

## Dimension-driven Geometry in CAD: A Survey

Dieter Roller, François Schonek, Anne Verroust

► **To cite this version:**

Dieter Roller, François Schonek, Anne Verroust. Dimension-driven Geometry in CAD: A Survey. Theory and Practice of Geometric Modeling, 1989, <10.1007/978-3-642-61542-9\_32>. <[http://dx.doi.org/10.1007/978-3-642-61542-9\\_32](http://dx.doi.org/10.1007/978-3-642-61542-9_32)>. <hal-01224960>

**HAL Id: hal-01224960**

**<https://hal.inria.fr/hal-01224960>**

Submitted on 5 Nov 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Dimension-driven Geometry in CAD : A Survey \*

Dieter Roller, François Schonek ‡ and Anne Verroust ‡  
HEWLETT-PACKARD GmbH,  
Postfach 1430, D-7030 Böblingen, F.R.G.  
‡LIENS, Ecole Normale Supérieure,  
CNRS URA 1327  
45 rue d'Ulm, 75230 PARIS CEDEX 05, FRANCE

## Abstract

In this paper we present different approaches occurring in recent literature to solve the problem of parametrizing mechanical designs by dimensions. We describe the approaches showing their advantages and their drawbacks. We conclude that the reviewed approaches each have different strengths. It is proposed that a method should be investigated which would combine valuable properties of the AI approaches along with the benefits of the global approach.

## 1 Introduction

We are interested here in an important problem occurring in CAD : the introduction of automated geometric variation in CAD systems for mechanical design. More precisely, we investigate the methods which try to handle the following problem :

given a model with a sufficient number of geometric constraints and a topological or approximate geometric description, we want the precise model to be evaluated automatically. The constraints allowed are principally the metric relationships between pairs of geometric entities : distance between two points or between parallel lines, angle between lines, etc....

It is easy to see the possible applications for such a tool. For example :

during the conception of mechanical designs, this allows a component part to be designed by successive trials and frees the designer from a time-consuming part of the computation.

it also allows a family of objects to be created and stored as derivatives of the original part. This means only dimensional parameters need to be stored for the derivatives.

The following approaches differ from one another in the type of dimensional constraints allowed, in the way they compute the solution and in the detection of subconstrained or overconstrained parts of the drawing or model.

---

\*This work has been supported in part by HEWLETT-PACKARD GMBH

Note that to rigidify a planar figure having  $n$  vertices,  $2n - 3$  constraints (length or angle constraints for example) are necessary (a point and a direction are supposed fixed). But this number is not a sufficient condition to eliminate the subconstrained or the overconstrained figures because if a part of the figure is overconstrained and the rest of it subconstrained, we can still obtain the correct total number of constraints for the whole figure ( cf. Figure 1 ).

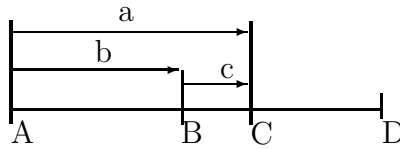


Figure 1: The position of C is overconstrained and D is not constrained.

An “ideal” method should satisfy at least the following important requirements :

1. it should detect the non-consistent constraining schemes, i.e. the under- and over-constrained cases
2. it should be reliable. Given a set of constraints and the description of the topology of the object, the method should give a solution if there is one and give all the possible solutions if the user asks for them
3. it should be interactive. This means it should be fast enough to provide appropriate feedback to the user at every step of the design
4. it should allow modifications of constraints during the construction of the object and the effect of the modification should be independent of the order of insertion of the previous constraints
5. it should allow a wide range of types of constraint and be easily adaptable to additional new types of constraint (for particular applications)
6. it should not be restricted to either 2D or 3D
7. it should be able, even with user interaction, to handle drawings coming from conventional CAD data bases

We shall classify the approaches investigated into three families : the primary, the algebraic and the A.I. approaches. The detailed description of these methods will be the subject of the following sections.

## 2 The Primary Approaches

We call them “primary approaches” because the types of constraint allowed are limited and the methods used to solve the problem are based on well understood algorithms. All the methods described in this section detect the overconstrained schemes.

