Understanding Naturally Conveyed Explanations of Device Behavior

by

Michael Oltmans

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2000

Author . . . .
Department of Electrical Engineering and Computer Science
August 18, 2000

Certified by . . . .
Randall Davis
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Understanding Naturally Conveyed Explanations of Device Behavior

by

Michael Oltmans

Submitted to the Department of Electrical Engineering and Computer Science
on August 18, 2000, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

One of the research goals of the field of human computer interaction is to make interacting with a computer as natural as interacting with a person. In pursuit of this goal, this thesis demonstrates a system that understands naturally conveyed explanations of mechanical devices in a way that traditional tools do not support. When using a traditional mechanical CAD program, for example, the designer formally specifies a spring connected to a block and must set parameters such as the length of the spring and its rest length. This is in contrast with the way a designer would describe the system to a colleague, by sketching the block and spring and saying "the spring pushes the block." Designers should not be forced to encode the behavior of the device in its parameters if the parameters are not the primary concern. This thesis presents a system, called ASSISTANCE, that understands naturally conveyed behavioral explanations. It demonstrates its understanding by generating a behavioral model for the device and answering questions posed by the designer about its model. ASSISTANCE allows a designer to sketch, gesture, and talk about a design in a familiar fashion, thus lowering the barriers to communication between the designer and the computer.

Thesis Supervisor: Randall Davis
Title: Professor of Computer Science and Engineering

Acknowledgments

- I would like to thank my advisor Randall Davis for many helpful discussions and for giving me a blend of guidance and freedom.
- I would like to thank Christine Alvarado for countless discussions, pieces of advice, and for creating ASSIST, all of which made this thesis and ASSISTANCE possible.
- I also extend my thanks to the people around the AI Lab and the Laboratory for Computer Science who took the time to describe mechanical devices for the survey.
- I would like to thank Ford Motor Company for their financial support.
- I would like to thank my officemates Mark Foltz and Christine Alvarado and our honorary officemates who receive solace from our couch, for the countless hours of tech support, emotional support, and fun loving goodness that make life in office 808 so great.
- Finally, I would like to thank my family and friends for encouraging, prodding, teaching, and putting up with me. Especially Amy Antman for the the cookies, proof reading, encouragement, and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Lowering the bar

When mechanical designers explain a device to a colleague, they do so by sketching and verbally explaining the behavior of the device. When they enter the same design into a computer, though, they must use a mouse and keyboard, and they are forced to encode the behavior of the device by specifying coordinates and parameters of components, instead of describing the behavior naturally and directly. This thesis presents a system that provides a more natural style of interaction with the designer by understanding descriptions that are conveyed using familiar media like sketches and verbal explanations. The system uses its knowledge of mechanical devices and its knowledge about the manner in which designers describe mechanical devices to understand the description and generate a model of the device. This combination of natural inputs and knowledge centered interpretations of the devices makes describing a mechanical design far easier and far less formal than is possible with current tools. In turn, this less formal interaction allows the designer to incorporate the computer into the early stages of the design cycle.

As a simple example, consider a spring attached to a block which is positioned next to a ball. The spring pushes the block which pushes the ball. In a traditional system the designer would select the components from a toolbar and place them into the model, as in Figure 1-1. Next the designer would specify a variety of parameters, such as the rest length of the spring, so that the device will exhibit the desired behavior.

Contrast this to the way a designer would describe this simple device to a colleague. The description would probably consist of a quick hand drawn sketch (like the one shown in Figure 1-2) and a brief spoken description like "the block pushes the ball." This is the type of description that designers use; this is the type of description that computers should understand.

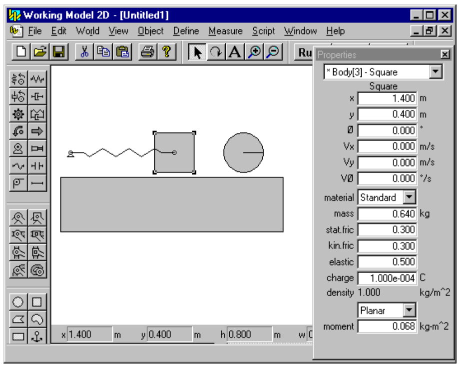


Figure 1-1: **A block and spring constructed in CAD:** The designer selects components from templates and specifies parameters.

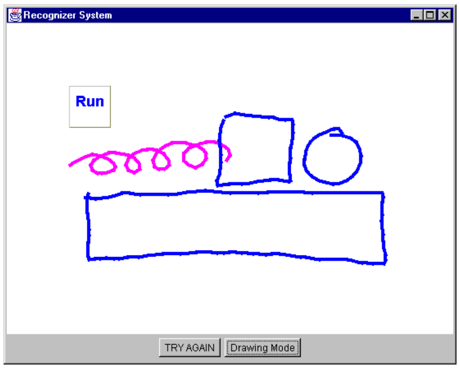


Figure 1-2: **A block and spring described in Assistance:** The designer sketches the block and spring and says "the block pushes the ball." From this description the system generates a simple causal model in which the spring causes the block to move and that in turn causes the ball to move.

This raises the question of what it means for a computer to understand a description of a device. In this thesis we consider a computer to understand a device description if it can generate an accurate model of the structure and behavior of the device. The behavioral part of the model should be capable of describing what actions cause which other actions and should be a synthesized representation, rather than a simple recording of the description given by the designer. The behavioral part of the model will be referred to interchangeably as the causal or behavioral model of the device.

Despite having unnatural interfaces, CAD systems are still useful because they allow designers to simulate and analyze the operation of a device. The solution described here uses a sketching and speech based interface to generate both a CAD model and an explicit behavioral model of the device. This has been implemented as two complimentary computer systems called ASSIST (A Shrewd Sketch Interpretation and Simulation Tool) and ASSISTANCE (ASSIST Augmented with Naturally Conveyed Explanations). The first system, ASSIST, transforms a sketch into a model that can be simulated. ASSIST is described briefly in Section 4.2. A more detailed description can be found in (Alvarado, 2000). The second system, ASSISTANCE, allows the designer to describe the behavior of a device sketched in ASSIST by sketching annotations, such as arrows, and verbally describing the behavior. ASSISTANCE is the topic of this thesis.

1.1 The device description understanding task

In this thesis the device description understanding task is defined as the problem of generating a structural and behavioral model of a mechanical device. The descriptions understood by ASSISTANCE are in the form of sketched and verbalized annotations made in reference to a sketch of the device.

The understanding of a device's behavior from sketches and verbal descriptions explores a variety of important issues ranging across mechanical device reasoning, natural language understanding, and multimodal interface design.

1.2 A brief example

This section gives a brief overview of how a designer interacts with ASSISTANCE to describe a Rube Goldberg device that cracks an egg. The egg-cracker works as follows: when the user pulls up the stopper the spring releases and pushes the ball which rotates the seesaw and then the knife falls and cracks the egg (Figure 1-3). The designer sketches the device on a digitizing sketchpad or a whiteboard that records the strokes drawn on it and speaks into a microphone to describe the device. A transcript of the utterances and gestures made by the designer are shown in Figure 1-3.

After describing the device the designer can query ASSISTANCE about its model of the device. This interaction is depicted in Figure 1-4. The responses that ASSISTANCE gives are often similar to the descriptions provided by the designer, however, they are constructed from the behavioral model that it generated. ASSISTANCE is not simply parroting the designers description; it has understood the designer's gestures and speech, and built a causal model of the device.

1.3 Why designers need ASSISTANCE

Three main observations motivate providing ASSISTANCE to designers. First, while current design tools are beneficial during the detailed design process, they are hard to use during the early, conceptual stages of design. The primary problem with CAD systems and formal device reasoning systems is that they are too detailed for rapidly changing initial designs. Mechanical CAD tools use input media that are different from the media designers use when designing away from the computer and when communicating with other designers.

Second, design rationale capture is becoming increasingly important and ASSISTANCE sets the stage for capturing information from the early design stages that might be useful in the construction of a comprehensive rationale capture system.

Finally, the representations generated by ASSISTANCE could provide a stepping stone upon which further design tools can be constructed.

1.3.1 Current design tools do not suit early design

The majority of tools that designers have at their disposal do not fit the needs of the designer in the early design stages. Consider how some of the common tools would represent the behavior of a spring attached to a block. CAD tools, for example, force the designer to think in terms of parts and parameters instead of the overall structure of a device and its behavior. For example, the designer would set the length of the spring and its rest length to describe how it will behave. Evidence for the unsuitability of CAD tools in the early design stages has been noted for architectural design by Branda (1989) and in the mechanical engineering domain by Ullman *et al.* (1990).

Other tools have been developed for performing qualitative analysis of designs with incomplete information (de Kleer and Brown, 1984; Iwasaki and Simon, 1986). While these systems are intended to allow analysis and examination of designs before they have been fully parameterized, they require hand generated models composed in a formal modeling language[1]. These systems free the designer from specifying specific parameter values by allowing the designer to express qualitative relationships between parameters, such as the relationship that the rest length of a spring is longer than its current length. However, the designer is still unable to explicitly describe the device's behavior and is forced to indirectly encode it in the device's parameters by specifying relations, e.g. the rest length of the spring is longer than its current length.

Tool:	**Type of description:**
Mechanical CAD	spring length = 2.3cm spring rest length = 3.0cm
Qualitative Reasoner	(< (length spring) (rest-length spring))
ASSISTANCE	"The spring pushes the block"

Table 1.1: **Three ways to specify the behavior of a spring and block:** Only ASSISTANCE allows the designer to explicitly mention the behavior of a block attached to a spring (pictured in Figure 1-1)

ASSISTANCE addresses these problems by enabling designers to work directly with a device's behavior, allowing the designer to say "the spring pushes the block." This not only eliminates the need to choose relationships between parameters, but eliminates the need to define the parameters altogether and frees the designer from dis-

[1]However, see (Stahovich *et al.*, 1996), for an interesting exception that uses sketches and state transition diagrams.

cussing the device in a formal language.

1.3.2 Current design tools are unnatural

ASSISTANCE is important because it uses natural input media. Designers use pencils and whiteboard markers when they describe their devices to colleagues, but are forced to interact with CAD tools using mice and keyboards. Although some CAD systems allow the designer to use a digital sketchpad of sorts, the designer must still select components from a template before drawing each part, quite unlike sketching with paper and pencil (Ullman *et al.*, 1990). Furthermore, the only way the designer can specify the desired behavior of the model is by specifying parameters that will cause the model to demonstrate the desired behavior when simulated; there is no explicit representation of the behavior. This is in contrast to the way a designer uses ASSISTANCE to directly specify the behavior with verbal and sketched descriptions. The familiarity designers have with describing devices by sketches and speaking greatly reduces the effort required to record an initial concept or incomplete design, increasing the likelihood that early design models will be recorded.

