

Supporting Creative Mechanical Design

Kun Sun and Boi Faltings

Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology (EPFL)
IN - Ecublens, 1015 Lausanne, Switzerland
Fax: +41-21-693-5225
e-mail: sun@lia.di.epfl.ch

Abstract

Knowledge-based CAD systems limit designers' creativity by constraining them to work with the prototypes provided by the systems' knowledge bases. We investigate knowledge-based CAD systems capable of *supporting* creative designs in the example domain of elementary mechanisms.

We present a technique based on qualitative explanations which allows a designer to extend the knowledge base by demonstrating a structure which implements a function in a creative way. Structure is defined as the geometry of the parts, and function using a general logical language based on qualitative physics. We argue that the technique can accommodate any creative design in the example domain, and we demonstrate the technique using an example of a creative design.

The use of qualitative physics as a tool for extensible knowledge-based systems points out a new and promising application area for qualitative physics.

Introduction

We consider those knowledge-based CAD systems where designers can compose designs from a library of *prototypes* ([Gero,90]) which the CAD system "knows" how to instantiate and adapt during the design process. Such systems limit designers' creativity by constraining them to designs which can be constructed as combinations of prototypes provided by the systems knowledge base. Creative ideas often fall outside this scope and thus cannot be accommodated in such a system.

We would like to propose that a design is creative if cannot be composed exactly from the prototypes in the system's knowledge base. The CAD system *supports* creative design if it allows the designer to define novel prototypes to cover his ideas. It *is* creative if it discovers new prototypes by itself. New prototypes are created by envisioning the prototypes of knowledge base in a different environment.

The techniques we describe in this paper allow designers to extend a prototype base by providing:

- a model of the function that their creative ideas address, expressed in a general logical language based on qualitative physics.
- a geometric model of a device that implements this function.

The system envisions the qualitative behavior of the device and identifies the behavior which implements the function that the designer intends. This allows the system to explain and generalize the idea, and define a new prototype for it. The new prototype can then be instantiated in any novel device the designer might want to construct with it. Note that the example device given by the designer can be very different from the device where the creative idea will be used. For example, a behavior observed in rocks could be reused in the design of a mechanism. If the representation languages used for structure and function are sufficiently general, our technique is guaranteed to cover any creative design which designers might propose.

Creativity is generally associated with extending a space of design alternatives or variables ([Gero,92, Sargent,92]). Such extensions are always possible only if the space of possible prototypes is unbounded. This is the case only if

- (i) the set of possible functions is infinite,
- (ii) the set of structures which can implement them is infinite, and
- (iii) there is no context-free mapping between primitives of structure and function which would allow composition of any prototype from a small set of primitive ones.

One such domain is designing geometric shapes which implement kinematic functions. For kinematic function, we define a generative qualitative representation language which allows an infinite set of different expressions. We represent shapes by polygons, which allow an infinite set of shapes. Kinematic functions are generated by contacts between shapes, and there thus exists a direct mapping between elementary functions

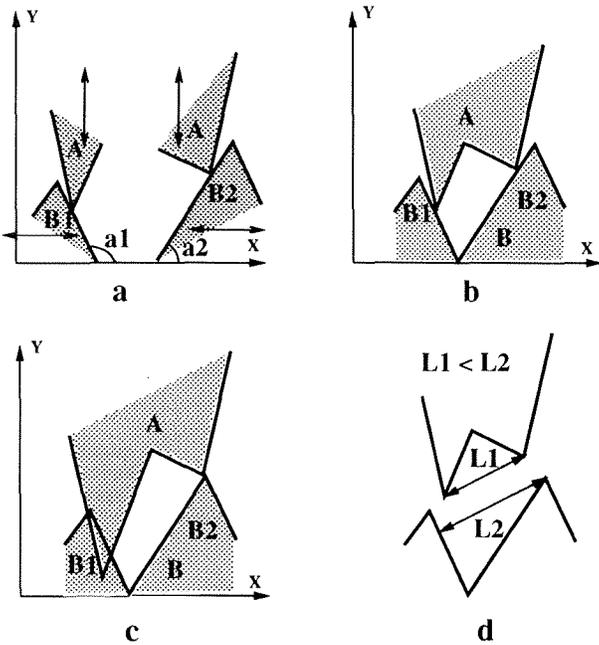


Figure 1: Composing prototypes into a novel device.

and structures. To satisfy requirement (iii), we must show that this mapping is not context-free, i.e. that the function of a particular structure is not independent of the context it is used in. Because of interactions through the geometry of the shapes, this is the case. As an example, consider the design of a device which prevents a block from dropping in the negative Y-direction, a function which could be represented as:

- Device (A, B)
behavior: $Y(A) \neq \ominus$

This function can be composed from two prototypes implemented by single contacts, shown in Figure 1 (a) and specified as follows:

- Device-1 (A, B₁)
behavior: $Y(A) = \ominus \rightarrow X(B_1) = \ominus$
- Device-2 (A, B₂)
behavior: $Y(A) = \ominus \rightarrow X(B_2) = \oplus$

The classical abductive reasoning says that if we know β and $\alpha \rightarrow \beta$, then the best explanation for β is that α is true. It means that creative design can also be considered as finding the best explanation for the design goal.

