ELSEVIER

0010-4485(95)00027-5

Computer-Aided Design, Vol. 28, No. 3, pp. 207-216, 1996

0010-4485/96 $15.00+0.00

FAMING: supporting innovative mechanism shape design

Boi Faltings and Kun Sun

A popular saying claims that 'innovation is 1% inspiration and 99% perspiration'. In this paper, we present a method for automating most of the perspiration involved in innovative design. We restrict our attention to innovative design processes which can be structured into three steps: *discovery* of a new technique, *understanding* it, and *generalising* it to fit the problem at hand. The method we developed automates the understanding and generalisation phases which involve most of the perspiration.

We present the FAMING system which demonstrates the method for the design of part shapes in 2D elementary mechanisms, also called *kinematic pairs*. We believe that the results are generalisable to other domains with similar characteristics, in particular any problem where geometry plays an important role.

Keywords: intelligent CAD, qualitative reasoning, case-based reasoning

INTRODUCTION

Most computer programs concern *deductive* tasks such as *analysis*, where a single answer follows from the input data. In contrast, design is an *abductive* problem which often has infinitely many correct answers. As an example, consider the function of an escapement, the central element of mechanical clocks. Its function is to regulate the motion of a *scape wheel* to advance one tooth per oscillation of a pendulum. Since the pendulum has a fixed period of oscillation, this means the wheel moves at constant speed and can drive the hands of a clock. There are many mechanisms which can be used to implement an escapement function. *Figure 1* shows three different mechanisms which satisfy this function, and many more variations could be designed.

Because of the abductive nature of design, an apparently closed problem like designing part shapes for an escapement mechanism still leaves much room for innovation and creativity: even though the problem has been studied for hundred of years, novel devices are still being designed in the watch industry to this day. In this paper, we present the FAMING (Functional Analysis of Mechanism for Inventing New Geometries)* program for supporting creative design of part shapes in higher kinematic pairs. This domain has proven to be particularly interesting for research because it demands creative design solutions but can still be formalised with a small knowledge base.

Since an abductive problem has many possible answers, there cannot exit an effective procedure for reliably solving it by computer. The only known method for solving abductive problems in general is to *search* the space of potential solutions. For example, there is a very successful program which invents novel automatic transmissions for automobiles by searching the space of all physically possible topologies allowed by a certain technology[1]. This technique is successful in producing innovations because (i) a computer can search through large numbers of candidates, and (ii) a computer is much better than people at providing the correct analysis of a device's behaviour.

While transmissions can be modelled by a fixed set of parameters, kinematic pairs require more sophisticated models and design methodologies. Before describing the approach we have developed to support shape design, we are going to review several design paradigms and their applicability to design of kinematic pairs.

Model-based design

Work in *model-based design*[2-4] has attempted to develop systems which systematically search the space of combinations of a set of *models* to produce novel designs. A model is a structural element with a fixed behaviour. For achieving the desired function, the models can be *inverted* and thus propose a potential structure. For example, the system of Williams[2] proposes a design for refilling a punch-bowl by composing qualitative models of pipes and behaviour of liquids. Subramanian[5] describes a program which can propose kinematic chains. These methods work well when the interactions between the individual models can be precisely identified (the *no-function-in-structure* principle of qualitative modelling[6]).

Artificial Intelligence Laboratory (LIA), Swiss Federal Institute of Technology (EPFL), IN - Ecublens, 1015 Lausanne, Switzerland
Paper received: 3 October 1994. Revised: 28 April 1995

* FAMING also means 'invent' in Mandarin Chinese.

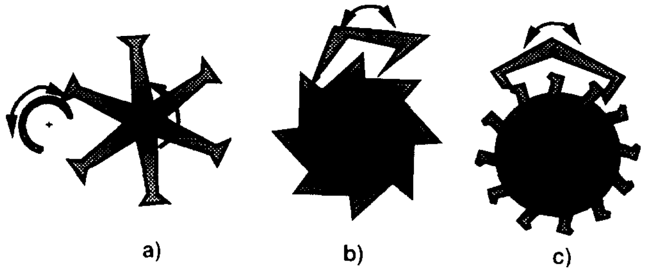


Figure 1 Three different kinematic pairs implementing an escapement function

Kinematic pairs achieve their function through the possible contact between elements of the part shapes. Because all contacts have to be integrated into a common geometry, they interact in very unstructured ways: any contact can *subsume* (inhibit) any other one and make it impossible. This makes it impossible to model kinematic function in a context-free formalism which would allow a model-based design strategy. Joskowicz and Addanki[7] have attempted to apply methods of model-based design to kinematic pairs. However, their method is capable to obtaining a solution only if by accident the specifications define a set of non-interacting contact relations. It is incapable of producing devices such as escapements or ratchets.

Prototype-based design

The highly context-dependent nature of mechanical shape design means that the space of mechanisms with interesting functions is sparse, as shown in *Figure 2*. If designs which are good for particular functions are to be described by combinations of predicates, this often means that predicates are invented specifically for one particular device. Rather than identifying combinations of properties which are responsible for a particular function, it is more appropriate to model *prototypes*[8] of complete designs.

Prototypes are usually parameterised to allow their

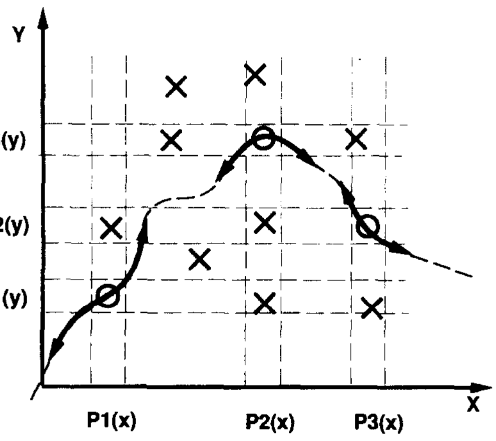


Figure 2 In highly context-dependent domains such as shape design for mechanism parts, the set of 'good' designs (circles) among the 'bad' designs (crosses) is often very sparse. Predicates P1–P3 and Q1–Q3 which distinguish good from bad often must be invented with a particular example in mind: P1–Q1, P2–Q3 and P3–Q2 cover the three positive examples shown here. In a prototype-based system, designs are found by varying the prototypes, as indicated by the arrows. The space of 'good' designs will often have a complex shape such as indicated by the dashed line, of which only the parts around the known prototypes are well understood

integration in varied design contexts. For many kinematic pairs, one can identify several key parameters which can be varied without changing the function. There exist commercial tools for supporting such design, for example ProEngineer or ICAD[9,10]. However, these tools only allow instantiation of preprogrammed prototypes and thus do not permit innovation without involving a programmer.

