreting the user's intention. Pen color was used in the sketches for several purposes:

- to refer back to existing parts of the sketch or link parts of the sketch together,
- to indicate a new topic as shown in the red and blue current paths in Figure 3,
- to reflect real world colors of objects.

The importance of ink color changes provides evidence that our dialogue system needs to recognize when a user changes ink color and similarly be able to generate appropriate computer ink color changes for the things it draws.

The speech observations from the second study echo the findings in the first study. First, the participants' speech was disfluent, especially when they appeared to be thinking about what to say. Figure 4 contains two typical fragments of speech from the study. Second, the responses to questions reused some of the vocabulary contained in the question. Finally, concurrent speech and sketching always referred to the same objects. This last observation is particularly relevant to our current work because it provides an interpretation for simultaneous input from different modalities.

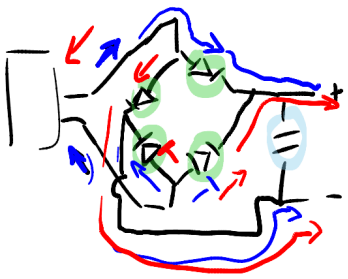
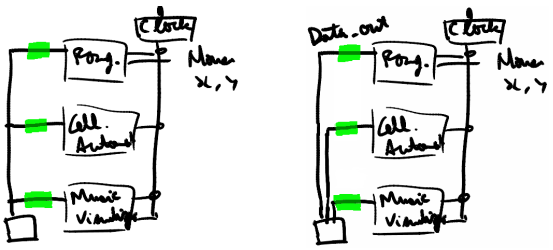


**Figure 3. A sketch from User Study 2 of an AC/DC transformer.**

| | |
|---|---|
| Experimenter: | so then what's what's um this piece what's that |
| Participant: | that would be the mux for the data input actually |
| Participant: | that was a uh uh yeah a memory bank with five hundred and twelve um yep five hundred and twelve bits this ah i could that i had read and write access to |

**Figure 4. Fragments of the conversation accompanying Figure 2. Notice the disfluencies and repeated words.**

Interesting answers resulted from the questions posed by the experimenter in the study. While the questions were simple, they produced lengthy, in-depth replies that went beyond simply answering the question. The participants also revised the sketches in response to the questions to make corrections or clarifications, as illustrated in Figure 5. The observed responses suggest that engaging the user in a conversation will do more than just resolve uncertainties in the physics simulation; we hypothesize that asking the user questions will engage them more deeply in the sketch and help them correct errors or clarify the design.



(a) Original (b) Revised

**Figure 5. Left: the original sketch, right: after revision. One data output line in the original image has been replaced by three in the revised image.**

## MULTIMODAL DYNAMIC DIALOGUE SYSTEM

The findings from the user studies have guided the design of the system. The current domain for the system is simple mechanical devices constructed from bodies, springs, pulleys, weights, pivots, and anchors. This domain enables the users to sketch a variety of devices, while still limiting the complexity of the physics. We have previously built several sketching systems using a similar domain [4].

The system's goal is to understand the design well enough to be able to simulate it. This is accomplished by using the power of the multimodal dialogue to resolve uncertainties and fill in missing details. The system analyses the physics and asks the user dynamic questions based on the current state and the user's previous answers. We plan to compare the human-computer interaction of our system with the human-human interaction in our studies and determine the effectiveness of the multimodal dialogue. Figure 6 illustrates the system components.

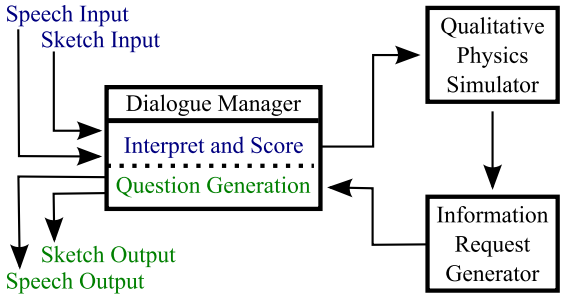



**Figure 6. The components of the multimodal dialgoue system.**

Figure 7 shows a screen shot of the user interface. Based on the observations about color changing in our user study, we provide the user with several pen and highlighter colors to choose from. At the bottom of the window the computer's outgoing speech and the user's recognized speech are displayed. The interface also processes the sketching input and output and the speech recognition and synthesis. The interface is written in c-sharp to easily interact with these components. The input and output data are processed in the core of the system, which is written in Java.