## 2.1 Fitzgerald's Approach

Fitzgerald [6] follows Requicha's approach [12] to solve the problem of overconstrained figures. He uses the fact that a set of length constraints along a given axis can be structured in a tree if no point is overconstrained along this axis, and the fact that this set generates a cycle in the other case (cf. Figure 2). Thus Fitzgerald's approach only allows the specification of horizontal or vertical lengths between two points. The special case of the circle can be represented using this type of constraint. Fitzgerald reasons

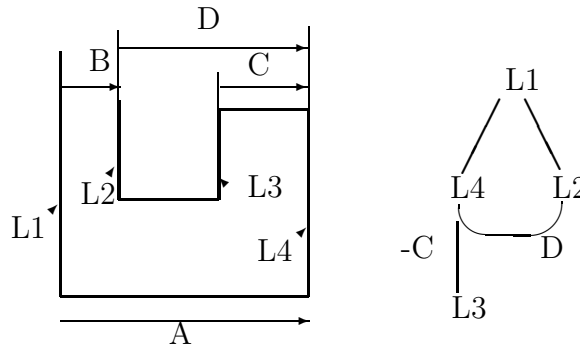


Figure 2: The set of constraints and the corresponding graph ( Fitzgerald )

identically along the horizontal and vertical axes. Thus, with this hierarchical structure, the set of subconstrained points along one of the axes is known at any time and the emergence of an overconstrained point can be detected.

This approach is simple and safe. The restriction introduced on the type of allowed constraints enables easy checking of the overconstrained points. This solution is very useful for industrial drawings having only vertical or horizontal constraints. However the model cannot be extended to other types of constraint because the simplicity of the situation relies on the choice of only horizontal and vertical constraints.

## 2.2 3D Constructive Approach

In [7], Gossard, Zuffante and Sakurai describe a 3D design system managing dimensions and tolerances. The way the objects are created determines the possible variations in dimensions.

Dimensional constraints are represented by relative position operators. These operators use the dimension value to position one face relative to another. Relative position operators introduced are parallel-distance and intersection-angle. Extensions to other operators that would represent other types of dimensional constraints are possible.

The geometric models are described in a new way by combining CSG and boundary representation ( B-rep ). This representation is based on an object graph consisting of nodes and branches (cf. Figure 3). In fact the solids are defined by unions and intersections of half spaces (delimited by planes). In addition, they are assembled using relative position operators.

They introduce the following innovation :

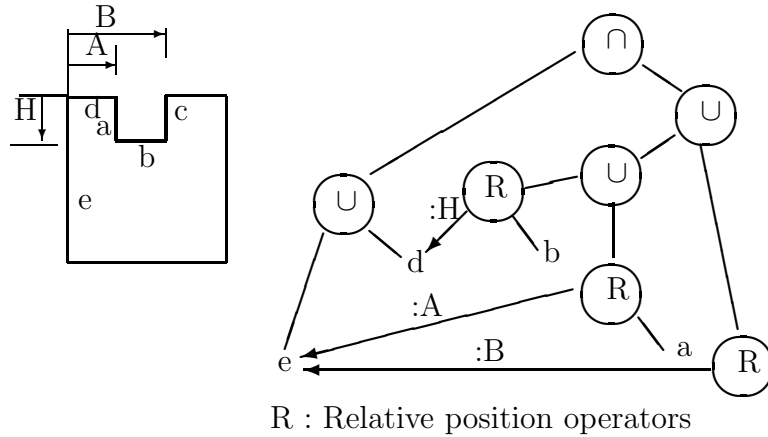


Figure 3: Example of object graph ( Gossard, Zuffante, Sakurai )

- if a change of value modifies the topology of the object, the system accepts it and automatically rebuilds the object with its new topology.

This approach is also interesting in that it can be applied directly to solid modeling, an important tool in CAD. It is difficult to see though how the object graph can be generated automatically for objects previously stored in conventional CAD data bases.

### 2.3 Macro-programming

Cugini, Devoti and Galli [2] introduce languages for the definition and the manipulation of parametrized figures.

The figures are defined by a graph called “Relations graph” (cf. Figure 4) where

- the vertices represent the lines and the points of the drawing and
- the edges represent the topological and dimensional relations.

Some of the possible relations between points and lines are :

- a point belongs to another point or line
- a point is at a distance  $x$  from another point or line
- a line is tangent to another line
- a line passes through a point
- a line makes a given angle with another line...

The same information can be translated into the “Data Definition Language”.

To manipulate relations graphs easily, they introduce the concept of a “macro”. A macro is a structure associating a key to a relations graph. They use rotations, symmetries, scaling and parameter modifications to assemble the macro images. To obtain the graphic construction corresponding to these macros, they use the “graphic parametric builder”. The function of the graphic parametric builder is to receive macros and parameters as input and return a macro.