Case-based design

Case-based reasoning[11] is a technique where past solutions are reused or adapted to solve new problems. In *case-based* design[12], specific design precedents are reused for new problems. This approach maintains the advantages of the prototype-based approach, but simplifies knowledge acquisition, as a library of cases can be built up by simply recording earlier designs. FAMING, the system we describe in this paper, uses geometric models of mechanism part shapes as its cases.

Because designs can never be reused exactly, a key issue in case-based design is how to *adapt* a case to a new problem. For adaptation, it is crucial to know what changes can be made to a design case without perturbing its function. This knowledge, called *adaptation knowledge*, must be provided in addition to the model of the structure stored in the design case[13].

In FAMING, adaptation knowledge is provided by associating with each case an *interpretation* in terms of *structure–behaviour–function* (SBF) models[14]. An SBF model represents an *understanding* of the case: it defines how each element of the structure is responsible for aspects of the behaviour, and how aspects of the behaviour are in turn responsible for the function of the device. SBF models for case adaptation were first used in the KRITIK program of Goel[15], which adapts a nitric acid cooler design to cooling sulfuric acid. In FAMING, the user must only provide a formal expression of the kinematic functions which he considers important in the case. The program then automatically constructs the intermediate behavioural model and the links between it and structure on one side and function on the other.

In order to correctly express function, the SBF model must be a *qualitative* model. A qualitative model[16] differs from a quantitive one in that it specifies properties which hold over *ranges* of parameter values. This makes it possible to express functions which hold over a range of situations, such as 'block any counterclockwise motion'. Qualitative models also make it possible to determine *all* possible behaviours of a device, not only a particular snapshot valid for certain input parameters. This is useful for expressing negative specifications, such as 'part A should never move counterclockwise'. Finally, a third reason for using qualitiative SBF models is that their discrete nature allows problem-solving by search among all possibilities.

Case adaptation using SBF models

FAMING translates the understanding embodied in the SBF model to the structural level for use in adapting the case to new specifications. This *structure–behaviour*

(SB) inversion is possible due to the use of qualitative behaviour models. It consists of mapping each property in the behaviour model to *constraints* on the shapes which ensure that the property is satisfied. The resulting set of constraints on the structural model allows *generalisation* of the structure in the case. Additional constraints on the behaviour, such as numerical bounds, can also be formulated as structural constraints.

Finally, the case is *adapted* to fit a new problem. A first solution is obtained either by *combination* of the structures in several cases, or by *modification* of a single case's structure. Both operations are carried out while maintaining the validity of all constraints defined in the generalisation stage. Since the composition may require taking into account further compositional constraints, these are discovered through renewed qualitative simulation and corrected by modification operators.

Innovation in case-based design

Innovation and creativity can arise in case-based design through adaptation and combination of cases which might yield very different results from what was previously known. As observed for example by Kolodner[17], another powerful source of innovation is the *reinterpretation* of an existing case using a different SBF model. *Figure 3* shows an example where a ratchet device is reinterpreted using a different SBF model than originally intended. A combination of two such devices results in an innovative design of a *forward-reverse-mechanism*, a device which transforms an oscillating input motion into a rotation which advanced two steps forward and one step backward with each oscillation (this example is described in detail in Reference 18). The design obtained using FAMING is much simpler than those found in the literature and thus a truly creative solution. When the set of known shapes is large, such reinterpretation is the source of much of the innovation in design.

FAMING: an interactive design tool

While FAMING could be used in a fully automatic way, our research has shown that the complexity of the domain puts *fully* automatic design beyond the capabilities of current-way computers for all but very simple adaptations. Our tool is therefore an interactive one where search is avoided by asking the designer himself to make certain critical choices:

- which cases and which functions should be used, and
- in case of modification of a single case, which dimensions the system should attempt to modify, and
- in case of case combination, which features of the shapes should be unified.

These choices, especially the specification of cases and the functions which they should be used for, are the 1% inspiration that the designer himself is asked to provide. Given this information, our method can automate the remaining 99% perspiration required to make the designer's idea in practice by proposing a suitable adaptation. Because of the difficulty people have in envisioning kinematic behaviour, such adaptations often take months to produce for human designers. Using the computer tool makes it possible to explore many more ideas in less time, and consequently be much more creative.

Since FAMING is only a research prototype, it is subject to many limitations. Shapes are limited to 2D polygons (but could be extended easily to include circular arcs as shown in Reference 19). Since the design only takes into account qualitative criteria, the function of the result is correct only in qualitative terms. A subsequent *optimisation* stage is necessary to obtain a design which satisfies all requirements of manufacturability, durability etc.

QUALITATIVE SBF MODELS USED IN FAMING

Figure 4 shows the structures which implement the SBF models in FAMING, and how they are related through reasoning processes. The structure of a device is represented by a *metric diagram*, a geometric model of vertices and connecting edges. From the geometric model, deductive algorithms are used to compute a *place vocabulary*, a complete model of all possible qual-