1.3.3 Computer based design tools have benefits over pencil and paper

Despite their shortcomings, existing systems provide many important benefits by proving a means to perform simulations, visualizations, and parameter estimations. After the initial design is done on pencil and paper, these benefits are worth the effort required to enter the design into the computer. ASSISTANCE simplifies this process by allowing the designer to effectively express the design to the computer at an early stage of the design process and move more fluidly between conceptual designs and formal models.

Computers also allow for the modification and reuse of previous designs, something not easily done with pencil and paper. ASSISTANCE complements this by combining a paper like interface with the advantages of a computer model that can be edited or reused.

1.3.4 Involving computers in early design encourages design rationale capture

Design rationale capture is becoming increasingly important. As devices become more complex and involve more designers, better documentation about the design, and about the evolution of the design will be necessary (Moran and Carroll, 1996). Having records of old versions of a design can provide insights into why the device behaves the way it does and can help prevent the reintroduction of old flaws in future design modifications.

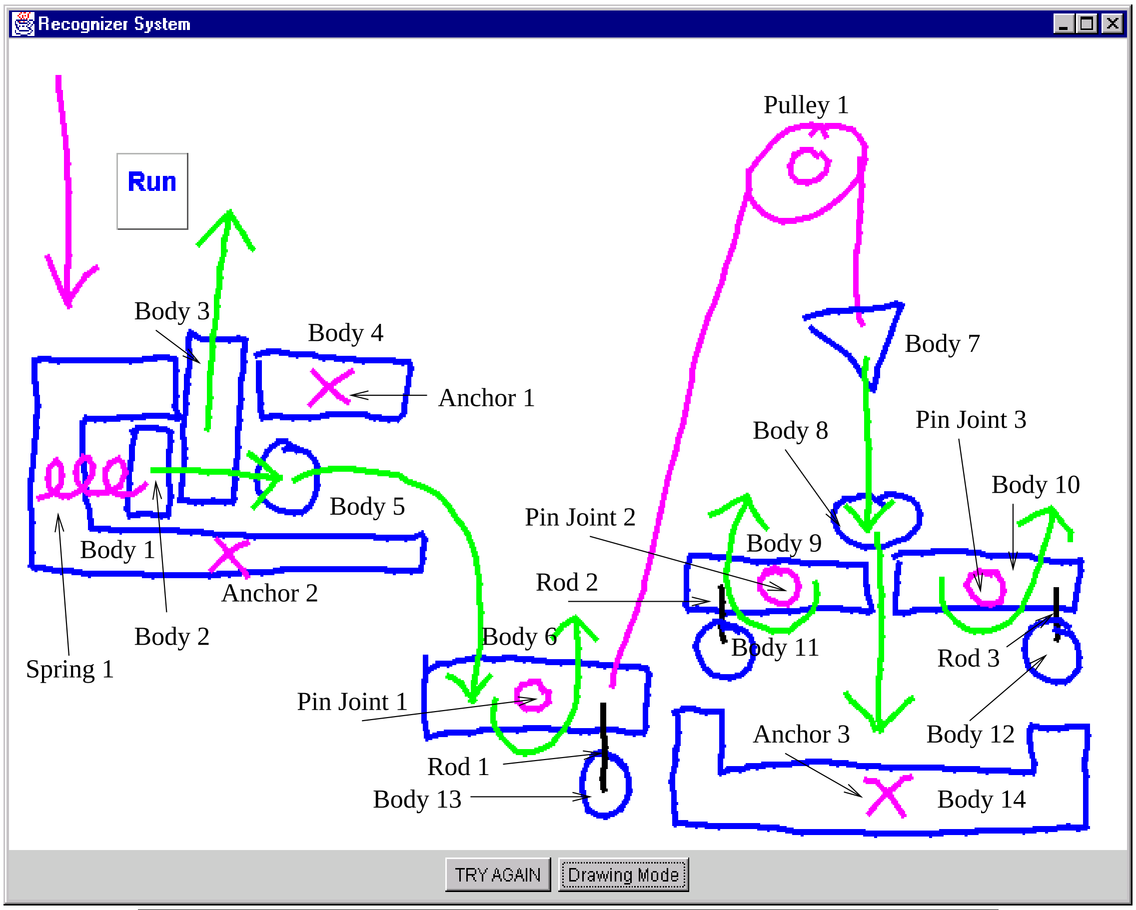
ASSISTANCE can be seen as a first step in building comprehensive design rationale capture systems by collecting and organizing the structure and behavior of devices, even early in the design process. One of the largest obstacles to effective design rationale capture is that the information is often more trouble to collect than it is worth. ASSISTANCE greatly decreases the formality and tedium of generating snapshots of a design that are useful in understanding a device's rationale.

1.3.5 Behavioral models might aide intelligent design tools

ASSISTANCE is an important first step in building better and more natural design tools. One possibility that ASSISTANCE makes available is a design assistant that uses the causal structure from ASSISTANCE to suggest alternate designs that have the same resulting behavior. One such system that could be integrated with ASSISTANCE is SketchIt by Stahovich *et al.* (1996) which suggests alternate designs for a sketch.

1.4 Contributions

The primary contribution of this work is the demonstration that it is possible to use naturally conveyed explanations to reduce the impedance of interacting with computerized design tools. ASSISTANCE demonstrates this by understanding explanations of devices that are conveyed using familiar media, such as sketching and speaking. However, the use of natural input media is not the full story; the designer is also able to express device behaviors explicitly and does not have to translate them into a set of parameters that produce the desired behavior when simulated. For example, the designer is able to say, "the spring pushes the block" instead of being forced to explicitly assign a value for the rest length of the spring. The goal of ASSISTANCE is to make the computer act like a naive colleague and not to make the designer act like an intelligent mouse.



when body 3 moves up spring 1 releases
body 2 pushes body 5
body 5 falls
body 6 rotates
body 7 falls
body 7 pushes body 8

Figure 1-3: **Describing the egg-cracker:** The arrows in the sketch are gestures describing the motion of the bodies. The bodies are shown in blue and the constraint components are shown in pink. Below the sketch is a transcript of the utterances that made up the verbal part of the explanation. (The labels shown in the figure are automatically generated by the system, however, the current implementation often positions them in ways that are hard to read. Because of this the figures in this thesis use hand positioned labels for clarity. For a more complete explanation of this issue and a sample of the original program display refer to Appendix B.)

Designer:	What is body 2 (the block) involved in?
ASSISTANCE:	When body 3 (the stopper) moves up body 2 (the block) pushes body 5 (the ball).
Designer:	What is body 7 (the knife) involved in?
ASSISTANCE:	When body 6 (the seesaw) rotates body 7 (the knife) moves. Also when body 7 (the knife) moves body 8 (the egg) moves.
Designer:	What is body 9 (the left egg support) involved in?
ASSISTANCE:	When body 8 (the egg) moves body 9 (the left support) rotates.

Figure 1-4: **Exploring the causal model:** This shows a transcript of a verbal interaction between the designer and ASSISTANCE

Chapter 2

The guiding principles of ASSISTANCE

The goal of ASSISTANCE is to demonstrate that naturally conveyed descriptions can be understood by a computer system. ASSISTANCE provides such a demonstration by understanding naturally conveyed explanations of mechanical devices and generating behavioral models for them. In order to realize this goal, ASSISTANCE makes use of a number of observations about the way designers describe devices and about the way these observations can be utilized.

2.1 Principles

While many ideas have been combined in order to make ASSISTANCE, several basic principles and observations are fundamental. These are:

- Natural input media facilitates interaction
- Behavior is more important than parameters in early design
- Local inferences about the device model improve understanding
- Explanation order indicates the causal order

- Overlapping information sources, such as topological, spatial, verbal, and temporal, lead to richer representations

2.1.1 Natural input media facilitates interaction

In Assistance there are two distinct notions about the meaning of natural inputs. The first notion is that natural inputs come from input media similar to those routinely used by designers when interacting with their colleagues. The second is that the device is described in a language that emphasizes the same concepts that designers use when describing devices to their colleagues.

Natural input media facilitate interaction between the designer and the computer at an earlier stage by reducing the effort required by the designer to enter a design into the computer. Designers are familiar with sketch and speech based interactions because they use them in daily interactions with colleagues and during design.

2.1.2 Behavior is more important than parameters in early design

The second notion of natural input is concerned with the type of information being conveyed, independent of the communication media. It has been observed that in the early stages of design, designers pay more attention to the behavior of the device than the properties of the individual components (Baya and Leifer, 1996). While device parameters are important, it appears that most descriptions are phrased in behavioral terms and do not explicitly mention parameter values or even include explicit references to parameters. This indication is further supported by the descriptions gathered in an informal survey, which is discussed in Section 3.5. Therefore, Assistance focuses on understanding natural language indications of behavioral aspects of a device rather than phrases relating parameter values.

2.1.3 Local inferences about the device model improve understanding

In addition to interpreting an explicit device description, ASSISTANCE is capable of using simple heuristics to directly infer information from the structural model and the sketch. These heuristics look at localized portions of the topology of the device and the geometric properties of the components involved to make assertions about the possible behaviors of a component.

For example, ASSISTANCE uses context and its knowledge of mechanical components to infer that a lever might be balanced when it has opposing forces acting on it. This method of using context is more efficient than an approach that finds all the forces acting on all the bodies regardless of whether or not it makes sense to talk about them being balanced. In addition, the local, context based approach more closely represents the intuition that a lever can balance if there are opposing forces applied to it. Because ASSISTANCE is meant to be used in the conceptual stages of design, the informal model suggested by the local balance rule is more appropriate than the more formal approach of finding all the possible forces.

Another way that local inferences are used is in the determination of qualitative relationships between parameters. There are heuristics that analyze the described behaviors and suggest qualitative relationships between parameters that will cause the device to achieve those behaviors. For example, in a device consisting of a spring attached to a block, it is possible to infer whether or not the spring is compressed from information about the direction the block moves and the fact that the motion is caused by the spring. These heuristics do not attempt to take into account all of the parameters of the device but rather just the local interactions between components.

2.1.4 Explanation order indicates causal order

Many of the tasks performed by ASSISTANCE can be simplified by observing that designers tend to follow certain patterns in the way they describe their designs. One important convention observed in an informal survey (described in Section 3.5) was that designers generally describe the behavior of a device in the order in which the behaviors occurs. One consequence of this observation is that ASSISTANCE can look first at the inputs that occur within a small window to determine the causes and effects of a given action. Another is that local rules can be used to identify multiple inputs representing the same physical event, without having to search the entire input stream. While not all descriptions follow this convention, it is obeyed often enough to make a good default assumption.

2.1.5 Overlapping information sources lead to richer representations

Device descriptions often contain events that are described using several information sources. The overlap in these sources allows ASSISTANCE to improve its understanding of the event by combining them. There are three primary categories of input sources:

Topological information consists of the attachments of bodies to constraints and the connections between bodies. Topological information is naturally conveyed in the sketch of the device and interpreted by the sketch understanding program, ASSIST.

Spatial information deals with the geometric properties of the components and the spatial relationships between them. It is also conveyed by the sketch of the device.

