Proceedings of DETC98
1998 ASME Design Engineering Technical Conference
September 13–16, 1998, Atlanta, GA

DETC98/DTM-5652

COMPUTING DESIGN RATIONALES BY INTERPRETING SIMULATIONS[*]

Anand Raghavan
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
anandr@andrew.cmu.edu

Thomas F. Stahovich[†]
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
stahov@andrew.cmu.edu

ABSTRACT

We describe an approach for automatically computing a class of design rationales. Our focus is computing the purposes of the geometric features on the parts of a device. We first simulate the device with the feature in question removed and compare this to a simulation of the nominal device. The differences in the simulations are indicative of the behaviors that the feature ultimately causes. We then use fundamental principles of mechanics to construct a causal explanation that links the feature to these behaviors. This explanation constitutes one of the rationales for the feature. We have implemented a program that can construct these kinds of causal explanations and have tested it on various examples.

KEYWORDS: Design Rationale Capture, Interpreting Simulations, Causal Reasoning, Qualitative Reasoning.

INTRODUCTION

The objective of our research is the creation of methodologies for capturing design rationales. A design rationale is an explanation for why a device is designed the way it is. These explanations are useful for a variety of purposes including resolving conflicts in collaborative design, performing product redesign without introducing undesired side effects, and modifying a design to improve manufacturability. Previous efforts to build design rationale management tools have frequently focused on providing an infrastructure for storing, indexing, and retrieving human generated rationales. By contrast, we focus on automatically computing a class of rationales. Thus our work complements this previous work, further shifting the burden of rationale management from the designer to the computer.

Human generated design rationales are often expressed in terms of geometric features, for example, “the *notch* on part X is intended to …” In fact, work by Knuffer and Ullman [Knuffer90] indicates that questions about the construction, purpose, and operation of features are among the questions most frequently asked by professional engineers during a redesign exercise. In the design speaking-aloud protocol studies they conducted, over 25% of the questions concerned features. Motivated by these observations, this project focuses on techniques for automatically computing the purposes of features on the parts of a device.

Our approach identifies purpose by interpreting simulations of the device. The focus is on interpreting kinematic and dynamic simulations to identify purely mechanical functions of a feature. We identify purpose by simulating how removing the feature alters the behavior of the device. The alteration is a clue to which of the device’s behaviors were produced by the feature. Our program works from this clue and uses fundamental principles of mechanics to establish a causal connection between the feature and its behaviors. This causal explanation is one of the rationales for the feature.

This approach is based on two assumptions: (1) if the designer bothered to create a geometric feature on a part, it most likely has some intended purpose, and (2) the simulations that the designer performs are intended to evaluate the “important” parts of the design, and thus many of the important design considerations are implicit in the simulations. One of the ad-

---

[*] Support for this project was provided by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-96-1-0750.

[†] Please address correspondence to this author.

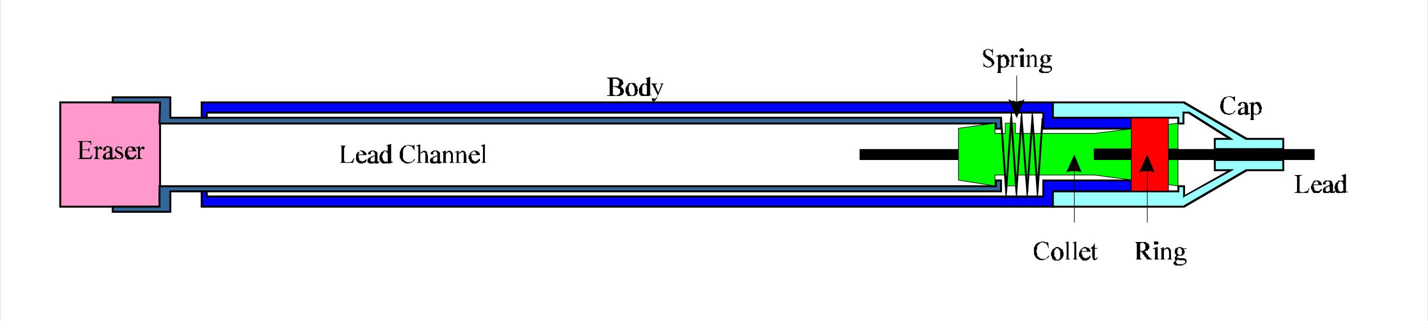

Figure 1: Mechanical Pencil.

vantages of this approach is that the complexity of the physics is taken care of by numerical simulators, rather than the reasoning engine, as is the case with some of the other approaches to reasoning about physical systems, such as qualitative physics.

The domain of kinematic and dynamic simulations was selected because it is rich enough to illustrate the challenges involved in inferring rationales from simulations. The methodologies developed, however, should generalize to other kinds of simulations, thus providing the basis for future computational design tools capable of capturing additional kinds of rationales.

Numerical simulations are becoming increasingly common in everyday engineering practice. This is occurring for a variety of reasons. First, software developers are now providing seamless integration between simulation tools (e.g., dynamic simulation and finite element analysis) and solid-modeling packages, thus making these tools much easier to use. Second, the cost of using these tools is decreasing because now even inexpensive personal computers are sufficiently powerful to run them. Thus, the raw materials (simulations and numerical analyses) needed by our design rationale capture tools will be increasingly more available.

Currently we have implemented only the causal analysis portion of our design rationale capture system, however, we have a model for how the completed system will likely be used. When a design is nearly complete, the designer will run our system to automatically generate rationales for all of the features on all of the parts. The system will present these rationales to the designer for further editing and augmentation. If our system is unable to find a rationale for a feature, it will prompt the designer for one. This will either focus the designer's attention on documenting a subtle part of the design or, if there really is no rationale, will indicate an opportunity to simplify the design. Our techniques will clearly not eliminate the need for human generated rationales, however, they will move a step closer to ensuring that the designer's time is used efficiently in documenting just those parts of the design that require subtle explanations.