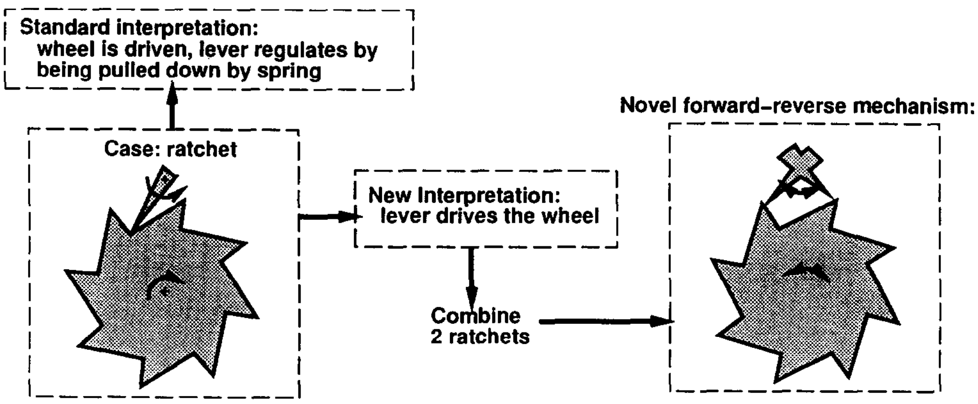


Figure 3 When viewed by a designer, artifacts are interpreted in terms of structure-behaviour-function (SBF) models. Innovation often results from reinterpreting cases with different models than originally intended

itative behaviours of the device. The place vocabulary can be formalised as a qualitative representation of the device's *configuration space*. Configuration space is a compact representation of the constraints on the part motions. Its usefulness for modelling mechanism kinematics has been argued for example in Reference 20. In order to be fast and reliable, FAMING directly computes a qualitative configuration space, called a *place vocabulary* and modelled using *behaviour predicates*. The *function* of a device is formulated using logical expressions on these behaviour predicates. Functional expressions can thus be *matched* automatically to the qualitative behaviour.

Structure: metric diagram

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 see Reference 19 for ways to extend it to include circular arcs. A metric diagram represents two objects, each of which has one well-defined degree of freedom.

Interesting aspects define *shape features*, which may involved both objects. For example, the fact that the top of the ratchet's lever is able to touch the wheel (v_1 touching v_2 in *Figure 4*) would define a shape feature. It is defined by:

- a set of vertices and edges, in this case v_1 and v_2.
- the metric parameters associated with them, in this case ϕ_1, r_1, ϕ_2 and r_2.
- a set of constraints which must hold simultaneously for the shape feature to be present, in this case $\{|d - r_1| < r_2\}$.

Qualitative behaviour

Textbooks explain kinematic behaviour qualitatively by sequences by *kinematic states*. A kinematic state is

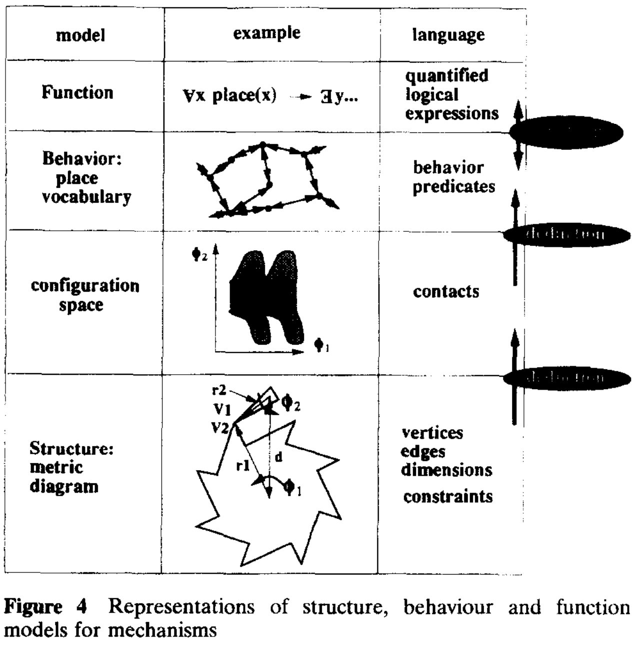


Figure 4 Representations of structure, behaviour and function models for mechanisms

defined by a contact relation and directions of part motion. Examples of kinematic states of a ratchet device are shown in *Figure 5*.

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 environment is obtained by combining each node of the place vocabulary with assumed motions and keeping only the states and transitions consistent with the external forces and motions. We have developed and implemented complete algorithms to compute place vocabularies for arbitrary 2D higher kinematic pairs in fixed-axis mechanisms. These have been used to compute environments for a large variety of practical mechanisms, such as a mechanical clock[21,22].

Qualitative motions

Each part in a kinematic pair has exactly one degree of freedom, so that the motion of a kinematic pair can be modelled by a vector of two parameters. Qualitatively, each value is modelled by its *sign*: +, 0 or −, so that a qualitative motion is a *qualitative vector* consisting of two such signs. As a shorthand, we shall use '∗' to denote the set of values $\{+, 0, -\}$.

Due to the fact that the information in a qualitative model is incomplete, the qualitative motion will often be *ambiguous*, i.e. admit several different vectors. For each state x of the device, we define a set $\mathcal{M}(x)$ which specifies all qualitative motion vectors possible in state x.

External influences

A kinematic pair can be actuated either by applying a force or momentum, in which case there could be an opposing force which prevents motion, or by forcing a

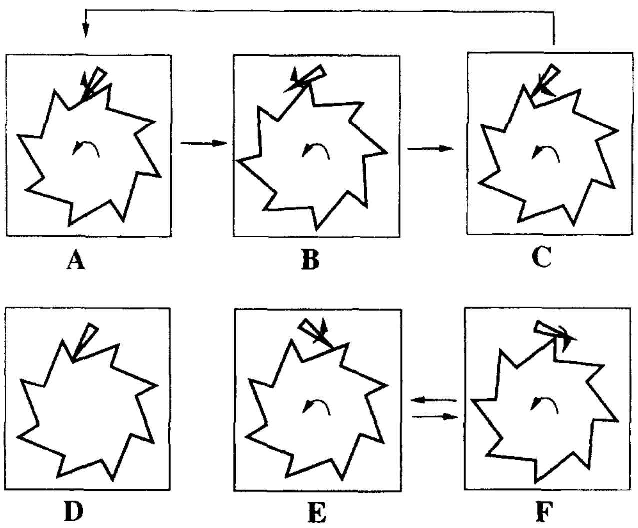


Figure 5 Examples of kinematic states and transitions in a ratchet. States A–C represent a cycle of behaviour in which the wheel can turn counterclockwise. State D can be reached from state A by reversing the wheel's motion; it blocks any further clockwise motion. States E and F represent a cycle which does not allow the ratchet function and is normally avoided by applying a counterclockwise force on the lever

particular motion, which cannot be prevented by any counteracting force. We assume that externally imposed motions and forces are independent of the device state, and represent them by sets of qualitative vectors $\mathcal{M}_{ext}$ and $\mathcal{F}_{ext}$. The set $\mathcal{M}_{ext}$ specifies all motions which are *consistent* with the external influence. The set $\mathcal{F}_{ext}$ denotes the set of actual force vectors which might be applied, and is often ambiguous because a function might be required under a range of circumstances.