We can expect the knowledge-based CAD system to be capable of proposing a composition where $B_1 = B_2 = B$, based on the following abductive reasoning:

- $Y(A) \neq \ominus$ if $Y(A) = \ominus$ implies a contradiction.
- $X(B) = \ominus$ and $X(B) = \oplus$ is a contradiction.

However, not all ways of composing the two devices actually lead to the composed function. For example, the composition of Figure 1 (b) is legal, but in Figure 1 (c) both contacts can not occur simultaneously and thus the device does not implement the desired function. This problem occurs whenever a contact can be *subsumed* by another. To satisfy the composed function, the device has to satisfy a *compositional* constraint (Figure 1 (d)) which become apparent only when the context of the device is known. But this also means that the composition of several prototypes defines a new and different prototype which was not known before. Thus, in designs which involve geometry there is no direct mapping between functions and structures, and such situations allow creative designs *even within fixed representation languages for structure and function*.

Now assume that a designer has decided to implement the support function by combining the two prototypes, a design which the CAD system does not know about. As the system's knowledge is insufficient to guarantee that the prototype composition will work as expected - the system does not know the difference between cases (b) and (c) in Figure 1 - the designer himself must draw a correct solution. By qualitative analysis of the device, our system then computes a causal *explanation* of the way it implements the given function. In particular, this explanation will show the additional constraint which has to be satisfied in order to avoid the subsumption in Figure 1 (c). This explanation is now used to automatically define a new prototype which is added to the systems knowledge base.

The system could be made creative by itself if it were provided with a mechanism for exploring possible functions and geometric structures. However, such a process must be guided by an evaluation of the interestingness of functions, known to be a very hard problem in the learning community. When a desired function is given, it is sometimes possible to use this to guide the process of exploration ([Faltings,92b]).

A more complex example of a kinematic pair is a clock escapement, shown in Figure 2. Using the theory of qualitative kinematics ([Faltings,90, Forbus *et al.*,91]), the qualitative behavior and function of any kinematic pair can be computed. Furthermore, it is possible to *invert* the computation to determine the limits up to which it is valid ([Faltings,92a]) and thus the explanation needed to define a new prototype. Our techniques concentrate on the *conceptual* design stage where creativity takes place. Subsequent *detail design* can further optimize the dimensions to accommodate non-qualitative specifications.

Our work builds on the idea of *prototypes* ([Gero,90]), but is also influenced by work on design systems using model-based abductive reasoning (for example: [Williams,90], [Neville and Weld, 92], [Sycara and Navinchandra,90]).

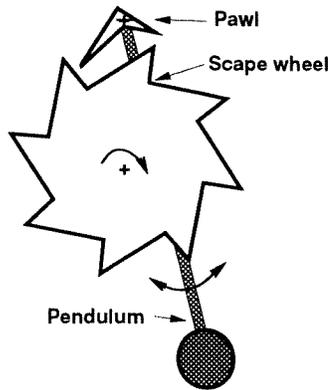


Figure 2: An escapement design produced using our system. The wheel is driven clockwise, and the pawl is attached to a pendulum which creates an oscillation with constant period.

However, existing work on model-based design has been based on the use of context-independent prototypes which do not consider the geometry of their composition. Design of kinematic chains by composing kinematic pairs has been studied in [Subramanian,93]. For certain classes of kinematic pairs, Subramanian gives an algorithm to determine their dimensions and arrangements in a chain under the assumption of uniform motion. Synthesis of kinematic pairs with considerations of geometric shapes has been studied by Joskowicz and Addanki ([Joskowicz and Addanki,88]), but they only treated the trivial cases of convex objects where no subsumptions and thus no compositional constraints can occur.

We first discuss the representation formalisms underlying our technique: a language for representing kinematic *function* based on qualitative physics, and a formalism for representing shape features. We then show how the analysis of a particular device can be generalized to define the shape features which are responsible for the its function and thus define a new prototype, and how this prototype can be used in design. Finally, we give an example of a creative design in which creative ideas developed by observing a ratchet mechanism are used to design a novel kind of forward-reverse mechanism.

Prototypes for kinematic pair design

Prototypes for kinematic pair design consist of a qualitative function and a shape feature which implements it. Both must be expressed in a well-defined representation language so that they can be composed by the CAD system. For creative design, these languages must be generative and capable to express any possible function or structure. In our system, we use a hierarchy of representation languages, as shown in Figure 3. In this section, we present the languages we use for

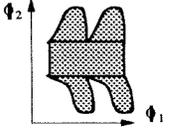
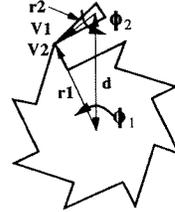
model	example	language
environment of use	$F = (+1 -1)$	qualitative vectors
functional feature	$\forall x \text{ place}(x) \rightarrow \exists y \dots$	quantified logical expressions
place vocabulary		behavior predicates
configuration space		contacts
metric diagram		vertices edges dimensions constraints

Figure 3: Representations used in our CAD system. Polygonal shapes are represented by a metric diagram consisting of vertices, edges and their dimensions. This structure defines a configuration space of the device, which shows the possible contacts between parts. Behavior predicates model qualitative features of the configuration space, and make up a place vocabulary model of the space. Finally, functions are defined as logical expressions involving the environment of u and the place vocabulary.

modeling qualitative function and shape features.