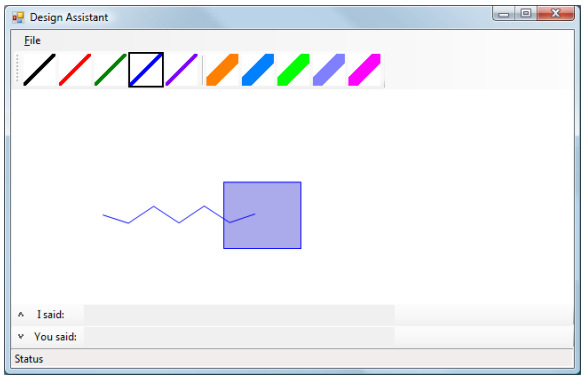



**Figure 7. The user interface of the multimodal dynamic dialogue system. The initial configuration of the block and the spring is shown.**

We use the Microsoft Speech Recognizer because it provides high-quality recognition results without requiring significant training. For the speech synthesis, we chose AT&T's Natural Voices for its realistic speech output. The incoming stylus data can easily be captured in c-sharp. We wrote our own

stroke generator so that the computer's strokes had the same rendering characteristics as the user's.

### Qualitative Physics Simulator

The physics simulator acts as a kind of inference engine, taking the current state of the world and trying to predict the next state. When the next state cannot be determined unambiguously, the system creates a set of possible questions to ask the user. The answers provide additional data that is used to update the system's model and allow it to continue simulating the device.

The physics simulator itself has two important properties: it is qualitative, and modest. As a qualitative simulator it uses only directions of velocities and accelerations, not their magnitudes. This is still useful, as the system is designed to allow users to describe early-stage designs, a stage when requiring them to enter precise velocities and masses would detract from this goal.

The simulator is modest in the sense that we have made a number of simplifying assumptions. For example, the system handles rotations and translations of bodies, but we prohibit simultaneous translation and rotation, and we do not handle friction. Our goal for this sub-problem was to develop a simulator that was sufficient to identify physical ambiguities and to generate sensible questions, rather than creating one capable of making extensive and subtle inferences.[1]

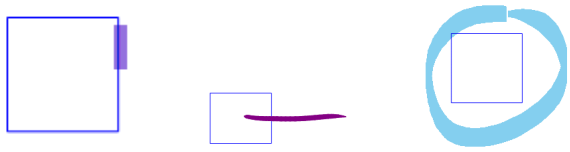
## MULTIMODAL INPUT

Speech recognition is done with the Microsoft Speech Recognizer in dictation mode, allowing the user more flexibility in expression. The flexibility for the user makes it more challenging for the system to understand their intentions. Our approach to dealing with this problem is discussed in more detail below.

Sketch recognition is handled by a low level stroke recognizer developed in our group [27] that returns primitive shapes (e.g., lines, arcs, ellipses, and polylines). We add a higher level classification of these primitives to categorize the strokes as either a location, path, or selection. A location can be used to indicate a point on an object or a new position for an object. A path can be used to show how a particular object moves. A selection stroke identifies a particular object. Figure 8 shows examples of each of the types of strokes.

The user's speech and sketching potentially overlap temporally and in content. The first step in figuring out what the user intended is to find corresponding speech and sketching. The user studies we conducted provide two key insights about segmentation: concurrent speech and sketching are about the same topic, and a pause indicates a new topic. The system uses these facts to group the concurrent speech and sketching together, and to wait for a pause in both input modalities before attempting to process the input. The system currently waits for a 500 millisecond pause in both modalities before processing the user's input.

[1]In any case, no matter how powerful the physical reasoning capabilities of the system, it would eventually encounter situations where it could not infer the answer, and had to ask the user. Hence the ability to ask questions is necessary; we focus here on doing it multimodally.



(a) Location (b) Path (c) Selection

**Figure 8. Illustration of the three different types of input strokes.**

### Determining the User's Intention

After the system receives speech and/or sketching it must determine the user's intention. We constrain the interpretation task in two ways. First, in most cases the user's speech is an answer to a question posed by the system. Second, the system expects the answer to fall within a known variety of possibilities for each modality. While the user's answer does not have to match any expected answer exactly, the expected answers help the system interpret the user's response and determine if it is valid. Acquiring information from the user is a six step process:

1. Ask the user a question.
2. Match and score the user speech against the expected speech.
3. Match and score the user sketching against the expected sketching.
4. Pick the best scoring combination.
5. Evaluate the best scoring combination to see if it makes sense.
   (a) If the match is successful, go to the next step.
   (b) If not, ask the user a follow-up question with more guidance about the expected answer. Go to step 2.
6. Generate statements about the new information and update current state and the physics appropriately.

We illustrate these steps using the very simple example in Figure 7, which shows a block connected to a spring.

The system runs the simulator, determines that the situation is ambiguous (is the spring in tension or compression?), and generates an appropriate question for the user: "Which way does this spring move?" (Figure 9). We first consider how the system interprets the user's answer, then discuss the generation of the speech and sketching output in the next section.