Place vocabulary

The place vocabulary represents the set of possible contact relations. We represent place vocabularies using a set of *behaviour predicates* which characterise 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 behaviour predicates **point-place**(x), **edge-place**(x) and **face-place**(x), which specify that x is a place with 2, 1 or zero contacts, respectively.

For each place, the place vocabulary defines the allowed qualitative directions of motion. The predicate **allowed-motion**(x,d) specifies that motion d, where d is a qualitative vector, is possible everywhere in place x. It is often more useful to make use of the set of *admissible motions* $\mathcal{M}_{adm}(x) = \{d|\textbf{allowed-motion}(x,d)\}$. For each link between states, the directions of motion which can cause a transition are defined by the predicate **transition**(x,y,d), which specifies that motion d can cause a transition from place x to y.

Behaviour = Envisionments of kinematic states

A qualitative *behaviour* of a kinematic pair is an ordered sequence of *kinematic states*. A kinematic state is characterised by a particular contact relationship (place) and a qualitative motion. The set of all possible behaviours of a kinematic pair can be modelled by connecting all possible kinematic states in a graph whose arcs represent all possible transitions. Such a graph is called an *envisionment*[5,23] and is the result of the qualitative simulation procedure. The fact that the envisionment represents *all* possible behaviours, not just the behaviour for a certain input, is an important advantage of qualitative simulation over numerical techniques.

The envisionment of a kinematic pair is obtained from the place vocabulary in two steps. First, for each place we compute the set of *consistent* motions. Next, for each pair of kinematic states such that their underlying places are connected, we compute the set of possible transitions.

We define the *envisionment functions* $\mathcal{M}$ and $\mathcal{T}$ as the combination of all these considerations: $\mathcal{M}(x, \mathcal{F}_{ext}, \mathcal{M}_{ext})$, applied to a place x, returns the minimal set of qualitative motions in the place x, and $\mathcal{T}(x, \mathcal{F}_{ext}, \mathcal{M}_{ext})$ returns the set of possible transitions from place x to other places.

A language for specifying function

There has been much recent work within the model-based reasoning community on formalisms for representing function in design[24]. Chandrasekaran[25] and his students have investigated the notion of a *functional representation*. Iwasaki *et al.*[26] have developed a language called CFRL based on qualitative physics. Another major work is that of Tomiyama[27], who defines function using notions of qualitative process theory. All these proposals consider function to be a *causal* relation between an *environment* (or context) and a particular *behaviour*, and are concerned mainly with vocabularies for specifying these causal connections. Behaviour is defined for example as a set of relations between parameters (functional representation[25]), a set of active processes and views[27], or a precise sequence of states, each specifying a particular set of relations[26].

In kinematic pairs, there is only one form of causality, that of pushing on a part contact. In any particular state, the *function* of the device is given by the inference rules defined by the part contacts, expressed by the **allowed-motion** behaviour predicates. Many important functions, however, are properties of *sequences* of states. For example, a function might be that from any initial state, motion in a particular direction will eventually lead to a particular state, or that a certain behaviour cannot occur anywhere.

Such specifications can only be specified as logical *conditions* on the set of *all possible behaviours*. These are formulated using the behaviour predicates and allow quantification over states and qualitative motions and forces.

We therefore formulate functions in two levels:

- a *functional feature* defines a property of a *particular* state of set of states, and thus always takes at least one state as an argument. It furthermore takes the external influences $\mathcal{M}_{ext}$ and $\mathcal{F}_{ext}$ as implicit additional arguments. Functional features are similiar to causal process descriptions (CPD) used in CFRL[26].
- a *device function* defines a property of the entire behaviour. It consists of a logical expression in functional features where all states are bound by quantifiers, and a specification of the $\mathcal{M}_{ext}$ and $\mathcal{F}_{ext}$ assumed for this device function.*

For example, some functional features our current prototype system uses are:

- **a blocking-place**(x): $(\forall d \in \mathcal{M}_{ext}) \neg \textbf{allowed-motion}(x,d)$(a place blocks motions if all external motions are disallowed).
- **possible-path**($\mathcal{S}$): $\mathcal{S} = (x_0, x_1, x_2, \ldots, x_n)$ $(x_0 = x_n) \vee (\forall i < n) x_{i+1} \in \mathcal{T}(x_i, \mathcal{F}_{ext}, \mathcal{M}_{ext})$($\mathcal{S}$ 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).

A place vocabulary can only fulfil the requried 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 indivdiual states. We use an explicit representation of the *kinematic topology*[28] of the mechanism to formulate properties relating to the topology of be-

*Recall that we assume the external influences to be independent of the mechanism state.

haviour. An example of a functional feature defined on the basis of kinematic topology is:

- **cycle-topology** ($\mathscr{C}$, d_1, d_2): if the first or second object has rotational freedom, the cycle $\mathscr{C}$ involves d_1 rotations of the first or d_2 rotations of the second object

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

As an example, the functions of a ratchet can be defined as follows:

- for $\mathscr{M}_{ext} = \{(+,*)\} \wedge \mathscr{F}_{ext} = \{(*,+)\}$: $(\exists\ \mathscr{C})$ **cycle**$(a, \mathscr{C}) \wedge$ **cycle-topology**$(\mathscr{C}, 1, 0) \wedge \neg(\exists x)$ **blocking-place**$(x) \wedge$ **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 $\mathscr{M}\ (x) = \{-, *)\} \wedge \mathscr{F}_{ext} = \{(*,+)\}$: $(\forall y)|(\exists \mathscr{S} = (a\ , \ldots,\ y))$**possible-path**$(\mathscr{S})| \Rightarrow \{\neg(\exists\mathscr{C})$**cycle**$(y, \mathscr{C}) \wedge (\exists z)($**blocking-state**$(z) \wedge$ **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, they overgenerate behaviours. It is therefore only possible to define *necessary*, but never *sufficient* specifications of behaviour 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.

