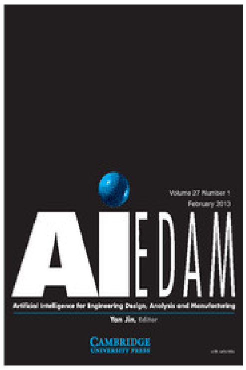


Prototype system for supporting the incremental modelling of vague geometric configurations

X. Guan, A.H.B. Duffy and K.J. MacCallum

Artificial Intelligence for Engineering, Design, Analysis and Manufacturing / Volume 11 / Issue 04 / September 1997, pp 287 - 310
DOI: 10.1017/S0890060400003231, Published online: 27 February 2009

Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1997), *11*, 287–310. Printed in the USA.

Prototype system for supporting the incremental modelling of vague geometric configurations

X. GUAN,[1] A.H.B. DUFFY,[2] AND K.J. MACCALLUM[2]

[1]Industrial Systems and Control Ltd., 50 George Street, Glasgow G1 1QE, U.K.

[2]CAD Centre, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, UK

(RECEIVED April 1, 1996; ACCEPTED November 18, 1996; REVISED January 15, 1997)

Abstract

Few existing Computer Aided Design (CAD) systems provide assistance to designers in developing geometric concepts at the early design stages. Instead they require a high level of precision and detail suited to detail design. To support the early geometric design, a CAD system should provide utilities for the rapid capture and iterative development of vague geometric models. This paper presents a pilot system that is being developed based on such a vision. The system has adopted minimum commitment modelling and incremental refinement as the guiding principles. The representation of geometric configuration is based on a parametric and constraint-based geometric design model, and provides a uniform representation of the approximate and precise size and location parameters. A constraint-based mechanism has been developed for processing geometric information. The use of the system in assisting the development of a geometric configuration is also demonstrated. Finally, features and limitations of the system as well as relations to relevant works are discussed, and based on this a number of key research directions are established.

Keywords: Vague Geometric Modelling; Early Design Support Systems; Geometric Configuration; CAD; Geometric Design

1. INTRODUCTION

Early stages of design is characterized by "back-of-the-envelope" design activities that involve, among others, the rapid and iterative generation, exploration and evaluation of geometric concepts. These concepts describe the various *geometric configurations* of a product, each of which consists of the rough or precise geometry of the constituent objects and their spatial or topological relationships in forming the product. Only the most suitable concepts are chosen and developed into full models suitable for downstream processes.

Geometric design information is the set of facts that are specified and used to describe or derive the geometric properties or concepts of a product. Early geometric information may be classified into four types including shape, size, location, and orientation (Guan & MacCallum, 1995). The shape of an object may be described as 1D, 2D, or 3D generic primitives or defined through specific design features. The size of an object may be given in the form of, usually, linear or nonlinear inequalities, equalities as well as ranges, relations among various design parameters such as width, depth, etc. The location and orientation of an object may be described explicitly in spatial relationships or, more often, expressed implicitly in design sketches in relation to other objects or a chosen datum.

A distinctive characteristic of the early geometric information and the corresponding geometric configuration concepts is their associated *vagueness*. As an inherent part of a process of evolving ideas from abstract to concrete, such vagueness reflects a designer's desire to explore the overall form of a concept, to illustrate abstract concepts such as function, or to illustrate concepts in ways that give economy of effort. It may also reflect lack of knowledge or certainty of some aspects of the geometry at these stages of the design.

The current generation of Computer Aided Design (CAD) systems are in general incapable of representing and manipulating such vague geometric information encapsulated in early design sketches. They usually require a level of pre-

Reprint requests to: Dr. A.H.B. Duffy, Department of Design, Manufacture and Engineering Management, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, UK. Phone: +44 141 548 3134; Fax: +44 141 552 3148; E-mail: alex@cad.strath.ac.uk.

cision and detail that is most often unavailable until a later stage. On the other hand, while aiming to support the earliest stages of form design, computer supported sketching systems are simply electronic sketch pads that record a designer's 2D sketches. They provide little support for modelling the geometry implied by the sketches. One research direction in this area has, therefore, been the interpretation of diagrams and sketches. For instance, a pen-based prototype diagramming environment for early stages of design, the Electronic Cocktail Napkin system (Gross, 1996), provides facilities for recognition and parsing of 2D diagrams, and for establishing relations found in a diagram as constraints on elements of the schematic drawing. However, in general, the initial geometric configuration of a product is most often conducted outside existing CAD systems using pencil and paper (Fig. 1). Only when they are fully developed, the corresponding geometric designs are transferred into such a system for, for example, detailed analysis, visualization, etc. Reflecting an early commitment and trial-and-error approach to design, these systems are therefore unable to provide the required support for early geometric design.

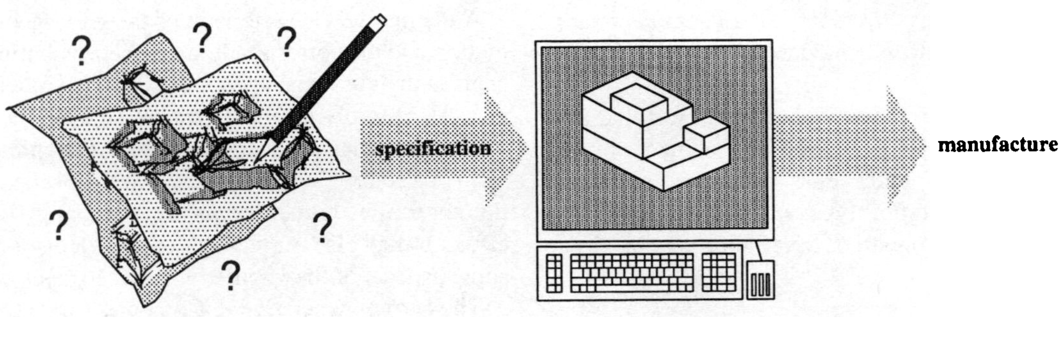
Clearly, if it is to support the early geometric configuration design, a CAD system should provide utilities for the rapid capture and evaluation of vague models, and support the further iterative and incremental development of the captured models. This paper presents our pilot investigation into the development of such a system—GEMCON. The system considers rapid and qualitative spatial configuration as a significant element of early geometric design, where a designer investigates the structural or topological organization of a product without committing to unnecessary details. It aims to reason about, maintain and represent the configuration solution space defined by geometric constraints rather than only a distinct solution point satisfying the constraints. Furthermore, constraints may be introduced into a configuration incrementally as a design progresses and use of the system does not require a user to follow a specific, fixed, or predefined sequence. Concepts behind the system are presented first (Section 2), followed by a description of the representation framework (Section 3), and the processing mechanism (Section 4) developed. Examples of using the system are described in Section 5. Finally, a discussion is presented in Section 6.

2. VAGUE GEOMETRIC MODELLING

The major goal of the GEMCON system is to support the incremental modelling of vague geometric configuration. More specifically, it seeks to support a user in establishing geometric configurations using vague, along with precise, geometric information including shape, size, location and orientation, and in gradually evolving the configurations into concrete and precise models. This requires the development of (a) a framework that is capable of representing geometric configurations with evolving levels of vagueness or precision, and (b) a reasoning mechanism that supports the manipulation on the model.

Our own observation into early design and that from existing research have led us to an overall philosophy, which has provided the conceptual basis for the system. This philosophy encompasses our view on a number of important aspects summarized below (Guan & MacCallum, 1996).

- *The relation between the user and the system.* The system should play, in general, the role of an Intelligent Design Assistant (MacCallum et al., 1985), which adopts a role secondary to the user, but can actively participate in the design process.
- *A parametric and constraint-based computational model of geometric design.* Geometric design is viewed as a process of establishing the geometric properties of a product to the extent that is required for physically manufacturing the product. The geometric properties are defined by a set of parameters that characterize or describe the geometric configuration of the product, that is, its overall shape and size and the shape, size, location, and orientation of its constituent components. These properties are completely and uniquely defined if values of all the parameters are fully and uniquely



early, unsupported stages *detailing, supported stages*

Fig. 1. Geometric design support.

defined. Values of the geometric parameters are specified through constraints. An activity or action of constraint manipulation that changes the possible values of the relevant geometric parameters is considered as a commitment toward the geometric design of the product.

- *A minimum commitment modelling principle.* This principle states that a commitment that is modelled in a CAD system should not be greater than that desired and requested by the user. In other words, the system should strive to facilitate the capture of the solution space defined by a piece of geometric constraint (vague or precise) rather than that of a specific solution point in the space.
- *An incremental refinement principle.* This complements the minimum commitment modelling principle by requiring the system to support the gradual refinement of the modelled design solution space in steps that are sufficiently small to maintain commitment at a minimum.

The GEMCON system is being built to provide a platform for exploring and experimenting with the above ideas. As a pilot study, the scope of investigation has so far been limited to the class of geometric configuration problems based on simple geometric information described in this paper. This was done on the basis that a) because of the great complexity involved in the problem, for the time being, there is a need to simplify the investigation into the implications, issues and challenges related to the above ideas, b) such primitive geometric information could provide a reasonable resource for modelling simple, abstract, and rough geometric configurations as occur during the early stage back-of-the-envelope type of design. In the following sections, the representation and processing mechanism developed for the system will be described.

3. REPRESENTATION

The framework for representing a geometric configuration within GEMCON is shown in Figure 3 using the OMT notation (Fig. 2) developed by Rumbaugh et al. (1991). For clarity and simplicity, only the key entities are included. The representation can be viewed as consisting of three parts: *geometric structure*, which provides the required elements for representing a geometric configuration consisting of objects at different levels of details; *parameterized geometric model*, which provides the parametric representation of the approximately or precisely defined geometric properties of each object; and *constraint models*, which capture high-level geometric design constraints that define the parameters characterizing the geometric properties of objects. Each of these three parts is explained below in more detail.

3.1. Geometric structure

As will be described later, an entity called *geom* is used to encapsulate the geometric properties of an object. To model a geometric configuration with different levels, each *geom* object, no matter at which level, has a component structure that contains other *geom* objects that the "parent" *geom* object may have as its components. A component structure is represented by a directed acyclic graph (DAG). For example, object A contains objects C and D can be represented as shown in Figure 4. Note that any objects that are not contained by another object, here A and B, are considered to be contained by a root *geom* object, which provides the basic space for configuring the objects.