Linguistic information includes the verbal descriptions of behaviors and the links between them.

When information from two input sources is combined it forms a more complete description of the physical event that it represents. For example, consider a block attached to a spring and a description that consists of the phrase "the spring decompresses" and an arrow indicating the motion of the block. The verbal phrase conveys the information that the spring is decompressing instead of being stretched beyond its rest position. The topological information representing the connection of the spring to the block implies that the spring is the cause of the motion of the block. Finally, the arrow annotation indicates the path taken by the block. The combination of each of these sources leads ASSISTANCE to a more complete representation of the described event.

2.2 Making ASSISTANCE flexible

The device description task is of course too large and complex to be completely solved by ASSISTANCE. In order to allow for future explorations and the testing of new approaches to the problem, ASSISTANCE has been designed to be highly modular.

2.2.1 Modular architecture

To achieve the flexibility necessary to explore other aspects of the description understanding task, ASSISTANCE follows the lead of systems like *Hearsay II* (Lesser and Erman, 1977) which attempt to separate the implementations of knowledge sources from one another. The knowledge sources in ASSISTANCE perform:

- Recognition of the device sketch
- Recognition of gestures and utterances
- Interpretation of recognized gestures and utterances
- Synthesis of interpretations

Each of these knowledge sources builds on the inferences drawn by the others but makes minimal use of the exact inferences made by the other sources. In this way each of these knowledge sources can be modified or replaced or new sources can be added without greatly affecting the others.

2.2.2 Handle noisy input

ASSISTANCE must also be flexible in how it handles its inputs. The sketch, and especially the speech recognition software, occasionally misinterpret their inputs. As-

SISTANCE attempts to cope with these inaccurate interpretations by allowing for the straightforward retraction of an input and all of its consequences in the event that the input is determined to be erroneous.

2.3 What ASSISTANCE does not attempt to do

This section identifies some of the tasks that are outside the scope of the current implementation of ASSISTANCE. These limitations on the full device description task were necessary in order to get a foothold on the task and to begin exploring the issues that it poses.

2.3.1 Understanding a description versus understanding a device

Understanding a device description is distinct from inferring the complete physical model of the device. ASSISTANCE attempts to form a model for the behavior that the designer describes, but does not attempt to infer complex behaviors of the device that go beyond the explicit explanations.

The full device understanding task is quite complex and requires large amounts of knowledge about the intended use of a device. Because of this complexity, ASSISTANCE would be a natural compliment to such a system. The designer can sketch a device and let the device understanding system attempt to infer as much of the device's behavior as possible. ASSISTANCE could then fill in the gaps by asking the designer to describe parts of the device that the other system did not understand from the sketch alone.

2.3.2 Restricted language structures and domain

Another limitation of ASSISTANCE is that the designer must use a constrained subset of natural language, because understanding unconstrained natural language, even in a limited domain, is too difficult. Instead of attempting to grapple with unconstrained language the grammar incorporates only common language constructs that are useful in describing mechanical systems.

One example of this limitation is that the physical components must be referred to by preassigned labels, e.g. "body 1." This requirement simplifies the grammar significantly and removes a large number of the ambiguities that make natural language difficult to understand. There has been work done on interpreting deictic references (e.g. Cohen *et al.*, 1998) and the incorporation of this feature is planned for future implementations.

The decision to simplify the range of parsable sentences has some support from the observation that multimodal interactions use speech differently than unimodal systems. The use of speech in multimodal systems is abbreviated and syntactically simpler than in unimodal systems (Oviatt, 1999). For example, instead of attempting to verbally describe the path traced by a moving object, the designer can simply draw an arrow indicating the path. Many of the difficulties of understanding natural language are simplified by integrating information from the sketch and from sketched gestures.

2.3.3 A benevolent teacher makes explanations easier to understand

ASSISTANCE also assumes that the designer describing the device is providing a straightforward mechanical description of the device. ASSISTANCE does not currently understand common descriptive mechanisms such as analogies or metaphors. This goes against the principle that the designer should be able to use the same descriptive language to communicate with the computer that is used to communicate with a colleague. But even with restricted language the descriptions accepted by ASSISTANCE are closer to a designer's natural description than a description that consists entirely of parameter specifications.

Chapter 3

An example of ASSISTANCE at work

This chapter explains at an information processing level the type of reasoning that ASSISTANCE performs, focusing on the types of inferences the system makes rather than on the mechanisms used to implement them. The discussion of the mechanism–a rule system–can be found in Chapter 4.

3.1 Inputs to the system

There are three input sources to ASSISTANCE: the device sketch, gestures, and verbal utterances. The device sketch is the hand drawn sketch made by the designer that represents the structure and geometry of the device. The sketch is parsed into its component parts, such as bodies, springs, pin joints, and pulleys by ASSIST. For a detailed description of how this is done see (Alvarado, 2000). The parsed representation of the device is used by ASSISTANCE to make simple inferences about the device's structure and to provide a context within which to understand the description of the device.

The sketched gestures are made by the designer while describing the device's behavior. The primary gestures are arrows indicating motions of objects and pointing gestures that refer to specific locations in the sketch. ASSIST parses the pen strokes into arrows and pointing gestures and passes them to Assistance to be interpreted in the context of the current device description. Examples of this interpretation by Assistance are given in Section 3.3.

The verbal utterances are natural language sentences that describe the device's behavior. They describe the events that occur during the operation of the device and the causal relationships between these events. For example, the sentence "When the stopper moves up the spring releases" describes two individual events, the motion of the stopper and the action of the spring. It also indicates that the motion of the stopper leads to the action of the spring. The sentences are parsed according to a grammar that was designed for the device description task. The parsed result is then interpreted by Assistance.

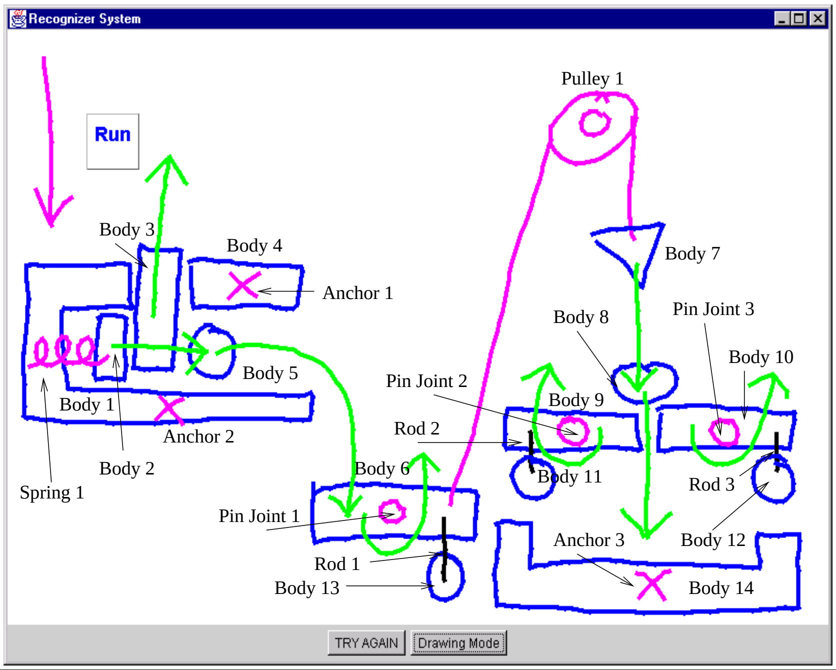
3.2 The models generated by the system

Two different models are generated in the process of understanding a device description: structural and behavioral. The structural model is generated by Assist from the device sketch. It represents the geometry of the physical components and the topological connections between them. The generation of this model is described briefly in Section 4.2 and more extensively in (Alvarado, 2000).

The behavioral model is generated by Assistance from the description of the device. The behavioral model contains descriptions of the events that occur during the operation of a device and indicates the causal relationships between these events. The construction of this model is one way that Assistance demonstrates its understanding of the device description. Assistance demonstrates its understanding to the designer by using the behavioral model to answer questions posed by the designer about the causal structure of the device.

3.3 How ASSISTANCE interprets the description of a Rube Goldberg egg-cracker

An example will demonstrate how ASSISTANCE interprets a description and generate a behavioral model for it. A sketch of an egg-cracker and a transcript of the description of it is shown in Figure 3-1. We describe the inferences made from the annotations and the knowledge required to make them.



when body 3 moves up spring 1 releases
body 2 pushes body 5
body 5 falls
body 6 rotates
body 7 falls
body 7 pushes body 8

Figure 3-1: **The description of the egg-cracker:** The arrows in the sketch are gestures made by the designer and they describe the motions of the bodies. Below the sketch is a transcript of the utterances that made up the verbal part of the explanation.

3.3.1 Making inferences from the device's structure

Assistance begins by making several static observations about the egg-cracker. For example, it infers that body 6 (the seesaw) is a lever because it has an oblong shape and only one degree of freedom. This inference supports other inferences such as the observation that the lever may be balanced because there are opposing forces acting on it, e.g. the rope pulls it up and the weight pulls it down. Assistance uses the topological structure of the device to determine possible forces on a body. Components such as the pulley and the rod with a weight attached to it are examples of the types of components that apply forces on an object.

3.3.2 Generating the behavioral model from the description

The following description shows the inferences made by Assistance after each annotation it receives.

Utterance: "When body 3 (the stopper) moves up spring 1 releases"

From this utterance Assistance deduces that the motion of body 3 (the stopper) causes the motion of body 2 (the block) and updates the behavioral model to indicate this. This deduction is based on three inferences from the sentence. First, Assistance constructs a representation for the motion of body 3 (the stopper) based on a straightforward interpretation of the clause "body 3 moves up".

Second, Assistance infers the motion of body 2 (the block). The clause "spring 1 releases" implies that the objects connected to the spring might be moving. Because spring 1 is anchored on one end Assistance infers that body 2 (the block) is moving and generates a representation for this motion. This is an example of an inference based on physical reasoning about the device.

Finally, Assistance infers the causal connection between these two motions from the structure of the "when" statement, in which the first clause is a condition for the second. This is an example of using linguistic properties to infer the causal structure of a device's behavior.

Gesture: The arrow for body 3 (the stopper)

From this arrow, ASSISTANCE generates a second event representation for the motion of body 3 (the stopper), which describes the path it moves along. Then ASSISTANCE links this representation with the representation generated by the utterance above. This linkage is inferred by recognizing that the two descriptions describe the same type of motion and involve the same object. Similarly, the arrow for body 2 (the block) generates a second representation for its motion and links them together. The links between event representation will later be used to merge them into a single, detailed representation of each event.

Utterances: "body 2 (the block) pushes body 5 (the ball)" and "body 5 (the ball) falls"

From these sentences, ASSISTANCE extends its model to include the fact that the motion of body 2 (the block) causes the motion of body 5 (the ball).