The question posed to the user is known to have several possible responses: The user might say "it expands," "the spring gets longer," or they might simply draw a line to indicate the direction the spring moves in. Alternatively, they might combine speech and sketching and say "it moves in this direction" and draw a stroke. As illustrated, the response from

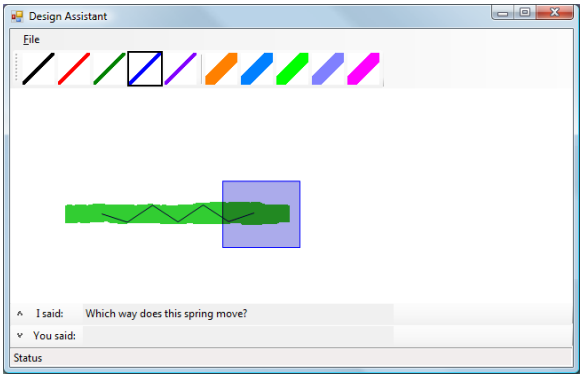



**Figure 9. The system asks the user what direction the spring is going to move in: "Which way does this spring move?"**

| | |
|---|---|
| it moves in this direction | No |
| it moves this direction | no |
| it moves and this direction | know |
| it moved this direction | noe |
| it moves on this direction | new |
| it moves its direction | noh |
| it moved its direction | knew |
| it moves his direction | nau |
| it moved his direction | dough |
| it owns this direction | doe |

**Table 1. The n-best lists from the speech recognizer for two speech phrases.**

the user could be speech only, sketch only, or a combination. The system can understand any of these alternatives.

We use these expected inputs to calculate scores for the incoming speech and sketching. For both types of input we get an n-best list of interpretations. For the recognized speech, we compare each of the options in the n-best list with each of the expected speech phrases, and score each based on how well it matches, with a discount based on the position in the n-best list. Table 1 shows the n-best list for two speech phrases. The top entry in the n-best list receives 100% of its match score, each subsequent entry's score is decreased by 10%. Similarly, we get different interpretations for each stroke, and compare each stroke to the expected strokes to calculate scores. Strokes are compared to the expected strokes based on possible target shapes and expected stroke types (location, path, or selection).

We then check the cross-modal consistency of the two inputs. For example, we calculate the spring direction that a stroke indicates and determine whether this is consistent with the user's speech. In Figure 10, the user drew a stroke indicating the spring compresses, while saying "it expands." The system notices this and asks a follow-up question to resolve the inconsistency.

Replies may also be insufficient or at odds with what the simulator knows about the physics of the situation. An insufficient match arises if, for example, the user says "it moves in this direction" without drawing a stroke (Figure 11). Without a stroke, the speech cannot be translated into a direction. The user's input may also make sense on the surface, but the underlying physics is impossible or at least impossible for our physics simulator to compute. This can occur, for example, if the system asks the user for the point of collision and the point the user specifies is impossible.

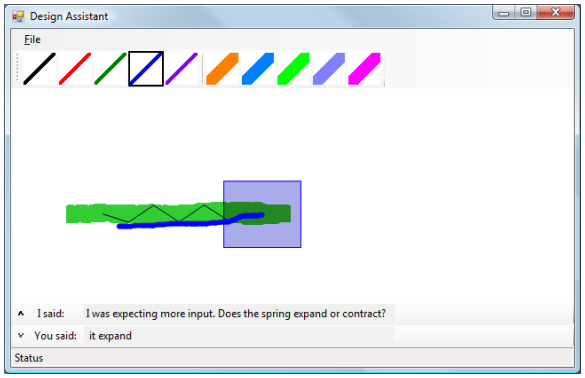



**Figure 10. The system asks the user if the spring expands or contracts, and the user provides a conflicting answer by drawing the shown stroke and saying "It expands."**

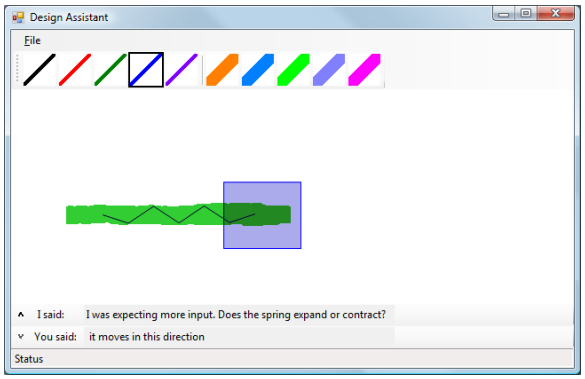



**Figure 11. The user provides an insufficient answer to the computer's question. The computer again asked "I was expecting more input. Does the spring expand or contract?" This time the user answered "it moves in this direction," but did not draw a stroke.**

The final result is used to update the state of the system, in this case setting the force exerted by the spring on the block in the appropriate direction (as indicated by the blue band in Figure 12). In other cases, properties of the stroke itself are used to compute the update, as for example, when the angle of a stroke is used to update the velocity direction of a body.