Each *geom* object is associated with a geometric configuration space (GCS) (which is currently the minimum orthogonal bounding box of the *geom* object). The GCS provides a physical boundary or space to enclose all the *geom* objects contained by the *geom* object to which the GCS is

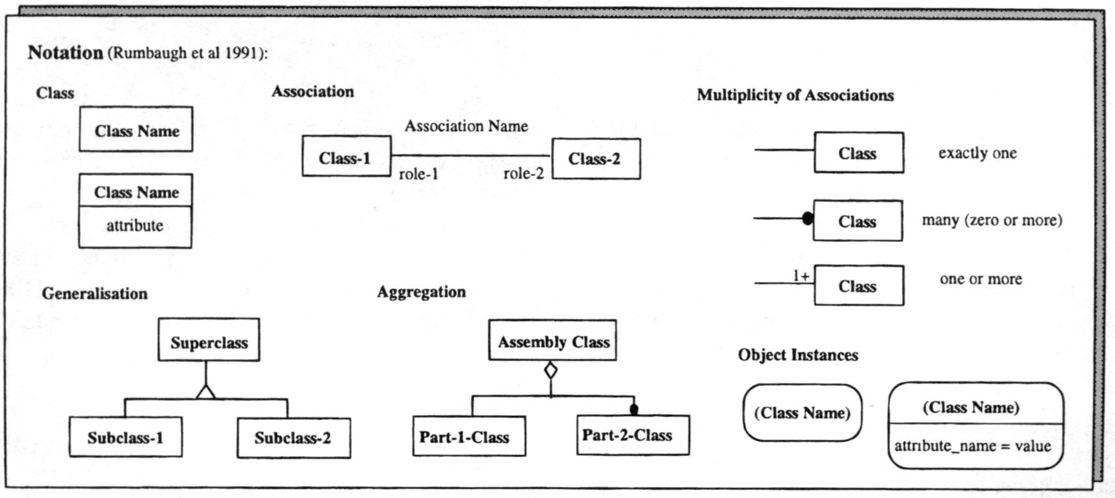


Fig. 2. The OMT notation (Rumbaugh, 1991).

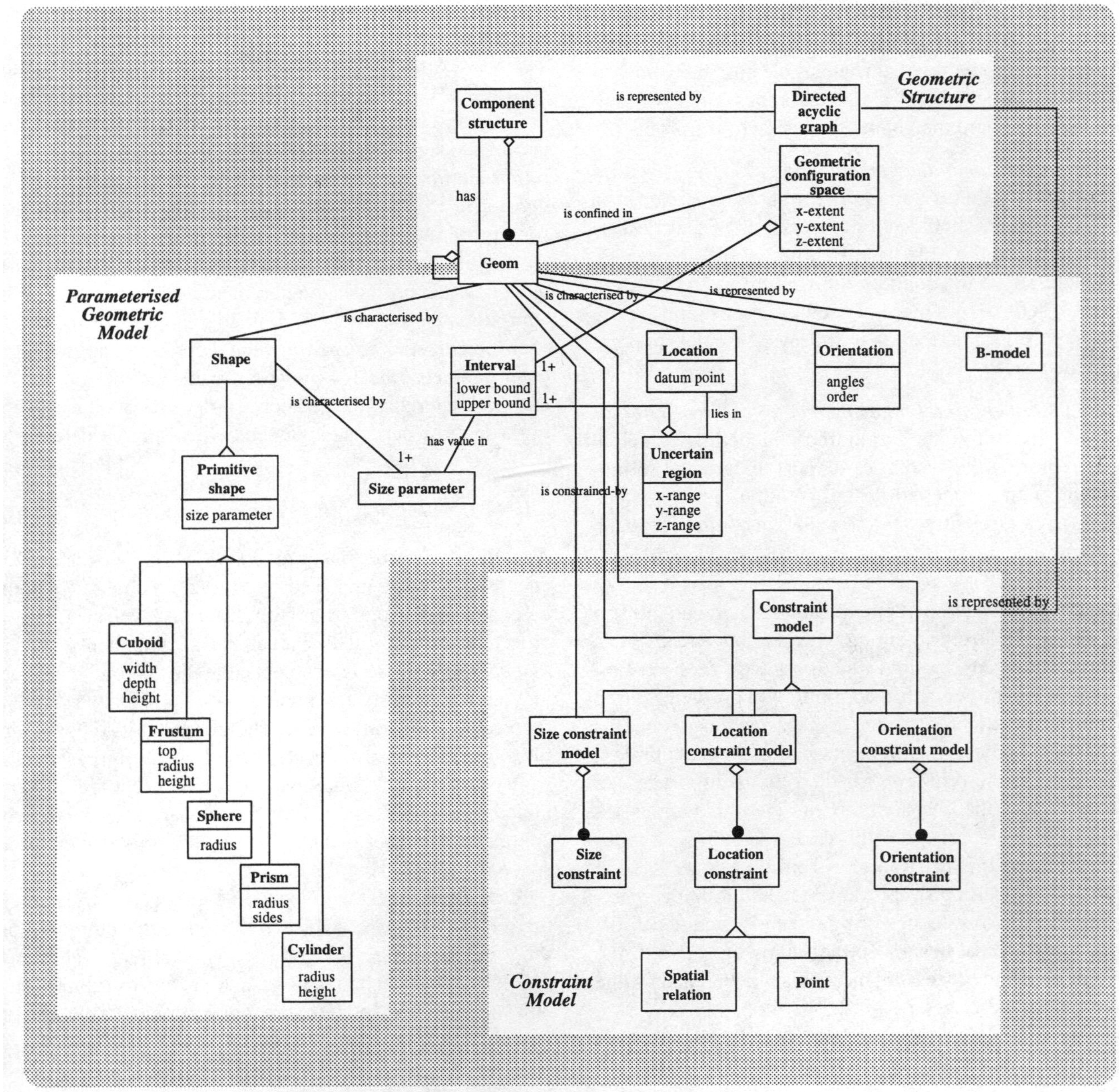


Fig. 3. Representation structure.

associated. With respect to Figure 4, for example, the GCS associated to the *geom* entity representing object A provides the space for enclosing the two *geom* entities representing objects C and D, respectively, while the GCS associated to the root *geom* entity provides the space for enclosing the two *geom* entities representing objects A and B. A GCS is currently a cuboid represented by three intervals corresponding to the three axes of the coordinate system.

3.2. Parameterized geometric model

An entity called *geom* is used to capture the geometric properties of a physical object. It encapsulates the inherent attributes of shape and size, and arrangement attributes of orientation, and location of the object. The shape may be any primitive—cuboid, frustum, sphere, prism, cylinder—whose size is characterized by size parameters[1] such as width, depth, etc. as shown in Figure 3. The value of a size parameter can be defined approximately or precisely and is represented by an *interval*. An interval is a bounded set of real numbers represented by a lower and an upper bound (Moore, 1979).

[1]There is a parameter, *sides*, for a prism that is not a size parameter. It defines the number of sides of the prism.

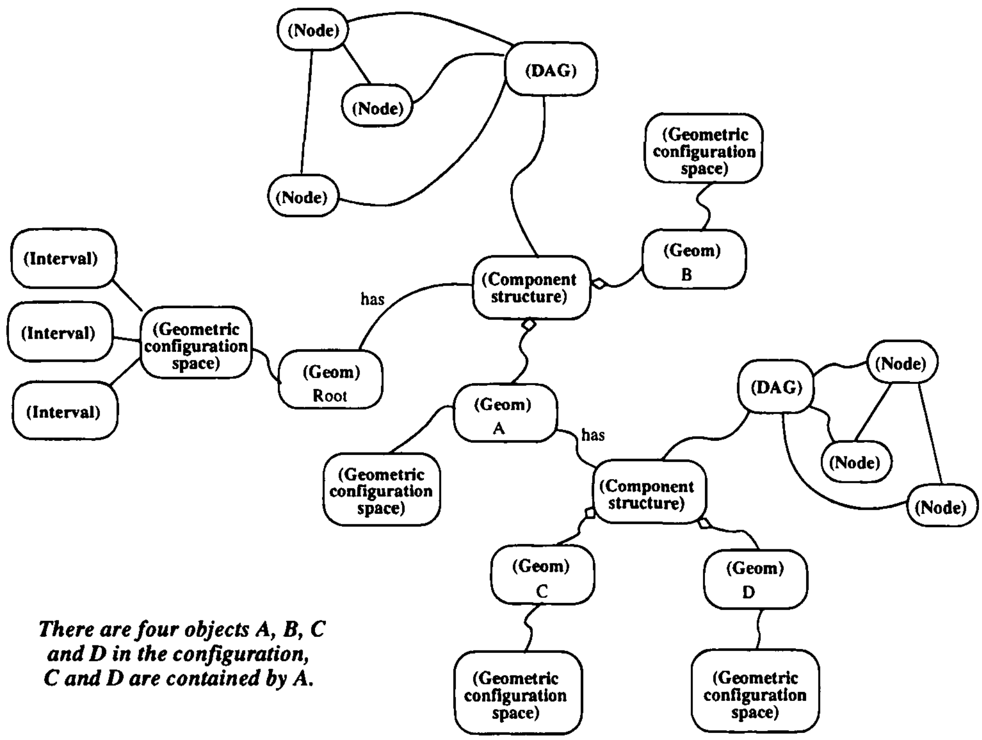


Fig. 4. Representation of a geometric structure.

The orientation of an object is characterized by the rotation the object has with respect to a global coordinate system. This is determined by the rotation angles, the coordinate axis about which a rotation is carried out, and the order of rotations about the axes.

The location of an object is characterized by a parameter, called a datum point (which is chosen as the geometric center of the object). This datum point lies in a 3D cubic *uncertain region* (UR), which captures the approximation or uncertainty associated with the location. An uncertain region is represented by three intervals along, respectively, the X, Y, and Z axes of the coordinate system associated with the corresponding geometric configuration space which specify the allowable x, y, and z coordinates of the corresponding datum point.