ASSISTANCE interprets the "pushes" phrase as two motion events where the first causes the second. The first is the motion of body 2 (the block), which now has three separate representations. This additional representation is linked with the other two. The second is the motion of body 5 (the ball). The causal link is inferred by the fact that *pushing* is the act of one object causing another object to move.

At this point ASSISTANCE's behavioral model represents the fact that the motion of body 3 (the stopper) causes the motion of body 2 (the block), which causes the motion of body 5.

Utterances: "body 6 (the seesaw) rotates," and "body 7 (the knife) falls"

From these utterances, ASSISTANCE infers a causal link between the motions of body 6 (the seesaw) and body 7 (the knife) because a rope and pulley connect them. The upward motion of the end of the seesaw generates slack in the rope and allows the downward motion of the knife. ASSISTANCE uses this piece of physical reasoning and the topology of the device to infer the causal link between the two motion events.

Gestures: The arrows for body 8 (the egg) and its two supporting levers

From these three arrows ASSISTANCE infers that there is a motion event associated with each of them. It also infers that the motion of the egg causes the motions of the two levers. This inference is made because the path followed by the egg brings it into contact with the levers. This geometric calculation is done by observing that there is an intersection of the space traced by the egg as it falls and the space occupied by the two levers.

ASSISTANCE also observes that the motions of the two levers are rotations because the bodies have a rotational degree of freedom but do not have any translational degrees of freedom.

3.3.3 Event canonicalization

As mentioned above ASSISTANCE generates more than one representation for some events because of overlapping input sources. The multiple representations are linked if the representations have consistent properties. If there are contradictory properties, such as simultaneous motions in two different directions, then the two representations describe distinct events. By transforming these sets of representations into a canonical form ASSISTANCE gains a more complete description of each event.

3.4 How ASSISTANCE demonstrates its understanding

After ASSISTANCE processes the description, it has both a structural model and a behavioral model of the device. ASSISTANCE can demonstrate its understanding of the device by composing answers to questions about the causal structure of the device and by adjusting the parameters of the components in the structural model so that the simulation of the device is closer to the behavior that the designer described.

3.4.1 Browsing the model

ASSISTANCE uses its knowledge of the causal structure of the device to describe the events that a component is involved in and the immediate causes and effects of those events. An example of an interaction between a designer and ASSISTANCE is presented in Figure 3-2.

Designer:	What is body 2 (the block) involved in?
ASSISTANCE:	When body 3 (the stopper) moves up body 2 (the block) pushes body 5 (the ball).
Designer:	What is body 7 (the knife) involved in?
ASSISTANCE:	When body 6 (the seesaw) rotates body 7 (the knife) moves. Also when body 7 (the knife) moves body 8 (the egg) moves.
Designer:	What is body 9 (the left egg support) involved in?
ASSISTANCE:	When body 8 (the egg) moves body 9 (the left support) rotates.

Figure 3-2: **Exploring the causal model:** A transcript of a verbal interaction between the designer and ASSISTANCE.

3.4.2 Improving the structural model

Although the primary focus of ASSISTANCE is on understanding the behavior of the device, ASSISTANCE is capable of making improvements on the structural model produced by ASSIST. With the knowledge that body 2 (the block) is propelled by spring 1, ASSISTANCE is able to adjust the default spring parameters assigned by ASSIST so that it is in compression. The device model then produces the correct behavior for the block's motion when the device is simulated.

3.5 A survey of actual device descriptions

The preceding description was modeled after the descriptions that were collected for the egg-cracker in an informal survey of how people describe mechanical devices. The goal of the survey was to gather data about the way people describe simple mechanical devices. The transcripts gathered from the survey provided an excellent resource from which to identify the types of gestures, phrases, and behaviors people use to describe mechanical devices. The transcripts were also used used to identify some of the difficulties and guidelines that are discussed in the previous chapter.

3.5.1 Survey Methodology

Five people participated in the survey, of which two had significant mechanical design training and experience. All of the subjects were members of the MIT community. Each subject was shown the same three designs: a simple box with a latching lid, the Rube Goldberg egg-cracker, and a self resetting lever (see Figure 3-4). The subjects were asked to explain each device through gestures and verbal explanations. The results were recorded and later transcribed. A transcript of one of the explanations of the egg-cracker is shown in Figure 3-3.

- when this (the stopper) is pulled up and removed then the spring releases pushing this ball here which may or may not hit this (wall)
- falls down here and pushes down on this (the lever) counter-acting the weight presumably
- so allowing this (the pulley rope) to go up and then this to fall down (knife)
- pushing on this (the egg) perhaps pushing it through so it falls there (the pan).

Figure 3-3: **The transcript of an explanation of the egg-cracker:** The text in parenthesis identifies the object being referenced

3.5.2 Experimental results

The first important observation made from these descriptions is that they do not include explicit references to the component parameters. There is no mention, qualitative or quantitative, of spring constants or dimensions. A second observation is that the language is abbreviated and makes use of sketched gestures and indications instead of verbal accounts of the same information. A third, and final observation, is that the order in which the device is described corresponds with the order that the action occurs in the device.

All of these observations have been exploited in ASSISTANCE to ensure that the descriptions that it understands are as close to the descriptions that designers actually give.

The complete set of transcripts can be found in appendix A.

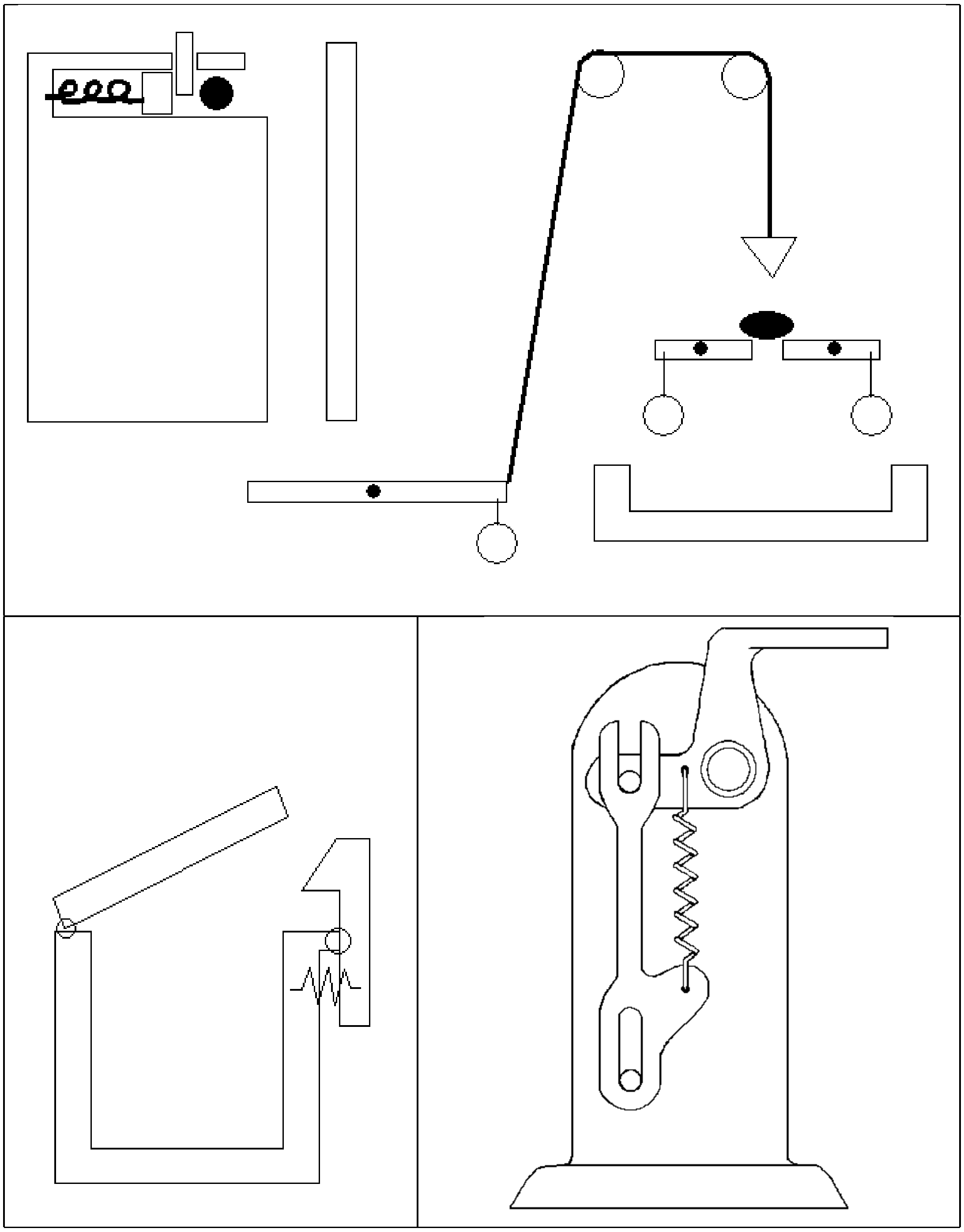


Figure 3-4: **The diagrams used in the survey:** Top: The egg-cracker. Lower left: A latching box. Lower right: A self resetting lever.

Chapter 4

The implementation of ASSISTANCE

The basic structure of the implementation of ASSISTANCE is shown in Figure 4-1. The designer begins by using ASSIST to sketch the device. After ASSIST parses the sketch, it is converted into a format understandable by ASSISTANCE. ASSISTANCE's rule system then uses the designer's description of the device to generate the information required to build a model of its behavior. After the device has been completely described, a separate module synthesizes the inferences made by the rule system to generate a causal structure for the device being described. The designer can then verbally query ASSISTANCE about the causal connections in the model to confirm that the model is accurate.