Quantitative constraints on behaviour

In many cases, purely qualitative specifications are insufficiently precise to specify a device function. For example, the specification of the forward–reverse mechanism (shown in *Figure 3*) must mention the fact that the forward motion is to be twice the reverse motion. Such constraints refer to particular configurations of the device, represented in the place vocabulary as 0-dimensional **point-places**. In order to allow their specification, the functional language contains the function **component** (x,i) which returns the ith coordinate of the configuration represented by **point-place** x. Constraints involving precise positions can be formulated on these coordinates.

INVERTING THE FBS MODEL

Adapting a case C to a novel problem requires

- understanding what aspects of the device are relevant to the interesting function, i.e. constructing an *interpretation* of its behaviour B in terms of the functional specification F, and
- using this understanding to construct a generalisation which either does not change these essential aspects, or changes them in the way that is intended. This is accomplished by an abductive *inversion* of behaviour to structure (SB inversion) which defines a set of *constraints* on the structural model.

Matching behaviour to functional specification

F is a quantified logical expression of behaviour predicates. C implements F by its behaviour B, therefore there exists at least one instantiation of the quantified variables in F with individuals of B such that the behavioural predicates of B satisfy F. Replacing all the quantified variables in F using this substitution, we obtain a *conjunction* of instances of behaviour predicates which define the way that F is implemented in C:

$$\text{functional feature } F \Rightarrow \text{behaviour-pred}_1 \wedge \text{behaviour-pred}_2 \wedge \ldots$$

As an example, suppose that the functional specification of a device contains the conditions that there exists a blocking-place:

$$\text{For } \mathscr{M}_{ext} = \{(-,*)\}, \mathscr{F}_{ext} = \{(*,*)\}: (\exists x)(\textbf{point-place}(x) \wedge \textbf{blocking-place}(x))$$

Suppose furthermore that the designer has selected a device which has a place P which qualifies as a blocking place. Unification of the functional specification with the place vocabulary substitutes P for x, thus transforming it into

$$\textbf{blocking-place}(P)$$

Replacing the **blocking-place** predicate by the full expression in its definition and expanding the quantification over all motions, we obtain the following conjunction of behaviour predicates:

$$\neg\textbf{allowed-motion}(P,(-,-)) \wedge$$
$$\neg\textbf{allowed-motion}(P,(-,0)) \wedge$$
$$\neg\textbf{allowed-motion}(P,(-,+))$$

The presence of quantification in F is essential for allowing innovation and creativity: if F did not contain any quantified individuals, it could at most admit a finite number of equivalent* behaviours which could be enumerated in a straighforward way. 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.

Structure-behaviour inversion

Each behaviour predicate in the conjunction is implemented by particular aspects of the object shapes. Using a trace of their computation, it is possible to determine the limits up to which the behaviour predicates remain valid. These limits define *constraints* on the shapes. The constraints, taken together, define a

* In the sense that the aspects satisfying the function are identical.

qualitative shape feature which is associated to the functional feature. That is:

behaviour-pred$_1$ $\wedge$ behaviour-pred$_2$ $\wedge$ ... $\Rightarrow$
constraints on shapes $\Rightarrow$ shape feature

Thus, the behaviour predicate $\neg$**allowed – motion**(P,($-$,$-$)) can be translated into constraints on positon of vertex v_1 when the device is in configuration P. The **point-place**(P) predicate translates into conditions for the particular touch being physically possible, i.e. not ruled out by other contacts.

Reversing the causal chain of the analysis thus establishes a mapping from functional features to shape feaures, and we call such a process *causal inversion*. More details on the mapping between shape and qualitative behaviour can be found in Reference 29.

CASE ADAPTATION

The final stage of the design process is to adapt the case(s) to the new problem. Cases can be adapted either by combining several cases, or by incrementally modifying one single case. Either way, the interpretation of the cases define a set of *structural constraints*. Some of these structural constraints fix the existence and connectivity of structural variables, and others restrict their relative values. Taken together for all desired functions, the structural constraints define a *constraint satisfaction problem* whose solution is the adapted case.

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. However, the cases themselves already provide partial solutions. Recent studies have shown that the iterative repair methods seem to be very efficient for solving large constraint satisfaction problems[30,31]. Cases can be used as initial solutions for such repair algorithms, as proposed by Pu and Purvis[32]. In the same spirit, our prototype solves the CSP by a combination of case *combination*, where value combintions which solve partial problems are combined into a new solution, and case *modification*, where an iteration of local repairs involving single variables is used to incrementally refine an initial assignment of values. The *topology* of the constraint network is decided by querying the user: in case combination, the user is asked to identify which parts of the shapes can be reused in the two devices. If the problem turns out to be unsatisfiable, he is asked to provide additional degrees of freedom by adding further variables.

In this section, we illustrate both case combination and modification for the example of designing an *escapement* mechanism. An escapement is a device where the constant-period oscillation of a pendulum regulates the rotation of a wheel such that it advances by one tooth for each oscillation. This function can be formally specified by stating that both extreme positions of the pendulum fall within *blocking-places* for the wheel's motion, and that there are paths between successive instances of these blocking places:

(1) $\exists \mathscr{X} = \{x_0, x_1, x_2, \ldots, x_{n-1}\}$
($\forall x_i$)**partial-blocking-place**(x_i, $(+,*)$) $\wedge$ **cycle-topology**($\mathscr{X}$, 1, 0)

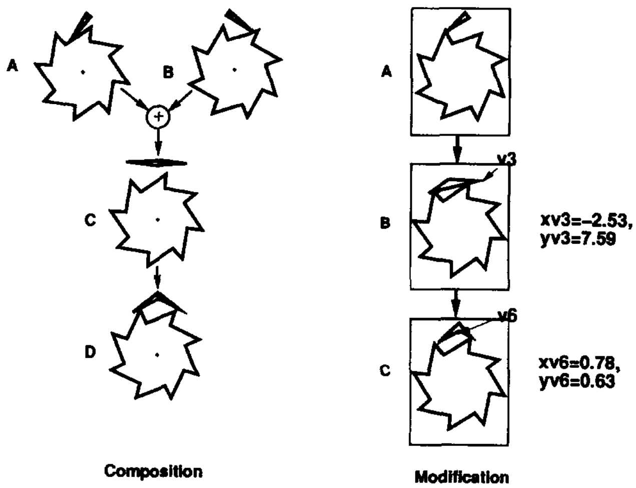


Figure 6 Design of an escapement by (left) composing two ratchet devices and (right) incrementally modifying a single ratchet