A boundary model (B-model) can be constructed to represent an object. A boundary model refers to a geometric model constructed via the boundary representation scheme developed in the field of geometric modelling (Requicha, 1980; Spatial Technology, 1992).

Figure 5 illustrates an example of the parameterized model of an object with a cuboid shape. The three pieces of size information—width between 15 and 18, depth approximately 10 and height exactly 20—are given through independent size constraints (discussed in Section 3.3), and are converted into three intervals shown in the figure. The method for deriving size intervals from independent size constraints is discussed in Section 4.3. Note that the degree of approximation used for converting *depth* $\approx$ 10 is assumed to be 1. The orientation of the object is not specified and is thus given a default setting. The location of the object is not specified either. The object is therefore assumed, by default, to be moveable in the entire geometric configuration space. The method used for deriving UR bounds from spatial relations is illustrated in Section 4.2.

3.3. Constraint models

A size constraint model, an orientation constraint model, and a location constraint model are associated with each *geom* object to hold all the constraints specified for all the *geom* objects contained by this object. These constraints are used to derive the values of the corresponding geometric parameters.

- *Size Constraints:* The system currently supports independent size constraints, each of which constrains only one size parameter. They are of inequality type (including $x \approx a$, $x < a$, $x \leq a$, $x > a$, $x \geq a$), range type ($x = [a_1, a_2]$), and equality type ($x = a$). Here, x denotes a size parameter, a, a_1, and a_2 real numbers.
- *Orientation Constraints:* The system currently supports the rotation of objects in multiples of 90 degrees around the coordinate axes.
- *Location Constraints:* There are two types—point and spatial relation. A point constraint specifies the coordinate position of an object in relation to the world coordinate system. A spatial relation, on the other hand,

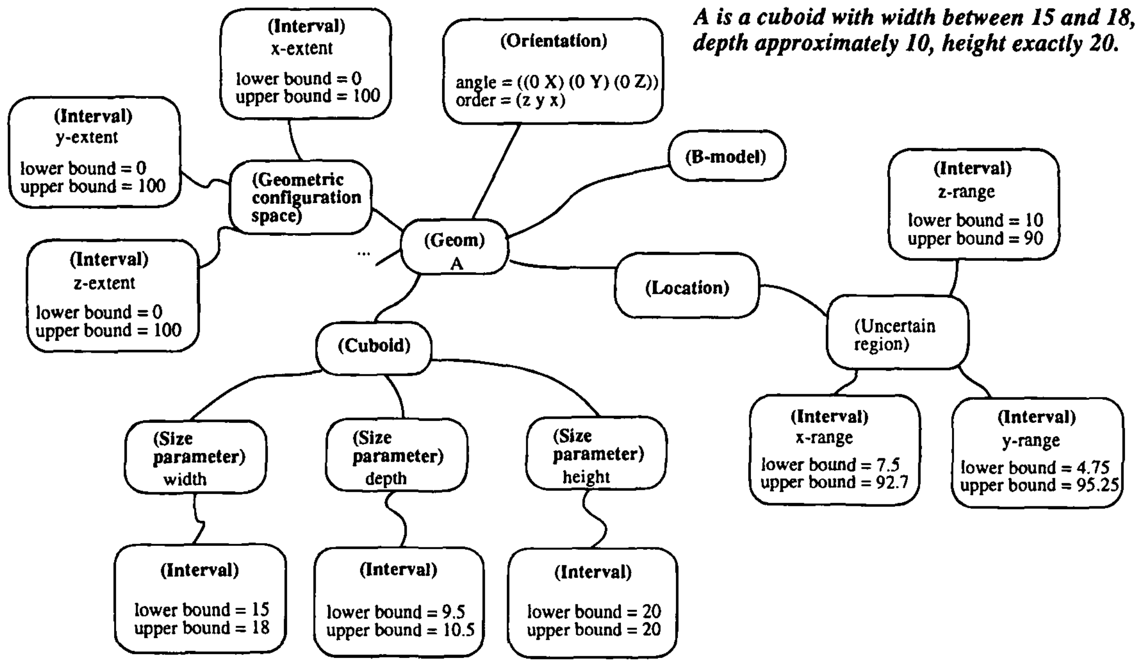


Fig. 5. Parameterized representation of an object with a cuboid shape.

constrains the location of an object in relation to other objects in the same level of configuration. The types of spatial relations currently supported are shown in Figure 6, where "above-orth-dist" stands for *above with an orthogonal distance* (the same notation applies to the others). It can be observed that, the definition of spatial relations as shown in Figure 6 means that {*above, above-orth-dist, below, below-orth-dist*} relations only affect the URs of the relevant objects in the Z direction, while {*right, right-orth-dist, left, left-orth-dist*} and {*behind, behind-orth-dist, front, front-orth-dist*} only affect the URs in the *X* and *Y* direction, respectively.

Each constraint is associated with an *importance* factor and a *time stamp*, which are used in the reasoning mechanism for resolving constraints (Section 4). The importance factor is a number that indicates how important it is for the associated constraint to be satisfied. The time stamp records the time when the associated constraint is established in the configuration process.

As an example, Figure 7 shows the representation of constraints related to an example geometric configuration.

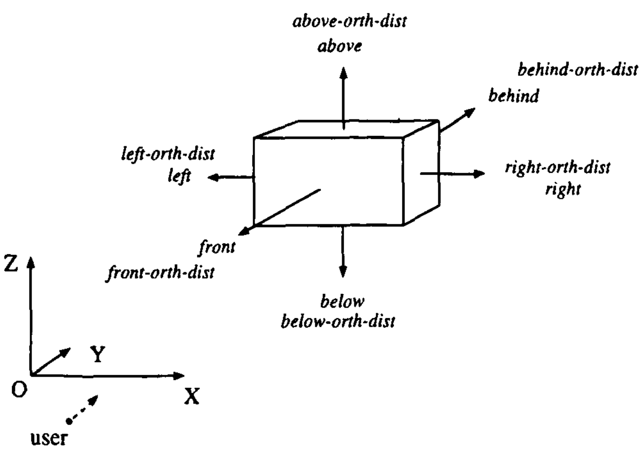


Fig. 6. Definition of spatial relationships.

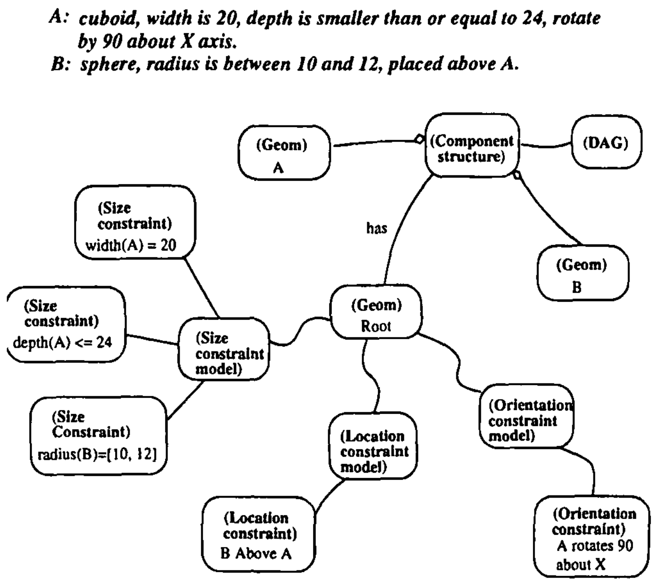


Fig. 7. Representation of constraint models.

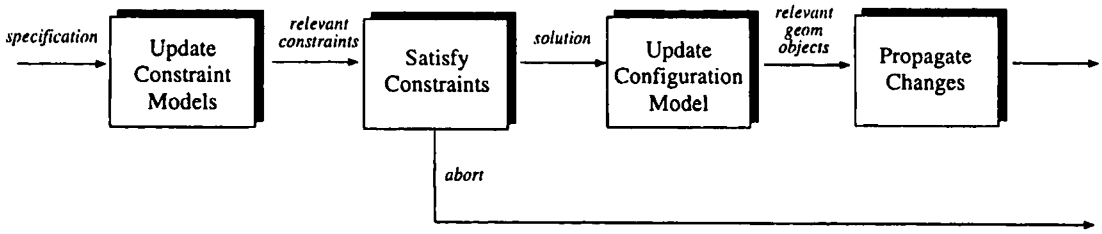


Fig. 8. Key stages of geometric information process.

To summarize, the framework shown in Figure 3 represents geometric configurations consisting of objects at different levels of details. The geometric properties of each object are encapsulated by a *geom* entity. They are characterized by size, orientation, and location parameters defined by a set of size constraints (of inequality, range, and equality types), orientation constraints, as well as spatial relations. Approximate or precise values for size and location parameters are captured by size ranges and location uncertain regions, respectively.

4. PROCESSING OF GEOMETRIC INFORMATION

A constraint-based mechanism, consisting of the key stages shown in Figure 8, has been developed to support the manipulation of geometric configuration represented via the above framework. This mechanism is applied to process size, orientation, and location information related to a geometric configuration. Each of the stages is described briefly in the following.

4.1. Update of constraint models

Constraints are created based on information on the size, orientation and location of objects, and are integrated into the relevant constraint models. When a new constraint is specified that constrains an entity that has already been constrained, instead of always taking the newly specified one, the system currently acts upon one of the following resolving strategies:

- IMPORTANCE—For a given set of constraints on the same parameter or object, if they have different *importance* factors, then always select the most important constraint, that is, the one with greatest *importance* factor, otherwise refer to the user for a decision.
- TIME—For a given set of constraints on the same parameter or object, always select the most recently specified constraint, that is, the one with the most recent *time stamp*.
- USER—For a given set of constraints on the same parameter or object, always refer to the user for a decision.
- AUTOMATIC—For a given set of constraints on the same parameter or object, always resolve using a combination of the above strategies. As an example, the automatic strategy for resolving size constraints is defined as that shown in Figure 9. There, an additional strategy, REFINEMENT, is available that resolves size constraints based on the Boolean intersection of the value sets of the given constraints. If a nonempty intersection exists, then the system uses the intersection set as the value range of the corresponding size parameter, otherwise it refers to the user for a decision.

The default strategy is set to be AUTOMATIC.

Once updated, a set of relevant constraints are extracted and passed to the satisfaction process.