The system uses a table to represent the kinds of responses it can handle, with each line of the table indicating which of the expected strokes and speech must present, not present, or have a specific value. These requirements can also be marked as optional or required. Each row in the table indicates what type of input it represents: success, insufficient, conflict, or other. A simple example table is shown in Table 2. If the user's input matches a row in the table other than success, we ask a follow-up question that indicates explicitly the type of answer the system was expecting. If possible, we specify what the system determined was missing

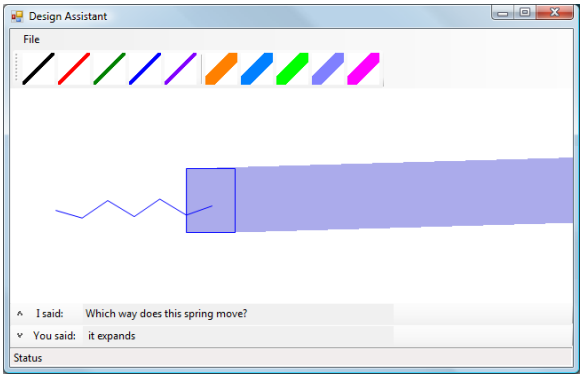



**Figure 12. The computer asks: "Which way does the spring move?" and the user provides an acceptable answer by saying "it expands." The system then updates the velocity of the body accordingly (the path of the block is now shaded).**

from the answer. Figure 11, for example, shows the computer's response when it is missing some input: It says "I was expecting more input. Does the spring expand or contract?"

| Expands Speech | Contracts Speech | Multimodal Speech | Direction Stroke | Result |
|---|---|---|---|---|
| Not Present | Not Present | Present | Not Present | Insufficient |
| Not Present | Present | Optional | Positive Value | Conflict |
| Present | Not Present | Optional | Negative Value | Conflict |
| Not Present | Present | Optional | Negative Value | Success |
| Present | Not Present | Optional | Positive Value | Success |

**Table 2. Part of a simplified table of expected inputs for a question about the direction a spring moves.**

The combination of the low level recognizers, our matching and scoring functions, and our consistency checking table allow the system to determine the user's intended behavior. We have yet to test the system on real users, but in the author's experience the speech recognizer is correct roughly 80% of the time, and the sketch recognizer is correct at least 95% of the time.

## MULTIMODAL OUTPUT

Another key feature of our system is the ability for the computer to output realistic, simultaneous speech and sketching, in the same way the user can use both modalities in their input. Composing this multimodal output involves determining the output for each modality and then coordinating the timing of these outputs. In an orchestra score, each instrument has notes to play but the score is not complete without the coordinated combination of the individual instruments. Similarly, the system's output is composed of generated sketch output and synthesized speech, but it is not complete without the synchronization of these individual modalities. We have created a simple language to easily write a multimodal score. We begin by discussing how the individual outputs are generated.

### Speech Output

Our speech output is generated using AT&T's Natural Voices speech synthesizer, which produces high quality speech. Generating the speech output is straightforward: a sentence or a sequence of phrases is sent to the speech synthesizer at a specific time. The system pieces together multiple output phrases to form the computer's speech output.

Ideally we would like to know how far along the speech synthesizer is with a particular utterance to time the sketch output accordingly. However, the feedback from the synthesizer is limited, and we know only whether it is actively producing speech or not. Our solution is to time our expected output phrases in an initialization step and use those estimated times to calculate the approximate progress of the synthesizer. Details of the coordination with the sketching are discussed below.

### Sketch Output

Users identify objects in a sketch by highlighting, circling, or otherwise marking the object being discussed. Computer generated output must also identify objects in the sketch. We accomplish this by creating a sketch synthesizer that can graphically indicate objects in the sketch. The current synthesizer indicates objects by circling them, but in the future it will also support indicating by using points, by filling in objects, and by highlighting them.

One of our goals for the sketch synthesizer is that it be realistic in the same way that the speech synthesizer is. In other words, its output should look plausibly human-generated instead of obviously machine generated. We identify and single out an object in the scene by generating an ellipse that encircles it. We then generate a set of points at a fixed interval on the ellipse, which are then modified randomly so that the ellipse will have some variation and error, producing an ellipse that looks human-drawn. Next we pick times for each of the points that correspond to an appropriate pen speed. In the future we hope to vary this speed slightly so that the computer can match the length of the concurrent speech. Finally, as the user study results indicated that pen color consistency is important, we ensure that all of the strokes for a particular topic are the same color.

