Proceedings of the 2001 Workshop on Perceptual User Interfaces (PUI'01), Orlando, FL, May 2001.

Naturally Conveyed Explanations of Device Behavior

Michael Oltmans and Randall Davis

MIT Artificial Intelligence Laboratory
{moltmans, davis}@ai.mit.edu

Abstract. Designers routinely explain their designs to one another using sketches and verbal descriptions of behavior, both of which can be understood long before the device has been fully specified. But current design tools fail almost completely to support this sort of interaction, instead not only forcing designers to specify details of the design, but typically requiring that they do so by navigating a forest of menus and dialog boxes, rather than directly describing the behaviors with sketches and verbal explanations. We have created a prototype system, called ASSISTANCE, capable of interpreting multimodal explanations for simple 2-D kinematic devices. The program generates a model of the events and the causal relationships between events that have been described via hand drawn sketches, sketched annotations, and verbal descriptions. Our goal is to make the designer's interaction with the computer more like interacting with another designer. This requires the ability not only to understand physical devices but also to understand the means by which the explanations of these devices are conveyed.

1 Introduction

When a mechanical designer explains a device to a colleague, s/he does so with sketches and verbal explanations of the device's behavior. When specifying the same device in a CAD system, however, the interaction is not nearly as natural, in either the medium of expression or the content expressed. The designer must use a mouse and keyboard to specify a substantial body of detailed information (e.g., spring constants) that is not the primary concern in early design stages. This state of affairs remains true even a decade after work indicated that the formality and

rigidity of CAD systems can significantly hinder the early stages of the design process [17].

We have been working to remove this barrier between designers and their tools by shifting the emphasis away from parametric specifications and towards multimodal explanations of behavior. We have constructed a system, called ASSISTANCE, that allows the user to describe a device's behavior using hand drawn sketches, sketched annotations, and verbal descriptions phrased in the same vocabulary designers routinely use when talking to one another. From this information ASSISTANCE generates a model that represents how the components move and what causal relationships exist between those movements. As we illustrate, this process both provides a more natural interface for the designer and allows the system to infer some useful details about the design of the device. In the near term the model constructed by the system will be used to inform a mechanical simulator, so the sketched device can be animated, while in the longer term we envision systems using ASSISTANCE's representations to reason about design rationale by tracking changes in both the design's structure and behavior during the early stages of design.

This paper reviews an example of the system in operation, explains what knowledge is required to support the inferences it makes, and examines both its capabilities and limitations.

2 Describing structure and behavior

Enabling designers to describe devices to a computer as naturally as they would to a colleague requires understanding descriptions of both structure and behavior. For a mechanical device, the structure represents the device's components and their physical interconnections while the behavior represents how the components move and the relationships between these motions. The problem of understanding structural descriptions using natural media like hand drawn sketches has been explored in our group [1] and a few other efforts (e.g., [9]) but there has been very little work on understanding similarly natural descriptions of behavior.

To make description feel natural, we have to attend to both the medium of expression and the content being expressed. Designers of all sorts feel natural drawing and talking about their designs, particularly in the early, conceptual stages of the process. Designers also find a particular kind of content natural at this stage: they typically pay more attention to the behavior of the device than the properties of the in-

dividual components [2]. This demands that we enable descriptions to be phrased in the sort of language and vocabulary typically employed.

As a trivial yet instructive example, consider a spring attached to a block positioned next to a ball. In a traditional CAD system (Fig. 1) the designer would select the components from a tool bar and position them, and would then have to specify a variety of parameters, such as the rest length of the spring, the spring constant, etc.

Contrast this to the way someone would describe this device to a colleague. As we discovered in a set of informal experiments, the description typically consists of a quick hand drawn sketch (e.g., Fig. 2) and a brief spoken description, "the block pushes the ball." We have built a tool that augments structural descriptions by understanding these sorts of graphical and verbal descriptions of behavior.

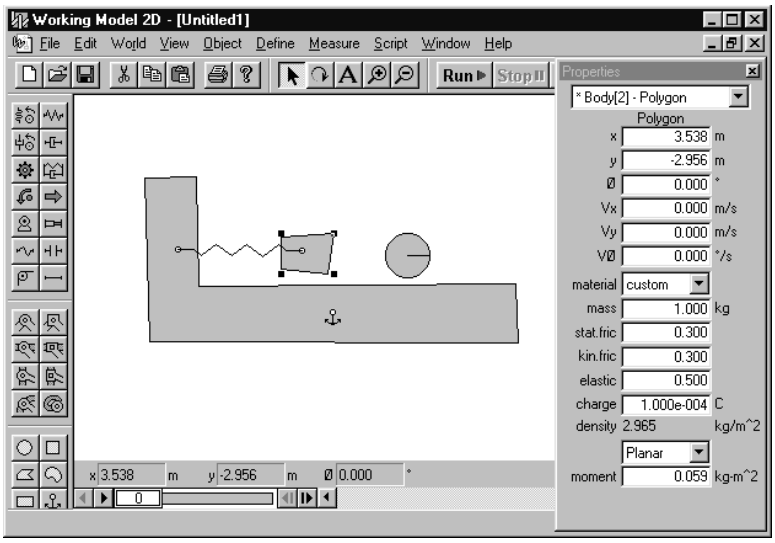


Fig. 1. A block and spring described using a CAD tool.