4.2. Satisfaction of constraints

Constraints obtained in the above process are satisfied to generate values for the corresponding geometric parameters. Size and orientation constraints handled by the cur-

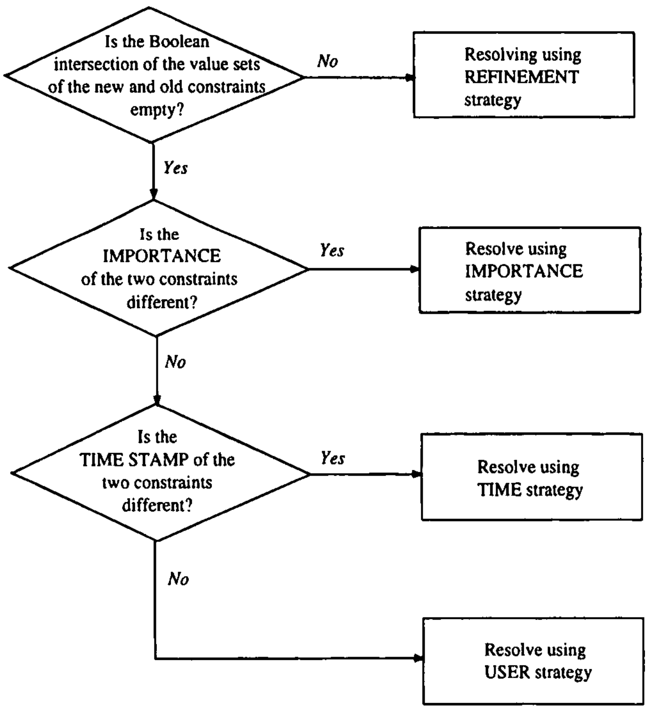


Fig. 9. The AUTOMATIC strategy for resolving size constraints.

rent system include only independent ones that do not involve more than one size parameter or object. Therefore, this process does not need to solve size and orientation constraints.

The location of an object in a configuration is given either in spatial relations or in a point and is represented by a datum point that lies in a UR defined by three ranges. To derive the bounds of the URs of objects, the solving process first formulates a set of constraints on these bounds from the spatial relations or points, and then proceeds to solve the formulated UR constraints.

4.2.1. Formation of UR constraints

This subprocess interprets location constraints into constraints on the bounds of the corresponding URs (referred to as UR constraints). As an example, Figure 10 illustrates the formulation of UR constraints in a simple situation where object G_p is specified to be *above* object G_r and G_p, and G_r have no *above* or *below* relations with other objects in the same configuration. Note that the size of an object used here is that of the orthogonal bounding box of the object. When the location of an object is not specified,

---

GIVEN: above(G_p, G_r), where G_p is the primary object and G_r the reference object, and G_r and G_r have neither above nor below relation with other objects in the same configuration,

the bounds of geometric configuration space GCS in the Z direction: $GCS_z = [\underline{GCS}_z, \overline{GCS}_z]$,

the size of G_r in the Z direction: $SIZE_{rz} = [\underline{SIZE}_{rz}, \overline{SIZE}_{rz}]$,

the existing values of the bounds of the UR of G_r in the Z direction : $UR'_{rz} = [\underline{UR'}_{rz}, \overline{UR'}_{rz}]$,

where $\underline{UR'}_{rz} = \underline{GCS}_z + \underline{SIZE}_{rz}/2$ and $\overline{UR'}_{rz} = \overline{GCS}_z - \underline{SIZE}_{rz}/2$,

the size of G_p in the Z direction: $SIZE_{pz} = [\underline{SIZE}_{pz}, \overline{SIZE}_{pz}]$,

the existing values of the bounds of the UR of G_p in the Z direction: $UR'_{pz} = [\underline{UR'}_{pz}, \overline{UR'}_{pz}]$,

where $\underline{UR'}_{pz} = \underline{GCS}_z + \underline{SIZE}_{pz}/2$ and $\overline{UR'}_{pz} = \overline{GCS}_z - \underline{SIZE}_{pz}/2$.

TO DERIVE: new constraints on $[\underline{UR}_{pz}, \overline{UR}_{pz}]$ and $[\underline{UR}_{rz}, \overline{UR}_{rz}]$.

FOR THE LOWER BOUNDS:

IF the space between GCS_z upper bound and G_r placed at its corresponding UR lower bound is larger than G_p

THEN form the following constraints:

$$\underline{UR}_{pz} = \underline{UR}_{rz} + (\underline{SIZE}_{pz} + \underline{SIZE}_{rz})/2$$
$$\underline{UR}_{rz} = \underline{UR'}_{rz}$$

ELSE declare spatial conflict.

FOR THE UPPER BOUNDS:

IF the space between GCS_z upper bound and G_r placed at its corresponding UR upper bound is larger than G_p

THEN form the following constraints:

$$\overline{UR}_{pz} = \overline{GCS}_z - \underline{SIZE}_{pz}/2$$
$$\overline{UR}_{rz} = \overline{UR'}_{rz}$$

ELSE IF G_r is not point fixed and its corresponding UR upper bound can be reduced to accommodate G_p

THEN form the following constraints:

$$\overline{UR}_{pz} = \overline{GCS}_z - \underline{SIZE}_{pz}/2$$
$$\overline{UR}_{rz} = \overline{UR'}_{rz} - \underline{SIZE}_{pz}$$

ELSE declare spatial conflict.

Fig. 10. Formulation of UR constraints from an above relation in a simple situation.

Table 1. *Conversion of independent size constraints to value ranges*

Constraints	Value
$x = a$	$[a, a]$
$x < a$	$[\underline{\text{default}}, a)$
$x \leq a$	$[\underline{\text{default}}, a]$
$x > a$	$(a, \overline{\text{default}}]$
$x \geq a$	$[a, \overline{\text{default}}]$
$x \approx a$	$[a - \text{degree}_{\text{approx}}/2, a + \text{degree}_{\text{approx}}/2]$
$x = [a_1, a_2]$	$[a_1, a_2]$

the object is assumed, by default, to be moveable in the entire geometric configuration space. Following Retz-Schmidt (1988), the object whose location is determined by a spatial relation, here G_p, is called a *primary object*, and the object to which the relation is specified, here G_r, is called a *reference object*.

A graphical interpretation of the definition of UR via an *above-orth-dist* relation can be found in the sample configuration session in Section 5 (see Fig. 13).

4.2.2. Solving of UR constraints

The set of UR constraints derived from spatial relations or point positions is solved through a generic constraint solver, currently CLP(R) (Heintze et al., 1991). This generates values for the corresponding bounds of the URs.

4.3. Update of configuration model

The relevant uncertain regions are updated according to the bound values derived from the above satisfaction process. The rotation angles, axes, and order that describe the orientation of an object in relation to the global coordinate system are also modified according to any new orientation constraints.

To update the value of a size parameter constrained by an independent size constraint, the system uses Table 1 to generate the bounds of the corresponding value range. Note that because $x\Delta a$, where $\Delta \in \{<, \leq, >, \geq\}$ only defines one value bound for a size parameter x, a default lower bound and upper bound, here denoted by $\underline{\text{default}}$ and $\overline{\text{default}}$, respectively, are used to provide the missing bound. They are provisional and can be changed by a user. The round brackets ")" and "(" in the ranges for $x < a$ and $x > a$, respectively, mean that the number a is not included in the corresponding value ranges.[2] To convert $x \approx a$, a default degree of approximation, $\text{degree}_{\text{approx}}$ (≥ 0), is introduced. It defines the width of the interval as shown in Table 1.

[2] In the current system, exclusion of a in $x < a$ or $x > a$ is treated as that the upper bound of $x < a$ or the lower bound of $x > a$ takes $a - \delta$ or $a + \delta$, respectively, where δ is a very small positive number in relation to a.

4.4. Propagation of changes

Finally, changes to objects are propagated throughout the relevant parts of the whole configuration model to maintain the global consistency. Currently, this includes two aspects:

- A reevaluation of the relevant set of location constraints. This is necessary because any changes in size and orientation lead to changes in the UR bounds of the relevant objects.
- A propagation to the sublevel components of a changed object. This currently includes a reevaluation of all location constraints existing among the component objects in response to the changed parent object (the container).

5. USE OF THE SYSTEM

The GEMCON system has been implemented using Harlequin Lisp Works (Common Lisp and CLOS) (Harlequin, 1994) running on a Silicon Graphics platform. To highlight the underlying ideas, the major modelling operations developed are briefly described and their use in constructing and evolving a geometric configuration model is illustrated.

5.1. Major modelling operations

Table 2 presents the set of operations that can be used for developing geometric configurations. Initial construction of models can be carried out through a type 2 operation, *make-geom*, which takes shape, size, location, and orientation information on the object to be modelled. A default is used where a value is required but not specified.

Further development of the initial model can be approached incrementally through type 2 and 3 operations. Overall, these operations support three types of modification to an initial model: *incremental*, which adds missing items to a model to make it less incomplete; *refinement*, which fine-tunes a model by reducing associated approximations; and *retraction*, which removes an existing item from a model.

The general flow of control implemented for these operations is shown in Figure 11. When an operation is invoked, the system first checks the correctness of the input information. A faulty input is sent to a correction process, which allows the user to amend the input or to abort the whole operation. If the input appears correct, it is then processed through the constraint-based mechanism described earlier. Whenever an operation is aborted as a result of spatial conflict or the user's change of plans, a clean-up process is invoked to ensure that no changes have been made to the corresponding model by the operation aborted.