The way the dimensional relations between points or lines are computed to obtain the whole figure is not explained and no reference is given on the subject. Their interest lies in the introduction of these macro-images and in the Data-Base oriented approach to the storage problem.

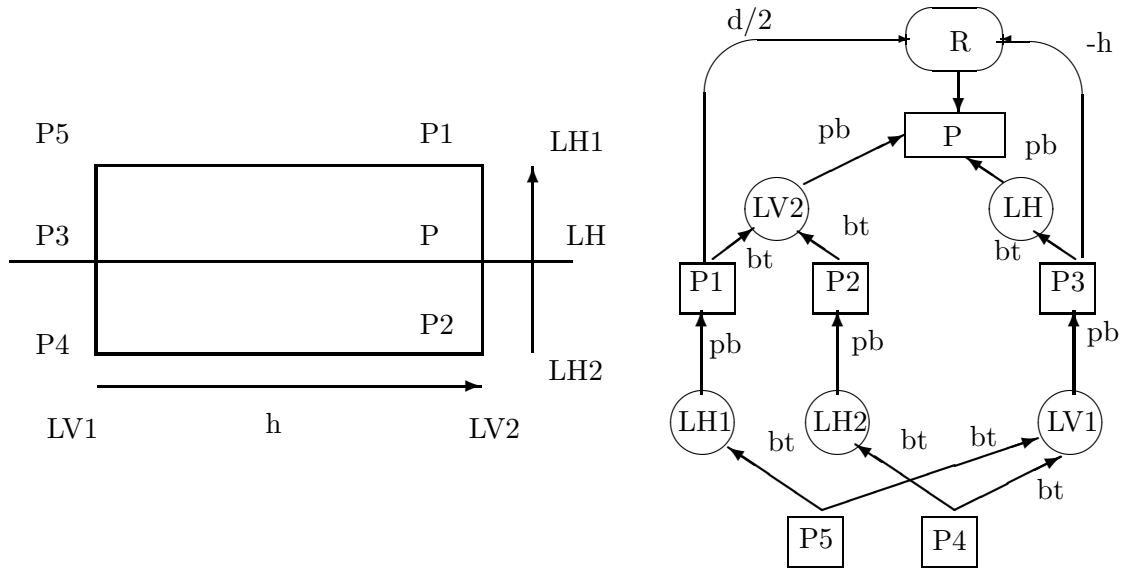


Figure 4: Example of a relations graph ( Cugini, Devoti and Galli )

### 3 Algebraic Approaches

#### 3.1 The Global Approach

The principle of this type of method is to translate every constraint into an equation. The problem is then the resolution of this system of (non-linear) equations where the unknowns are the coordinates of the characteristic points of the model. Several versions of this approach ( [4], [5], [9], [10] and [8] ) have been proposed. We will describe Light's method and Chyz's preprocessing approach [3] in more detail. The main interest found in these types of approaches is the range of possible constraints :

- distance between two points
- angular constraint
- distance between two parallel lines
- distance along an axis
- equal linear distance constraint
- area constraint, etc....

In fact, if a constraint can be expressed by an equation of the coordinates of the characteristic points, this constraint can be added. It is obvious therefore, that this approach is applicable to 2D and 3D problems.

The principle of the resolution of the system is mainly based on the Newton-Raphson iterative method. This method is an extension of Newton's method.

$$\text{Let } F(X) = 0 \text{ be the system, with } X = (x_i)_{i \leq m} , F = (f_j)_{j \leq m}.$$

The sequence  $(X^n)$ , with  $X^0$  chosen arbitrarily, is built as follows :

$$X^{n+1} = X^n - [F'(X^n)]^{-1} F(X^n)$$

with  $F'(X^n) = (\frac{\partial f_i}{\partial x_j}(X^n))_{i,j \leq m}$

Let  $\Delta X$  be equal to  $X^{n+1} - X^n$ .

$$\text{Then, } \begin{cases} F'(X^n) \Delta X &= -F(X^n) \\ X^{n+1} &= \Delta X + X^n \end{cases}$$

we have :

$$J \Delta X = F, \text{ J being the jacobian matrix.} \quad (1)$$

To avoid divergence, the computation of the solution is made in several steps. For example, if we want to change a dimension from 10 to 20, this will be made by successively computing the solutions for 15, 17, 18 and 19 before 20.