4.1 ASSISTANCE uses sketch and speech based media

As discussed earlier, ASSISTANCE makes use of input media that are commonly used by designers in their interactions with their colleagues. In our current implementation the designer can draw on one of two surfaces. The first is a digitizing sketchpad with a built-in LCD display; the second is a whiteboard with digitizing hardware from

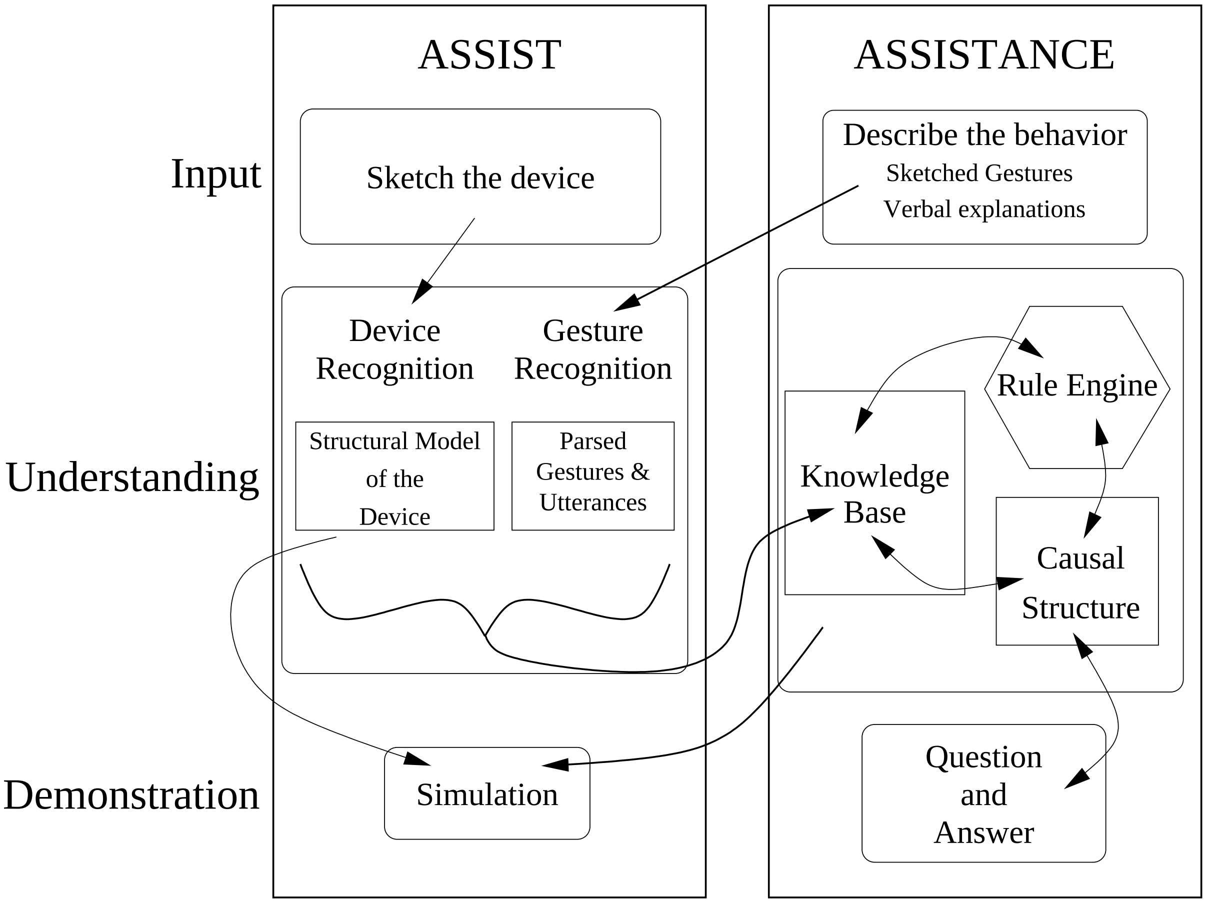


Figure 4-1: **An overview of the modules of Assistance:**

Virtual Ink that provides a way to turn an ordinary whiteboard into a computer input device. A device called a Mimio, attached to the whiteboard records the strokes made on the whiteboard and sends them to a computer. Alternatively, it is possible to project a computer display on the whiteboard and use the Mimio mouse, an inkless whiteboard marker, as the mouse input to the computer. Both the sketchpad and the whiteboard interfaces combine the natural interface of drawing with the ability to transparently record the designer's sketch.

The verbal input is collected by off-the-shelf speech recognition software from IBM, using a lapel microphone or headset.

4.2 ASSIST is used to sketch the device

The designer begins the description process by sketching the device with ASSIST (A Shrewd Sketch Interpretation and Simulation Tool)(Alvarado, 2000; Weisman, 1999; Muzumdar, 1999). The purpose of ASSIST is to parse freehand sketches of mechanical devices into a CAD model that can be simulated and analyzed. It assists designers in the early stages of design by providing the naturalness of a sketching environment with the ability to get immediate feedback from the simulations.

ASSIST uses its knowledge of mechanical devices to choose from multiple ambiguous interpretations of sketches. For example, consider drawing a simple truck like the one in Figure 4-2. ASSIST can determine that the larger circle, representing the wheel, is a body and the smaller one, representing the axle, is a pin joint. It does this by using its knowledge of drawing conventions of mechanical devices to determine that a small circle inside a body is usually a pin joint. This frees the designer from the distracting task of choosing part templates from a menu and placing them into the model.

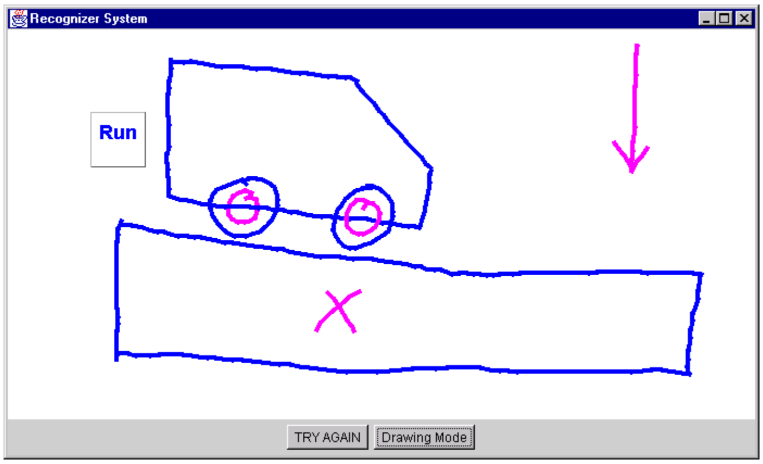


Figure 4-2: **Assist performs semantic disambiguation:** The wheel and the pin joint are not confused even though they have the same shape. ASSIST indicates its understanding by drawing constraints in pink and bodies in blue.

The context knowledge used by ASSIST is represented in modules that contain sets of recognizers that can be activated or deactivated as a group. Each active recognizer attempts to fit the sketched strokes into a possible interpretation. When there is more than one possible interpretation for a stroke, it ranks the interpretations according to a set of heuristics about drawing conventions and knowledge about mechanical objects. The recognizers in the mechanical device module recognize solid bodies, springs, dampers, rods, pin joints, pulleys, and anchors.

4.3 A structural model is constructed from the sketch

Once the device has been sketched using Assist, the designer verbally tells the system to begin interpreting the speech and sketched gestures as annotations and not as part of the device sketch. The need to explicitly switch from sketching mode to annotation mode is an artifact of the current implementation and will hopefully be fixed in future implementations. Ideally, designers would be able to switch back and forth between sketching without explicitly notifying the computer. Entering the description mode causes the system to convert the model generated by Assist into a model more suited for the reasoning performed in Assistance. The structural representation used by Assistance has four general classes of facts:

1. **Physical bodies:** The first class of facts represents physical bodies. Assistance generates a fact in its knowledge base for each body in the model. These facts assign an automatically generated English name to the object such as body 1 and encode geometric information about the body such as its size and location.

2. **Physical constraints:** The second class of facts represents physical constraints. All of the mechanical components that apply constraints on the motion of the bodies are in this class. This includes springs, dampers, rods, pin joints, pulleys, and anchors. Constraint objects also have English names and geometric representations.

3. **Attachments:** The third class of facts represents attachments that provide the topological structure of the device. The attachment facts indicate which con-

straint components are attached to which bodies. This information is extracted from the ASSIST model.

4. **Common device types:** The fourth class of facts indicates compound components that are made up of more basic components. For example, levers cannot be recognized from their shape alone; levers are generally oblong shaped objects that are limited by one of the constraint components to have only one degree of rotational freedom.

4.4 The description is translated into a behavioral model of the device

In this section we describe the process of interpreting the gestures and utterances. The inputs are recognized in ASSIST and interpreted by a rule system that encodes ASSISTANCE's knowledge of gestures, utterances, and ways to infer the causal structure of the behavior of the device from these annotations.

4.4.1 Gesture recognition

The first part of the description interpretation process is the recognition of the gestures and utterances. There are a small number of sketched gestures that designers commonly use, including arrows and pointing gestures. These can be easily recognized with ASSIST's recognizers. When ASSIST recognizes an annotation gesture it passes the gesture to ASSISTANCE, where it is interpreted in the context of the current device explanation.

4.4.2 Utterance parsing

Before understanding the verbal utterances they must be recognized and parsed. Speech recognition and parsing is performed by off-the-shelf speech recognition software that parses utterances against a grammar containing phrases commonly used in device descriptions. The grammar uses tags to annotate words and subphrases with labels that provide structural and semantic clues about the recognized utterance. Additionally, the exact text that corresponds to the tag can be extracted if Assistance needs more precise information. Figure 4-3 shows an example of the associations created by parsing a simple sentence against a fragment from a tagged grammar.

⟨SIMPLE-PHRASE⟩	=	⟨NP⟩ {subject} ⟨VP⟩ {simple-sentence}
⟨VP⟩	=	(⟨V-none⟩ | ⟨V-np⟩) {verb-phrase}
⟨NP⟩	=	the ⟨N⟩
⟨V-none⟩	=	(moves | falls | slides) {moves}
⟨V-np⟩	=	(pushes | propels) {propels} ⟨NP⟩ {direct-object}
⟨N⟩	=	(block | ball) {noun}

Parsing: "The block pushes the ball"

Tag	*Matched text*
simple-sentence	"the block pushes the ball"
subject	"the block"
noun	"block"
verb-phrase	"pushes the ball"
propels	"pushes"
direct-object	"the ball"
noun	"ball"

Figure 4-3: **A sample grammar:** The top box shows a simple tagged grammar, with tags indicated in braces; the second shows the mappings from tags to the text they matched.

An utterance is parsed against the grammar into a hierarchical structure that mirrors the parse tree, but includes only the tagged nodes. The tags help map from words to concepts and thus simplify the interpretation of the sentence. For example, the verbs "propels" and "pushes" both map to the *propels* tag since they have the same meaning. However, the parse tree also contains the exact word choice so more detailed information can always be retrieved if required.

As mentioned previously, the system does not attempt to fully understand unconstrained natural language, focusing instead on a select subset of common sentence structures used in device descriptions.

4.4.3 The rule system

The rule system encodes and implements the inferences that ASSISTANCE uses to generate the behavioral model from a device description.

The rule system is a simple forward chainer. It uses a justification based truth maintenance system to record the inferences generated by the rules and to record the reasons for the inferences (Forbus and de Kleer, 1993). The truth maintenance system also simplifies the retraction of statements that were misunderstood by the speech recognition system and helps with debugging and displaying explanations for the inferences.

The primary role of the rule system is to translate the description annotations into a causal model of the device. In addition, it indicates several conceptual components that are made from several parts, such as levers. It is not involved in the recognition of gestures from the sketch data or the parsing of the verbal utterances, which is performed by ASSIST. This decomposition maximizes the benefits of a rule system by constraining its tasks to drawing logical, local inferences and not trying to use it as the central control structure of the system.