5.2. A Sample modelling session

This section illustrates the use of the system through four scenes that shows the interaction between the user and the system in the process of constructing and evolving a specific geo-

Table 2. *A selection of configuration modelling operations*

Type	Operation	Brief description
1. Establish configuration environment	set-gcs-size	Modify the size of the geometric configuration space of a given geometric object
	set-default-shape	Change the default shape type
	set-size-approx	Set the degree of approximation to be used in representing approximately specified value of a size parameter
	set-size-lb	Set the default lower bound of the interval representation for the value of a size parameter
	set-size-ub	Set the default upper bound of the interval representation for the value of a size parameter
	set-strategy	Choose the strategy to be used by the system in resolving constraints
2. Explore component geometry	make-geom	Create a geometric model based on vaguely or precisely specified geometric information
	delete-geom	Delete a given geometric object
	add-scs	Modify the size of a given geometric object
3. Explore spatial arrangement	add-ocs	Modify the orientation of a given geometric object
	specify-rot-order	Specify the order in which rotations are performed
	add-lcs	Modify the location of a given geometric object
	refine-ur	Refine the location uncertain region of a geometric object
	delete-constraint	Delete a geometric constraint
4. Inspect configuration model	display-geom	Display information related to a given geometric object
	display-strategy	Display the strategy used currently for resolving a given type of constraint
	explain-strategy	Give brief explanation of a given constraint resolving strategy
	display-env	Display variables that define the configuration environment

metric configuration model. For ease of description, operations in the scenes are numbered. Where applicable, the figure number that corresponds to an operation is also included after the operation number for ease of reference. Input to the system is presented in **bold** (see Figs. 12, 15, and 18–20). The > symbol at the beginning of each input is the prompt of the Lisp Listener through which the system is invoked. To save space, layout of some of the textual output from the system is changed and unimportant texts are omitted.

5.2.1. Working with vague information

Initially, the user has the following information about a geometric configuration:

> It contains two objects. The first one is a cuboid with its *width*, *depth*, and *height* each being exactly 20. The second object is placed *above* the first one, and is also a cuboid with a *width* between 16 and 18, *depth* about 5, and *height* no more than 14. The orthogonal distance between the two objects is 5.

Scene 1 (Fig. 12) shows the modelling of such a vaguely specified configuration. Operation [1.1] and [1.2] create *geom1* and *geom2* to represent the two objects described above. The size ranges of the objects captured in the system and their UR bounds are shown by [1.4]–[1.7]. The generated configuration model therefore corresponds to the solution space illustrated in Figure 13. Namely, *geom1* is an exactly sized cuboid located in the UR of ([10.0, 90.0], [10.0, 90.0], [10.0, 75.0]). Note that, for minimum commitment, an object whose location is not specified is assumed to be moveable in the entire space of the corresponding GCS (here ([0.0, 100.0], [0.0, 100.0], [0.0, 100.0])—see Operation [1.8]), as is the case for *geom1* (except in the Z direction,

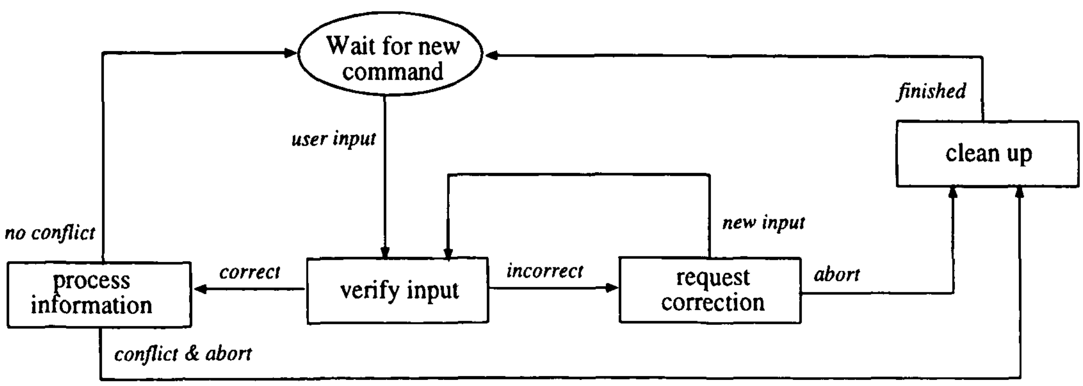
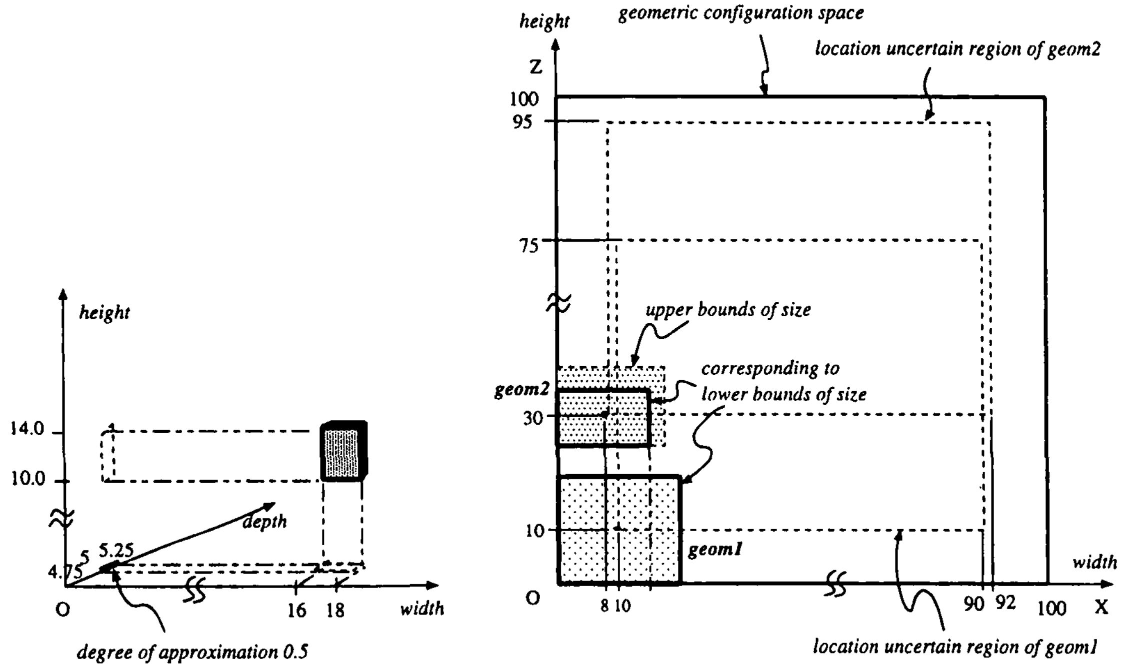


Fig. 11. Flow of control for geometric manipulations.

```
> (make-geom :size '(((width = 20)) ((depth = 20)) ((height = 20))))          [1.1]
GEOM1

> (make-geom :size '(((width = 16 -> 18)) ((depth ~= 5)) ((height <= 14)))    [1.2]
             :location '(((above-orth-dist geom1 5))))
GEOM2

> (display-geom geom0)                                                         [1.3] (Figure 14)

> (display-geom geom1 :aspect 'size-value)                                     [1.4]
<Size of GEOM1>
     WIDTH:
            Lower Bound — 20
            Upper Bound — 20
     DEPTH:
            Lower Bound — 20
            Upper Bound — 20
     HEIGHT:
            Lower Bound — 20
            Upper Bound — 20

> (display-geom geom1 :aspect 'ur)                                             [1.6] (Figure 13(b))
<Location uncertain region of GEOM1>
         X-RANGE:
                  Lower Bound — 10.0
                  Upper Bound — 90.0
         Y-RANGE:
                  Lower Bound — 10.0
                  Upper Bound — 90.0
         Z-RANGE:
                  Lower Bound — 10
                  Upper Bound — 75

> (display-geom geom2 :aspect 'size-value)                                     [1.7] (Figure 13(a))
<Size of GEOM2>
     WIDTH:
           Lower Bound — 16
           Upper Bound — 18
     DEPTH:
           Lower Bound — 4.75
           Upper Bound — 5.25
     HEIGHT:
           Lower Bound — 10.0
           Upper Bound — 14

> (display-geom geom2 :aspect 'ur)                                             [1.7] (Figure 13(b))
<Location uncertain region of GEOM1>
         X-RANGE:
                  Lower Bound — 8.0
                  Upper Bound — 92.0
         Y-RANGE:
                  Lower Bound — 2.375
                  Upper Bound — 97.625
         Z-RANGE:
                  Lower Bound — 30
                  Upper Bound — 95

> (display-env)                                                                [1.8] (Figure 13)
*DEFAULT-GCS-SIZE* ....... ((0.0 100.0) (0.0 100.0) (0.0 100.0))
*DEFAULT-DEPTH* ......... (15 0 15.0)
*DEFAULT-HEIGHT* ......... (10.0 10.0)
*DEFAULT-RADIUS* ......... (10.0 10.0)
*DEFAULT-WIDTH* ......... (20.0 20.0)
*DEFAULT-SHAPE-TYPE* ......... CUBOID
*DEFAULT-IMPORTANCE* ......... 0.0
*DEGREE-OF-APPROXIMATION* ........ 0.5
*DISPLAY-LEVEL* ......... 1
*CURRENT-SC-RESOLVING-STRATEGY* ......... AUTOMATIC
...
```

Fig. 12. Interaction Scene 1.

where it is constrained by *geom2*). On the other hand, the solution space of *geom2* includes all the cuboids whose width, depth, and height have values in the ranges of [16,17], [4.75, 5.25], and [10.0, 14.0], respectively, and whose datum point is located within the UR of ([8.0, 92.0], [2.375, 97.625], [30.0, 95.0]). The value ranges of depth and height of *geom2* are defined based on the default *degree-of-approximation* (0.5) and lower bound (10.0) of **default-*



(a) The size solution space of geom2 (not to scale).

(b) The location solution space of geom1 and geom2 (OXZ projection, not to scale).

Fig. 13. Solution space of the geometric configuration model established by Operations [1.1] and [1.2] and shown by Operations [1.4]–[1.8].

*height** shown by operation [1.8]. A wire-frame display of the boundary model of a configuration instance included in the solution space, which corresponds to the minimum size and lower-left-front corner of the corresponding UR, is shown in Figure 14. This is generated by operation [1.3] using a geometric modeller, ACIS (Spatial Technology, 1992), integrated in the system. For ease of viewing, labels are added to the display manually to identify the corresponding *geom* objects.

5.2.2. Further development of objects

While the user could continue to add into the configuration other objects at the same level as *geom1* and *geom2*, he/she may also choose to focus on more detailed development of an existing object, say *geom1*. This can be a) creation and iterative modification of its subcomponents or b) iterative modification of its size, orientation, and location.