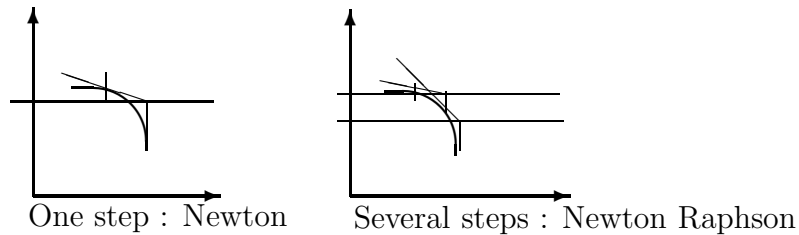


Figure 5: Graphical description of Newton-Raphson iterative method

Some improvements have been proposed to solve the particular cases :

- When the Jacobian matrix is singular, the Newton-Raphson method is not applicable directly. In these cases, Light uses a modified version of Doolittle's method which solves (1) for some solutions.

The principle of Doolittle's method to solve a linear equation  $Ax = b$ , with A non singular, is to make a LU decomposition :

$$\begin{cases} A &= LU \quad \text{where L is lower triangular with unity elements} & (2) \\ Ly &= b \quad \text{on the diagonal and U is upper triangular} & (3) \\ Ux &= y \quad \text{with non zero elements on the diagonal.} & (4) \end{cases}$$

When A becomes singular, the modified Doolittle's method can give L and U, but only (3) and (4) remain valid and U is different. For each subconstrained variable there is a zero column and for each linearly dependent equation a zero row in U. The unconstrained parameters are easily detected. By setting them to zero, it is then possible to solve the system for the other unknowns.

- When only some of the variables have been modified Lin, Light and Gossard [10] propose an algorithm to isolate the minimum subset of the constraints affected by this change.

This algorithm is divided in two processes :

1. a backward propagation to create a one-to-one bound coordinate constraint equation relationship.

2. a forward propagation to determine sensitive coordinates and equations of the given dimension changes.

The power of this global approach resides in the large range of dimensional constraints that can easily supported. Another important fact is that cyclic constraint situations can be handled through the simultaneous solution of the constraint equations.

The drawback of this approach is :

- that the result is an approximation of one of the possible solutions. In fact, the solution obtained is intrinsically dependent on the “arbitrary” value chosen for  $X^0$ . Figure 6 illustrates this problem. Thus, by a bad choice of the first value, the method can converge to an unwanted solution and, what is more problematic, the solution may change, with no possibility to control it, according to the different numerical values entered.

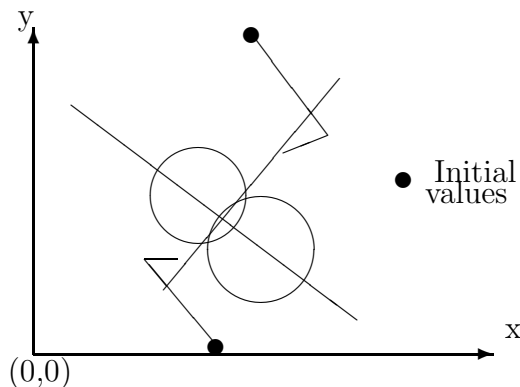


Figure 6: Importance of the choice of the first value

### 3.2 Chyz’s Preprocessing Approach

To solve the first requirement mentioned in the introduction, Chyz [3] proposes a pre-process for this method. His solution is a 2D approach but can be extended to the 3D problem. He introduces a graph structure to verify whether a drawing is “well” constrained or not (e.g. if there are enough constraints w.r.t. the number of unknowns in all the parts of the drawing).

Chyz’s model is based on the following remark : if the points and the lines present in the figure are the vertices of the graph and if the geometrical and the topological relations form the oriented edges of the graph, then, when the drawing is stiffened, exactly two edges end at each vertex of the graph. This is illustrated by Figure 7. For the special case of the circle, three edges are necessary : the first one gives the radius and the others fix the position of the center of the circle.

Chyz proposes an algorithm which translates assembly operation between figures into a graph operation. The graph corresponding to the assembly is the union of the two graphs (the “master” and the “slave”) in which the direction of some edges has been reversed and one of the edges fixing the “slave” graph has disappeared.