(2) $\exists \mathscr{Y} = \{y_0, y_1, y_2, \ldots, y_{n-1}\}$
($\forall y_i$)**partial-blocking-place**(y_i,$(+,*)$) $\wedge$ **cycle-topology**($\mathscr{Y}$, 1, 0)

(3) for $\mathscr{M}(x) = \{(*,0), (*,-)\} \wedge \mathscr{F}_{ass} = \{(+,*)\}$:
($\forall x_i \in \mathscr{X}$)**possible-path**($x_i$,$y_i$), $y_i \in \mathscr{Y}$ (when the pendulum swings from right to left or is stationary, there exists a path from place x_i to y_i)

(4) for $\mathscr{M}(x) = \{(*,0), (*,+)\} \wedge \mathscr{F}_{ass} = \{(+,*)\}$:
($\forall y_i \in \mathscr{Y}$)**possible-path** ($y_i$, $x_{\mathrm{mod}(i+1,n)}$), $x_i \in \mathscr{X}$(when the pendulum swings from left to right or is stationary, there exists a path from place y_i to the place following x_i in the cycle $\mathscr{X}$)

(5) for $\mathscr{M}(x) = \{(+,0)\}$:
$(\forall x)\neg(\exists\ \mathscr{C})$**cycle**($x$, $\mathscr{C}$)(the wheel is prevented from rotating a full cycle whenever the pendulum does not move).

Case combination

Assume that the designer has noticed that a ratchet device, when used in the environment of an escapement, can implement the desired **partial-blocking-states**. She decides to compose two ratchet devices and identifies their blocking states as the interesting functions, thus creating constraints on their composition. The left half of *Figure 6* shows a trace of the design process.

The functional features have been mapped into two shape features, each defined as a set of constraints on the metric diagrams of single ratchet devices A and B. A first composition (C) results in a non-functional device, as there is a cycle of states where the lever does not move, but the wheel is free to turn; this is a contradiction of specification (5). The transitions in this cycle of states have conditions associated with them. By SB inversion, these are translated into constraints on the positions of the vertices on the lever. Solving the modified constraints satisfaction problem results in design (D), which satisfies all the specifications.

A general and complete, but still somewhat inefficient method for solving the complex constraint satisfaction problems resulting from kinematic constraints has been proposed by Haroud and Faltings[33]. FAMING currently uses a more pragmatic solution using incremental refinement operators, called *modification operators*, which refine an initial candidate solution by modifying one parameter at a time.

Modification operators

A constraint satisfaction with continuous variables defines a *feasible region* within which all combinations of variable assignments satisfy the constraints. Assume for example that we have a problem with three variables x_1, x_2 and x_3 and the constraints:

$$C1: x_1 + x_2 > x_3 \text{ and } C2: |x_1 - x_2| > x_3 - 10.$$

Assume that there is an initial solution candidate where $x_1 = 2$, $x_2 = 26$ and $x_3 = 30$. This candiate does not satisfy C1. We can now generate three modification operators for this problem, one for each variable, by *projecting* the constraints into *feasible intervals* of each variable assuming that all other variables retain the same values. For example, if we would like to change x_1, we project both constraints onto x_1:

$$C1': x_1 > 4 \text{ and } C2': x_1 > 56 \vee x_1 < 6$$

and thus obtain the bounds of its feasible intervals as [4..6], [56..∞]. A modification operator would now change x_1 into one of the feasible intervals.

While some important successes with applying iterative refinement methods to constraint satisfaction problems have been reported in the literature[30,31], these methods are incomplete: the correct solution might require simultaneous modification of several paramters. In FAMING, we avoid the most serious incompleteness problems by always considering simultaneous modification of *pairs* of parameters which define the position of a vertex.

Topological modifications are proposed when the constraints formulated by metric predicates are contradictory: adding an additional vertex gives two additional degrees of freedom to resolve the contradiction. A vertex can be removed if placing it on the straight line between its two neighbours is consistent with all metric constraints.

Modification operators are indexed by their effect on the place vocabulary: *changing* the appearance of a state in the place vocabulary, or making a state *appear* or *disappear*. Based on matches between possibilities and active goals, the system computes a finite set of potentially applicable modification operators for proceeding with the design.

Generation and application of modification operators must be controlled by domain knowledge to avoid excessive search. Since it is very difficult to formulate such domain knowledge, our current system asks the user to choose the discrepancy to modify among a list of suggestions, the modification operator to apply among a list of suggestions, and any topological changes which might be required to create additional degrees of freedom.

One possible trace of an incremental modification where an escapement is obtained from a ratchet mechanism is illustrated in the right side of *Figure 6*. The ratchet (device A) already provides a cycle of blocking states which can be used to satisfy either the functional specification (1) or (2) of the escapement. However, it does not satisfy specification (5), and specifications (3) and (4) cannot be evaluated.

The system matches the cycle of blocking states of the ratchet to specification (1), and we assume that the user chooses to first satisfy specification (2). The first subgoal is then to create the cycle of places it requires, in a way that they satisfy the **partial-blocking-place** property. The user chooses to change the position of vertex $v3$ among the variables proposed by the system.

Solving the constraints added to resolve the discrepancies with specifications (2), (3) and (4) results in the values $x_{v3} = -2.53$, $y_{v3} = 7.59$, as shown in *Figure 6 (right) B*.

Discovering and satisfying compositional constraints

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. While it is possible to formulate all possible compositional constraints for guaranteeing that a particular device is subsumption-free, their number grows as $\mathcal{O}(n^d)$, where n is the number of possible contacts and d is the number of degrees of freedom of the device. In order to limit this complexity, our prototype generates compositional constraints only when they have been observed to be violated in a proposed solution.