Metric diagram representation of shape

Shapes are represented using a *metric diagram*. The metric diagram consists of a symbolic structure which defines vertices, edges and metric parameters for the positions of the vertices. In our current implementation, the metric diagram is restricted to polygons, but can be extended to include circular arcs. A metric diagram represents several objects, each of which has a well-defined degree of freedom.

Using the metric diagram, a *shape feature* (which may involve several objects) is defined by:

- a set of vertices and edges,
- the metric parameters associated with them,
- a set of constraints which must hold simultaneously for the shape feature to be present.

For example, the shape feature which corresponds to the possibility of the top of the ratchets lever being able to touch the wheel (Figure 3) can be expressed as follows:

- must exist: vertices v_1, v_2
- constraints: $|d - r_1| < r_2$

A language for modeling qualitative kinematic function

For supporting creative design, it is crucial to be able to model any function that a designer might consider. In this section, we present a generative language capable of representing any kinematic function. *Function* is a property of behavior caused by certain external influences on the device. For example, the function of a ratchet is to block the motion of wheel in one direction when the pawl is forced downwards and to not block it in the other direction. We define a set of *behavior predicates* and a formalism for expressing external influences on mechanisms. Functions are then defined by logical expressions connecting external influences and behavior predicates. The language is similar to languages like CFRL ([Iwasaki *et al.*, 93]), but allows general logical expressions which are required to represent many mechanical functions.

Representing qualitative kinematic behavior

Textbooks on the subject explain kinematic behavior qualitatively by sequences of *kinematic states*. Examples of kinematic states of a ratchet device are shown in Figure 4.

In qualitative physics terminology, a graph of kinematic states and transitions is called an *envisionment*. It can be computed based on a *place vocabulary*, a graph where each node represents a different combination of contact relationships, and each arc represents a potential transition between them. The envisionment is obtained by combining each node of the place vocabulary with assumed motions and keeping

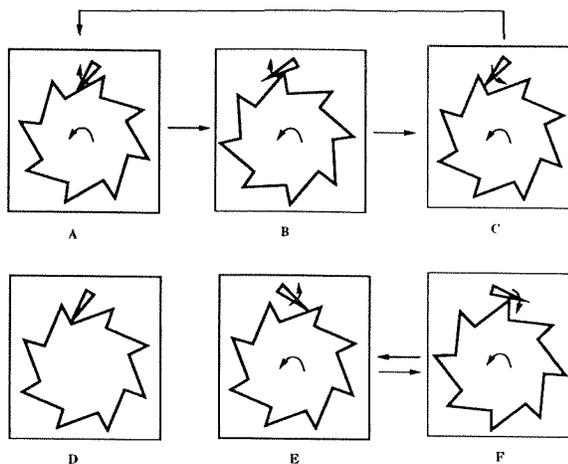


Figure 4: *Examples of kinematic states and transitions in a ratchet.*

only the states and transitions consistent with external forces and motions. We have developed and implemented complete algorithms to compute place vocabularies for arbitrary two-dimensional higher kinematic pairs in fixed-axis mechanisms. These have been used to compute envisionments for a number of mechanisms, such as a mechanical clock ([Forbus *et al.*,91]).

Behavior predicates We represent place vocabularies using a set of *behavior predicates* which characterize places, their features and their connectivity. For a kinematic pair, the place vocabulary defines a graph containing three types of kinematic states, corresponding to two, one and no contacts, and identified by the following behavior predicates:

- **point-place(x)**: the contacts in x hold only in a single configuration.
- **edge-place(x)**: the contacts in x hold in a one-dimensional set of configurations.
- **face-place(x)**: x is a place without any contacts and two-degrees of freedom.

For each place, the place vocabulary defines the allowed qualitative directions of motion:

- **qualitative-motion(d)**: d is a qualitative vector (d_0, d_1) whose components indicate the direction of motion of each object: $d_i \in \{-, 0, +\}$.
- **allowed-motion(x,d)**: motion d is possible everywhere in place x .

For each link between states, the place vocabulary defines the directions which can cause a transition:

- **transition(x,y,d)**: motion d can cause a transition from place x to y .

Qualitative motions The kinematic states of a mechanism are obtained by combining each place x with its maximal set of possible qualitative motions:

$$\mathcal{M}_{all}(x) = \{m \mid \text{allowed-motion}(x, m)\}.$$

In an actual behavior, only those motions $\mathcal{M}(x)$ which in fact *caused* by an external influence actually occur. The set of transitions between states is the set:

$$\mathcal{T}_m = \{ (x, y) \mid (\exists d \in \mathcal{M}(x)) \text{ transition}(x, y, d) \}.$$

More details on envisioning mechanisms using place vocabularies can be found in [Nielsen,88].