This method leaves the following points open :



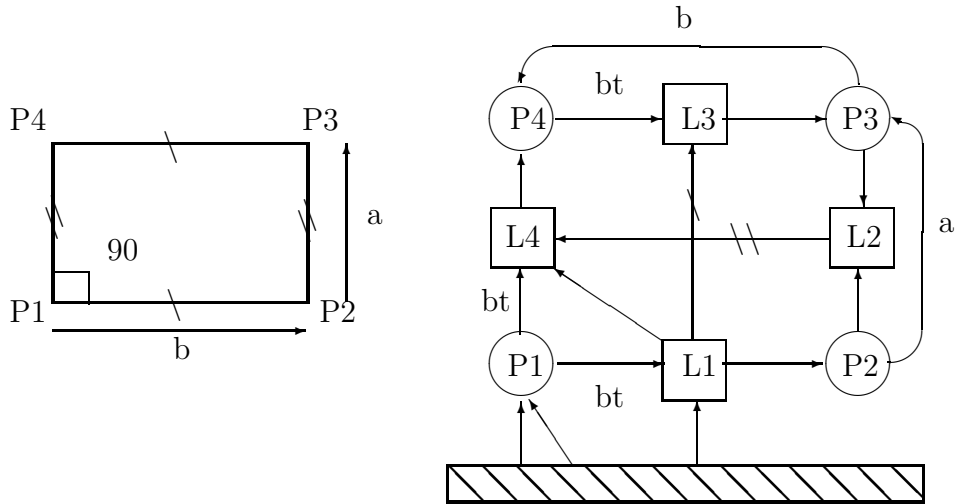


Figure 7: Graph corresponding to a rectangle ( Chyz )

- The control of such a graph structure seems to be expensive in time and gives reasons for doubt about the interactivity of the whole process.
- The introduction of this graph structure does not solve all the cases of singularity : Figure 8 shows for example an ill conditioned case that would not be detected by Chyz's method.

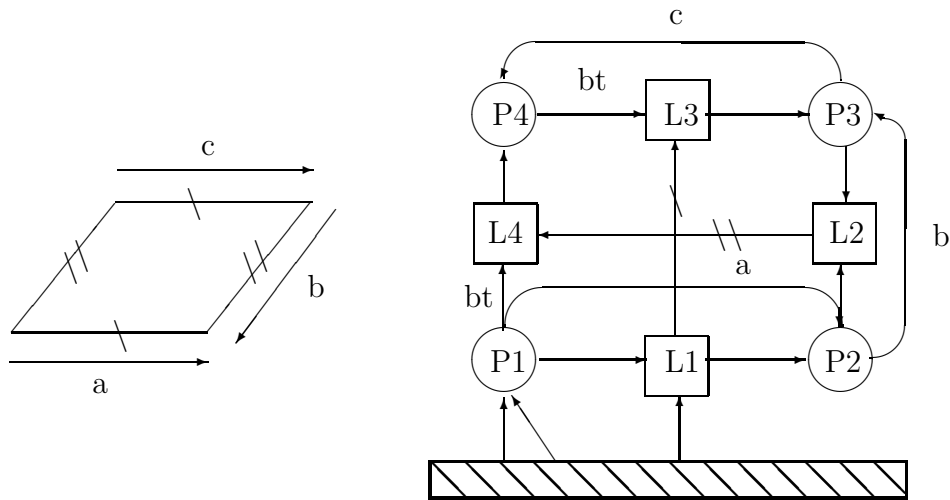


Figure 8: The Jacobian is singular and the graph is correct (Chyz)

## 4 The Artificial Intelligence Oriented Approaches

We have found three methods using inference to parametrize CAD mechanical designs. Sunde has proposed two different methods [13], [14], [15] and Aldefeld has another solution [1]. As yet these methods solve only the 2D problem.

## 4.1 First Approach : PICME System

In [13] Sunde describes a first method, using an expert system to serve as a substitute for the constructions made by the industrial designers in their drawings. The rules introduced are the translation of what a designer does to draw the figure. For example, to define a triangle by its three sides the system will do the geometrical construction, giving the third point by intersecting two circles. Let us describe the method more precisely :

The topological description of the figure and the constraints on it are represented by “constraining relations” between objects. The objects considered are the points, the line segments, the arcs and the circles present in the figure.