4.4.4 The rules

The rules encode several broad classes of knowledge necessary to understand device descriptions. These categories include information that tell ASSISTANCE how to:

- interpret English phrases
- interpret sketched gestures
- combine language and gestures
- employ conventions prevalent in device behavior descriptions
- infer causal relationships between events

There is often confusion about the use of visual information in discussions of diagrammatic reasoning systems. Part of this confusion is the lack of distinction between topological and spatial information. In this discussion, spatial information is anything that is represented as a set of coordinate points, while topological information refers to the symbolic interconnections of the objects represented by the spatial representations. Topological information includes information about which components are connected to other components and information about the topology of annotations, such as which body an arrow originates from.

Rules that interpret natural language phrases

There are two primary types of linguistic rules. The first type translates sentences into the events they represent, while the second type identifies phrases with multiple clauses and breaks them down so they can be more easily handled by other rules.

Sentence interpretation rules convert single clause sentences into the events that they describe. The tags help categorize the sentences, allowing similar phrases to be parsed by the same rules. For example, the two phrases "body one moves" and "body one falls" are both tagged as motions and can thus be processed uniformly. If necessary, more specific rules can use the actual word rather than its semantic category.

The system is also capable of handling multiclause phrases that describe more than one event like, "when someone moves body one the spring releases." This type of utterance is handled by rules that break the two clauses into independent utterances. The separate clauses will then trigger rules that will determine the events represented by each clause. A subsequent rule then understands the relationship represented by the complex statement relating the two clauses and applies it to the newly generated event facts.

As an example of both types of linguistic rules, consider the following sentence, "when someone moves body 3 the spring releases." First, a rule splits the composite sentence and asserts the two clauses "someone moves body 3" and "the spring releases" as new sentences. Other rules translate these sentences into a motion event and a spring event. Finally, a rule asserts a causal link between these two events.

Rules that represent sketched gestures

There are similar rules for converting gestures to events. However, due to the more limited range of gestures these are less complex. The most common gesture is the arrow. Rules recognize arrows and associate a motion event with the body that is located at or near the tail end of the arrow. The fact that the arrow represents the path that the motion follows is also recorded. This is an example of a combination of topological knowledge about the connection between the arrow and the body it refers to and spatial knowledge about the coordinates of the path that the motion follows.

Rules that combine language and gestures

Another set of rules combines information from verbal and sketched annotations. There are two primary types: rules that combine the two types of information directly, and rules that link multiple representations of the same event.

One rule that combines the two types of information directly is the one used to recognize references to the length between two points of interest, as in the phrase "the distance from here to here..." and pointing at two different points on the sketch. This is processed by combining two temporally proximate pointing gestures and an utterance, and generating the associated distance observation.

Another way in which linguistic and sketched information are combined is by merging different representations for the same event. ASSISTANCE does not have rules that recognize events from complicated combinations of annotations from multiple input sources. Instead, ASSISTANCE generates an independent event fact for each annotation and then merges them together. The merging is done according to the compatibility between the described events and does not explicitly depend on the input sources. In the egg cracker (see Figure 3-1) the motion of the block is described three times: in the utterance about the spring releasing, in the utterance about the block pushing the ball, and with the arrow indicating its motion.

The event merging rules look for events of the same type that refer to the same components and that came from annotations that were made in the same time frame.

The time range is determined by the order the annotations where made in and not absolute times between the annotations. This metric was chosen because the absolute times between annotations varies widely. This method of generating event representations combines the semantic content gathered from two or more input types into a single canonical event.

Rules that take advantage of conventions used in device descriptions

The system is also capable of identifying common conventions used in device sketches. For example, it recognizes that oblong bodies with a single degree of rotational freedom are often considered to be levers. This information is useful because there are certain behaviors and properties of levers that the system can then infer.

Another common convention that the system makes use of is that design sketches often show the device in equilibrium. Combining this heuristic and the above knowledge about levers allows us to observe that a lever with opposing forces being applied to it is intended to be balanced.

Rules that infer causal relationships between events

The causal links that make up the behavioral model are generated by the rule system and stored in the knowledge base. There are several classes of rules that assert causality: linguistic, spatial, and topological.

The linguistic rules extract causal clues from phrases and language constructs that imply it. For example, the rule that understands the utterance "body 2 pushes body 5" asserts that the motion of body 2 causes the motion of body 5. There are also utterances that explicitly mention causal relationships such as the statement, "if body 3 moves then the spring releases."

Another class of causal rules are spatially based. For example, it is possible to infer that one body propels another body by examining the paths followed by each of the bodies. If the first body's trajectory leads to the second body's location, there is a good chance the the second body's motion is caused by the first body. This rule makes use of the size of the bodies and the path traced by the arrow. For example, this rule is used in the egg-cracker example (Figure 3-1) to infer that the motions of the two levers holding the egg are caused by the motion of the egg on its trajectory into the pan.

The process of inferring causality from topological information is similar to the process of inferring it from spatial information, but instead of using geometric properties, it uses the physical connections between the components. For example, when a body attached to one end of a pulley moves, the body attached to the other end also moves.

4.4.5 Multiple event representations are canonicalized

At the completion of the description, ASSISTANCE generates a more complete causal model for the device in which each event has a single canonical representation. As mentioned previously, there are often several representations for the same event. In order to generate a unified causal structure, these sets of equivalent events must be reduced to a single, canonical, event. Identifying the sets is a straightforward process because rules have already identified pairs of equivalent events. ASSISTANCE transitively gathers these pairs together.

The generation of the canonical representation is done by unifying the properties of the individual events into a single event. Different representations often contain different information about an event. For example, a description that contains an arrow and the phrase "body 5 falls" will have at least two event representations. The representation generated from the arrow provides the spatial information about the trajectory of the motion whereas the representation from the phrase provides a descriptive verb that can be used to describe the motion more specifically than "moves." The rules that assert that the two event representations are of the same physical event ensure that they do not contain conflicting pieces of information, such as motion in two different directions at once.

4.5 ASSISTANCE can demonstrate its understanding

ASSISTANCE provides several mechanisms that allow the designer to verify that the model is accurate. One mechanism is direct query, in which the designer can ask the system what role a part plays or what causes a certain event. The system constructs a response that describes a small portion of the behavioral model surrounding the events that involve the object in question. An example of this interaction is shown in detail in Figure 3-2.

The second method is by making simple parameter adjustments in the simulation model generated by ASSIST. Because ASSIST does not have a method to determine the parameters of a device's components it chooses default values for them that are not context dependent. For example, all of the springs in an ASSIST model are in their neutral position. ASSISTANCE uses its behavioral knowledge of the spring and its attached bodies to infer the qualitative state of the spring. The rest length of the spring is then proportionally adjusted from its depicted length to match this qualitative state. The adjustments are asserted by the rule system and are applied to the model when ASSISTANCE is asked to simulate the model.

Chapter 5

Related work

ASSISTANCE draws on research in a variety of fields ranging from human computer interaction and multimodal interface design to qualitative reasoning and knowledge representation. This chapter describes the relationship of ASSISTANCE to these other fields and their contributions to ASSISTANCE.

5.1 Design rationale capture

There are many interpretations of what design rationale is and what should be included in a comprehensive design rationale system. A good overview of the various approaches has been collected by Moran and Carroll (1996). One of the primary tradeoffs against which design rationale capture systems can be measured is the amount of effort required to capture the rationale verses the payoff that the rationale provides. On one end of the spectrum are systems that record verbatim parts of the design process, such as video or audio tape of design meetings. These systems have a minimal capture cost but can be hard to access due to the unstructured nature of the stored information. On the other end of the spectrum are systems that require the explicit encoding of the issues, possible solutions, and arguments for and against each solution (e.g. Conklin and Yakemovic, 1996). These systems have rich representations from which to extract information about the design process but have a much higher collection cost.

One approach towards design rationale capture advocates collecting information that allows for the answering of questions instead of attempting to anticipate the questions in advance and collecting answers to them (Gruber and Russell, 1996). Gruber and Russell have termed this approach generative design rationale. This approach fits nicely with ASSISTANCE because ASSISTANCE gathers information about the behavior of the device, which might provide a basis upon which to construct tools to answer questions about the reasons for why such a design was chosen.

Another piece of work that fits the definition of a generative model is the work of Myers *et al.* (1999). The goal of their work is to reduce the cost of gathering the rationale by directly analyzing a series of design changes and simulation results. Information about which parameters are being modified and the qualitative results of these changes is used to infer the goal of the changes and to infer which subcomponents are being analyzed. This attempts to balance the tradeoffs between capture cost and ease of use. While this approach is consistent with the goals of ASSISTANCE in that it is easy to capture rationale, the technique is only plausible in the latter stages of design when the overall behavior has been determined and the parameters must be properly tuned. This difference in focus suggests that the two systems are highly complimentary. ASSISTANCE could be used to describe the desired behavior. Then the system written by Myers *et al.* could infer the relationships between the parameters being adjusted and the desired behavior.

5.2 Explanation understanding

There has also been work done on understanding explanations (Borchardt, 1993). Borchardt's system used textual utterances to describe the device behavior and reconstruct the causal relationships involved in its behavior. The central insight of Borchardt's work is to base the representation around changes instead of on the states that the system passes through. While ASSISTANCE does not explicitly use the representations proposed by Borchardt, it does make use of this insight in its emphasis on behavior and causal structure.

Building on the explanation understanding task, ASSISTANCE added the extra dimension of sketched input. Borchardt's system did not directly make use of visual information, which is important in building a comprehensive model of a device. By interpreting sketches ASSISTANCE is able to directly analyze the spatial properties of the device where Borchardt's system is forced to reason about them symbolically.

5.3 Inferring behavior from a model

Another approach to the device understanding problem is to infer the device's behavior directly from the sketch without the behavioral description. This is similar to the approaches followed by both Brand (1997) and Dar *et al.* (1999). Their systems attempt to interpret the behavior of devices from images or sequences of images of a device in action. Similar work has been done to understand diagrams of devices (Narayanan *et al.*, 1995). ASSISTANCE could be integrated with any of these system to fill in any gaps that remain in the models generated by these other systems.

Chapter 6

Future work

ASSISTANCE is an important step in understanding the device description task. However, the description understanding task is just one step in the full device understanding task, which determines the behavior of a device from a sketch and some contextual information. This chapter describes some possible future directions for ASSISTANCE and the explanation understanding task and some proposed directions for the more general device understanding task.

6.1 Improvements to ASSISTANCE

There are a number of improvements that could be made to ASSISTANCE that build on the current architecture and representations. The primary changes are based around improving the naturalness of the communication with the computer system, including converting this interaction into a dialog.

6.1.1 Expand the language processing capabilities

There are several ways that the existing language processing capabilities of the system could be improved. ASSISTANCE would be much easier to use if it had a much larger vocabulary and more behavioral terms. Currently, the designer must be aware of which phrases and words the system understands and must tailor the explanation to those terms. As the language processing capabilities improve this will become less of a concern.

The language handling capabilities could also be improved if the system could understand deictic references, such as, “this body slides to here”. Deictic references are an obvious way of interacting with a multimodal system such as ASSISTANCE. To do this effectively the event canonicalization procedures should be incorporated more directly into the reasoning module instead of being in the synthesis stage. However, this should not require substantial changes to the structure of the system other than the addition of the rules required to unify the reference with the diagram.