Scene 2 illustrates the use of the system in assisting these two aspects of configuration development. Part A (Fig. 15) shows the creation and iterative modification of the subcomponents of *geom1* to achieve a specific configuration goal. Operation [2A.1] makes a subcomponent, *geom3*. After that, in [2A.2], the user attempts to create, and place above *geom3*, a subcomponent with a height of 30. Because 30 is larger than the space (which is $20 - 10 = 10$) left between *geom3* and the corresponding configuration space (the orthogonal bounding box of the parent object *geom1*, here the same as *geom1* itself), creating the subcomponent would cause a spatial conflict. When notified of the situation, the user withdraws the operation and makes a smaller object instead ([2A.3]). An example of the model defined after [2A.3] is graphically displayed in Figure 16a. Figure 16b and 16c show the local geometric structure of *geom1* and global structure of the resulting configuration, respectively, where *geom0* corresponds to the most *geom* object described in Section 3.1.

Next, the goal of the user is to change the height of *geom3* to 18 while maintaining its current orientation. To do this, the user first tries to change the height directly ([2A.7]). Because this would cause a spatial conflict as detected by the system, the user then explores another way: first altering the orientation of *geom3* ([2A.8]), then changing its *height* ([2A.9]). This, however, has altered the orientation of *geom3* (Figure 17a), which is not desirable. If, at this point, the user immediately changes the orientation of *geom3* back to the original, the operation would fail as expected (see [2A.11] for demonstration). One correct strategy would be to make use of the space left in the other direction, as illustrated by the next few steps. The user first puts *geom5* behind *geom3* ([2A.12]) and then tries to reorient *geom3* ([2A.13]). The reorientation is still unsuccessful because there is an above relation between *geom3* and *geom5*. So, to achieve the original configuration goal, all the user needs to

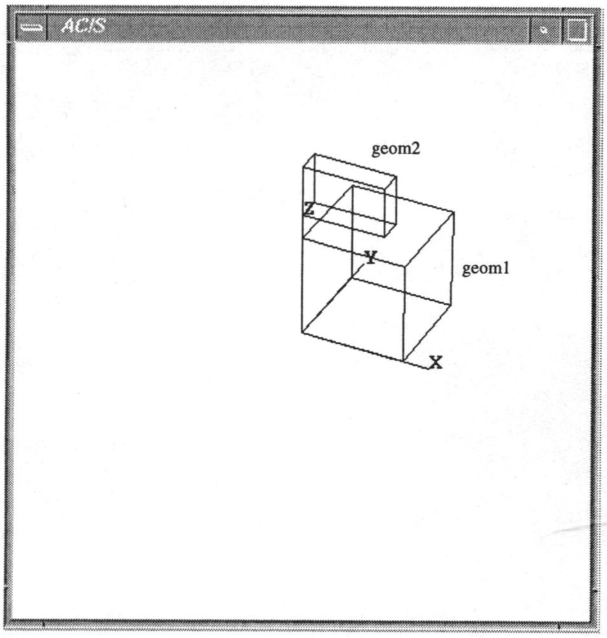


Fig. 14. An instance of the geometric configuration model.

do now is to remove the above relation ([2A.14]) and then rotate *geom3* back to its original orientation ([2A.15], Figure 17b).

The illustrated is only one way of approaching the targeted configuration iteratively using the available operations. The user could achieve the same configuration goal in several other ways or sequences of using the operations, for example

- specifying *geom5* behind *geom3* → deleting the existing *geom5* above *geom3* relation → increasing the height of *geom3* to 18;
- deleting the *geom5* above *geom3* relation → specifying *geom5* right *geom3* → increasing the height of *geom3* to 18;

and so on.

Part B of Interaction Scene 2 (Fig. 18) shows the use of the system in modifying the size ([2B.1]), orientation ([2B.2]), and location ([2B.3]–[2B.8]) of *geom1* itself. The uncertain region of *geom1* in the Z direction is refined from [10, 75] ([2B.3]) to [30, 75] ([2B.5]) through the operation *refine-ur* ([2B.4]) by raising its lower bound. Later in Operation [2B.7], it is replaced by that derived from an above relation. This is because, in the current system, the UR of an object is allowed to be changed directly through *refine-ur* only if its location is not dependent on any other objects. In other words, UR bounds derived from spatial relations supersede those from *refine-ur*.

5.2.3. Use of resolving strategies

As discussed earlier, the system supports a number of strategies for resolving constraints. Scene 3 (Fig. 19) shows how this is used in the case of size modification. In Operation [3.3], the user tries to change the depth of *geom7*. Because the default resolving strategy is AUTOMATIC (see Operation [1.8]) and the existing constraint, *depth* = [24,27] specified in [3.1], already satisfies the new constraint, *depth* ≥ 8 [i.e., the Boolean intersection of these two constraints is not empty (refer to Fig. 9)], the depth is not changed (compare [3.2] and [3.4]). Thus, if the user's intention was to enlarge the existing solution space for *depth*, then it was not achieved. To do so, the user needs to use one of the other strategies, TIME, IMPORTANCE, or USER. After switching to TIME ([3.5]), the user succeeded in enforcing the new constraint ([3.6]).

5.2.4. Dynamic change of configuration space and deletion of object

A geometric configuration space can be modified dynamically if necessary, e.g. to accommodate larger objects. Scene 4 (Fig. 20) shows how this can be done. In Operation [4.1], the user tries to modify the width of *geom7* to 110, which fails because it exceeds the size of the configuration space of *geom7*-100 (defined by the default shown in Operation [1.8]). This, however, is accommodated ([4.3], Fig. 21a) after the user changes the size of the GCS to 300 along the *X*, *Y*, and *Z* directions ([4.2]).

Finally, objects can be deleted from a configuration as illustrated by Operation [4.5]. The upper most level of the geometric structure so far and the graphical display of an instance of the corresponding model are shown in Figure 21b and 21c.

6. DISCUSSION

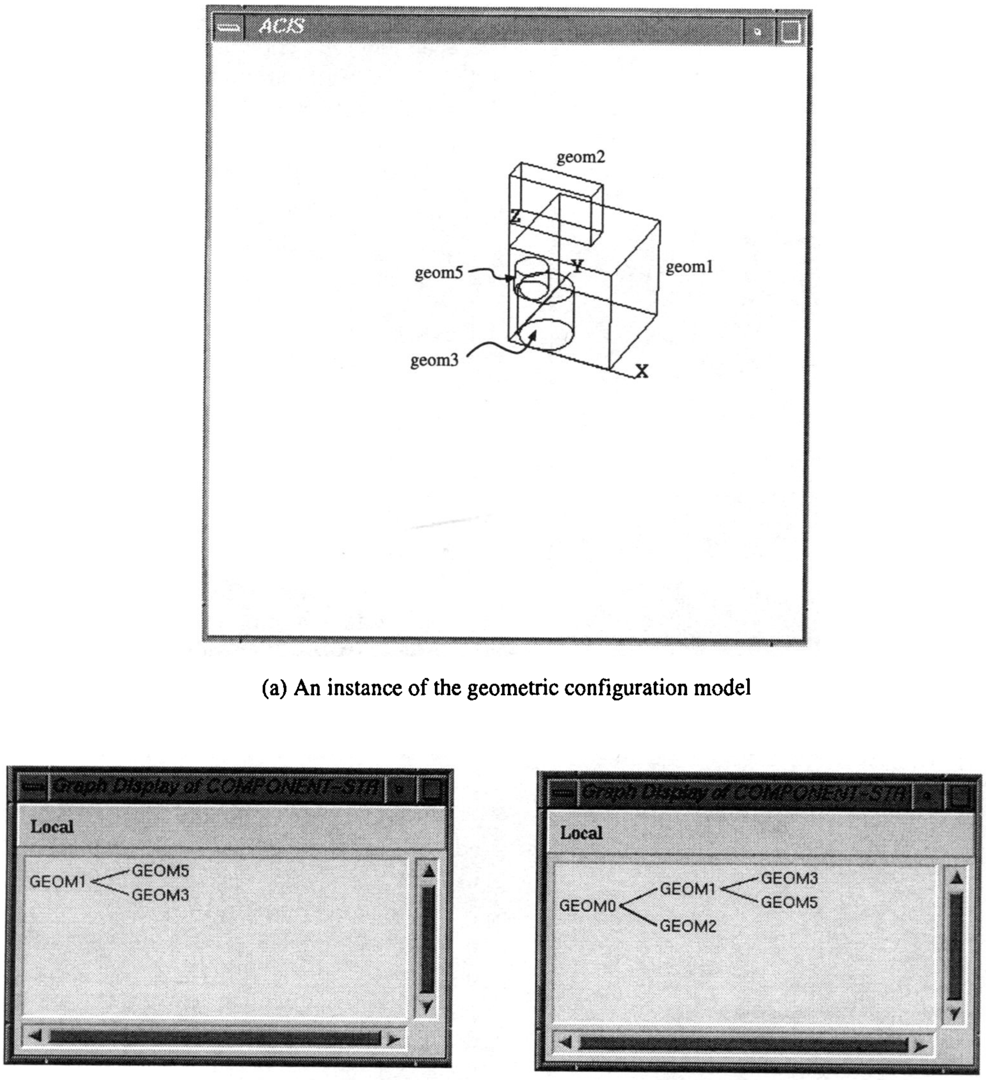
6.1. Incremental modelling of vague geometry

In this paper, we have presented a prototype system for supporting the early stages of geometric design following the conceptual basis presented in Section 2. Within the current scope of investigation, the system supports the user in approaching the development of a geometric configuration in a minimum commitment way by supporting incremental and iterative definition and evolution of configurations using vague size and location information. Two observations can be made on the type and content of the information used in the sample session:

- The use of the system, more specifically the modelling operations, does not demand complete or precise knowledge or information of an object. Rather, bits and pieces of vague information, introduced by inequality type of independent size constraints and spatial relations, or precise information can be used to initialise a configuration model, which can be built up gradually.

```
> (make-geom :shape 'cylinder                                    [2A.1]
             :size '(((radius = 5)) ((height = 10)))
             :contained-by geom1)
GEOM3

> (make-geom :shape 'cylinder                                    [2A.2]
             :size '(((radius = 5)) ((height = 30)))
             :location '(((above geom3)))
             :contained-by geom1)
There is not enough space available in the Z direction for placing GEOM4.
Please choose from the following:
      0 : Abort
PICK: 0
Please confirm:
      0 : Abort
      1 : Allow
PICK: 0
GEOM4 is deleted.

> (make-geom :shape 'cylinder                                    [2A.3]
             :size '(((radius = 3)) ((height = 5)))
             :location '(((above geom3)))
             :contained-by geom1)
GEOM5

> (display-geom geom0 :level 2)                                  [2A.4] (Figure 16(a))

> (display-geom geom1 :aspect 'local-cs)                         [2A.5] (Figure 16(b))

> (display-geom geom0 :aspect 'global-cs)                        [2A.6] (Figure 16(c))

> (add-scs geom3 '(((height = 18))))                             [2A.7]
Using ((HEIGHT GEOM3) = 18).
Reasons are:
            The current conflict resolution strategy is set as:
                       AUTOMATIC
            ((HEIGHT GEOM3) = 18) is more recent than ((HEIGHT GEOM3) = 10)
There is not enough space available in the Z direction for placing GEOM5 in relation to GEOM3.
Please choose from the following:
      0 : Abort
PICK: 0

> (add-ocs geom3 '(((rotate 90 (y wcs)))))                       [2A.8]

> (add-scs geom3 '(((height = 18))))                             [2A.9]

> (display-geom geom0 :level 2)                                  [2A.10] (Figure 17(a))

> (add-ocs geom3 '(((rotate 0 (y wcs)))))                        [2A.11]
Using (ROTATE GEOM3 0 (Y WCS)).
Reasons are:
            The current conflict resolution strategy is set as:
                       AUTOMATIC
            (ROTATE GEOM3 0 (Y WCS)) is more recent than (ROTATE GEOM3 90 (Y WCS))
There is not enough space available in the Z direction for placing GEOM5 in relation to GEOM3.
Please choose from the following:
      0 : Abort
PICK: 0

> (add-lcs geom5 '(((behind geom3))))                            [2A.12]

> (add-ocs geom3 '(((rotate 0 (y wcs)))))                        [2A.13]
Using (ROTATE GEOM3 0 (Y WCS)).
...
There is not enough space available in the Z direction for placing GEOM5 in relation to GEOM3.
Please choose from the following:
      0 : Abort
PICK: 0

> (delete-constraint above0)                                     [2A.14]

> (add-ocs geom3 '(((rotate 0 (y wcs)))))                        [2A.15]

> (display-geom geom0 :level 2)                                  [2A.16] (Figure 17(b))
```

Fig. 15. Interaction Scene 2—Part A.



(a) An instance of the geometric configuration model

(b) Local component structure

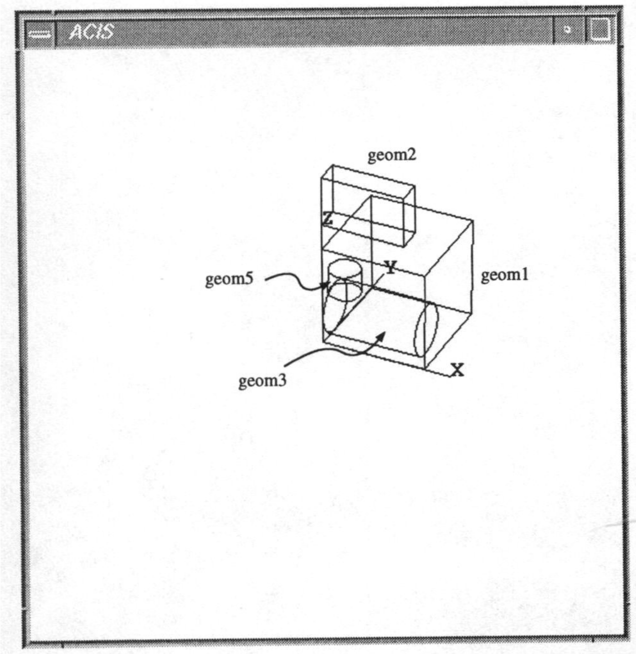
(c) Global component structure

Fig. 16. Addition of subcomponents to geom1.

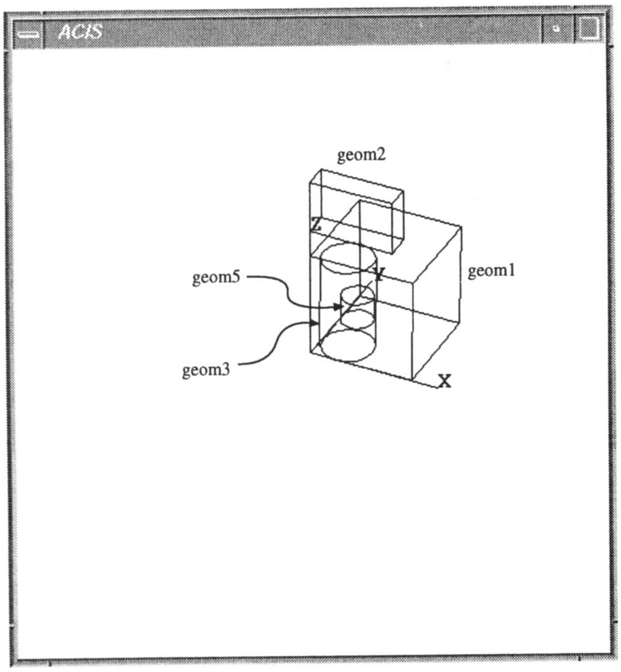
- A piece of information used could simply be:
  - an *addition* of something about an object which has not been defined,
  - a *refinement* to or redefinition of something already defined (coarsely or not),
  - a *retraction*, from the model, of something already defined.

Figure 22 illustrates the process in the sample session in developing the specific geometric configuration, where a directed line denotes the application of the associated operation to the directed *geom* object. The process is clearly *iterative* and *nonsequential*, as shown by those lines linking objects in various stages of the process. It is interative in the sense that the user can return to a previously defined *geom* object carrying out modifications. It is also nonsequential in that the user can define any *geom* object at any stage. The modelling operations can be invoked in different sequences/orders to different objects to achieve the same goal (Scene 2A). Therefore, they are in this sense flexible and support iterative and nonsequential configuration design.

Use of the configuration system requires cooperation between a user and the system. As demonstrated, the user is responsible for supplying information and for directing, driving, and controlling the configuration process in terms of determining a particular sequence of achieving a configuration goal or exploring a configuration (in other words, determining to invoke which operation at a given point of a configuration session), making important decisions in resolving conflicts or inconsistencies. The system, on the other hand, assists the user in the process by taking care of the



(a). Height of geom3 is changed but orientation is altered as well.



(b). Height of geom3 is changed and orientation is maintained.

Fig. 17. Configuration of *geom3* and *geom5* with *geom1*.

reasoning and propagation of the effect of a modelling operation, consistency checking and model presentation in various ways. The role of the system is consistent with that of the general Intelligent Design Assistant described in MacCallum et al. (1985).

Finally, we observe that the parametric and constraint-based model of geometric design is descriptive in the sense that it is close to an intuitive description of geometry and provides a natural description of design or engineering constraints. The pilot study has also justified the feasibility of the corresponding representation framework to handle simple geometric configurations investigated.

6.2. Relation to other work

This section relates the work presented in this paper to the relevant existing works from computer-based design support systems, including systems that adopt a minimum commitment based approach, spatial layout systems, as well as constraint-based geometric design systems.

6.2.1. Existing systems adopting minimum commitment approach

In a previous paper (Guan & MacCallum, 1996), we have proposed the adoption of the minimum commitment principle in early geometric design support systems, and have reviewed its application in a number of areas including computer-based planning systems (Weld, 1994), engineering design (Asimow, 1962; Dym, 1994), and a few computer-based design and manufacturing planning systems.

Minimum commitment, known as least commitment, is adopted, in a limited way, to permit ranges of values or choices in a number of design and manufacturing planning systems. For example, Baykan and Fox (1992) consider the constraint propagation mechanism used in their constraint based 2D layout system—WRIGHT—as a least commitment-based approach in two aspects: it only removes from variable ranges those values that violate a constraint, and it selects constraints to be satisfied by design units instead of choosing specific locations.

ALADIN is an expert system developed for aiding materials, in particular aluminum alloys, design (Farinacci et al., 1992). It treats alloy design as a planning problem and uses a combination of least commitment and overcompensatory methods for planning. The relevant design variables can take ranges of values that "are kept as broad as possible until more data is present to force them to be restricted, which allows the system to avoid trial-and-error in selecting values."

Based on a design by least commitment approach, Mäntylä et al. (1989) proposed the use of *relaxed feature models* in a generative process planning system to avoid over-specifying geometric models to not restrict the subsequent manufacturing options. Following the design by least commitment approach, Hel Or et al. (1994) proposed the use of *soft constraints*, that is, constraints that need not be satisfied exactly, in a *probabilistic-constraints* scheme for implementing a relaxed parametric design paradigm.

Encouraging least commitment design practice is also one of the precepts driving the development of a feature-based thin-walled component design system reported in (Nielsen,

```
> (add-scs geom1 '(((height = 30))))                          [2B.1]

> (add-ocs geom1 '(((rotate 90 (y wcs)))))                    [2B.2]

> (display-geom geom1 :aspect 'ur)                            [2B.3]
<Location uncertain region of GEOM1>
        X-RANGE:
                Lower Bound — 15.0
                Upper Bound — 85.0
        Y-RANGE:
                Lower Bound — 10.0
                Upper Bound — 90.0
        Z-RANGE:
                Lower Bound — 10
                Upper Bound — 75

> (refine-ur geom1 :z- 30)                                    [2B.4]

> (display-geom geom1 :aspect 'ur)                            [2B.5]
<Location uncertain region of GEOM1>
        X-RANGE:
                Lower Bound — 15.0
                Upper Bound — 85.0
        Y-RANGE:
                Lower Bound — 10.0
                Upper Bound — 90.0
        Z-RANGE:
                Lower Bound — 30
                Upper Bound — 75

> (make-geom)                                                 [2B.6]
GEOM6

> (add-lcs geom1 '(((above geom6))))                          [2B.7]

> (display-geom geom1 :aspect 'ur)                            [2B.8]
<Location uncertain region of GEOM1>
        X-RANGE:
                Lower Bound — 15.0
                Upper Bound — 85.0
        Y-RANGE:
                Lower Bound — 10.0
                Upper Bound — 90.0
        Z-RANGE:
                Lower Bound — 20
                Upper Bound — 75
```

Fig. 18. Interaction Scene 2—Part B.