In addition to strokes, the computer can also draw pie wedges to indicate a small range of possible angles to the user. For example, after a collision between two bodies, we need to know what direction the bodies move in (due to the qualitative nature of the physics simulator this cannot be calculated). Specifying a small range of angles using a stroke proved to be difficult. Pie shaped areas clearly indicate the allowable range of angles, and although the pie wedges are shapes that the user cannot draw, it seemed to be the best solution. The pie areas are timed to appear in a similar manner as the strokes.

### Synchronizing Outputs

Any non-trivial use of simultaneous sketching and speech requires synchronizing the two modalities. For example, use of two deictic gestures in the same sentence ("Does this

| Function | Annotation |
|---|---|
| Associating a word or group of word with one stroke | ( ) |
| Associating a word or group of word with one or more strokes | { } |
| Short pause (1 s) in the output | <short pause> |
| Long pause (3 s) in the output | <long pause> |
| Clearing all the computer generated strokes | <clear strokes> |
| Clearing the last computer generated stroke | <clear stroke> |
| Make the specified words singular or plural depending on the number of strokes | * |

**Table 3. The timing annotations for the speech and sketching output.**

block hit this block?") is impossible without close coordination of the two modalities.

Coordination of output modalities was also present in our user studies. In particular, the participants would pause their speech or sketching to keep the two modalities synchronized. This helped to ensure that the two modalities were always referring to the same topic and objects.

We have created a small language that expresses the chronological relationship between the system's speech and pen strokes. Table 3 indicates the supported functions and their annotation. Parentheses and braces are used to indicate a group of words that should be concurrent with a stroke or a group of strokes, respectively. The language provides a way to delete the last stroke or all the strokes that have been drawn, allowing one question to include a sequence of events like circling objects, erasing the strokes, and circling an object a second time. This allows one object to be singled out from a group of indicated objects. The pause functions insert a delay into the outgoing speech and sketching; this allows the user a few seconds to absorb the system's output.

Several steps are followed to generate the multimodal output. The first step is to break the speech string into pieces based on the stroke associations. The critical issue is whether the specified stroke or strokes will be entirely contained in associated speech. If the strokes require more time, subsequent speech will need to be delayed.

We cannot obtain the exact duration of an utterance from the speech synthesizer; instead we must estimate it. Overestimating the speech duration could cause a stroke that was intended to finish within that spoken phrase to finish during the next phrase. If we underestimate the speech duration, the same stroke would cause a delay in the start of the next speech phrase and the synchronization would happen as intended. For this reason, we underestimate the speech phrase durations.

Once we have determined the timing information for each phrase we can recombine the segments. Speech fragments in consecutive segments can be recombined if there is no constraint on the beginning of the second phrase (not waiting for a stroke to be drawn). The segments are timed relative to each other, which allows the entire output to be shifted so that it starts at a specific absolute time. The calculations attempt to keep speech together as much as possible, so that it sounds smooth and natural, while still maintaining the appropriate sketch alignment.

The questions the system asks range from simple: "Do {these two bodies} collide?" to complex: "(These two) (bodies) collide (here.) <long pause><clear strokes>Where on (this body) does the contact occur?" Both of those utterances are accompanied by strokes that identify the bodies in question, and for the second question the collision region.

## DYNAMIC DIALOGUE

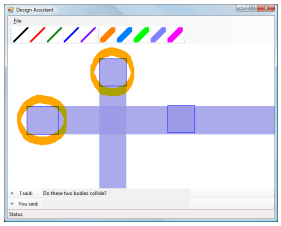
An important component of the dialogue we produce is its dynamic nature. The questions the computer asks in the conversation are not fixed and are not based on a set of fields that need to be filled in to run a database query. Instead, the questions are derived from the physics simulator and the information that the user provides.

Our system allows users to draw any mechanical device they want, within the limits of the physics noted above (e.g., no simultaneous rotation and translation). The questions asked depend entirely on the device that the user has drawn and the answers they provide to the system's questions. If the physics simulator cannot generate any questions, the system asks a generic question and allows the user to explain what happens next.

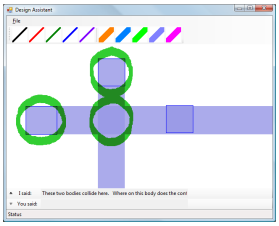
The dynamic nature of the conversation the system produces is illustrated by the sequence of snapshots in Figure 13. Figure 13(a) contains three bodies: a left body that has a velocity to the right, a middle body that has a downward velocity, and a right body that has no velocity. There are several possible collisions that may occur; the system cannot figure out what collisions will or will not occur because the velocities do not have magnitudes. We illustrate two possible outcomes to show the dynamic nature of the dialogue.