EXAMPLE

We illustrate our approach with the help of the mechanical pencil shown in Figure 1. In this example, the goal is to explain the purpose of the taper on the collet blades (Figure 2). Figure 3 shows what happens when one presses and releases the eraser end of the pencil.[1] To identify the taper's purpose, we contrast this with a simulation of the behavior that results when the taper is removed (Figure 4). For the class of devices we consider, the important differences between two simulations are those between the final states. In the final state of the nominal simulation the lead has advanced a few millimeters, but in the final state of the modified simulation the lead has had no net displacement. Apparently, the purpose of the taper is to somehow make the lead advance. The question is how?

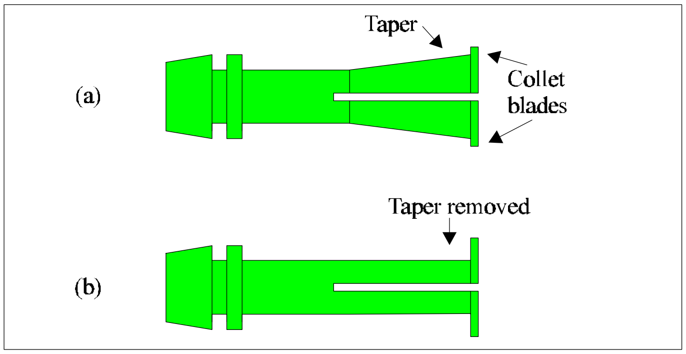

Figure 2: Collet blades (a) nominal geometry (b) geometry with taper removed.

To answer this, we look for the first point at which the lead's motion is qualitatively different in the two simulations. This occurs right at the beginning: the lead's velocity is initially zero in the nominal simulation while it is initially negative in the modified one. Next we look for the existence of forces which could explain this difference. In the nominal simulation the collet applies a friction force to the lead (f_{cl}), but there is no corresponding force in the modified simulation (Figure 5). Because this force is in the right direction to prevent the lead from moving, the next step is to look for a path by which the taper causes this force. If such a causal path can be found, we have identified the rationale for the taper. The

---

[1] Because of the way the simulator works, all of the velocities are computed *relative* to the eraser end of the pencil which thus appears stationary. (See the "Implementation" section.)

following is the causal explanation our program constructs (i.e., this text is direct output from the program):

The taper feature forms the surfaces that cause wedging and produce the force Ncr. The wedging forces Nrc and Ncr cause each other because wedging occurs between the interacting surfaces of the ring and the collet. The force Nrc causes the moment MNrc because the two form a force-moment pair. The moment MNrc causes the moment MNlc because it is the only moment that is opposite in direction to MNlc and the part they act on is in equilibrium. The moment MNlc causes the force Nlc because the two form a moment-force pair. The force Nlc causes the force Ncl because of Newton's third law. The force Ncl causes force fcl because normal forces cause friction forces. The force fcl causes the lead to move forward in the nominal simulation.

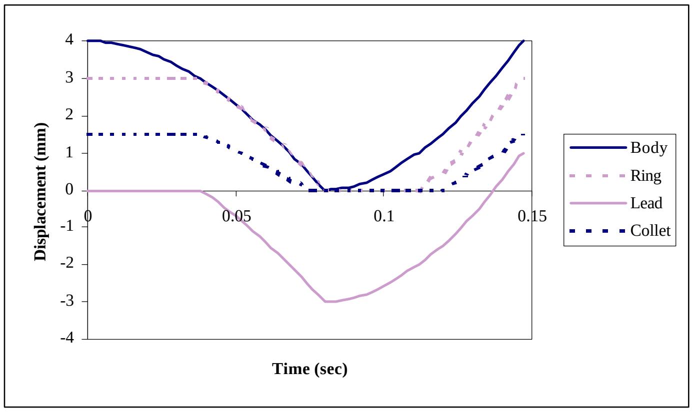

Figure 3 : Nominal Simulation[1] (collet displacement is in degrees)

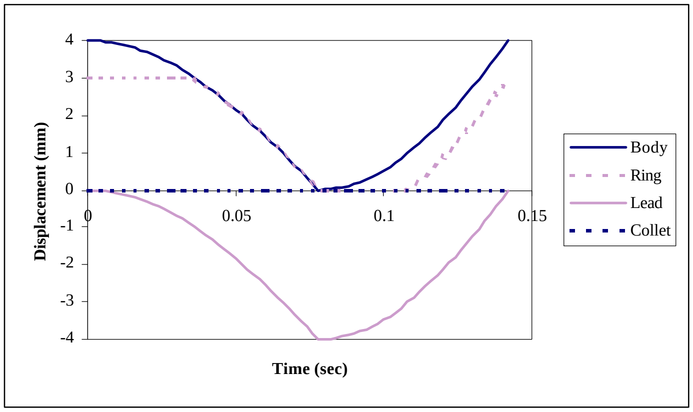

Figure 4 : Modified Simulation[1] (collet displacement is in degrees)

The forces used in this explanation are defined in Figure 5, but for convenience we can summarize the explanation as follows: The taper on the collet interacts with the ring to produce a wedging force (i.e., the ring is press-fit onto the collet). This force applies a moment to the collet which is balanced by the moment of a force that the lead applies to the collet. By Newton's third law, there must be an equal and opposite force that the collet applies to the lead, and that force gives rise to a friction force (f_{cl}) which eventually causes the lead to advance. Thus, our techniques for identifying the important differences in the simulations, combined with the fundamental principles of mechanics, have allowed our program to correctly identify that the purpose of the taper is to cause the collet to grip the lead so that it can be advanced.