3 Overview and capabilities

To use the system a designer first sketches the device, using a system called ASSIST [1], which interprets the sketch and generates a representation of device structure. The designer then switches to an explanation

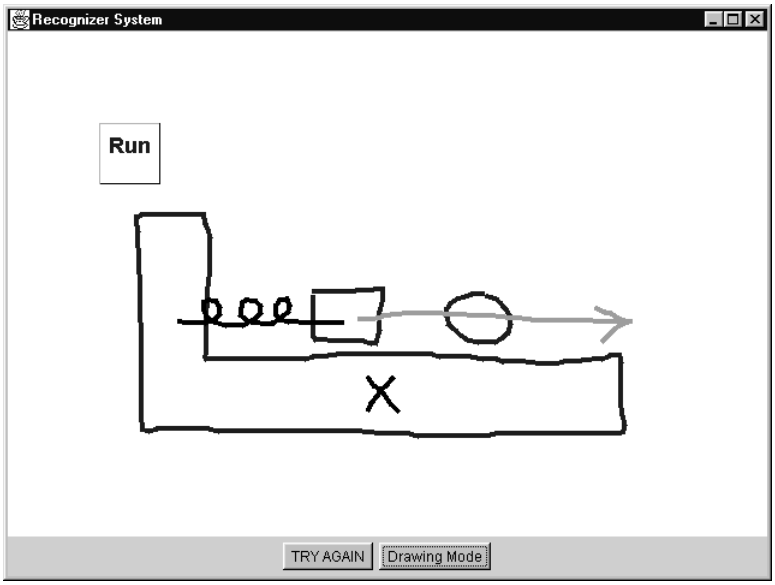


Fig. 2. A more natural medium of description.

mode and explains the device's behavior by drawing arrows, speaking, and pointing. After each of the designer's explanation fragments (i.e., each utterance and gesture) the system interprets that explanation fragment, updating its model of devices. At any time the designer can verbally ask the system to explain its causal model.

ASSISTANCE can currently understand descriptions of two dimensional kinematic devices that use rigid bodies, pin joints, pulleys, rods, and springs. It takes spoken natural language and hand-drawn sketches as input and generates a causal model that describes the actions the device performs and the causal connections between them.

We take "understanding" in this context to mean the ability to generate such a causal model, that accurately reflects the behavior description given by the designer. The system's task is thus to understand the designer, without attempting to determine whether the designer's description is physically accurate.

The representations ASSISTANCE generates are not a verbatim recording of the designer's description. To demonstrate that it has understood an explanation (and not just recorded it), ASSISTANCE can construct simple explanations about the role of each component in terms of the events that it is involved in and causal connections between events. Fur-

ther evidence of the system's understanding is provided by its ability to infer from the behavior description what values some device parameters (e.g., spring constants) must take on in order to be consistent with the description. The query and parameter adjustment capabilities were designed to provide a mechanism for the system to describe its internal model and to suggest how such representations could be used in the future.

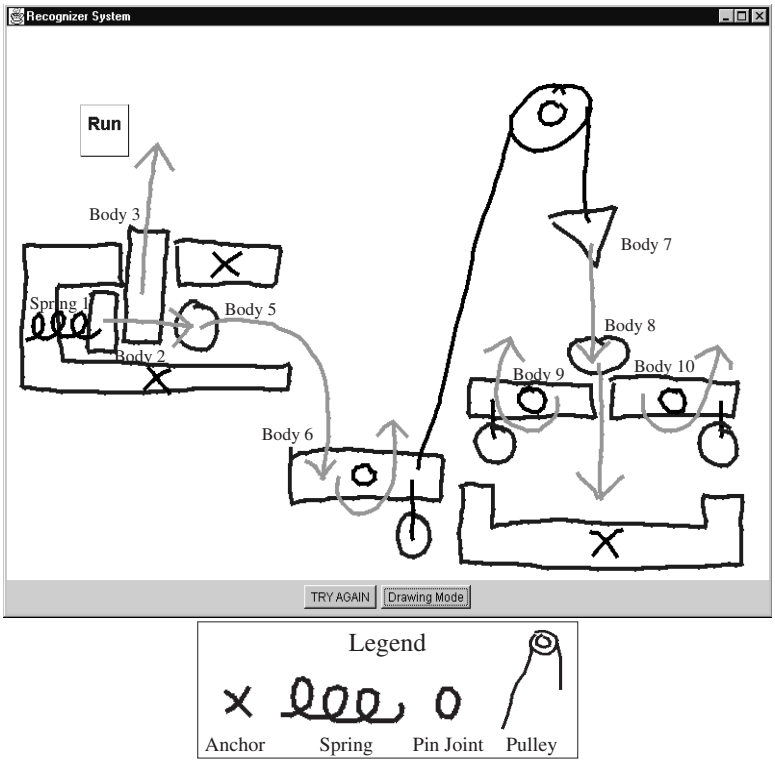
The current implementation of ASSISTANCE is made tractable by taking advantage of a number of sources of knowledge and focusing the scope of the task. We currently focus on two-dimensional kinematic devices, thereby limiting the vocabulary and grammar necessary to describe a device, making the language understanding problem in turn tractable. We then take advantage of two characteristics of informal behavior descriptions: they typically contain overlapping information and they are often expressed in stereotypical forms. We use the multiple, overlapping descriptions of an event–the same event described in a verbal explanation and in a sketched annotation–to help infer the meaning of the description. We also combine multiple descriptions to produce a richer description than either one provides alone. Finally, we use knowledge about the way designers describe devices to simplify the process of interpreting their descriptions (e.g., mechanical device behavior is frequently described in the order in which it occurs).