Representing external influence In kinematic pairs, external influences can be either *forces*, represented by a set \mathcal{F}_{ass} of qualitative vectors, or *motions*, represented by a set \mathcal{M}_{ass} , also consisting of qualitative vectors. Since a qualitative force vector causes a qualitative motion in the direction of the same vector, the set of possible motions \mathcal{M}_f caused by external forces is then given as:

$$\mathcal{M}_f(x) = \{v \mid v \in \mathcal{F}_{ass}\}$$

The actual set of motions $\mathcal{M}(x)$ to be considered in state x is then:

$$\mathcal{M}(x) = \mathcal{M}_{ass}(x) \cap \mathcal{M}_f \cap \mathcal{M}_{all}(x)$$

Formulating functions Functions are properties of behavior under certain environment. In our system, they are the assumed forces and motions. Therefore, qualitative functions can be defined as *logical conditions* on place vocabularies without first constructing the qualitative behavior. Using logical expressions on the behavior predicates which represent the place vocabulary, a set of functions can be defined as required for the application, and extended whenever required to express a new specification. For example, some functions our current prototype system uses are:

- **transmitting-place**(x, dir_1, dir_2):
 $(\forall d = (d_1, d_2))\{d_1 = dir_1 \Rightarrow d_2 = dir_2\} \wedge \{d_2 = -dir_2 \Rightarrow d_1 = -dir_1\}$
- **blocking-place**(x):
 $\neg(\exists d \in \mathcal{M}(x))\text{allowed-motion}(x, d)$
 (a place blocks motions if it does not allow any of the assumed motions).
- **partial-blocking-place**($x, dirs$):
 $\neg(\exists d \in dirs)\text{allowed-motion}(x, d)$
 (a partial blocking place blocks the specified motions)
- **possible-path**(x_0, x_n):
 $(x_0 = x_n) \vee \exists S \triangleq (x_0, x_1, x_2, \dots, x_n) (\forall i < n)(\exists d \in \mathcal{M}(x_i))\text{transition}(x_i, x_{i+1}, d)$
 (There is a path from place x_0 to place x_n whenever there is a sequence of places with transitions between them under at least one assumed motion)

- **cycle**(x_0, C):
 $C = (x_0, x_1, x_2, \dots, x_n, x_0) (\forall x_i \in C)(\exists d \in \mathcal{M}(x_i))\text{transition}(x_i, x_{mod(i+1, n+1)}, d)$
 (there is a cycle of states C such that transitions between subsequent states are consistent with assumed and allowed directions of motion.)

A place vocabulary can only fulfill the required functions if the number of states and their connectedness is sufficient. Reasoning about such *topological* features is difficult in the place vocabulary itself, since it is based only on individual *boundaries* of shapes which cannot be modified individually. We use an explicit representation of the *kinematic topology* ([Faltings *et al.*,89]) of the mechanism to detect cases where the topology of particular object shapes would not permit the specified function. An example of a function defined on the basis of kinematic topology is:

- **cycle-topology**(c, d_1, d_2): if the first or second object have rotational freedom, the cycle involves d_1 rotations of the first or d_2 rotations of the second object. This predicate is defined directly on the kinematic topology of the mechanism.

which can be defined formally using similar behavior predicates as those which define place vocabularies.

The function of a ratchet can now be defined qualitatively as follows:

For all starting states a in which the orientation of the lever is in the interval $[0..π]$ (pointing to the left such that the moment gravity exerts on it is positive):

- for $\mathcal{M}_{ass}=\{(+, *)\} \wedge \mathcal{F}_{ass}=\{(*, +)\}$ (the '*' stands for either +, 0 or -):
 - **cycle**(a, C) \wedge **cycle-topology**($c, 1, 0$)
 - $\neg(\exists x)\text{blocking-place}(x) \wedge \text{possible-path}(a, x)$
 (assuming that the wheel turns counterclockwise and the lever is forced onto it, there is a cycle of states where the wheel can rotate, and no reachable blocking state from any starting state a.)
- for $\mathcal{M}(x)=\{-, *\} \wedge \mathcal{F}_{ass}=\{(*, +)\}$:
 - $(\forall y)\text{possible-path}(a, y) \Rightarrow \{\neg\text{cycle}(y) \wedge (\exists z)(\text{blocking-state}(z) \wedge \text{possible-path}(y, z))\}$
 (assuming that the wheel turns clockwise, no reachable state leads to a cycle and all states can eventually lead to a blocking state).

Note that due to the ambiguities inherent in qualitative envisionments, the formalism always overgenerate behaviors. It is therefore only possible to define *necessary*, but never *sufficient* specifications of behavior and, consequently, function. For example, we can express the specification that clockwise motion leads to a blocking state only in an indirect manner: if there is no possibility to cycle, and there is at least one reachable blocking state, the device must eventually reach this state.