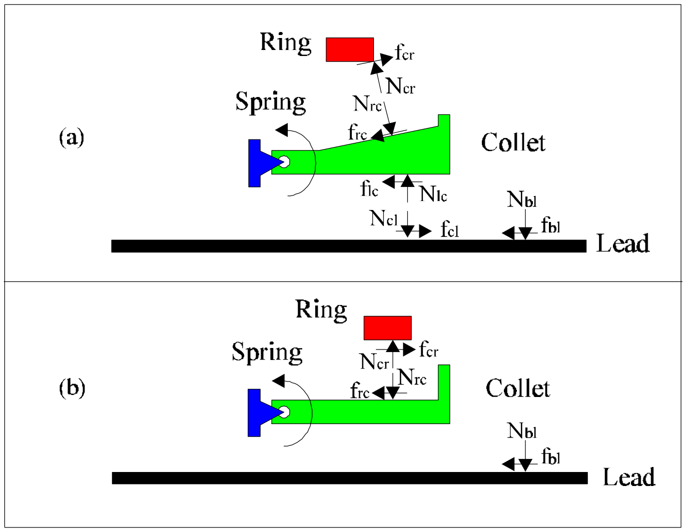

Figure 5 : Forces in the pencil (a) nominal simulation (b) modified simulation. The first subscript denotes the body that applies the force, the second denotes the body to which the force is applied. "c"=collet, "r"=ring, "b"=body, and "l"=lead. "N" denotes a normal or wedging force, "f" a friction force, and "M" the moment of a force. For example "Nrc" is the normal force of the ring on the collet, and "MNrc" is the moment of that force. (See Figure 8 for discussion of collet model.)

INTERPRETING SIMULATIONS

Simulations describe *what* happens but not *why*. Thus, a simulation does not directly indicate which of the device's many behaviors are caused by a given feature on a given part. To tease out the answer to that question we remove the feature from the part and compute a new simulation (with all other properties the same). The differences between the modified and nominal simulations indicate which behaviors are ultimately caused by the feature. However, the comparison still does not indicate *how* the feature causes those behaviors. A separate causal analysis is still necessary. This analysis is possible precisely because we have both ends of the causal path: the feature is at one end and the identified behaviors are at the other.

One of the challenges in implementing this approach is determining which of the differences between the modified and nominal simulations are significant. There are likely to be differences in force magnitudes, velocities, accelerations, etc., at every instant of time. Many of these differences are insignificant, such as those resulting from the small change in mass that occurs when the feature is removed. The unfortunate consequence is that direct numerical comparison of the traces of the state variables is not particularly informative. To provide a working definition for which differences are important, we focus on devices whose function is to change state – devices whose purpose is for the parts to start in one position and end up in another. For these devices, differences between the final states of the simulations are the significant differences. At the end of the nominal simulation of the pencil, for example, the lead has advanced a few millimeters, but at the end of the modified simulation, the lead has had no net displacement. This difference between the two end states is indicative of the purpose of the taper on the collet and must be explained causally.

The construction of the causal path relies on a simple but useful insight: for mechanical systems, force causes motion. Thus we can establish cause and effect by examining how the input forces (and motions) propagate through the device. We do this by applying the fundamental principles of mechanics such as the fact that normal forces cause friction forces, rather than the other way around.

The next two sections expand upon this discussion, describing in detail how we compare two simulations and how we construct causal explanations.

Analyzing Differences

In order to abstract away the insignificant differences in the traces of the state variables between the two simulations, we make qualitative comparisons. For each simulation we qualitatively describe the motion of each body in terms of its net displacement. The qualitative displacement is either positive, negative or zero. If there is a difference in the qualitative displacement of a body between the two simulations, the difference is an indication of a behavior of the removed feature and must be explained. If more than one body exhibits a difference in qualitative displacement, then there is more than one behavior to be explained.

If none of the bodies have different qualitative displacements, the program concludes that the feature has no kinematic or dynamic purpose. In this case, the program (when completely implemented) would prompt the designer for the rationale for the feature. There are three possible outcomes. The first possibility is that there really is no rationale, in which case the program has identified an opportunity to simplify the design by removing the feature. The second possibility is that there is a rationale, but it is a subtle one. In this case, the program would be prompting the designer for an explanation that only he or she knows. This kind of rationale is particularly valuable and the program would have performed a useful service by drawing the designer's attention to it. Our goal is for the program to document all of the obvious rationales and to focus the designer's attention on just the subtle parts of the design, which makes for efficient use of the designer's time. Furthermore, it is the subtle parts of the design that designers typically enjoy describing to others. The third possibility is that the purpose is related to physics in other domains such as heat transfer or fluid mechanics. Currently, when this third situation occurs, the designer will have to manually document the purpose of the feature, however our eventual goal is to extend our techniques to a variety of other physical domains.

Once we have found a part that has a different qualitative displacement in the two simulations, we must find the root cause of the difference. This root cause will occur at the first point the displacement-time plots differ. After this first difference in the plots, there may be a cascade of other differences, but those are likely to be consequences of the first difference. For example, once the path of a train is diverted by throwing a switch and changing the tracks, the entire subsequent path will be different.