4 An example

An example will help demonstrate the types of multi-modal explanations ASSISTANCE understands and the types of inferences it can make. Fig. 3 shows a Rube Goldberg-style egg-cracking device (adapted from [12]), along with an explanation of its behavior (the arrows and verbal statements).

From this explanation the system generates a model of the device's behavior. The model (described in more detail below) consists of 8 events, one motion event for each of bodies 2, 3, and 5-10.

Note that in the absence of the information provided by the verbal and gestural annotations of Fig. 3, a simulation of the device produces useless behavior: body 3 will simply drop the small remaining distance to the frame, and sit there, preventing the spring from propelling the ball, while the ball will similarly drop the small distance and remain in place. Allowing the designer to explain how the device should work allows the system to construct a model that can be used to inform a simulator, so that intended behavior of the current design can be visualized.



"When body 3 moves up spring 1 releases."
"Body 2 pushes body 5."
"Body 6 rotates."
"Body 7 falls."

Fig. 3. The explanation of an egg cracking device. (Labels of the form "body 3" are created by the system for each component; some have been removed in this figure for clarity.)

4.1 Inferences from device structure

ASSISTANCE is given a description of the device's structure (supplied by another system we have developed [1]) that specifies each of the objects in the figure and their connections.[1] As its first step ASSISTANCE does a degree of freedom analysis based on the interconnection information (e.g., anchors prevent both rotation and translation while pin joints allow rotation).

4.2 Generating the behavioral model from the description

The bulk of the work of ASSISTANCE lies in parsing the user's verbal description and sketched annotations, and providing a causal model of the device behavior. We walk through several inputs to illustrate this process in action, detailing the knowledge required to understand the description. The example illustrates ASSISTANCE's ability to infer motions of bodies, identify multiple descriptions of the same event, disambiguate deictic references, and infer causal links between motions.

"When body 3 moves up spring 1 releases" ASSISTANCE begins by breaking the utterance into its constituent clauses, which it then translates into events. A straightforward interpretation of the first clause ("body 3 moves up") generates a representation for the motion of body 3. The system then infers the motion of body 2 from the second clause ("spring 1 releases"), based on the observation that spring 1 is connected on the left end to an anchored body (body 1), hence in order for the spring to "release," body 2 must be moving. This is an example of an inference based on the physical structure of the device.

ASSISTANCE then infers a causal connection between these two motions because the two clauses are linked by a conditional statement ("When body 3 moves...") suggesting causality, in which the motion of the first clause is a precondition for the motion in the second. This is an example of using linguistic properties to infer a causal link between events.

[1] We use "objects" to mean any of the things in the sketch. We refer to objects such as springs, pulleys, pin joints, etc., as "functional components", or "components." We use the term "body" to refer to any other hunk of material in the sketch (e.g., everything other than the spring, pulley, pin joints, and arrows in Fig. 3).

The arrow at body 3 From the arrow at body 3, ASSISTANCE generates a second event representation for the motion of this component, describing its path. ASSISTANCE then links this representation with the representation generated by the utterance above, based on its ability to recognize that the two descriptions describe the same type of motion and refer to the same object.

"Body 2 pushes body 5", the arrow at body 2 ASSISTANCE interprets the "pushes" phrase as two motion events with the first (the motion of body 2) causing the second (the motion of body 5). The causal link is inferred by the fact that *pushing* is interpreted as the act of one object causing another object to move.

From the arrow at body 2 the system generates a second representation the motion of that body (the first resulted from "the spring releases" utterance). Recognizing that they refer to motions of the same object, the system marks the two representations as describing the same motion. These equivalence links between event representations are used later to merge two descriptions of the same event into a single, more detailed representation.