Creating and using new prototypes

Explaining functions New prototypes are defined by generalizing a particular device which implements

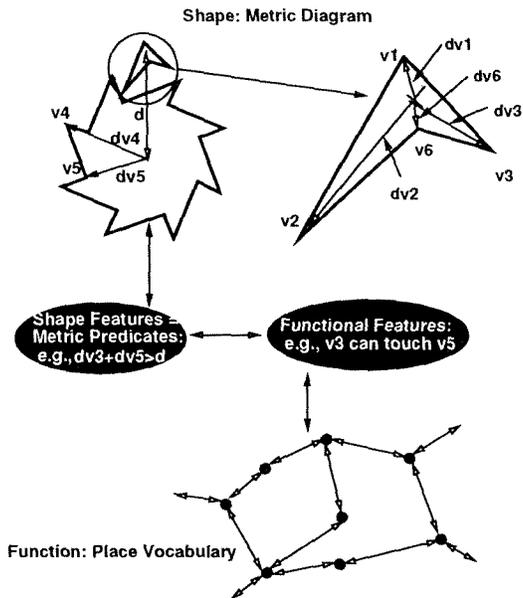


Figure 5: Analysis of a device defines a place vocabulary as a set of features, for example the possibility of touch between the tip of the lever and the bottom of the wheels teeth. Such features define metric predicates such as the one shown.

a novel function. The generalization is based on an explanation of the function in terms of the structure of the device.

A function is a quantified logical expression of behavior predicates defined on the place vocabulary. When the function is implemented in a device, the envisionment defines a set of behavior predicates. Among these predicates, there is at least one logical conjunction which satisfies the quantified condition defining the function:

functional feature \Rightarrow

behavior-pred₁ \wedge behavior-pred₂ \wedge ...

The essence of a creative idea is now formally defined by the choice of a particular conjunction of behavior predicates which satisfy the quantified logical expression defining the function. Note that in general, finding all conjunctive propositions which satisfy a quantified logical expression is a non-computable problem, thus putting creativity beyond the scope of algorithms.

Defining shape features Analyzing the behavior of a device using a place vocabulary produces a set of behavior predicates. Each of these predicates is computed based on certain properties of the geometry of the device being analyzed (Figure 5). The analysis which produced the behavior predicates can be generalized to constraints which the shape must satisfy in order for the behavior predicates to remain present.

These constraints, taken together, define a qualitative shape feature which is associated to the functional feature. That is:

behavior-pred₁ \wedge behavior-pred₂ \wedge ... \Rightarrow

constraints on shapes \Rightarrow shape feature

Reversing the causal chain of the analysis thus establishes a mapping from functional features to shape features, and we call such a process causal inversion. More details on the mapping between shape and qualitative behavior can be found in [Faltings,92a].

For any functional feature identified in the place vocabulary of a device, we can thus construct a corresponding shape feature which implements it. These shape features, indexed by the functions, form new prototypes which are added to the system's knowledge base.

Kinematic pair design with prototypes

A kinematic pair is specified by a conjunction of qualitative functions. Structures which satisfy these functions are obtained by combining prototypes from the systems knowledge base such that all required functions are covered. Combining prototypes means combining shape features using the following steps:

1. choose a unification of vertices and edges of the shape features defined in the prototypes.
2. instantiate the constraints associated with the shape features and find a solution to the resulting constraint network.
3. envision the solution to determine compositional constraints which must also be considered, add them to the constraint network and iterate from step (2).

This process poses two major difficulties: satisfying the dynamic constraint network, and discovering and adding compositional constraints.

Constraint satisfaction The constraint network for combining shape features is dynamic and involves many nonlinear constraints. No reliable and efficient method exists for solving such constraint networks. In our current prototype, we use a process of iterative refinement where an initial partial solution, given for example by the device used to define the prototype, is incrementally modified until all constraints are satisfied. The refinement process uses two types of modification operators:

- *dimensional* modifications, where the dimensions of parts are adjusted to fit the functional requirements, and
- *topological* modifications, where vertices are added to part shapes. A topological modification is always coupled with a dimensional modification to fix the dimensions of the new features.

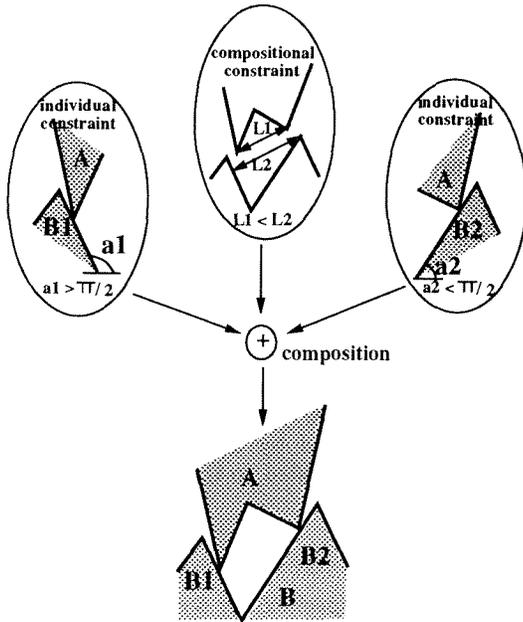


Figure 6: Inferring a compositional constraint from observing the behavior of a combination of shape features.