Once again, we use qualitative comparisons to identify the first difference between the two displacement-time plots. In this case there are two reasons we make the comparisons qualitatively. First, as before, the values of state variables may differ in insignificant ways at every instant of time, so that direct numerical comparison is not meaningful. Second, the notion of time itself is a problem. As a result of removing the feature, motions may be either faster or slower in the modified simulation. Thus, similar events may not occur at the same time in the two simulations. Our approach to solving these problems, is to divide each displacement-time plot into segments of uniform motion: periods during which the velocity remains strictly positive, strictly negative, or zero. Our task then reduces to looking for the first segment that differs. In Figure 6, for example, the first two segments of the nominal simulation match those of the modified one, but the third segment differs. In this case, the first point of the third segment must be analyzed to establish a causal explanation for the purpose of the feature.

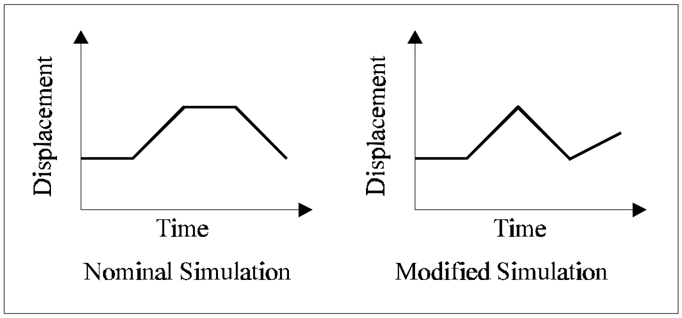

Figure 6 : These displacement-time plots differ starting at the beginning of the third segment.

Computing Causal Explanations

To construct a causal explanation for the purpose of a feature, we must causally explain the differences between the nominal and modified simulations. As the last section described, there is a very specific set of differences that must be explained. We need examine only those bodies whose qualitative displacements differ in the two simulations, and for each of those bodies (called anomalous bodies), we need analyze only the first instant of the first differing segment of the displacement-time plots. This section describes how we perform that analysis.

Just prior to the start of the differing segment, the velocity of the anomalous body is the same in both simulations; just after the velocities have different signs. This difference in velocities can occur only if the net force on the anomalous body differs at the start of the segment. We assume that this difference in net force is due to one or more forces (called anomalous forces) whose qualitative magnitudes are different in the two simulations.[2] By qualitative magnitude we mean the sign of the projection of the force onto the direction of motion [Stahovich97]. The qualitative magnitude is defined to be positive if the force has a component in the direction of motion, negative if the force has a component opposite the direction of motion, and zero if the force is perpendicular to the direction of motion. This force representation is particularly useful because the projection on the degree of freedom is the only component of the force that has any effect on the motion. This representation works equally well if the body rotates rather than translates, but in this case we reason about projections of the moments of the forces onto the axis of rotation.

Our task has now reduced to explaining the anomalous forces. In doing this, we always reason about why a force exists (i.e., why its magnitude is non-zero) rather than attempting the philosophically more difficult question of why it does not exist (i.e., why it has zero magnitude). If a force's qualitative magnitude is zero in the modified simulation and non-zero in the nominal simulation, we explain how the surfaces of the feature eventually cause the force to exist in the nominal simulation. If, on the other hand, the force's qualitative magnitude is zero in the nominal simulation and non-zero in the modified simulation, we explain how the surfaces created by removing the feature eventually cause the force to exist in the modified simulation. In the former case, the purpose of the feature is the same as the causal explanation for the existence of the force. In the latter case, the purpose is to prevent the causal path that explains the existence of the force. If the anomalous force is non-zero in both simulations (i.e., positive in one and negative in the other), then the feature has two purposes: one describes behaviors it was intended to produce, the other behaviors it was intended to prevent.

Reasoning About Forces

We use the fundamental principles of mechanics to construct the causal path that explains the existence of the anomalous force. Normally these principles are a-causal. However, we can infer the direction of causality because we know the two end points of the causal path: the surfaces of the feature (in the nominal simulation) or the surfaces produced by its removal (in the modified simulation) are at the start of the path and the anomalous force is at the end. By starting the reasoning at the anomalous force and working toward the feature, we know that we are always going opposite the direction of causality.

Our program uses a propagation process to construct the causal path for an anomalous force. It starts by looking for all of the forces that could directly cause the anomalous force. It then looks for all of the forces that could directly cause those forces and so on until reaching a force that is directly produced by the surfaces of the feature or the surfaces created by removing the feature.

To construct the causal path, the program uses knowledge of the laws of friction, Newton's third law, the laws of equilibrium, and a few other specialized principles of mechanics. The remainder of this section describes these.

Friction. In the usual case, normal forces cause friction forces. Hence, when looking for the cause of a friction force, the program usually concludes that the corresponding normal force is the cause. There are, however, two exceptions to this rule. The first exception, which is typically called self-energizing friction, occurs when the friction force and corresponding normal force apply opposing moments to a rotating body and the ratio of the normal force's moment arm to that of the friction force is less than the coefficient of friction. In this case the normal force and friction force mutually cause each other and the propagation of causality stops.

The second exception, which we call "wedging," occurs when the normal force is due to elastic deformation and the corresponding friction force prevents the two parts from moving relative to one another in a way that would relax the deformation and eliminate the forces. A sharply tapered peg press-fit into a similarly tapered hole is a prototypical example. As with self-energizing friction, when wedging occurs, the normal force and friction force mutually cause each other and the propagation of causality stops. (Wedging is detected by the simulator in the normal course of computing motion.)