At this point ASSISTANCE's behavioral model represents the fact that the motion of body 3 causes the motion of body 2, which in turn causes the motion of body 5.

"Body 6 rotates," and "body 7 falls" From these utterances ASSISTANCE infers a causal link between the motions of body 6 and body 7. This is based on its model of pulleys, which is simply that if two things are attached to either end of a pulley, and both of them are known to be moving, then one of the motions may have caused the other. ASSISTANCE uses this piece of physical reasoning and the topology of the device to hypothesize a causal link between the two motion events.

The arrows at bodies 8, 9, and 10 ASSISTANCE infers that there is a motion event associated with each of these 3 arrows. It also infers that the motion of the egg causes the motions of the two levers, based on the observation that the path followed by the egg brings it into contact with the levers.

Finally, ASSISTANCE observes that the motions of the two levers are rotations because the bodies have a single, rotational degree of freedom.

4.3 How ASSISTANCE demonstrates understanding

ASSISTANCE now has both a structural and behavioral model of the device. The system can demonstrate its understanding of the device by describing the events that a component is involved in and the immediate causes and effects of those events. An example is presented in Fig. 4.

Designer: What is body 2 involved in?
ASSISTANCE: The motion of body 3 causes the motion of body 2 which causes the motion of body 5

Fig. 4. ASSISTANCE demonstrates its understanding.

The system also demonstrates understanding by adjusting the parameters of the springs in the structural model so that the simulation of the device is closer to the behavior described by the designer. Consider the spring in Fig. 3: As drawn, is it currently compressed, stretched, or at its neutral position? With the knowledge that body 2 is propelled by spring 1, ASSISTANCE is able to infer that it must be compressed, allowing the program to modify the model, which then will produce the correct behavior when the device is simulated.

5 Implementation

The process by which ASSISTANCE builds its models can be split into four main components:

- Process sketch and speech input
- Translate these inputs into events and causal links
- Construct the causal structure
- Demonstrate the system's understanding of the explanation

We describe the overall architecture of the system, then discuss each of these components in detail.

5.1 System architecture

The basic structure and information flow in the system is depicted in Fig. 5. The sketch recognition system, ASSIST recognizes the raw sketch

data and produces a description of objects in the sketch and their interconnections. IBM's ViaVoice recognizes the acoustic data and produces a decorated parse tree of the utterances. The input to ASSISTANCE is thus descriptions of physical bodies, descriptions of arrows, and parsed textual phrases, rather than raw pixel and acoustic data. The information from ASSIST and ViaVoice is converted into propositional statements and used as the foundation for the reasoning performed by ASSISTANCE.

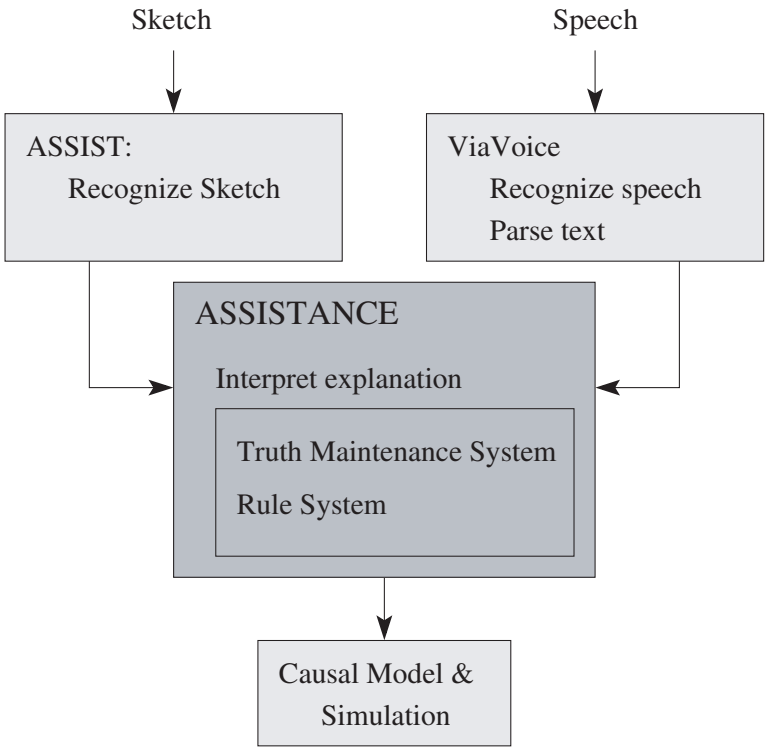


Fig. 5. The overall structure of the system.

ASSISTANCE is implemented with a forward-chaining rule-based system and truth maintenance system (TMS) taken from [7]. The rules represent the knowledge required to translate the parsed utterances and gestures into representations of events, and the knowledge required to infer causal relationships between events.