Dimensional modification varies the values of dimensional parameters, thus changing the appearance of one place and its properties (for example, inference rule). Topological modification are proposed when there is no dimensional modification which can satisfy additional constraints. More details about the computation of dimensional and topological modifications can be found in [Faltings and Sun,93].

Discovering compositional constraints As discussed in the introduction, combination of shape features often implies novel interactions which result in additional compositional constraints. In kinematics, the only interactions we have to consider are *subsumptions*, where one shape features makes the contact of another impossible or alters the way it occurs. Compositional constraints which ensure the absence of subsumptions can be formulated most easily once a subsumption has been observed. For example, if we observe the subsumption shown in the introduction (Figure 1), we can infer a novel compositional constraint as shown in Figure 6 by expressing the condition that the subsuming contact may not occur simultaneously with the subsumed contact as an algebraic inequality. Subsumption constraints are relatively simple expressions in the case of translational motion, but can be considerably more complex in the case of kinematic pairs involving rotations.

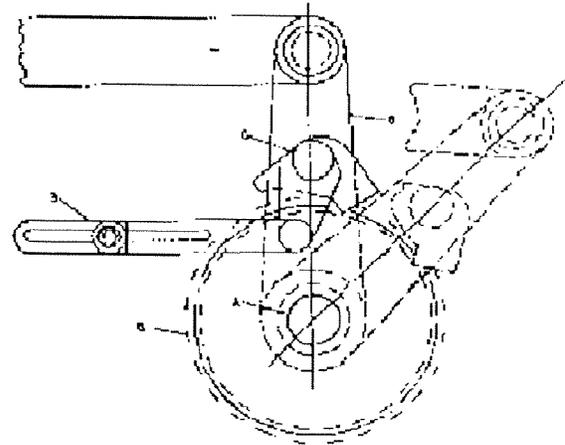


Figure 7: A forward-reverse mechanism as found in a popular mechanism book.

Creative design of a forward-reverse mechanism by composition

As an example of using composition for creative design, we show how a novel forward-reverse mechanism can be conceived by composing two ratchets.

A forward-reverse mechanism is used to transform an oscillating motion into a rotation which advances in one direction and, after a period of rest, reverses the motion to a lesser degree. A solution for this problem in the literature ([Newell and Horton,67]) is shown in Figure 7. It uses 4 parts and a friction-based mechanism which is problematic for maintenance.

Using the functional features defined earlier, the function of a mechanism with forward-reverse movements can be specified as follows:

1. (\exists array $\mathcal{X} = \{x_0, x_1, x_2, \dots, x_{n-1}\}$) of states such that:
 $(\forall x_i \in \mathcal{X}) \text{transmitting-place}(x_i, (+, -)) \wedge \text{array-topology}(\mathcal{X}, 1, 0)$
 (There is a array of states which transmits counterclockwise motion of input driver to clockwise motion of wheel.)
2. (\exists array $\mathcal{Y} = \{y_0, y_1, y_2, \dots, y_{n-1}\}$) of states such that:
 $(\forall y_i \in \mathcal{Y}) \text{transmitting-place}(y_i, (-, +)) \wedge \text{array-topology}(\mathcal{Y}, 1, 0)$
 (There is a array of states which transmits clockwise motion of input driver to counterclockwise motion of wheel.)
3. for $\mathcal{F}_{ass} = (0, +)$, $\mathcal{M}_{ass} = \{(-, +), (0, +), (+, +)\}$:
 $(\forall x_i \in \mathcal{X}) \text{possible-path}(x_i, y_i), y_i \in \mathcal{Y}$
 (When the input driver changes the motion from counterclockwise to clockwise, there exists a path from place x_i to y_i)
4. for $\mathcal{F}_{ass} = (0, -)$, $\mathcal{M}_{ass} = \{(-, -), (0, -), (+, -)\}$:

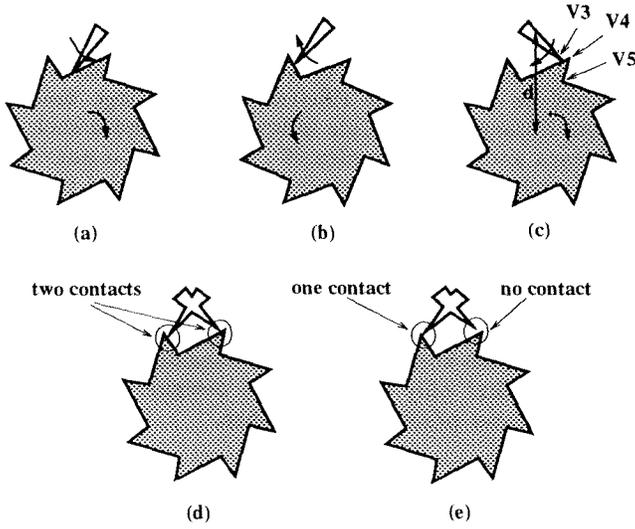


Figure 8: Creative design of forward-reverse mechanism by composing two ratchets. a), known device of start design. a), b) and c), interesting aspects of functional features discovered by envisionment. d), composition of shape features with subsumptions. e), iterative satisfaction of subsumption constraints.