Newton's Third Law. Newton's third law states that for every action there is an equal and opposite reaction. For our purposes we interpret this as saying that one of the forces of the action-reaction pair causes the other. The challenge is determining which is the cause and which is the effect. The pro-

---

[2] It may be possible that the forces change only in magnitude and not in direction (i.e., there is no change in their *qualitative* magnitude) but do so in such a way that there is still a change in the qualitative magnitude of the net force. If this situation occurs, our current program will be unable to produce a causal explanation. However, in the examples we tried this situation did not occur.

gram is able to disambiguate the answer because it always starts at the end of the causal path (the anomalous force) and works opposite the direction of causality. Thus, the force which is equal and opposite the anomalous force is the cause of the anomalous force. Similarly, a force which is equal and opposite to a second force which is known to cause the anomalous force is the cause of that second force, and so on. The program applies this principle only to normal forces; the causes of friction forces are determined using the friction rules above.

Equilibrium. As stated above, the only component of a force that affects the motion of a body is the projection onto the degree of freedom. Thus, when reasoning about equilibrium one need consider only the projections. (If the body rotates, the projection onto the degree of freedom is defined as the sign of the moment of the force about the axis of rotation.) To simplify the discussion, in the remainder of this section we will use the term “force” to mean “the projection of the force onto the degree of freedom.”

To see how the conditions of equilibrium provide a useful source of clues about causality, consider the block in Figure 7a which is subject to three forces, **A**, **B**, and **C,** which are in equilibrium. Imagine that we are looking for the cause of force **A**, i.e., the causal path has reached from the anomalous force to **A**, and we are trying to extend the path from **A** toward the feature. It is clear that **B** cannot cause **A** because they are in the same direction. **C** is the only force opposite **A**, so it must be the cause.

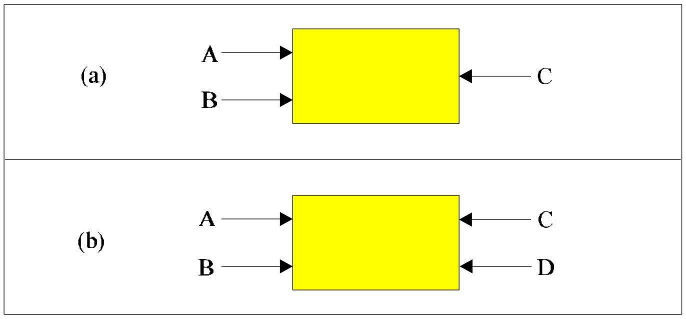

Figure 7 : (a) Three force equilibrium, (b) Four force equilibrium

If there are multiple forces opposite the force whose cause is being sought, the causal path is ambiguous. For example, if we add another force to the previous example as shown in Figure 7b, the cause of **A** is unclear: either **C** alone, **D** alone, or **C** and **D** together could be the cause. The program has two techniques that can often disambiguate causality in this situation. First, the program checks to see if any of the forces with opposite sign are actually caused by the force whose cause is being sought. Any such forces are eliminated as possible causes. For example, if **A** is known to cause **D**, then **D** could not be the cause of **A** because that would be equivalent to **A** causing itself. In this case, the only remaining possibility is that **C** is the sole cause of **A**. One way this could occur is if **A** is a normal force and **D** is the corresponding friction force (recall that **A** and **D** are projections of the actual forces onto the direction of motion – thus the actual normal and friction forces are not parallel). The other approach the program uses to resolve ambiguities is to check if any of the forces with opposite sign are caused by any of the other forces with opposite sign. If so those forces become a single cause. For example, if **C** causes **D** then **C** is the ultimate cause of **A**.

As described above, when the program reasons about equilibrium, it reasons about the projections of the actual forces onto the direction of motion. This requires the program to make use of rules which transfer causality from the projections of the forces to the actual forces and vice versa. There are rules for both translational and rotational problems. For example, when looking for the cause of a moment, the program looks for the cause of the force which produces the moment. Similarly, when looking for the cause of a force applied to a rotating body, the program looks for the cause of the force’s moment.

PROGRAM IMPLEMENTATION

Our causal analysis program is implemented as a forward chaining rule-based system. The rules encode the principles of mechanics described above. The system starts by identifying the anomalous forces (those that could be responsible for the differences in the end states of the simulations). It then chains from these forces until it reaches the feature in question. The chaining can branch because sometimes there is more than one possible cause for a force. We use a graph search algorithm to prune away any branches that do not connect the anomalous forces to the feature. The consequent of each rule generates a text string describing why each deduction was made. As the graph search algorithm prunes the branches, it assembles the text strings into a paragraph such as the one shown earlier in the pencil example.

The program works from a list of assertions describing the first instant that the displacement-time plots differ. The following facts from both the nominal and modified simulation constitute the complete set of facts that the program requires as input:

1. The name and motion type – rotation or translation – of each part.
2. The name of the feature.
3. The names of the surfaces of the feature.
4. The names of the surfaces created by removing the feature.
5. The list of parts whose final states differ in the two simulations and a description of the difference (e.g., positive net displacement vs. no-displacement).
6. The names of all of the forces.
7. The equal and opposite force of each force (Newton’s third law).
8. For translating bodies, the sign of the projection of each force onto the direction of motion.

9. For rotating bodies, the signs of the moments of the forces about the axis of rotation and the lengths of the moment arms.
10. The magnitude – either zero or non-zero – of each force.
11. The type – normal, friction, or wedging – of each force.
12. The name of the normal or wedging force that corresponds to each friction force.
13. The coefficient of friction for each friction force.
14. The name of the surface that applies each force.
15. The list of parts that are springs.
16. The names of those bodies that are in equilibrium.

Currently we have implemented only the causal analysis portion of our design rationale capture system. In the examples described here, we generated the simulations by hand and manually transformed the results into the form needed by our causal analysis program. The manual tasks included identifying and removing the features, comparing the final states of the simulations, and segmenting the displacement-time plots. We focused on the causal analysis problem first because that portion of the problem was the least explored. Of the remaining tasks, only identifying and removing the features present some degree of technical challenge.