The role of the TMS is to maintain a record of inferences: When a rule fires, the TMS records the rule's preconditions as justifications for the inference. When the system later attempts to build a complete causal model of the device behavior, these records of inferences permit flexible and efficient exploration of alternative interpretations.

5.2 Process the inputs

There are three inputs to ASSISTANCE: the structural model generated by ASSIST, the utterances recognized by the speech recognition system, and the sketched arrows.

The Representation of the structural model The structure representation provided by ASSIST[1] contains shape and location information about every object in the sketch. For functional components (e.g., springs, pin joints, etc.) ASSIST also indicates which bodies they are attached to and whether they are attached to the fixed plane. All objects are assigned a unique English name (e.g. "body 1", "spring 1") so that they can be referred to unambiguously.

ASSISTANCE uses this model to perform a degree of freedom analysis on each body, determining from its connections (e.g., a pin joint) whether it can rotate, translate, or neither.

Speech recognition and processing Speech recognition is handled by IBM's ViaVoice software, which parses the utterances against a grammar containing phrases we found commonly used in an informal survey of several designers explanations of devices. The grammar abstracts from the surface level syntactic features to an intermediate syntactic representation that explicitly encodes grammatical relations such as subject and object. These intermediate representations are used by the rules described below to generate semantic representations of the utterances. This type of intermediate syntactic representation is similar to the approach taken in [15].

The grammar is written using the Java Speech Grammar Format, which provides a mechanism for annotating the grammar rules with tags. These tags decorate the parse tree generated by the speech recognition system with both the surface level syntactic features and the intermediate syntactic representations mentioned above.

The grammar handles several basic sentence types: motions (e.g. "the block moves"), conditionals (e.g. "If body 1 moves up then Spring 1 releases"), and propulsions (e.g. "Body 2 pushes body 5"). The system

is also capable of handling deictic references to bodies, for example "This pushes Body 1." The interpretation of each of these is described in more detail below.

The grammar has intentionally been kept constrained, because our emphasis has not been on the language processing aspects of the system. Having a small grammar also helps the speech recognition system achieve a high level of accuracy. We are currently looking into the possibility of linking into a more powerful language processing system such as the START System [10].

Recognizing sketched gestures The sketched gestures currently handled by ASSISTANCE are arrows and pointing gestures. Both of these gesture types are recognized by ASSIST and converted into a symbolic representation that includes the object that they refer to; ASSISTANCE then reasons with the symbolic representations. For arrows, the referent is the object closest to the base of the arrow and for pointing gestures it is the object that is closest to the point indicated.

5.3 Translate inputs into events and causal links

In order to construct the causal structure of the device from the explanation, ASSISTANCE must first determine what events are mentioned in the explanation, then unify multiple representations of the same event. It then determines the causal relationships between pairs of events. The knowledge required to make these inferences is represented by rules that fit into several categories:

- Translate utterances into events
- Resolve deictic references
- Translate arrows into events
- Merge multiple representations of the same event
- Find causal connections between events

The rules are organized around knowledge of language patterns (the first two categories), knowledge of drawing conventions (the third category), and knowledge about physics and physical devices. By organizing the rules around such knowledge, and by writing them to be as general as possible within these categories, we achieve a degree of generality in the system's performance. As is common with rule-based systems, new rules must be added with care, but the task has to date been quite tractable.

Translate utterances into events There are currently three rules that translate utterances into motion events. These rules correspond to the three classes of verbs understood by the system: “moves,” “releases,” and “propels.” The rule for translating “propels” utterances, such as “Body 2 pushes body 3,” will illustrate how these rules work. The decorated parse tree for this sentence is shown in Fig. 6. The parse tree contains the structure of the sentence and syntactic tags that indicate the parts of speech and the roles played by each part of the parse (e.g. subject).

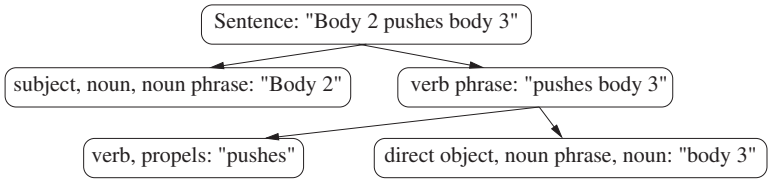


Fig. 6. Decorated parse tree for ”body 2 pushes body 3.”

To process this utterance the rule begins by identifying the sentence as a “propels” utterance, by the “propels” tag on the verb. Then it uses the structure of the parse tree to bind the “subject” to “body 2” and the “direct object” to “body 3.” Finally, it finds the physical objects corresponding to these bodies by matching the names to the bodies. The rule then asserts one motion event for the subject and one for the object.