- ($\forall y_i \in \mathcal{Y}$) **possible-path**($y_i, x_{mod(i+1,n)}, x_{mod(i+1,n)} \in \mathcal{X}$)
 (When the input driver changes the motion from clockwise to counterclockwise, there exists a path from place y_i to the place $x_{mod(i+1,n)}$ in the array \mathcal{X} .)
5. ($\forall x_i \in \mathcal{X}$) ($\forall y_i \in \mathcal{Y}$) { **dist**₁($y_i, y_{mod(i+1,n)}, (+, *)$) < **dist**₁($x_i, x_{mod(i+1,n)}, (-, *)$) }
 (The counterclockwise motion angle of wheel in each period is smaller than its clockwise motion angle.)

Discovering functional features in the envisionment

Assume that the designer has noticed that a ratchet device, when used in the environment of forward-reverse mechanism, can achieve some of the required functions. In an envisionment of the ratchets behavior in the environment of the forward-reverse mechanism, the designer specifies the following correspondences to the functional specifications:

1. there is one array of states, Figure 8 (a), in which the lever can drive the wheel. The counterclockwise motion of the lever turns the wheel clockwise. Therefore, it can be used to satisfy functional specification (1).
2. there is another array of states, Figure 8 (b), in which the lever can also drive the wheel. The clockwise motion of the lever pushes the wheel counterclockwise, which meets the requirement of specification (2).

3. when the lever changes the motion from counterclockwise to clockwise, there is a possible path from one state of array \mathcal{X} to one state of array \mathcal{Y} in the same period, which fulfills specification (3).
4. when the lever changes the motion from clockwise to counterclockwise, there are no paths for satisfying specification (4), since from one state of array \mathcal{Y} , it returns to one state of array \mathcal{X} in the same period.
5. when the wheel is driven by the lever, clockwise and counterclockwise motion angles of the wheel are always equal, which means that specification (5) cannot be satisfied.

The designer now searches functional features to satisfy specifications (4) and (5). For solving the discrepancy with specification (5), the designer has to add a set of states where motion of the lever results in a clockwise motion of the wheel:

$$(\exists \text{array } \mathcal{Z} = \{z_0, z_1, z_2, \dots, z_{n-1}\}) \\
\forall z_i \{ \mathbf{transmitting-place}(z_i, (+, *)) \} \wedge \\
\mathbf{array-topology}(\mathcal{Z}, 1, 0) \\
(\text{States } z \text{ transmits some motion of the lever to a clockwise motion of the wheel.})$$

The envisionment of the ratchet in fact contains such an array of states z , shown in Figure 8 (c). We assume that the designer decides to use this set of states to make it possible to satisfy specification (5). However, (5) refers to the transitions defined in specification (4), which are not yet satisfied in the simple ratchet device. Assume that the designer decides to satisfy specifications (4) and (5) by creating paths passing through intermediate states chosen from the array \mathcal{Z} . He communicates this to the system by identifying in the envisionment of a ratchet the states $z \in \mathcal{Z}$ and the transitions which should be connected to states in \mathcal{X} and \mathcal{Y} , thus defining the shape features and their relative positions used in the design.

Defining a new prototype

The new prototype consists of the functional feature, defined by the logical expression given earlier, and the corresponding shape feature which implements it. The shape feature is defined by the explanation underlying the envisionment. For example, the existence of state z in Figure 8 (c) can be translated to the existence of vertices v_3, v_4, v_5 , the center distance d , and the following constraints:

$$\begin{aligned} \text{C1: } & \sqrt{x_4^2 + y_4^2} - \sqrt{x_3^2 + y_3^2} > 0 \\ \text{C2: } & x_3^2 + (y_3 - d)^2 - \frac{(d \times (x_5 - x_4) + x_4 \times (y_5 - y_4) + y_4 \times (x_4 - x_5))^2}{(y_4 - y_5)^2 + (x_5 - x_4)^2} > 0 \\ \text{C3: } & (x_3 - x_4) \times (y_5 - y_4) - (y_3 - y_4) \times (x_5 - x_4) > 0 \\ \text{C4: } & x_3 \times (x_4 - x_5) + (y_3 - d) \times (y_4 - y_5) > 0 \end{aligned}$$

Composing prototypes

The functional features have been mapped into two shape features, each defined as a set of constraints on the metric diagram of a single ratchet device. Furthermore, the identified transitions impose constraints on the relative positions of the shape features in the combined device. For the output member, these can be satisfied by one and the same object, but the input driver has to be a composition of two levers implementing specifications (1)-(3) and specification (4)-(5), respectively. The resulting composed device is shown in Figure 8 (d).