Another way in which the language capability can be improved is by gathering more behavioral clues from the names that the designer uses to refer to the device components. The names that designers use to refer to components often convey a considerable amount of information about their behavior. For example, referring to a body as a stopper implies that it is capable of preventing the motion of one or more nearby objects. Other examples include latch, piston, and counterweight.

6.1.2 Make the interaction into an interactive dialog

One important difference between describing a device to ASSISTANCE and to another designer is that the other designer can provide feedback about what is understood and what is unclear. Adding this capability to ASSISTANCE would help improve the accuracy of the models generated by ASSISTANCE, because it could ask the designer for further explanations when it is confused. The system could also ask questions to confirm its current representations, which may help the designer understand the source of the confusion.

In order to control this type of dialog and know when to interrupt the designer, the system must be able to assess the accuracy of its current beliefs. There are at least two possible approaches, an explicit numeric certainty value or by having rules that recognize conditions that are known to have ambiguous interpretations. As one example, ASSISTANCE might ask the designer if one event actually causes another event.

6.1.3 Enable the designer to manipulate the sketch directly

When freed from the static nature of standard sketching media, such as paper and whiteboard, there are new ways in which explanations can be conveyed. The best example of this occurs when an annotation describes the motion of an object the sketch could update to reflect the new location of the object. While this does not need to be an accurate simulation, it does require some analysis of the physical constraints on a body. For example, while the motion of an unconstrained body is easy to track by following a sketched arrow indicating its motion, the motion of a body that is jointed to another body requires a combination of rotation about the joint and translations by both of the bodies. One possible solution that avoids some of this complexity is to allow designers to redraw the components in their new configuration. However, while redrawing one or two simple components is reasonable, redrawing complex bodies and constraints may not be desirable.

6.2 Develop design tools that use ASSISTANCE models

One of the goals of Assistance is to generate an accurate causal behavior model of the device. This information could be used in automated design tools that assist the designer in the adjustment of design parameters. The Assistance model would describe the desired behavior while the other system iteratively adjusted the parameters and compared the simulated behavior to the desired behavior. This is not a trivial task, but there is prior research in the area of functional specification and reasoning that might provide guidance. For example see the work of Vescovi *et al.* (1993).

One project that could benefit from the models generated by Assistance is SketchIt (Stahovich *et al.*, 1996). In SketchIt the designer describes the geometry of the device with a stylized sketch and the behavior with a state transition diagram. From these representations, SketchIt is able to understand the operation of the device and suggest alternative designs with the same qualitative behavior as the original. A natural extension would be to use ASSISTANCE to specify the behavior instead of the state transition diagram.

6.3 Use ASSISTANCE to aide design rationale capture

One of the motivations behind ASSISTANCE is its application to design rationale. Most design rationale systems suffer from the same root problem: they are more trouble to use than they are worth. This is often because the person in charge of recording the rationale is not the person who will need to refer to the rationale in the future. Also the process of encoding the information is often time consuming and disconnected from the actual design process.

ASSISTANCE begins to address these issues by reducing the cost of generating the device model and by generating representations that can be used to support the design process instead of just recording it. With a series of models of the behavior of the device it may be possible to infer some of the reasons for changes that are made.

Chapter 7

Contributions

One of the goals of the field of human computer interaction is to make interacting with a computer as natural as interacting with a person. In the last several years computers have been given the ability to process many new kinds of input media, such as spoken natural language and hand drawn sketches. These new input media have reduced the disparity between interactions between people and the interactions between a person and a computer. However, the full potential of these tools has not yet been realized. A large part of this is due to the fact that the new input media are often employed as novel features that reimplement old interface paradigms. To fully realize the potential of new input media, a new paradigm must be embraced, and the computer must be able to understand the meaning of what the user is conveying instead of forcing the user to speak the same commands that used to be typed.

Assistance demonstrates that such an interface is possible by understanding naturally conveyed explanations of mechanical devices. 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 dissolves the core of traditional CAD interfaces and brings the computer into the designer's world. During conceptual design, designers talk about behaviors and not the parameters that lead to them, therefore Assistance focuses on understanding behavioral explanations instead of providing novel ways of specifying parameters.

ASSISTANCE's understanding of the device description is evident in its ability to answer questions about the causal chain of events that make up the behavior of the device and by the fact that the behavior is formally represented.

ASSISTANCE has taken an important step in lowering the barrier between human and computer interactions.

Appendix A

Transcripts from the study

This appendix contains the transcripts of four subjects who were asked to describe three devices each. For a description of the study see Section 3.5. From these transcripts a number of the common sentence structures and behaviors were identified. These were latter implemented in the rules and reasoning of Assistance.

A.1 The Rube Goldberg egg-cracker

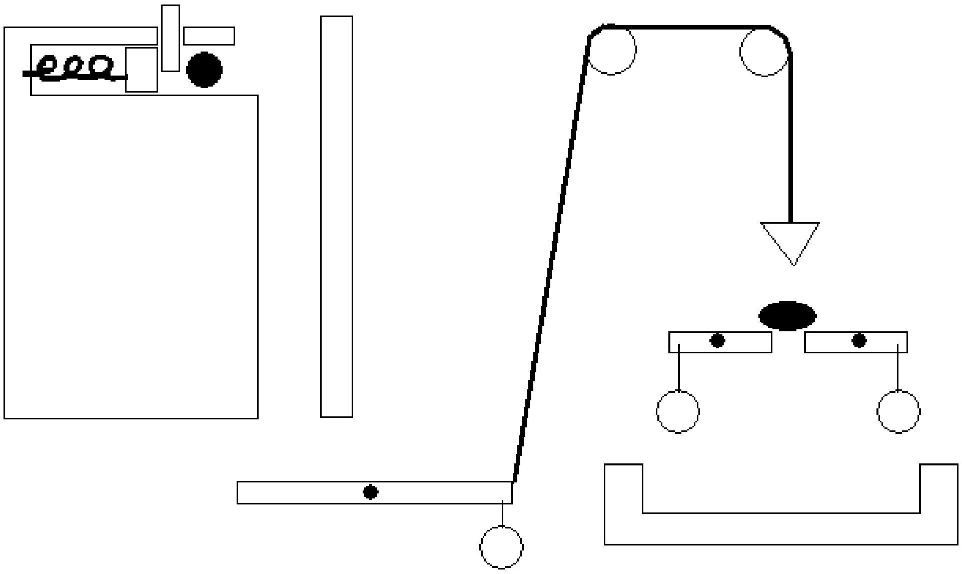


Figure A-1: **The egg-cracker:**

A.1.1 Subject A

- So this is a Rube Goldberg style machine that I designed to crack an egg and it works in a serial fashion. The action starts at the spring which pushes on the block thats prevented from pushing the ball until I pull the stopper out.
- When I pull the stopper out the spring extends pushing the block which hits the ball. The ball extends out to the right and hits the wall following that arrow.
- Then it will slide down the wall landing on one end, the left end of the lever. There the force of gravity will pull it down and also its momentum will push it down pushing the right end of the lever up which will let rope one extend out in the direction of the arrow.
- It will go around the pulleys, rope one that is, and extend down in the direction that I drew.
- After the pulleys at the end of the rope is attached a knife which will crack the egg and once the egg has been cracked it will be allowed to fall through and hit the pan due to the arrangement of weight two and weight three and the associated levers.

A.1.2 Subject B

- when this is pulled up and removed then the spring releases
- pushing this ball here which may or may not hit this
- falls down here and pushes down on this counteracting the weight presumably
- so allowing this to go up and then this to fall down
- pushing on this
- perhaps pushing it through so it falls there.

A.1.3 Subject C

- by pulling up on this pin causes this spring which is in tension to hit this ball
- which then is guided down this channel
- which then hits one end of a lever causing this end to move upwards
- which causes this cable here to move down and this knife would break this object.

A.1.4 Subject D

- so the first thing to happen is once this plunger is pulled up it releases the block with the coiled spring that pushes the ball
- the ball falls to the bottom of the channel and hits this pivoting rod
- the rod pivots up and the cord attached to it drops the weight at the other end to push the item in the middle of two other pivoting rods which then drops the item to the bottom
- the circular weights restore all the rods to the upright positions

A.2 The latching box

A.2.1 Subject A

- this is a box with a latching top
- you can put something in the box and then you can push the lid down and it will latch due to the lever
- once you have it latched you can open the lid if you want by pushing on the lever at the bottom location where I drew the arrow against the spring that will cause the latch to move away in the direction that I drew

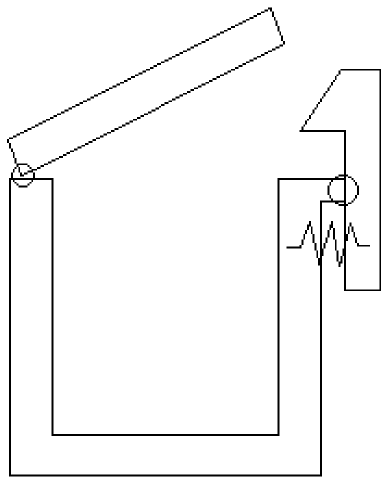


Figure A-2: **A latching box:**

- so the lid can be moved up

A.2.2 Subject B

- when the lid is pushed down it pushes this back so it swings back
- presumably enough such that it could go underneath
- the spring pushes it back and locks it
- you would have to push on this thing again to release it
- and then have something inside push it up

A.2.3 Subject C

- when the top or lid of this container is put into its closed position what will happen is it will hit this lever at this point
- and because of the shape of the lever here which is in a slant an angle by continually pressing down on the lid the lever will rotate in a clockwise direction until the lid passes the lip of this lever

- at which point the lever will then rotate in a counter clockwise direction over the top of the lid thereby locking in place
- in order to open the container one would then pull back on this lever making it rotate in a clockwise direction until the lid could pass beyond this point
- whereby it could be opened and then the lever can be released
- in which case it would rotate back into its original position
- we have a spring which is in compression which is what is holding that lever in place in its locked position.