Assume that in the first scenario the user intends for the middle body to collide with the left body, hitting it from above. But the initial situation is ambiguous: with the information given we can't determine what collisions will occur. As a result, the system begins by asking "Do these two bodies collide?" while circling the left and middle bodies (Figure 13(a)). The user answers "yes," and the system continues by asking a series of questions to determine what happens next (Figure 13). In this case the user indicates exactly where the collision occurs (Figure 13(c)) and then the direction of the velocity of the left body after the collision (Figure 13(e)). The left body moves off the screen and the other two bodies are positioned as shown in Figure 13(f).

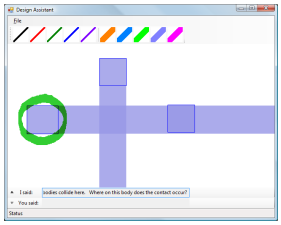
Alternatively, if the user indicates that the left and middle bodies do not collide, the system will ask questions about a collision between the left and right bodies (shown in Figure 14). The collision between those bodies results in Fig-



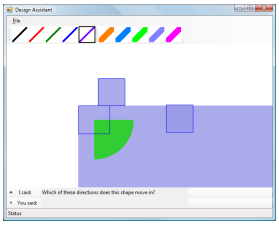
(a) System asks: "Do these two bodies collide?"



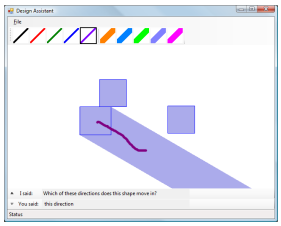
(b) After the user says "Yes" the system says "These two bodies collide here," while circling the collision location.



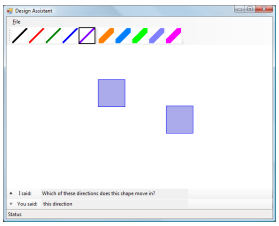
(c) The system continues: "Where on this body does the contact occur?"



(d) The user indicates the contact location and the positions of the bodies are updated. The system inquires: "Which of these directions does this shape move in?"

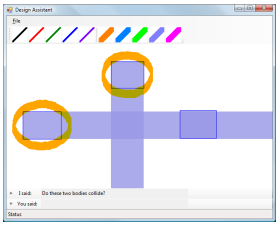


(e) The user replies using the shown stroke and says: "this direction."

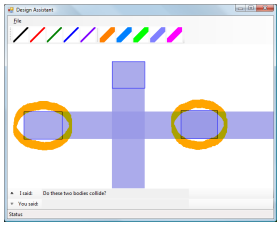


(f) The system moves the body off the screen.

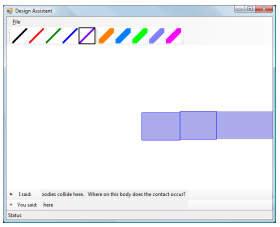
**Figure 13. A series of screenshots that indicate some of the questions and answers when the user says the left and middle bodies collide.**



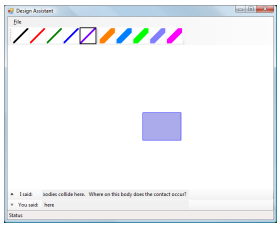
(a) System asks: "Do these two bodies collide?"



(b) After the user says "No" the system asks about the other collisions and asks: "Do these two bodies collide?"



(c) The user answers affirmatively. The system updates the positions accordingly: The middle body has moved off screen and the velocity of the left body has been transferred to the right body.



(d) The system updates the state again as the right body moves off screen.

**Figure 14. An alternative series of screenshots of questions and answers when the user says that the left and middle bodies do not collide.**

ure 14(c), in which the middle body has moved off the screen and the velocity of the left body has been transferred to the right body. The final positions in this case are illustrated in Figure 14(d) with only the left body still visible.

These two cases show how different the dialogue and the result can be based on the user's response to the system's questions. In one case the left body moved off the screen and in the other case it is the only body still visible.

## RELATED WORK

Multimodal user interfaces originated with Bolt's "Put-That-There" system. Working in the domain of rescue mission planning, Bolt's system used pointing gestures to resolve designated keywords in the speech [8]. The field has gradually grown to include more interesting and complex non-verbal input.

QuickSet [24] is a collaborative multimodal interface that recognizes sketched icons. The user can create and position items on a map using voice and pen-based gestures. For example, a user could say "medical company facing this way <draws arrow>." QuickSet uses a continuous speaker-independent speech recognition system like our system does. QuickSet differs from our system in several ways: it provides users a map to refer to, and does not provide the multimodal dialogue capabilities for the computer.