We have the following set of relations :

horizontal(ls)	parallel(ls1,ls2)	Where pt is a point, ls a line segment, ar an arc, cr a circle, ds a distance, and ag an angle.
vertical(ls)	orthogonal(ls1,ls2)	
reference(pt)	tangent(ls1,ar1)	
equal(ag1,ag2)	adjacent(ls1,pt1)	
direction(ls1,ag1)	distance(pt1,pt2,ds1)	
arc-angle(ar1,ag1)	symmetric(cr1,cr2,ls1)	
radius(ar1,ds1)	angle-between-lines(ls1,ls2,ag1)	
diameter(ar1,ds1)	directed-distance(pt1,pt2,ds1,dr1)	

These relations are translated into properties, premisses of the rules of the expert system. Let us consider how it works in this example (cf. Figure 9) :

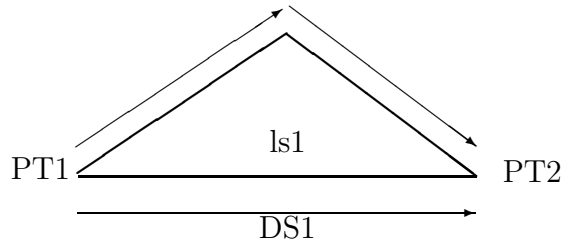


Figure 9: Construction of a triangle ( PICME )

horizontal(ls1)	⇒	known-direction(ls1)
reference(PT1)	⇒	known-location(PT1)
known-location(PT1)		
and distance-relation(PT1,PT2,ds1)		
and known(DS1)	⇒	known-to-be-on-circle(PT2)
known-location(PT1)		
and topological-relation(PT1,ls1)	⇒	known-start-location(ls1)
known-start-location(ls1)		
and know-direction(ls1)		
and topological-relation(PT2,ls1)	⇒	known-to-be-on-line(PT2)
known-to-be-on-circle(PT2)		
and known-to-be-on-line(PT2)	⇒	known-location(PT2)

A system according to this method provides a drawing only when it has been completely defined.

Some points of the algorithm are not explained :

- how the overconstrained parts of the drawing are detected
- what is done when several drawings correspond to the same set of constraints.

This method has been implemented in a commercial product.

## 4.2 Aldefeld's Approach

Aldefeld also uses an expert system in his proposed approach [1]. The set of constraints he handles is the same as in the PICME system.

The drawings are computed in two phases :

- During the first phase, a forward inferencing is used to propagate the constraints introduced by the user. This propagation is symbolic and is used to build a plan for the second phase of the computation.
- In the second phase, a point and a direction are fixed and the numerical values are computed. If they correspond to constraints entered in the first phase, the construction plan is used for the numerical computation. In the other case, the construction plan is modified using the corresponding history and if no overconstraint is detected, the numerical values are computed.

Aldefeld introduces these two phases to reduce the accuracy problems, to accelerate the computation of solutions when a large number of shapes very close to each other are drawn, and to give an explanation to the user when a point is overconstrained. The first phase can be viewed as a preparation and a control phase before the entire computation of the figure.

Although he recognizes that his solution is expensive in computation time (the system requires 10 minutes of CPU time to set up the construction plan for a figure with only 20 characteristic points), he points out important requirements for such methods and proposes solutions to most of them.

- he ensures that, for the same set of constraints, two different plans of construction, corresponding to two different orderings of constraints, lead to the same drawing, when a constraint is modified in their second phase.
- he handles the resolution of ambiguities during the computation of the drawing as follows : if the wanted solution is close to a given figure, when an execution process induces more than one solution, all the different solutions are computed and criteria like the similarities of the sequences of points or the fact that a point is located on the left or right side of a line, are used to choose the solution.
- he proposes a solution to handle global constraints, such as area constraints :
  - in the first phase, a construction plan is generated for a complete set of linear dimensions (i.e. global constraints are temporarily replaced by others)
  - and in the second phase, a dimensional value is relaxed and the construction plan is iterated until the global constraint is reached.

This solution may be expensive in computation time if the value of the desired constraint is very different from the first value computed.

We should make reservations to some of Aldefeld's choices :

- Before such a method can be commercially viable, it must be shown that phase 1 of the method can be drastically accelerated.
- Aldefeld does not explain much about the rules used and the computation. Nevertheless, this method seems to resemble the PICME model in the sense that the position of the characteristic points are sequentially computed from a fixed point and a flexible line.
- Furthermore, the decomposition in two phases is of interest only when a sequence of similar drawings have to be processed. The first phase does not detect the numerical cases of singularity, even though the cases of parallelism and orthogonality are handled (Aldefeld considers the parallelism constraint as a special constraint which is not a numerical case of an angular constraint).