Sacks and Joskowicz [Sacks97] have developed a dynamic simulator that will provide most of the simulation capabilities we need. This simulator can handle fixed-axis devices composed of rigid bodies and springs and subject to multiple time-varying contacts. Fixed-axis devices are those in which all of the parts either translate along fixed axes or rotate about fixed axes. The pencil is an example of a fixed-axis device if the lead channel is assumed to be fixed rather than the pencil body (to advance the lead, one would pull the pencil body backward rather than pressing the eraser forward). To use this simulator, it is necessary to approximate any flexible bodies as rigid bodies connected by springs and kinematic joints. For example, in many mechanical pencils the collet is implemented as a single piece of plastic. However, this kind of collet can be accurately modeled as two rigid blades connected to the inner cylinder of the pencil by revolute joints and rotary springs as shown in Figure 8. For our purposes, the only capability that this simulator lacks is the ability to detect wedging. Fortunately, because the simulator works from configuration space, detecting wedging is straightforward: it occurs when two configuration space boundaries intersect at a shallow angle, where shallow is defined as any angle smaller than the arctan of the coefficient of friction.

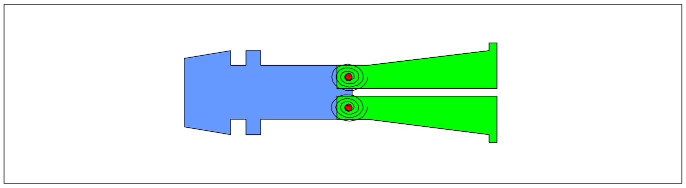

Figure 8 : Rigid body model of flexible collet blades. The blades are attached with pivots and springs.

In the context of this project, a feature is any embellishment to what would otherwise be a simpler part. Recall that one of our fundamental assumptions is that if the designer bothered to add a feature to a part, there must be a reason – the more complicated the feature, the more likely that there is a reason. Traditional feature recognition systems, which are typically intended to identify features used to plan manufacturing operations (e.g., [Sakurai90], [Sakurai95], and [Das96]), often work from libraries of prototypical features. However, for our purposes a feature can be an arbitrary chunk of geometry and thus library approaches may not provide a complete solution. In our case, the best approach to identifying features would be to first identify the “simple geometry” that underlies the actual geometry of the part. The difference between this simple geometry and the actual geometry would be the set of features that need to be explained. Work in multiresolution modeling (e.g., [DeHaemer91], [Heckbert94], and [Borrel95]) may form the basis of an approach for identifying the simple underlying geometry of a part.

Our particular application gives additional constraint in determining where to look for features. First, any surface that never touches other surfaces has no kinematic or dynamic purpose (other than, perhaps, avoiding contact) and needs no further analysis. Second, any surface which maintains static contact with another surface needs no further analysis because the purpose of that pair of surfaces is to provide static structural support.

RELATED WORK

There is a large and growing body of work in design rationale capture and construction. [Gruber91] and [Chung97] offer good overviews of work relevant to the work described here. However, much of that work is focused on tools for managing documentation that is human generated whereas our work aims to automatically compute rationales.

Our approach can be seen as similar in spirit to work of Gautier and Gruber ([Gautier93], [Gruber93]) which uses models of a device to automatically generate design rationales. They use compositional modeling [Falkenhainer91] and causal ordering techniques [Iwasaki86] to produce explanations for a device’s behavior. They reason about devices composed of components that are connected together at ports associated with parameters like temperature and pressure (“component-connection devices”). Constraints “inside” a component relate the values of the parameters at each of its ports. Depending on the context, their system selects different sets of constraints to describe a given component. For example, the constraints for a valve would depend on whether it is opened or closed. Their system can produce descriptions of what mode a component is in (which constraints are active), when the device changes state (mode change), and what causes a parameter value to change. In the domain of component-connection devices, the interesting behavior happens inside components, and components interact only through shared scalar parameters. In the domain considered here, behavior arises through interactions between the shapes of components. Hence, component-connection techniques do not apply here.

Garcia's Active Design Documentation (ADD) system computes rationales for parametric design problems [Garcia97]. This system works from an initial design model that describes both the artifact and the decision making process for selecting parameter values. The system generates rationales by comparing parameter values predicted by the decision making model with those actually selected by the designer. This system works from a decision making model constructed by a knowledge engineer, while our approach directly infers rationales from simulations.

There has been some previous work in trying to "understand" the behavior of mechanical devices (mechanisms). Forbus et. al. [Forbus91] and Faltings [Faltings92] describe systems that produce descriptions of the motions of the parts of a device. They decompose the device's configuration space into regions of uniform contact called "places," producing a "place vocabulary" for the device. They generate a description of the device's behavior by describing the sequence of places that are visited when the external inputs are applied to the device. Sacks and Joskowicz [Sacks93] describe a similar system that partitions configuration space into a region diagram (similar to a place vocabulary). They produce a description of the behavior by enumerating the sequence of regions that are visited when the external inputs are applied. These systems are intended to produce descriptions of what is happening and thus provide little explanation for why. Because they do not derive cause and effect relationships, these systems do not derive the purposes of the parts.