A.2.4 Subject D

- if you push the lid down the lid pushes against the angled latch which can pivot along the circular axis on the right
- and the spring on the other side of the latch is compressed by the downward motion on the lid which then allows the lid to pass underneath the latch
- then the spring pushes the latch back out which pushes the latch over the lid and locks the lid down
- to release it you would apply force to the spring end of the latch and pull the lid up

A.3 The self resetting lever

A.3.1 Subject A

- this is a self resetting lever mechanism
- the interface for this device is where I'm going to draw these arrows here and here

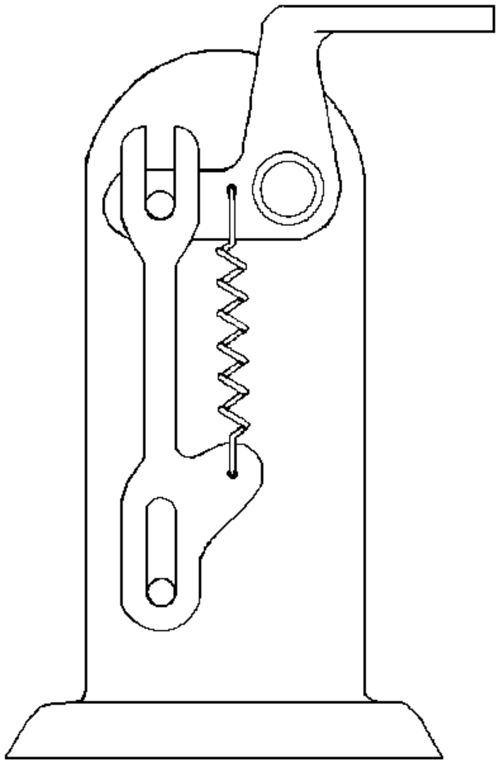


Figure A-3: **A self resetting lever:**

- and the idea is that I can apply a force in the direction of either of these arrows to that lever and the lever will apply a reaction force to the applied force
- it accomplishes that by tensioning the spring
- and the link mechanism is used to tension the spring when I push in direction 1
- so if I push with a force in direction 1 that will push down at the location of the top pin which will push the link down which will stretch the spring by pulling that end downwards at connection 2 that is
- connection 1 will also move downwards but will move downwards less so than connection 2
- so the spring will be stretched opposing the applied force

- now if I push in the direction 2 then the pivot will pull on connection 1 of the spring
- extending the spring upwards thereby creating a reaction force to the way that I pulled the lever
- and in this case connection 2 doesn't move at all.

A.3.2 Subject B

- so if you push down on this lever this pin goes up in an arc around this
- so it goes like this which would force this thing to pivot around that because it can't go up any more so it would go this way
- if you push it down too much it will disengage all together and break and you won't be able to disengage it
- so once it is up this way this thing would be like this
- so the spring will be extended so it will pull the spring back
- and then return it down to this position
- I guess this extra gap is so it could swing evenly... I don't understand why
- if you lift it up it pushes this down this way and the whole thing slides down turning
- now when you release the spring is again elongated so it will retract the spring and pull it back up to that position
- so both pushing and pulling will restore there
- *I don't like dynamite going off near me.*[1]

[1]...as the old Polaroid building next door is demolished with a wrecking ball. Sorry, no real dynamite.

A.3.3 Subject C

- if one rotates the handle in a counterclockwise direction what will occur is this pin will press this link downwards
- and by doing that the spring will be lengthened.
- and one can see that would be the case because of the 2 radii from the joint of the handle to the pin which presses down the link is greater than from the joint of the handle to the insertion point of the spring
- so therefore the lever is being pushed down further than the spring is being pushed down so thereby the spring is lengthening
- when the handle is released the spring will then contract returning the lever and link to this original position
- if one were to rotate the handle in a clockwise direction the link would also rotate slightly in a clockwise direction by having this pin move in this slot
- and the spring once again would be lengthened
- because of this point here would be rotating and increasing in the upwards direction whereas where the spring connects to the link would be moving in a downwards position due to the rotation of the link
- so the spring now is in tension and when the handle is released it will once again, the spring will cause it to return to this initial position.

A.3.4 Subject D

- If a force is applied to push the handle upwards, it pushes the secondary mechanism downward
- and once the force is removed the spring pulls the secondary mechanism back upwards

- which in turn pushes the lever back to its initial position
- similarly if force is applied downward the handle twists upwards allowing the spring to pull ...
- in this case the handle would just move on up and the spring alone would restore the handle to its original position.

Appendix B

Printing labels on sketches

The positioning of textual labels on diagrams such that they are not overlapping one another and not overlapping the boundaries of other elements of the diagram is a hard problem. If the labels are offset so that they do not overlap it often becomes unclear which element of the diagram they refer to. For clarity, the figures in this thesis use hand positioned labels instead of the automatically generated ones. A figure with the original label positions is given in Figure B-1 for reference.

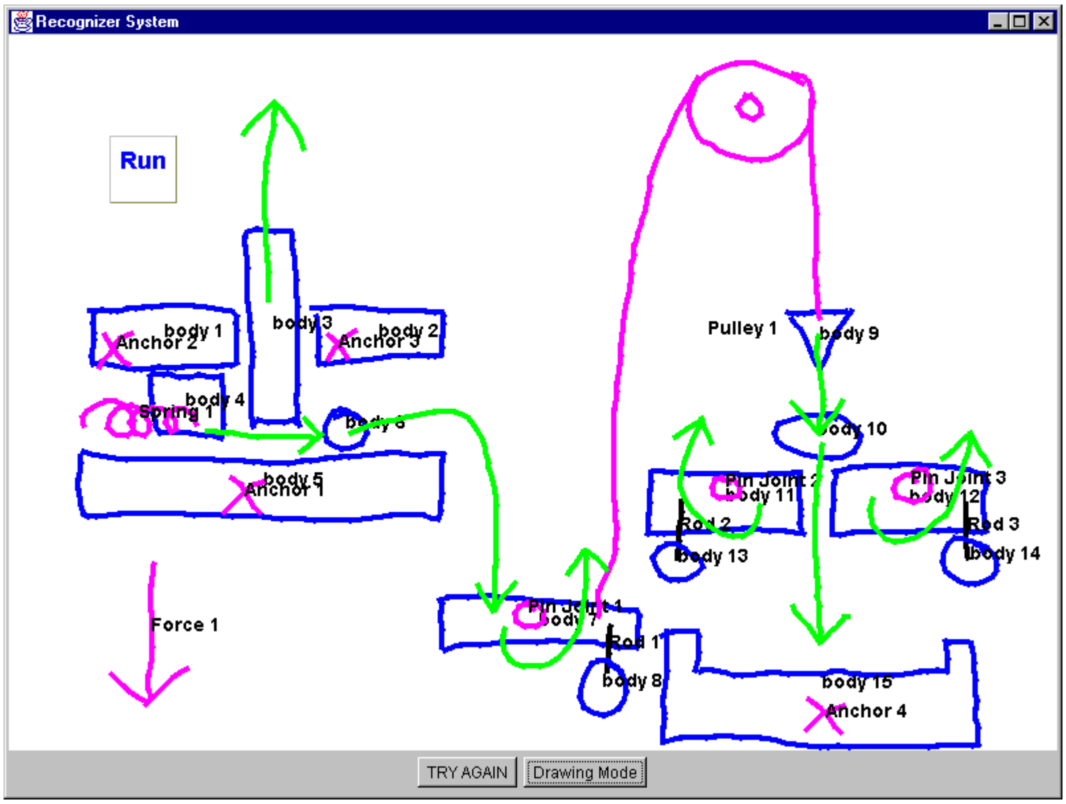


Figure B-1: **The original label positioning:**

Bibliography

ALVARADO, CHRISTINE. *A Natural Sketching Environment: Bringing the Computer into Early Stages of Mechanical Design.* Master's thesis, Massachusetts Institute of Technology, 2000.

BAYA, VINOD AND LEIFER, LARRY J. Understanding information management in conceptual design. In *Analysing Design Activity*, N. Cross, H. Christiaans, and K. Dorst, editors, pages 151–168. John Wiley and Sons, 1996.

BORCHARDT, GARY C. Causal reconstruction. Technical Report AIM-1403, Massachusetts Institute of Technology, February 1993.

BRAND, MATTHEW. Physics-based visual understanding. *Computer Vision and Image Understanding: CVIU*, **65**(2):192–205, February 1997.

BRANDA, EWAN E. *Drawing Inferences: Building Geometric Models With Hand-drawn Sketches.* Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, October 1989.

COHEN, P., JOHNSTON, M., MCGEE, D., OVIATT, S., PITTMAN, J., SMITH, I., CHEN, L., AND CLOW, J. QuickSet: Multimodal interaction for distributed applications. In *Proceedings of The Fifth ACM International Multimedia Conference (MULTIMEDIA '97)*, pages 31–40. ACM Press/Addison-Wesley, New York/Reading, November 1998.

CONKLIN, E. JEFFREY AND YAKEMOVIC, KC BURGESS. *A Process-Oriented Ap-*

proach to Design Rationale, chapter 14, pages 393–427. In Moran and Carroll (1996), 1996.

Dar, Tzachi, Joskowicz, Leo, and Rivlin, Ehud. Understanding mechanical motion: From images to behaviors. *Artificial Intelligence*, **112**:147–179, 1999.

de Kleer, Johan and Brown, John Seely. A qualitative physics based on confluences. *Artificial Intelligence*, **24**:7–83, 1984.

Forbus, Kenneth D. and de Kleer, Johan. *Building Problem Solvers.* MIT Press, Cambridge, Massachusetts, 1993.

Gruber, Thomas R. and Russell, Daniel M. *Generative Design Rationale: Beyond the Record and Replay Paradigm*, chapter 11, pages 323–349. In Moran and Carroll (1996), 1996.

Iwasaki, Yumi and Simon, Herbert A. Causality in device behavior. *Artificial Intelligence*, **29**:3–32, 1986.

Lesser, Victor R. and Erman, Lee D. A retrospective view of the hearsay-II architecture. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Raj Reddy, editor, pages 790–800. William Kaufmann, Cambridge, MA, August 1977.

Moran, Thomas P. and Carroll, John M., editors. Lawrence Erlbaum Associates, 1996.

Muzumdar, Manoj. *ICEMENDR: Intelligent Capture Environment for Mechanical Engineering Drawing.* Master's thesis, Massachusetts Institute of Technology, 1999.

Myers, Karen L., Zumel, Nina B., and Garcia, Pablo. Automated capture of rationale for the detailed design process. In *Proceedings of the Eleventh National Conference on Innovative Applications of Artificial Intelligence.* IAAI, AAAI Press, 1999.

Narayanan, N. Hari, Suwa, Masaki, and Motoda, Hiroshi. *Behavior Hypothesis from Schematic Diagrams*, chapter 15, pages 501–534. The MIT Press, Cambridge, Massachusetts, 1995.

Oviatt, Sharon. Ten myths of multimodal interaction. *Communications of the ACM*, **42**(11):74–81, November 1999.

Stahovich, Thomas F., Davis, Randall, and Shrobe, Howard. Generating multiple new designs from a sketch. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 1022–1030. AAAI Press / MIT Press, Menlo Park, August 1996.

Ullman, David G., Wood, Stephen, and Craig, David. The importance of drawing in the mechanical design process. *Computers and Graphics*, **14**(2):263–274, 1990.

Vescovi, Marcos, Iwasaki, Yumi, Fikes, Richard, and Chandrasekaran, B. CFRL: A language for specifying the causal functionality of engineered devices. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 626–633. AAAI Press, Menlo Park, CA, USA, July 1993.

Weisman, Luke. *A Foundation for Intelligent Multimodal Drawing and Sketching Programs.* Master's thesis, Massachusetts Institute of Technology, 1999.