### 4.3 Sunde's Second Approach

Rather interesting is the second model introduced by Sunde [14]. This method is also based on a A.I. approach, but it allows the user to obtain a drawing even when it is incompletely specified. Furthermore this model seems to be really interactive :

- the designer can see the evolution of his drawing every time he adds a new constraint ( cf. Figure 10 ). All the possible solutions triggered by the last rule will be displayed by the system.

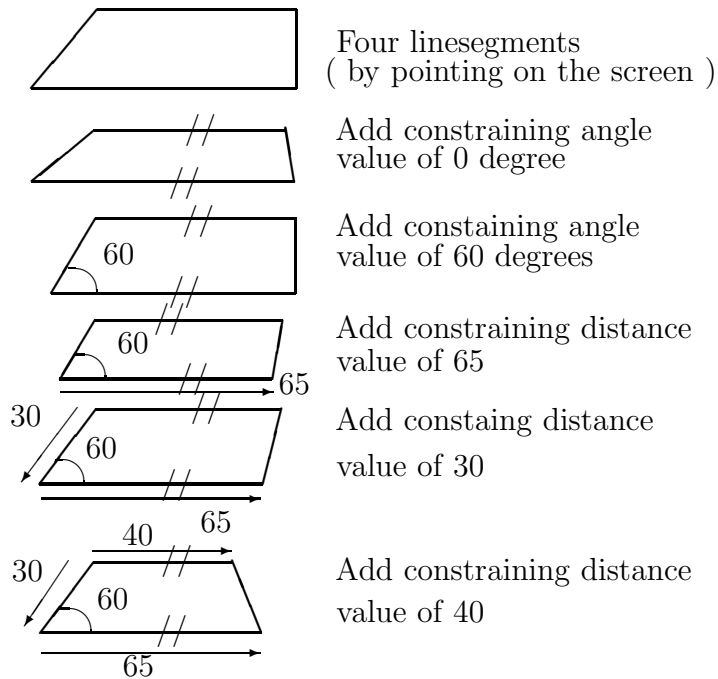


Figure 10: Example of an interactive sequence

- overdetermined cases are detected; in these cases the system stops and an explanation is provided to the user.
- during the session, the user can replace a constraint by another one. Then the system recomputes and redraws the figure ( cf. Figure 11).

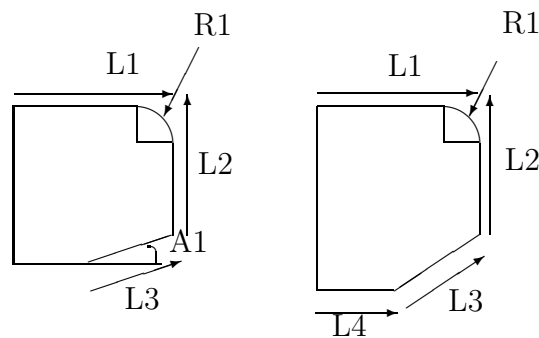


Figure 11: Modification of constraint

Let us see in more detail how this system works :

Two data types are introduced :

- CA-set : a set of segments with mutually constrained angles

- CD-set : a set of points with mutually constrained distances

Intuitively, each CD-set is a set of points entirely fixed in their own frame. Then the drawing is completely determined when a CD-set contains all the unknown points. Thus with this representation, we know at each step what part of the drawing is partially or totally constrained and how it is constrained.

The constraints specified by the user are translated into CA-set or CD-set as follows :

- when a distance is added, a CD-set containing the two points concerned is added.
- when an angle is added, a CA-set containing the two segments concerned is added.

The production rules work on these sets and combine them into larger ones. For example we have :

- If two CD-sets contain a common point and if the angle between them at this point is constrained, then the two CD-sets are combined into one CD-set.
- If two CA-sets have a segment in common, then they are combined into one CA-set.

With knowledge of these CD-sets and CA-sets, we know if the distance between two points is constrained (in this case, there exists a CA-set containing these two points) and if the angle between two segments is constrained (there exists a CA-set containing the two segments). Thus it is obvious to verify whether a particular angle (or distance) value is known or not. These rules are introduced to detect the overdetermined cases.

Sunde keeps record of the creation and growth of the CA-sets and the CD-sets. This is used :

- to accelerate the computation of the figure when a constraint has been modified
- and to produce an explanation when the user tries to constrain a distance or an angle that already has been constrained implicitly.