Focusing explicitly on managing multimodal dialogues, Johnston et al. describe MATCH in [19], which can combine multimodal inputs, including speech, sketch, and handwriting, in the domain of map-based information retrieval. The dialogue manager in MATCH enables a goal-directed conversation, using a speech-act dialogue model similar to [26]. This tool is not a sketching system and does not use a dynamic dialogue. Some recent work on multimodal reference resolution uses a greedy algorithm that uses linguistic and cognitive principles to efficiently resolve the references [11].

Several existing systems allow users to make simple spoken commands to the system [13, 20]. We had many instances of users writing words and speaking them, which is very similar to the types of input that [20] handles. Kaiser et al. describe how they can add new vocabulary to the system based on handwritten words and their spoken equivalents of the type that appear in Gantt schedule-charts [21].

Another system [10] allows users to query a real estate database with a multimodal user-driven dialogue (speech and

sketching). They use a probabilistic graph-matching approach to resolve multimodal references. In a user study, this approach proved effective in resolving ambiguous gesture inputs. Their study, like ours, highlighted the importance of disfluencies in the user's speech.

All of these systems have benefitted from a series of empirical studies of multimodal communication. Oviatt et al. document users' multimodal integration patterns across speech and pen gestures in [25].

There are several other related projects[14, 24] that involve sketching and speech, but they are focused more on a command-based interaction with the user. In our system, speech augments the sketching; in other systems, the speech is necessary to the interaction.

There has been significant work on multimodal output, but it has focused on generating combinations of speech, images, text, and gestures by an avatar or robot. The COMIC system [16] focuses on generating multimodal dialogues for an avatar including speech output and pointing gestures. Most relevant is their work on interleaving speech and avatar animation [15] which takes a similar approach in timing the outgoing speech and aligning the other modalities accordingly. However, the main focus of their work is on supporting parallel output and planning of the multimodal dialogue. Our system does not require this level of planning to produce the required output. We produce the strokes to display along with speech instead of avatar animations and speech.

The system in [9] is used to animate two agents that communicate using speech and coordinated gestures. The two output modalities are different from our system, but have some important similarities. In both systems, both modalities influence the combined output. Specifically, both systems must adjust the output based on the duration of different output events – the speech and the gestures and the speech and the sketching.

Multimodal output is also used in several other systems. The WIP system generates device instructions that are multimodal illustrated texts [6] containing images and text. The text includes references to the images. Another system [7] uses a set of rules and heuristics to produce a page layout of text and images. The structure and references in the text determine the sections and the formatting. Related sections of text are displayed using similar styles. This is analogous to our use of the same highlighter color when identifying similar objects. WIP and the system in [7] deal with text layout and images instead of our system's generated questions and identification of objects on a shared drawing surface.

Medical images, a text display and speech are coordinated by the multimodal system in [12]. The layout of the visual display provides constraints on the spoken output. Textual data is highlighted as it is verbally referenced; one of the constraints is that the text should be highlighted in coherent areas. The information displayed differs from our system, but the coordination between the display and speech is similar. In our system we focus on the highlighting parts of the dynamic sketched objects. Again, our output is a shared medium and we ask the user questions based on the system state rather than presenting fixed data to the user.

Giuliani [17] provides a way to specify speech and gestures for a human-robot interaction in an XML format. The format supports specific start and end time information for the gestures. Since our modalities require less information to generate the output, we can use our simple format and calculate the exact timing information as needed.

### Our Previous Work

Our previous system, ASSIST [4], lets users sketch in a natural fashion and recognizes mechanical components (e.g., springs, pulleys, axles, etc.). Sketches can be drawn with any variety of pen-based input (e.g., Tablet PC). ASSIST displays a "cleaned up" version of the user's sketch and interfaces with a simulator to show users their sketch in action.

ASSISTANCE[22] was a previous effort in our group to combine speech and sketching. It built on ASSIST[4] by letting the user describe the behavior of the mechanical device with additional sketching and voice input. More recently we built a system [2] that let users simultaneously talk in an unconstrained manner and sketch. This system had a limited vocabulary and could not engage the user in a dialogue, limiting its ability to interpret the user's input.

## CONCLUSION AND FUTURE WORK

In this paper, we highlight some of the reasons that a multimodal dynamic dialogue can strengthen communication with a user about a design. We discussed three important aspects of such a dialogue: multimodal input processing, multimodal output generation, and creating a dynamic dialogue. Our hypothesis is that a multimodal dialogue that incorporates these aspects will produce some of the same interaction characteristic we observed in our user studies – for example, user's refining and updating their designs and lengthy design explanations.