Shrobe [Shrobe93] describes a system that produces causal explanations for the behavior of linkages by interpreting kinematic simulations. The simulator is based on Kramer's TLA [Kramer90]. By examining the order in which the simulator solves the kinematic constraints, the system can decompose the linkage into driving and driven parts. The system then analyzes the traces of points on the driven members and angles of the driving members (i.e., crank angles) to look for interesting features (these are features of the traces, not geometric features on the parts). The system then uses geometric reasoning to derive causal relationships between the features. In one example, the system decides that the purpose of a linkage is to cause dwell. The explanation is that because the driving member moves in an arc whose radius is the same length as the driven member, the other end of the driven member does not move (dwells). This approach can detect some of the purposes of the parts of a device, but it is limited to kinematic behaviors. Also, it is limited to linkages and cannot handle the devices with time-varying contacts (variable kinematic topology) considered here. Finally, it cannot handle behaviors that depend on compliance, friction, inertia, etc.

[Doyle88] describes a system that hypothesizes a sequence of parts that could achieve a set of observable events. He generates possible causal connections between the inputs and observed outputs, but he does not consider the actual structure of the device – he treats it as a black box. He computes explanations for how the device might work, not necessarily how it does work. Thus, he does not compute design rationales.

Our approach is the computational equivalent of reverse engineering ([Ingle94], [Lefever96], [Otto96]) in that we work from a model of the device to infer the purpose of its parts. Our approach is also similar to Lefever and Wood's "Subtract and Operate" (SOP) technique for reducing part count [Lefever96]. SOP is the technique of removing a part from a device and then operating it to determine if the device still functions properly or if that part was necessary. However, SOP is performed by a human analyst using a physical device whereas our techniques are automatically performed by a computer program.

We infer causality by using principles of mechanics to analyze the propagation of force through a device. Previous work in other domains has indicated that there are a variety of "flows" that mark causality. For example, de Kleer describes a program that produces causal explanations of the small signal behavior of electric circuits by using constraint propagation techniques to propagate the electrical inputs through the circuit [deKleer79]. Stahovich has demonstrated that the flow of power through a device is another means of inferring causality [Stahovich93]. Additionally, Iwasaki has demonstrated that the order in which the governing equations must be solved indicates causality [Iwasaki86]. Both of the latter techniques are likely to be useful in our domain and will be explored in future work.

FUTURE WORK

After completing the other parts of our design rationale capture system as described in the "Implementation" section, we plan to extend this work in a number of directions. The planned extensions include developing methods for automatically deriving simulation models from raw geometric models and adapting our approach to problems that involve other domains of physics.

We currently assume that the simulation models are provided by the designer. There are several reasons for this. First, as described above, the simulations the designer runs are frequently intended to examine the important parts of the design. Thus, there is a virtue in working from the simulation models that the designer has created. Second, comparing these simulation models to the raw geometric models gives additional clues about what behaviors the designer intended each of the parts to exhibit. For example, a model in which the blades of the collet are connected to the inner cylinder of the pencil with pivots and springs indicates that the designer intended the blades to be compliant. Comparing the designer's simulation models with the raw geometric model will provide an additional low cost means of automatically capturing the design intent.

Our eventual goal is to develop techniques for automatically deriving simulation models from the raw geometric models. This ability is useful for several reasons. First, removing a feature from a part may require a substantial change to the simulation model. For example, removing the axial slit which forms the two blades of the collet (Figure 8) will eliminate the collet's flexibility, thus necessitating a new simulation model. Second, the ability to automatically generate simulation models will allow our system to examine parts of the design that the designer did not simulate.

For the domain of problems considered here, the two primary tasks in creating a simulation model are determining which component interactions can be idealized as kinematic joints and which parts must be modeled as flexible. The first task is straight forward and there are a number of published techniques for doing this such as [Joskowicz91] and [Gelsey87]. The second task is more difficult because every part is rigid if the forces are small enough, and conversely, all parts (except extremely brittle ones) are flexible if the forces are small enough. Apparently, to determine which parts are flexible, it is first necessary to know how large the forces are. A simple way to estimate the forces would be to compute an initial simulation with all parts assumed to be rigid. Any parts with large forces could then be modeled as flexible and the process repeated until all flexible parts are identified.

Although we have focused on interpreting dynamic simulations, we believe that our techniques will prove to be applicable to interpreting a wide range of simulations. To identify the behaviors of a feature, we simulate the device with and without the feature and examine the differences. Applying this approach in other domains requires having suitable simulators and a definition of what differences in the simulations are significant. Once we have used the two simulations to identify the behaviors of a feature, we search for a causal explanation that links the feature to those behaviors (that explanation forms the basis of the design rationale). For the dynamic simulations we consider here, we identify causality by examining how forces propagate through the device. We believe that we will be able to accomplish the same task in other domains by examining other kinds of physical "flows." (See "Related Work.") For example, interpreting thermal and electrical simulations may require reasoning about the flow (propagation) of heat and current through the device.

CONCLUSION

We have developed an approach for automatically computing a class of design rationales. The approach focuses on computing causal explanations of the purposes of the geometric features on the parts of a device. The approach uses a two step process. The first step is to simulate the nominal device and compare this to a simulation of the device with the feature in question removed. The differences in the two simulations are indicative of the behaviors that the feature ultimately causes. The second step is to use the fundamental principles of mechanics to construct the causal explanation that links the feature to these behaviors. This explanation constitutes one of the rationales for the feature. In the work described here, we use a dynamic simulator and hence can derive the mechanical (as opposed to electrical, thermal, etc.) purposes of features. We have implemented the portion of the program which constructs the causal explanations and have tested it on a variety of examples which includes the mechanical pencil described here, a door lock, and a single use camera.

While it is true that design rationales are necessary for performing a variety of common and important engineering tasks, it is also true that those who create this documentation are often not the ones who directly benefit from it. The all too frequent result of this misalignment between costs and benefits is that inadequate documentation is produced. This project, by creating techniques that automatically compute a class of design rationales, brings us one step closer to alleviating this problem.