The rule for “releases” utterances deals with springs, where “release” implies the motions of objects connected to either end of the spring. This rule is analogous to the one for “propels” except the bodies are those attached to the spring (which must be explicitly referenced by name or by a pointing gesture as described below).

The rule for interpreting “moves” utterances is analogous to the one for “propels” but does not have an object and only asserts one motion event.

Resolve deictic references In the current implementation, when objects are not referred to by name they must be accompanied by a pointing gesture. This allows the interpretation of phrases such as “when this moves up, body 1 pushes body 2.”

In addition to the pronouns "this" and "that" the system has a vocabulary of common component names like "block" and "ball" which the user can use to reference objects. However, since the system does not have representations for "balls" and "blocks" these references must also be disambiguated by pointing gestures. For example, the user can say "the block moves up" as long as she also points to the block in question.

The constraint that each reference has a corresponding pointing gesture means that the two can be matched in a straightforward manner, simply by keeping a list of references and pointing gestures and matching them in the order in which they occur.

While deictic references are common in explanations, our current requirement that the user disambiguate the referent is awkward and is one that we plan to eliminate in the future. There has been substantial work in both the literature on discourse theory and multi-modal interfaces [13] to indicate that this is possible. In particular the typical time delays between gestures and verbal utterances reported in [14] could be used to identify such multimodal references.

Translate arrows into events The translation of sketched arrows to motion events is straightforward because the recognition of the arrow includes the determination of the object it refers to: this is defined as the object at or near the tail of the arrow. A simple rule associates a motion event with this body and records the path depicted by the arrow.

Merge multiple representations of the same event We have found that in many explanations the motion of one body is described multiple times, often by multiple modalities. For example, in the egg cracker (see Fig. 3) the motion of body 2 is described three times: by the utterance "spring 1 releases", by the utterance "body 2 pushes body 5," and by the arrow. Initially each of these is represented individually.

To generate a unified causal structure, sets of equivalent events must be combined into a single, canonical event. This is done by unifying the properties of the individual events. In the example mentioned above, the representation generated from the arrow provides spatial information about the trajectory of the motion, while the verbal utterance indicates the causal connection between that motion and other motions.

Our current implementation assumes that each body can be involved in only a single motion event. This means that any two events

that involve the same physical object are actually different descriptions of the same event[2]

The occurrence of overlapping descriptions from multiple modalities is well known (see [13] for example), however, our problem is slightly more complex than is typical: We not only have redundant descriptions due to multiple modalities, we also have redundant explanations within a modality, as, for example, the two utterances describing the motion of body 3. This is why we perform the merging of events based on the semantic interpretations instead of the input sources.

Find causal connections between events After identifying the motion events, the system attempts to find causal connections between them. There are currently two classes of causal links in our system: definite and plausible.

Definite Causal Links Definite causal links result from verbal utterances that unambiguously describe a causal relationship between two events. If, for example, the user says, "if this moves up spring 1 releases," we take that to be an unambiguous statement of a causal relationship between the two events. Definite causal links are also constructed from "propels" utterances, to relate the two motion events with a causal relationship. There are currently two rules for asserting definite causes, one for utterances of conditional statements and one for "propels" utterances.

Plausible Causal Links Plausible causal links arise from less explicit indications of causality. Currently these links are inferred from spatial information about the trajectories traced by bodies and from the motions of bodies connected by a rod or pulley system. There is one rule for each of these cases. As an example, spatial information is used in the egg-cracker description (Fig. 3) to infer that the motions of the two levers holding the egg are caused by the motion of the egg on its way into the pan. If the first body's trajectory brings it in contact with the second body, it is plausible that the second body's motion is caused by the first, but not guaranteed: the second motion may have commenced

[2] Future implementations will relax this restriction by adding reasoning to determine when an event description refers to a new event rather than another aspect of a previously mentioned event. The decision to keep the events separate could be based on evidence of conflicting properties (e.g. direction) or by examining the amount of time since the last reference to the event.

before the first. This rule makes use of the size of the bodies and the path traced by the arrow.

As as a second example, if two bodies are attached to a rope that passes through a pulley and they both move, it is plausible that one of the motions is the cause of the other. However, this is not definite because there may be slack in the rope.

5.4 Construct the causal structure

After finding all the events and the causal relationships between them, ASSISTANCE has two remaining tasks: (i) find the set of consistent causal structures, and (ii) choose the causal structure that is closest to the designer's description.

Find the set of consistent causal structures The constraints that must be satisfied in order for a causal ordering to be considered consistent are: (i) each event must have exactly one cause (but can have multiple effects), and (ii) causes precede effects.

The truth maintenance system and its constraint propagation capabilities enable fast and efficient exploration of different possible causal structures, allowing the system to find those consistent with the constraints. This exploration of alternative causes is a forward-looking depth-first search with backtracking over the set of possible causal orderings. The search is constrained by limiting it to those events with no definite causes and multiple plausible causes. It proceeds by trying all the plausible causes of each event until each has a cause. Any event that does not have a cause can be hypothesized to be caused by an exogenous force (a later step minimizes the number of hypothesized exogenous causes).