Satisfying compositional constraints

Not all compositional constraints can be specified before composition, but many are only discovered when the composed device is envisioned. For example, the composed device has subsumptions as illustrated in Figure 8 (d), where tips of the levers touch two teeth of the wheel simultaneously. In this case, the composed levers are blocked from further clockwise movement. Therefore, the paths from state y to z in the replaced specification (4) are broken. A new subsumption constraint is added to avoid this behavior, starting a search for a better solution which satisfies all constraints.

We have seen that the constraints describing the state z and subsumptions involve v_3, v_4, v_5 and d , which are highly nonlinear. Their satisfaction is very difficult. We attack this problem by an incremental refinement. In this example, assume that the designer chooses to change vertex v_3 to search for a solution. By carrying out a region search, the system changes its position to $v_3 = (7.58, 24.00)$. This results in a new device as shown in Figure 8 (e). Renewed envisionment shows that it is in fact a functional forward-reverse mechanism, and the design is finished.

Conclusions

A main shortcoming of knowledge-based CAD systems is the fact that precoded design knowledge does not allow designers to express their creative ideas. In this paper, we have presented an implemented technique which shows that knowledge-based technology and creativity are not contradictory concepts. *Qualitative physics* provides the extensible representations needed to accommodate creative ideas in a knowledge-based system. Qualitative physics has exactly the functionality required for extending design spaces, as postulated by many researchers in creative design. Using more complete domain models such as developed for IICAD ([Kiriya *et al.*, 92]), the technique is applicable to more general domains than elementary mechanisms. This points to a new and as yet unexploited application of qualitative physics as a tool to support extensible knowledge-based systems.

Our current approach is geared towards *supporting* creative designs, not generating them automati-

cally. By automatically searching the space of possible geometric structures, it would be possible to construct a fully automatic "creative" system. Such search might be made more efficient using a large mechanism library and suitable indexing techniques such as [Sycara and Navinchandra, 90]. However, it is not clear how such a generation process could be guided to only furnish functionalities which are in fact interesting. One way to do this might be to provide such a search with a set of specifications taken from a standard mechanism textbook and ask it to find novel solutions which satisfy them. However, the combinatorial problems associated with such a search are considerable, and we consider that only supporting a designer's creativity has a greater practical importance.

References

- Boi Faltings, Emmanuel Baechler, Jeff Primus. Reasoning about Kinematic Topology. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, 1989
- Boi Faltings. Qualitative Kinematics in Mechanisms. *Artificial Intelligence*, 44(1), 1990
- Boi Faltings. A Symbolic Approach to Qualitative Kinematics. *Artificial Intelligence*, 56(2), 1992
- Boi Faltings. Supporting Creativity in Symbolic Computation. *Second International Round-Table Conference on Computational Models of Creative Design*, Heron Island, Queensland, Australia, 1992
- Boi Faltings, Kun Sun. Computer-aided Creative Mechanism Design. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, 1993
- Ken Forbus, Paul Nielsen, Boi Faltings. Qualitative Spatial Reasoning, the CLOCK Project. *Artificial Intelligence*, 51(3), 1991
- John Gero. Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine*, 11(4), 1990
- John Gero. Creativity, emergence and evolution in design. *Second International Round-Table Conference on Computational Models of Creative Design*, Heron Island, Queensland, Australia, 1992
- Y. Iwasaki, R. Fikes, M. Vescovi, B. Chandrasekaran. How things are Intended to Work: Capturing Functional Knowledge in Device Design. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, 1993
- Leo Joskowicz, S. Addanki. From Kinematics to Shape: An Approach to Innovative Design. *Proceedings of the AAAI*, St. Paul, August 1988
- Takashi Kiriya, Tetsuo Tomiyama, Hiroyuki Yoshikawa. Building a Physical Feature Database for Qualitative

Modeling and Reasoning, *AAAI Fall Symposium, Design from Physical Principles*, Cambridge, MA, 1992

Dorothy Neville, Daniel S. Weld. Innovative Design as Systematic Search, *AAAI Fall Symposium, Design from Physical Principles*, Cambridge, MA, 1992

John Newell, Holbrook Horton. Ingenious Mechanism for Designers and Inventors, Volume IV. Industrial Press INC, New York, 1967

Paul Nielsen. *A Qualitative Approach to Rigid Body Mechanics*. Ph. D. Thesis, University of Illinois, 1988.

P.M. Sargent. A computational view of creative steps. *Second International Round-Table Conference on Computational Models of Creative Design*, Heron Island, Queensland, Australia, 1992

Devika Subramanian. Conceptual Design and Artificial Intelligence. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambéry, France, 1993, pp. 800-809

Katia Sycara, D. Navinchandra. Index Transformation Techniques for Facilitating Creative Use of Multiple Cases. *Proceedings of the 12th IJCAI*, Sydney, 1991

Brian Williams. Interaction-based Invention: Designing Novel Devices from First Principles, *Proceedings of the National Conference on Artificial Intelligence*, Boston, 1990