The next step in our research is to conduct a user study of our system. The goal of this study is to test our hypothesis and determine how the human-computer dialogue compares to the human-human dialogue. In addition, there are several user interface issues we need to address including determining how to interrupt the user if we have a question.

## REFERENCES


1. A. Adler. Segmentation and alignment of speech and sketching in a design environment. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

2. A. Adler and R. Davis. Speech and sketching for multimodal design. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 214–216. ACM Press, 2004.

3. A. Adler and R. Davis. Speech and sketching: An empirical study of multimodal interaction. In *SBIM*

*'07: Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling*, pages 83–90, New York, NY, USA, August 2-3 2007. ACM.

4. C. Alvarado and R. Davis. Resolving ambiguities to create a natural sketch based interface. In *Proceedings of IJCAI-2001*, August 2001.
5. C. Alvarado and R. Davis. Dynamically constructed bayes nets for multi-domain sketch understanding. In *Proceedings of IJCAI-05*, pages 1407–1412, San Francisco, California, August 1 2005.
6. E. André, W. Finkler, W. Graf, T. Rist, A. Schauder, and W. Wahlster. *Intelligent Multimedia Interfaces*, pages 75–93. AAAI Press, 1993.
7. J. Bateman, J. Kleinz, T. Kamps, and K. Reichenberger. Towards constructive text, diagram, and layout generation for information presentation. *Computational Linguistics*, 27(3):409–449, 2001.
8. R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, pages 262–270, 1980.
9. J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of SIGGRAPH '94*, pages 413–420, 1994.
10. J. Y. Chai, P. Hong, and M. X. Zhou. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of 2004 International Conference on Intelligent User Intefaces (IUI'04)*, pages 70–77, 2004.
11. J. Y. Chai, Z. Prasov, J. Blaim, and R. Jin. Linguistic theories in efficient multimodal reference resolution: An empirical investigation. In *IUI '05: Proceedings of the 10th international conference on intelligent user interfaces*, pages 43–50, New York, NY, USA, January 2005. ACM Press.
12. M. Dalal, S. Feiner, K. McKeown, S. Pan, M. Zhou, T. Höllerer, J. Shaw, Y. Feng, and J. Fromer. Negotiation for automated generation of temporal multimedia presentations. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 55–64. ACM, 1996.
13. D. Demirdjian, T. Ko, and T. Darrell. Untethered gesture acquisition and recognition for virtual world manipulation. *Virtual Reality*, 8(4):222–230, September 2005.
14. K. Forbus, R. Ferguson, and J. Usher. Towards a computational model of sketching. In *Intelligent User Interfaces '01*, pages 77–83, 2001.
15. M. E. Foster. Interleaved planning and output in the COMIC fission module. In *Proceedings of the ACL 2005 Workshop on Software*, Ann Arbor, June 2005.
16. M. E. Foster and M. White. Assessing the impact of adaptive generation in the COMIC multimodal dialogue system. In *Proceedings of the IJCAI 2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Edinburgh, 2005.
17. M. Giuliani. Representation of speech and gestures in human-robot interaction. In *IEEE Ro-Man 2008 Workshop: Towards Natural Human-Robot Joint Action*, Munich, Germany, August 2008.
18. T. Hammond and R. Davis. LADDER, a sketching language for user interface developers. *Elsevier, Computers and Graphics*, 28:518–532, 2005.
19. M. Johnston, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. Match: An architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 276–383, 2002.
20. E. C. Kaiser. Multimodal new vocabulary recognition through speech and handwriting in a whiteboard scheduling application. In *IUI '05: Proceedings of the 10th international conference on intelligent user interfaces*, pages 51–58, New York, NY, USA, January 2005. ACM Press.
21. E. C. Kaiser. Using redundant speech and handwriting for learning new vocabulary and understanding abbreviations. In *Proceedings of the Eighth International Conference on Multimodal Interfaces (ICMI 2006)*, Banff, Canada, November 2006.
22. M. Oltmans and R. Davis. Naturally conveyed explanations of device behavior. In *Workshop on Perceptive User Interfaces*, 2001.
23. T. Y. Ouyang and R. Davis. Recognition of hand drawn chemical diagrams. In *Proceedings of AAAI*, pages 846–851, 2007.
24. S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions. In *Human Computer Interaction*, pages 263–322, August 2000.
25. S. Oviatt, A. DeAngeli, and K. Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. In *Conference Proceedings on Human Factors in Computing Systems*, pages 415–422. ACM Press, 1997.
26. C. Rich and C. Sidner. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapter Interaction*, 8(3–4):315–350, 1998.
27. T. M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: Early processing for sketch understanding. In *The Proceedings of 2001 Perceptive User Interfaces Workshop (PUI'01)*, Orlando, FL, November 2001.