The truth maintenance system can identify and record sets of inconsistent assumptions, enabling search to be terminated along other branches of the tree that include the same set of assumptions that led to the contradiction. Although the search is exponential in the worst case, the branching factor is small and for our current problems efficiency has not been an issue. (We believe that there may be more efficient search strategies that take advantage of the fact that causes and effects are generally described together, and refer to physical components that are nearby spatially. This may allow us to partition the search space and reduce the depth of the search.)

Choose the causal structure that is closest to the designer's description Finally, the system must choose from all the consistent

models the one that most closely matches the designer's description. Two heuristics are used to select the model: there should be a minimal number of events caused by exogenous forces, and the order of the events in the causal description should be as close as possible to the order in which they were described. This latter criterion is based on our empirical observation that people describe the behavior in the order in which it occurs.

6 Evaluation and future work

Criteria for evaluating a system such as ASSISTANCE include its usability and the range of inferences supported by its representations. We consider these by first comparing existing alternative methods for behavior explanation. We then evaluate the usability and expressiveness of ASSISTANCE and discuss the features that we believe are necessary for the growth of the system.

6.1 Existing alternatives

To date designers have had to choose between descriptions that were formal, constrained, and usable by an automated system, and those that were natural and unconstrained, but not easily automated.

On the formal end of the spectrum are CAD tools, which require the designer to describe the device with mathematical precision, using input media that are very different from the pencil and paper sketches used in the early design stages. Although some CAD systems claim to support sketching, they are still highly modal and force the designer to indicate what they are about to draw, instead of just drawing it. To date CAD systems have also supported the specification of behavior only through the adjustment of parameters, rather than via explicit descriptions of the intended behavior.

On the opposite end of the spectrum are written documents, person to person explanations, and verbatim recordings of explanations. The collection of this information imposes no constraints on the designer but also does not produce a representation usable by an automated reasoning system.

ASSISTANCE aims to combine the strengths of both of these approaches to create a system that gathers information from natural interactions and generate useful representations.

6.2 Usability issues

We have not yet performed a formal evaluation of ASSISTANCE's naturalness but can offer comments from our own experiences.

As we have demonstrated, ASSISTANCE is capable of interpreting the types of input media and language that designers use in the early stages of design. The explanations include many features commonly found in person-to-person explanations such as the use of natural language, sketching, simple deictic references, and the use of behavior-oriented explanations instead of parameter-oriented ones. The representations generated by ASSISTANCE are in a machine readable form suitable for use by other reasoning systems. By generating such representations ASSISTANCE is capitalizing on one of the primary advantages of CAD tools.

One area of the interface we hope to improve is its ability to provide feedback to the user about its current level of understanding. One way to do this would be to involve the computer in a dialog with the designer, in which the system asks for clarifications and asks questions about the roles played by different components. This would both provide the designer feedback about the system's current understanding of the explanation and offer some structure to the explanation which may provide additional constraints on the interpretation of the explanations. This inclusion of the computer as an active participant is an approach that is also advocated in [8] and fits into our overall goal of providing an interface that is as close as possible to person-to-person interactions.

A second improvement would be the extension of the reference disambiguation facilities. As mentioned earlier (Section 5.3) there has been previous work on this topic (e.g. [14, 13]) to provide guidance in expanding the system's current capabilities in this area.

A third improvement would be the extension of the natural language capabilities. The grammar of recognized utterances is currently too small to allow designers who have not previously used the system to describe a device easily. This difficulty is complicated by occasional errors in the speech recognition. Using a mature speech understanding system such as START [10] will alleviate some of these problems, by accounting for common language structures such as passive voice.

6.3 Expand reasoning abilities

Future work also needs to be done to expand the range of devices that the system can understand. In particular the limitation that each body

can be involved in only a single state transition precludes many common devices. We hypothesize that this does not pose any conceptual level adjustments to the architecture but it will involve reengineering some of the internal representations of the reasoning system. The ability to manipulate the components of the device directly (in addition to just describing them) could help this problem from a usability perspective by visually displaying the current state of the device. Without this feature the designer must visualize the new positions of components after each event occurs. As the chain of events involving the same component grows this will become a larger issue.

7 Related work

While a variety of work has explored the understanding of descriptions in individual modalities, and some multi-modal systems deal with direct manipulation tasks, relatively little work has attempted to interpret the sorts of multi-modal descriptions handled by ASSISTANCE.

7.1 Related description understanding systems

Borchardt [4], for example, parsed natural language descriptions of device behavior and from this reconstructed the causal relationships described in the text. His insight was to focus on the changes that occurred in the state of the device instead of the states themselves. This closely parallels our goal to focus on descriptions of behavior instead of descriptions of structure. The primary difference in our work derives from having sketched input; having explicit spatial models simplifies many descriptions. This changes the focus of the descriptions and allows them to be conveyed more naturally.