For example, simulating version B of the escapement shown in *Figure 6* shows that due to a subsumption, the required transitions between blocking states are still impossible. In this case, the additional constraint makes the system of constraints contradictory* and FAMING requires the user to introduce a new degree of freedom. We assume that the user chooses to introduce a new vertex $v6$ between $v2$ and $v3$ to create this new degree of freedom. The system proposes to place v_6 at $x_{v6} = 0.78$ and $y_{v6} = 0.63$ to satisfy all constraints, shown in *Figure 6 (right) C*. Simulation shows that there are no subsumptions which produce unexpected behaviour, and furthermore specification (5) also turns out to be satisfied. User interaction is indispensible for controlling the modification process: had the system started by attempting to satisfy specification (5), for example, a long and not very fruitful search would have resulted. The intuitions behind such choices appear very complex and we doubt that they could be formulated in a sufficiently concise way.

CONCLUSIONS

In this paper, we have described the FAMING system for supporting innovative design of kinematic paris. FAMING

*Meaning that no solution can be reached by modifying a single vertex — there may still be a solution by modifying several vertices simultaneously which is not found by the system.

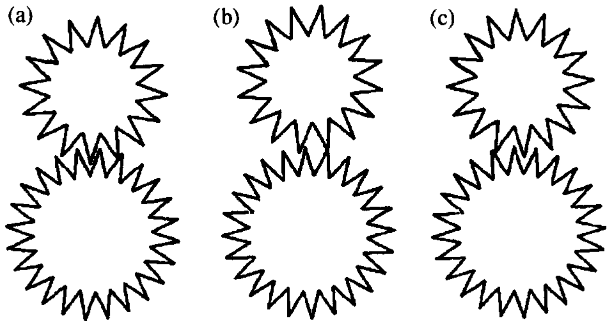


Figure 7 Three similar-looking gear devices with very different behaviours; (a) functions as a normal gear transmission, (b) the transmission ratio from the top wheel to the bottom is half that of the opposite direction, (c) motion can only be transmitted from the top wheel to the bottom one, the other direction will cause a jam

supports a particular approach to innovative design which is both computationally feasible and practical for designers to follow:

- the designer provides the innovative idea by suggesting modification, combination and reinterpretation of existing devices. He provides the 1% inspiration which he is best at.
- FAMING uses first-principle theories of qualitative kinematics to propose a design solution for the desired function following the idea suggested by the designer. The system thus contributes the 99% perspiration which the designer would like to avoid.

While we do not rule out the possibility of one day being able to systematically search the space of possible kinematic pairs, we consider the solution in FAMING a suitable compromise between what is computationally feasible with current-day computers and what is required by a designer. FAMING can provide a designer with an intuition about mechanism behaviour which he does not normally have. *Figure 7* shows three arrangements of gears which are almost indistinguishable to people, but implement very different functions. FAMING makes is possible to not only detect these possibilities, but to search for them in a systematic manner. As an example of the practical impact of the system, we have reproduced the development of a new escapement design for a major Swiss watch manufacturer within one afternoon. In current practice, this development has taken six months for a team of several people!

The work we have presented presents several novel contributions for AI in design research. In spite of the apparent simplicity of kinematic pairs, the scope of possible functions which can be constructed with them turned out to be much richer than what has previously been addressed in research on knowledge-based CAD systems. Consequently, certain aspects of the formalism we developed to formally represent qualitative function is significantly more powerful than previous proposals such as CFRL[26]. The infinite matching possibilities between function and behaviour allowed in our functional language also provide an explanation for why mechanism design is not a closed space of possible devices, but allows infinite possibilites for creativity and innovation.

Earlier work on designing mechanism part shapes[7] was not sufficiently general to design devices of practical interest, and FAMING is the first program to allow goal-directed design of mechanism part shapes. It could be integrated with systems for conceptual design of kinematic chains[5]. The techniques can be generalised to other domains where qualitative models have been investigated: thermodynamics[34], nonlinear control systems[35], electronic circuits[6] as well as many other engineering problems[36].

FAMING is to our knowledge the first system to demonstrate the use of case *adaptation* and *reinterpretation* for innovative and even creative design. This shows that the case-based design paradigm is not limited to routine design problems, as is often assumed. In fact, in the domain addressed by FAMING, case-based design proves to be a better tool for supporting innovation than methodologies such as model-based design which are intended for innovative design but are impractical because of computational intractability. This is due to two reasons. First, cases provide coherent starting points for design solutions and thus avoid much of the computational complexity imposed by geometric considerations. Second, and maybe more importantly, cases provide a communication framework where the designer can bring his extensive knowledge and intuitions to bear on the design process. Other work on architectural design[37] has lead to similar conclusions and make us believe that the combination of case- and model-based reasoning implemented in FAMING is a promising way to make AI techniques work for real-world design problems.

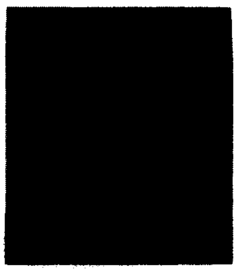
While the FAMING system is already useful for mechanism designers, the following additional work would remain to be done to make it a commercial tool. FAMING only considers qualitative kinematic function of devices, leaving aside considerations of dynamics and friction. These can be integrated by providing the appropriate qualitative and quantitiative models, and we have verified that at least the models used by watch designers can be integrated without great difficulty. Second, FAMING currently only deals with polygonal shapes. To handle most practical devices, the range of admissible shapes will have to at least include circular arcs. This extension had already been implemented in the kinematic modelling system of Faltings[19]. The conceputal design at the qualitative level has to be complemented by an optimisation which takes into account wear, material strength and manufacturing constraints. Since the qualitative behaviours are already correct, the problem will be convex and linearisable; there are many reliable techniques for solving this problem. The biggest obstacle for practical use is that the designer has to formally specify device function. Even if many common components of function can be preformulated, writing formal specifications for completely novel devices requires a high degree of abstraction which not all designers are comfortable with.