Sunde justifies the choices taken in his second approach :

- the number of facts needed for a pure deduction system are reduced. By reasoning on the CA and CD sets level and with the history, he avoids the storage of an important number of intermediate facts. He delays the computation of some values until the time they are needed. This approach is also useful for handling the modifications of constraints quickly.
- the verification is easily made by using simple rules of the production mechanism.

Although Sunde does not give the crucial rules to be used in this method (three induction rules on CA and CD sets among nine or ten needed are explained), some of them such as the rules related to the particular cases of the triangle can be easily added. Thus, this last approach seems to be a very promising one and a 3D extension should be possible. However, it could be difficult to handle the addition of a new type of constraint in the system : if a new type of constraint is added all the deduction rules have to be reconsidered and new rules involving this type of constraint have to be added. All the geometric cases have to be investigated and the task may be non-trivial.

## 5 Conclusion

Let us now examine how the different methods satisfy the requirements of the “ideal” solution mentioned in the introduction. We summarize their behavior in the following table (the numbers correspond to the order of the requirements given in the introduction).

requirements	Fitzgerald [6]	Gossard Zuffante Sakurai [7]	algebraic + Chyz [9] [3]	PICME [13]	Aldefeld [1]	Sunde [14]
(1) consistency	yes	yes	incomplete	yes	yes	yes
(2) reliability	yes	yes	no	yes	yes	yes
(3) interactivity	yes	yes	?	yes	no	yes
(4) modifications	yes	no	no	no	yes	yes
(5) extensionability	no	?	yes	?	?	?
(6) 2D-3D	possible	yes	yes	possible	possible	possible
(7) conventional data	no	no	no	no	nearly	no

Of course there is no best solution. The choice of one solution or another rather should depend on the priority of the requirements and the range of constraints in a particular system environment.

In our opinion though, a new method should be investigated which incorporates the properties of the AI approaches with the benefits of the global approach. Such a combined method could more closely meet the requirements outlined than any other of the approaches reviewed in this paper.

## References

- [1] Aldefeld, B., “Variation of geometries based on a geometric-reasoning method”, CAD, vol. 20,no. 3, April 1988.
- [2] Cugini, U., Devoti, C. and Galli, P. “System for Parametric Definition of Engineering Drawings”, MICAD’85.
- [3] Chyz W., “ Constraint Management for CSG ”, Master Thesis, MIT, June 1985.
- [4] Hillyard, R. and Braid I., “ Analysis of Dimensions and Tolerances in Computer-Aided Mechanical Design”, CAD, vol. 10, no 3, May 1978.

- [5] Hillyard, R. and Braid I., “ Characterizing non Ideal Shapes in Terms of Dimensions and Tolerances ”, Computer Graphics, vol. 12, no 3, August 1978.
- [6] Fitzgerald W., “Using Axial Dimensions to Determine the Proportions of Line Drawings in Computer Graphics”, CAD, vol. 13, no 6, November 1981.
- [7] Gossard, D., Zuffante, R. and Sakurai, H., “ Representing Dimensions, Tolerances and Features in MCAE Systems”, IEEE Computer Graphics and Applications, March 1988.
- [8] Lee, K. and Andrews, G., “Inference of the Positions of Components in an Assembly : Part 2”, CAD, vol. 17, no 1, January 1985.
- [9] Light R., “ Symbolic Dimensioning in Computer-Aided design”, Master Thesis, MIT, May 1979.
- [10] Light, R., Lin, V. and Gossard D., “ Variational Geometry in CAD”, Computer Graphics, vol. 15, no 3, August 1981.
- [11] Light R. and Gossard, D., “ Modification of Geometric Models through Variational Geometry”, CAD, vol. 14, no 4, July 1982.
- [12] Requicha, A. “Dimensioning and tolerancing”, PADL TM-19 Production Automation Project University of Rochester, May 1977.
- [13] Sunde, G., “Specification of Shape by Dimensions and other Geometric constraints”, IFIP WG. 5.2 on Geometric Modeling, Rensselaerville,NY, May 1986.
- [14] Sunde G., “A CAD System with Declarative Specification of Shape ”, EuroGraphics Workshop on Intelligent CAD Systems, April 21-24 1987, Noordwijkerhout, The Netherlands.
- [15] Sunde G. and Kallevik V., “ A Dimension-driven CAD system. Utilizing AI Techniques in CAD. ”, December 1987, Senter for Industriforskning, report no 860216-1.