Understanding device behavior was also an element in [16]. In that system the designer specified structure by indicating some elements of the topology of the device and described behavior with a state transition diagram. From these representations, his system was able to understand the operation of the device and suggest alternative designs with the same qualitative behavior as the original. To engineers, a state transition diagram is one form of natural explanation, and as such that work took one step in the direction we have pursued.

Another approach to the behavior understanding problem is to infer the device's behavior by observation, without a separate behavioral description. This was the approach taken in [12] to infer the behavior of a device from static diagrams. Similarly, [5] and [6] interpret the

behavior of devices from images and sequences of images of a device in action. These approaches are important because they could provide an initial guess at the behavior which can be augmented and repaired by explanations provided through a system such as ours.

7.2 Related multimodal systems

There has been a great deal of work done on the design and theory of pen and speech based multimodal interfaces (see [13] for an overview). This work has focused on improving recognition accuracy by combining multiple input modalities. It has also identified general properties of multimodal human computer interactions that can guide their design. [14].

Another body of work in the field of multi-modal interfaces has focused on recording human interactions in meetings[3]. The goal of that work is to generate annotated multimedia transcripts of meetings. The transcripts include the text of what was said, who said it, and links between the transcript and video sequences.

ASSISTANCE fits between these two bodies of work in its emphasis on being a silent observer but also understanding the content the user is conveying instead of just recording it in a structured manner. Another system which takes this approach is Rasa described in [11].

8 Contributions

ASSISTANCE demonstrates a new kind of interface for describing mechanism behavior. Rather than proposing better templates, buttons, or menus, ASSISTANCE adopts the interface that designers use everyday to communicate with their colleagues. Armed with knowledge about sketching, natural language, and mechanical devices, ASSISTANCE brings the computer into the designer's world. During conceptual design, designers talk about behaviors and not the parameters that lead to them, hence ASSISTANCE focuses on understanding behavioral explanations rather than providing ways of specifying parameters.

9 Acknowledgments

This work was supported in part by the Ford-MIT Collaboration, and in part by the MIT Oxygen Partnership.

References

1. C. Alvarado and R. Davis. Resolving ambiguities to create a natural computer-based sketching environment. In *IJCAI-01*, 2001.
2. V. Baya and L. Leifer. Understanding information management in conceptual design. In *Analysing Design Activity*, pages 151–168. John Wiley, 1996.
3. Michael Bett, Ralph Gross, Hua Yu, Xiaojin Zhu, Yue Pan, Jie Yang, and Alex Waibel. Multimodal meeting tracker. In *Proceedings of RIAO2000*, April 2000.
4. G. Borchardt. Causal reconstruction. Tech. Report AIM-1403, MIT, February 1993.
5. M. Brand. Physics-based visual understanding. *CVIU*, 65(2):192–205, February 1997.
6. T. Dar, L. Joskowicz, and E. Rivlin. Understanding mechanical motion. *Artificial Intelligence*, 112:147–179, 1999.
7. K. Forbus and J. de Kleer. *Building Problem Solvers*. MIT Press, Cambridge, MA, 1993.
8. Kenneth Forbus, Ronald Ferguson, and Jeffery Usher. Towards a computational model of sketching. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*, pages 77–83, January 2001.
9. M. Hearst. Sketching intelligent systems. *IEEE Intelligent Systems*, pages 10–18, May/June 1998.
10. B. Katz. From sentence processing to information access on the world wide web. In *AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, 1997.
11. D. McGee, P. Cohen, and L. Wu. Something from nothing: Augmenting a paper-based work practice via multimodal interaction. In *Proceedings of Designing Augmented Reality Environments*, pages 71–80. ACM Press, April 2000.
12. N. Narayanan, M. Suwa, and H. Motoda. *Hypothesizing Behavior from Device Diagrams*, chapter 15, pages 501–534. MIT Press, Cambridge, MA, 1995.
13. 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 gesture applications: State-of-the-art systems and research directions. In *Human Computer Interaction*, volume 15, pages 263–322. Addison-Wesley, 2000.
14. Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Speech, Haptic, & Multimodal Input*, pages 415–422, 1997.
15. M. Palmer, R. Passonneau, C. Weir, and T. Finin. The KERNEL text understanding system. *Artificial Intelligence*, 63(1-2):17–68, October 1993.

16. T. Stahovich, R. Davis, and H. Shrobe. Generating multiple new designs from a sketch. In *Proc. 13th AAAI*, pages 1022–1030, Menlo Park, August 1996. MIT Press.
17. D. Ullman, S. Wood, and D. Craig. The importance of drawing in the mechanical design process. *Computers and Graphics*, 14(2):263–274, 1990.