REFERENCES

1 Nadel, B A and Lin, J 'Automobile transmission design as a constraint satisfaction problem: first results' *in* Tong, C and Sriram, D (eds) *Artificial Intelligence in Engineering Design* Academic (1992)

2 Williams, B 'Intersection-based invention: designing novel devices from first principles' *Proc. National Conference on Artificial Intelligence* Boston (1990)

3 Neville, D and Weld, D S 'Innovative design as systematic search' *Proc. AAAI* Washington (1993)
4 Sycara, K and Navinchandra, D 'Index transformation techniques for facilitating creative use for multiple cases' *Proc. 12th IJCAI* Sydney (1991)
5 Subramanian, D 'Conceputal design and artificial intelligence' *Proc. 13th Int. Joint Conference on Artificial Intelligence* Chambery, France (1993)
6 DeKleer, J and Brown, J S 'A qualitative physics based on confluences' *Artificial Intelligence* Vol 24 (1984)
7 Joskowicz, L and Addanki, S 'From kinematics to shape: an approach to innovative design' *Proc. AAAI* St. Paul (1988)
8 Gero, J 'Design prototypes: a knowledge representation schema for design' *AI Mag.* Vol 11 No 4 (1990)
9 James, M 'Prototyping CAD' *CAD/CAM* (1991)
10 Wagner, M R *Understanding the ICAD System* ICAD Inc. (1990)
11 Kolodner, J L *Case-based Reasoning* Morgan-Kaufmann (1993)
12 Pu, P (Ed.) 'Special issue on case-based design' *AI EDAM* Vol 7 No 2 (1993)
13 Hua, K and Faltings, B 'Exploring case-based building design — CADRE' *AI EDAM* Vol 7 No 2 (1993)
14 Gero, J S, Lee, H S and Tham, K W 'Behaviour: a link between function and structure in design' *in* Brown, D, Waldron, M and Yoshikawa, H (eds) *Intelligent Computer-Aided Design* North Holland (1992)
15 Goel, A K 'A model-based approach to case adaptation' *Proc. Annual Conference of the Cognitive Science Society* (1991) pp 143–148
16 DeKleer, J and Weld, D (eds) *Readings in Qualitative Reasoning about Physical Systems* Morgan-Kaufmann (1990)
17 Wills, L M and Kolodner, J L 'Towards more creative case-based design systems' *Proc. 12th National Conference of the AAAI* Morgan-Kaufmann (1994)
18 Sun, K and Faltings, B 'Supporting creative mechanical design' *in* Gero, J and Sudweeks, F (eds) *Artificial Intelligence in Design '94* Kluwer Academic (1994)
19 Faltings, B 'Qualitative kinematics in mechanism' *Artificial Intelligence* Vol 44 No 1 (1990)
20 Sacks, E and Joskowicz, L 'Automated modeling and kinematic simulation of mechanisms' *Comput.-Aided Des.* Vol 25 No 2 (1993)
21 Forbus, K, Nielsen, P and Faltings, B 'Qualitative spatial reasoning: the CLOCK project', *Artificial Intelligence* Vol 51 No 3 (1991)
22 Nielson, P 'A qualitative approach to rigid body mechanics' *PhD Thesis* University of Illinois (1988)
23 Forbus, K 'Qualitative process theory' *Artificial Intelligence* Vol 24 (1984)
24 Chakrabarti, A, Holden, T and Blessing, L 'Representing function in design', *Workshop at the 3rd International Conference on AI in Design* Lausanne (1994)
25 Sembugamoorthy, V and Chandrasekaran, B 'Functional representation of devices and compilation of diagnostic problem-solving systems' *in* Kolodner, J and Riesbeck, C (eds) *Experience, Memory and Reasoning* Lawrence-Earlbaum Associates (1986)
26 Iwasaki, Y, Fikes, R, Vescovi, M and Chandrasekaran, B 'How things are intended to work: capturing functional knowledge in device design' *Proc. 13th Int. Joint Conference on Artificial Intelligence* Chambery (1993)
27 Umeda, Y, Takeda, H, Tomiyama, T and Yoshikawa, Y 'Function, behaviour and structure' *in* Gero, J S (ed.) *Applications of Artificial Intelligence in Engineering V* Vol 1 Springer-Verlag (1990)
28 Faltings, B, Baechler, E and Primus, J 'Reasoning about kinematic topology' *Proc. 11th Int. Joint Conference on Artificial Intelligence* Detroit (1989)
29 Faltings, B 'A symbolic approach to qualitative kinematics' *Artificial Intelligence* Vol 56 No 2 (1992)
30 Selman, B, Levesque, H and Mitchell, D 'A new method for solving hard satisfiability problems' *Proc. 10th National conference of the AAAI* Morgan-Kaufman (1992)
31 Minton, S, Johnston, M, Phillips, A and Laird, P 'Minimising conflicts: a heuristic repair method for constraint satisfaction and scheduling problems' *Artificial Intelligence* Vol 58 (1992)
32 Pu, P and Purvis, L 'Formalizing case adaptation in a case-based design system' *in* Gero, J and Sudweeks, F (eds) *Artificial Intelligence in Design '94* Kluwer Academic (1994)
33 Haroud, D and Faltings, B 'Global consistency for continuous constraints' *Proc. European Conference on Artificial Intelligence* Wiley (1994)
34 Forbus, K D and Whalley, P B 'Using qualitative physics to build articulate software for themodynamics education' *Proc. National Conference of the AAAI* Morgan-Kaufman (1994)
35 Zhao, F 'Extracting and representing qualitiative behaviour of complex systems in phase space' *Artificial Intelligence* Vol 69 No 1 (1994)
36 Milne, R and Travé-Massuyes, L 'Qualitative reasoning for design and diagnosis applications' Tutorial *National Conference of the AAAI* Washington (1993)
37 Hua, K, Smith, I, Faltings, B, Shih, S and Schmitt, G 'Adaptation of spatial design cases' *in* Gero, J (ed.) *Artificial Intelligence in Design '92*, Kluwer (1992)



Boi Faltings received his diploma from ETH Zurich, Switzerland and his PhD from the University of Illinois, USA. He is Professor of Computer Science at the Swiss Federal Institute of Technology in Lausanne, where he directs the Artificial Intelligence Laboratory. His main research interests are in the areas of qualitative reasoning, constraint satifaction, reasoning with abstractions, and applications in intelligent computer-aided design.

Kun Sun received an MSc in Computer Science from the Northwestern Polytechnical University in 1985 and a PhD from EPFL in 1994. His main research interests are in knowledge-based systems, particularly applied to design.