REFERENCES

[Borrel95] Borrel, P. et. al., "The IBM 3D Interaction Accelerator (3DIX)," IBM T.J.Watson Research Center, Yorktown Heights, NY, Research Report IBMC-20302, 1995.

[Chung97] Chung, P. and Bañares-Alcántara, R., editors, "Special Issue: Representation and Use of Design Rationale," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 11, No. 2, 1997.

[Das96] Das, D., Gupta, S., Nau, D., "Generating Redesign Suggestions to Reduce Setup Cost: A Step Towards Automated Redesign," *Computer-Aided Design*, Vol. 28, No. 10, pp. 763–782, 1995.

[DeHaemer91] DeHaemer, M. and Zyda, M., "Simplification of Objects Rendered by Polygonal Approximations," *Computer and Graphics*, Vol. 15, No. 2, pp. 175-184, 1991.

[deKleer79] de Kleer, J., "Causal and Teleological Reasoning in Circuit Recognition," Massachusetts Institute of Technology Ph.D. Thesis, 1979.

[Doyle88] Doyle, R., "Hypothesizing Device Mechanisms: Opening up the Black Box," MIT AI Lab. Technical Report No. 1047, 1988.

[Falkenhainer91] Falkenhainer, B. and Forbus, K., "Compositional Modeling: Finding the Right Model for the Job," *Artificial Intelligence*, Vol. 51, pp. 95-143, 1991.

[Faltings92] Faltings, B., "A Symbolic Approach to Qualitative Kinematics," *Artificial Intelligence*, Vol. 56, pp. 139-170, 1992.

[Forbus91] Forbus, K., Nielsen, P. and Faltings, B., "Qualitative Spatial Reasoning: The CLOCK Project," Northwestern University, The Institute for the Learning Sciences Technical Report No. 9, 1991.

[Garcia97] Garcia, A. and de Souza, C., “ADD+: Including Rhetorical Structures in Active Documents,” *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 11, pp. 109-124, 1997.

[Gautier93] Gautier, P. and Gruber, T., “Generating Explanations of Device Behavior Using Compositional Modeling and Causal Ordering,” *Eleventh National Conference on Artificial Intelligence*, 1993.

[Gelsey87] Gelsey, A., “Automated Reasoning about Machine Geometry and Kinematics,” in Proceedings 3rd IEEE Conference of AI Applications, pp. 182-187, 1987.

[Gruber91] Gruber, T., Baudin, C., Boose, J. and Weber, J., “Design Rationale Capture as Knowledge Acquisition Tradeoffs in the Design of Interactive Tools”, Stanford University, Knowledge Systems Laboratory, Report KSL 91-47, 1991.

[Gruber93] Gruber, T. and Gautier, P., “Machine-generated Explanations of Engineering Models: A Compositional Modeling Approach,” *Proceedings of the 1993 International Joint Conference on Artificial Intelligence*, 1993.

[Heckbert94] Heckbert, P. and Garland, M., “Multiresolution Modeling for Fast Rendering,” *Proceedings of Graphics Interface 94*, pp. 43-50, 1994.

[Ingle94] Ingle, K., *Reverse Engineering*, McGraw-Hill, Inc., New York, 1994.

[Iwasaki86] Iwasaki, Y. and Simon, H., “Causality in Device Behavior,” *Artificial Intelligence*, Vol. 29, pp. 3-32, 1986.

[Joskowicz91] Joskowicz, L. and Sacks, E., “Computational Kinematics,” *Artificial Intelligence*, Vol. 51, pp. 381-416, 1991.

[Knuffer90] Knuffer, T. and Ullman, D. , “The Information Requests of Mechanical Design Engineers,” *Design Studies*, Vol. 12, No. 1, pp. 42-50, 1990.

[Kramer90] Kramer, G., “Solving Geometric Constraint Systems,” in Proceedings *AAAI-90*, pp. 708-714, 1990.

[Lefever 96] Lefever, D. and Wood, K., “Design For Assembly Techniques In Reverse Engineering And Redesign,” *ASME Design Theory and Methodology Conference*, 96-DETC/DTM-1507, 1996.

[Otto96] Otto, K. and Wood, K., “A Reverse Engineering and Redesign Methodology for Product Evolution,” *ASME Design Theory and Methodology Conference*, DETC/DTM-1523, 1996.

[Sacks93] Sacks, E. and Joskowicz, L., “Automated Modeling and Kinematic Simulation of Mechanisms ,” *CAD*, Vol. 25, No. 2, pp. 106-118, 1993.

[Sacks97] Sacks, E. and Joskowicz, L., “Dynamical simulation of planar assemblies with changing contacts using configuration spaces,” Purdue University Computer Science Dept. Technical Report 97-008, 1997.

[Sakurai90] Sakurai, H. and Gossard, D., “Recognizing Shape Features in Solid Models,” *IEEE Computer Graphics and Applications*, pp. 22-32, 1990.

[Sakurai95] Sakurai, H., “Volume Decomposition and Feature Recognition: Part 1 - Polyhedral Objects,” *Computer-Aided Design*, Vol. 27, No. 11, pp. 833-843, 1995.

[Shrobe93] Shrobe, H., “Understanding Linkages,” in *Proceedings AAAI-93*, pp. 620-625, 1993.

[Stahovich93] Stahovich, T., Davis, R. and Shrobe, H., “An Ontology of Mechanical Devices,” *Working Notes, Reasoning about Function, AAAI-93*, pp. 137-140, 1993.

[Stahovich97] Stahovich, T., Davis, R., and Shrobe, H., “Qualitative Rigid Body Mechanics,” in *Proceedings AAAI-97*, pp. 138-144, 1997.