1991). Minimum commitment design is encouraged by supporting the use of abstract feature-forms that can be modified incrementally.

6.2.2. *Spatial layout systems*

The arrangement aspect of the GEMCON system is related to systems developed for spatial layout, in particular, those that permit the description of layout problems in high-level spatial relations and inequality/equality constraints, and represent value ranges for dimensional and locational variables. For instance, Chambon and Tollenaere (1987) described a rule-based expert system for sequentially placing mechanical components in 3D space. The model of a 3D component contains the description of its geometry and position in a space. The geometric description consists of two levels: a fixed solid geometry such as cylinders, "parallelepipeds," etc., and a parallelepiped enclosing the entity. The position of an entity is described by three coordinates, each of which is composed of "a reference that can be another coordinate or the absolute fixed reference" and "a gap that can be a number or an interval $[X_{min}, X_{max}]$." Qualitative relationships, such as *against_direction_x, centered_on, facing*, are used to describe the constraints on the placement of entities. The system first selects the entity to be placed based on certain if–then rules, and then for the selected entity suggests predicates consisting of qualitative relationships or actions that create new entities. These qualitative relationships are then checked for their compatibility with one another, and translated into a numerical position. A list of entities in conflict with the entity being placed is formed for freedom reduction. Conflict between two entities is then solved based on the strategy of "keeping the largest freedom space on the current entity" being placed. The system selects objects with an age criterion and tries to solve the conflict with the entities in the list sequentially starting from the first by finding a freedom parallelepipedic solution for each entity involved.

The LOOS system by Flemming et al. (1989) is a spatial layout system that uses a hierarchical generate-and-test ap-

```
> (make-geom :size '(((depth = 24 -> 27)) ((width ^= 5)))                [3.1]
             :location '(((left geom1)) ((left geom6))))
GEOM7

> (display-geom geom7 :aspect 'size-value)                              [3.2]
<Sizeof GEOM7>
    WIDTH:
            Lower Bound — 4.75
            Upper Bound — 5.25
    DEPTH:
            Lower Bound — 24
            Upper Bound — 27
    HEIGHT:
            Lower Bound — 10.0
            Upper Bound — 10.0

> (add-scs geom7 '(((depth >= 8))))                                     [3.3]

> (display-geom geom7 :aspect 'size-value)                              [3.4]
<Sizeof GEOM7>
    WIDTH:
            Lower Bound — 4.75
            Upper Bound — 5.25
    DEPTH:
            Lower Bound — 24
            Upper Bound — 27
    HEIGHT:
            Lower Bound — 10.0
            Upper Bound — 10.0

> (set-strategy 'sc 'time)                                              [3.5]

> (add-scs geom7 '(((depth >= 8))))                                     [3.6]
Using ((DEPTH GEOM7) >= 8).
Reasons are:
         The current conflict resolution strategy is set as:
                TIME
         ((DEPTH GEOM7) >= 8) is more recent than ((DEPTH GEOM7) = 24 -> 27)

> (display-geom geom7 :aspect 'size-value)                              [3.7]
<Sizeof GEOM7>
    WIDTH:
            Lower Bound — 4.75
            Upper Bound — 5.25
    DEPTH:
            Lower Bound — 8
            Upper Bound — 27
    HEIGHT:
            Lower Bound — 10.0
            Upper Bound — 10.0
```

Fig. 19. Interaction Scene 3.

proach based on rules. Given a set of design units, it tries to derive feasible layouts of orthogonal rectangles where no two rectangles overlap. A set of four spatial relations, *above, below, to the left* and to *the right*, are supported. Additionally, the lower and upper bound of the coordinates of each rectangle, which define the dimensional range of the rectangle, are represented explicitly. An extension of LOOS—ABLOOS—was later developed to enable the hierarchical decomposition of a layout task (Coyne, 1991). There, the LOOS methodology is applied recursively to layout subtasks at appropriate levels of abstraction within a hierarchy.

The WRIGHT system (Baykan & Fox, 1992) provides a larger set of spatial relations (38 in total) to describe the topological relationships between design units. These spatial relations are defined in terms of algebraic relations on lines. New relations may be defined through a grammar. The system also allows a user to specify *greater-than, greater-than-or-equal-to, less-than, less-than-or-equal-to*, and *equal-to* types of unary or binary constraints on the dimensions of design units. Design units as a type of architectural primitive objects include entities such as kitchen, door, window, wall, sink, sink-center, etc. The geometry of a design unit is modelled as a 2D rectangular shape represented by eight variables, that is, *north-line, south-line, east-line*, and *west-line* for its location, *length, width, area* for the dimensions, and *orientation*. The location and dimension variables are represented as intervals with an initial value of $[-\infty, \infty]$ and $[0,\infty]$, respectively. The value of the orientation variable is one of $\{0, 90, 180, 270\}$. Spatial layout in WRIGHT is viewed as generating configurations of design units that satisfy given spatial relations and limits on dimensions. Local propagation based on interval arithmetic is used

```
> (add-scs geom7 '(((width = 110)))                          [4.1]
Using ((WIDTH GEOM7) = 110).
Reasons are:
        The current conflict resolution strategy is set as:
                AUTOMATIC
        ((WIDTH GEOM7) = 110) is more recent than ((WIDTH GEOM7) ^= 5)
There is not enough space available in the X direction
for placing GEOM7.
Please choose from the following:
    0 : Abort
PICK: 0

> (set-gcs-size geom7 :all+ 300)                             [4.2]

> (add-scs geom7 '(((width = 110)))                          [4.3]

> (display-geom geom0 :level 2)                              [4.4] (Figure 21(a))

> (delete-geom geom7)                                        [4.5]

> (display-geom geom0 :aspect 'local-cs)                     [4.6] (Figure 21(b))

> (display-geom geom0)                                       [4.7] (Figure 21(c))
```

Fig. 20. Interaction Scene 4.

to maintain the consistency of a configuration. The approach is regarded by Baykan and Fox as a least-commitment approach as described earlier.

6.2.3. Constraint-based geometric design systems

In the GEMCON system, constraints provide a tool for describing and reasoning about geometric configuration information. Constraint handling has been used in variational/parametric geometry systems (Light & Gossard, 1982; Solano & Brunet, 1994). In general, a variational geometry system represents the geometry of an object by a set of characteristic points. Dimensional and other constraints, such as *tangency*, are treated as the defining relations between the characteristic points on the object, and are interpreted as nonlinear/linear equations on the coordinates of the characteristic points. While capable of modelling a single object with more complex shapes, these systems in general do not seem to provide suitable means for qualitatively exploring spatial arrangements of multicomponent products, nor a means of recording and using approximate information. Furthermore, constraints on the geometry of an object require to be fully defined for the object to be modeled in such a system.

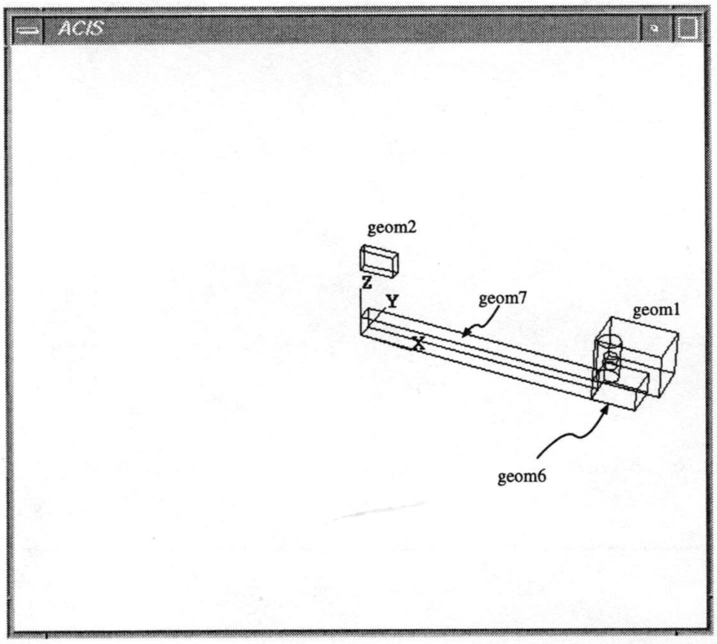
As discussed earlier, the *probabilistic-constraints* scheme reported by Hei-Or et al. (1994) incorporates *soft constraints* in a parametric geometric system. The amount of rigidity with which the constraint is to be satisfied is specified by a softness function. The relaxed parametric model is treated as a static stochastic system. The softness functions of the constraints are expressed as covariance matrices. Kalman filter is used to solve the corresponding parametric system. A simple 2D parametric modeller has been implemented to test the algorithm developed.

Constraints are also supported in the feature-based thin-wall component design system developed by Nielsen (1991). They are captured in the form of "design intent," which is the sum of all "restraints" imposed on design. Restraints set the target values or ranges for chosen geometric parameters and are used to guide the modification of the geometric model of a component. Four levels of certainty—*unsure, less sure, sure, definite*—are also attached to the lower and upper bounds of a value or to a target value to indicate how sure one is of the range or target value. Feature-forms are represented by a set of "virtual boundaries," which are geometric abstractions such as midplanes, centerlines, and locating points. In using the system, one need not supply information required for completely defining the feature-forms in 3D.

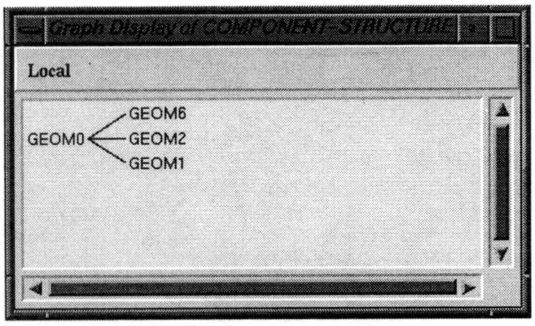
6.2.4. Uncertain shapes

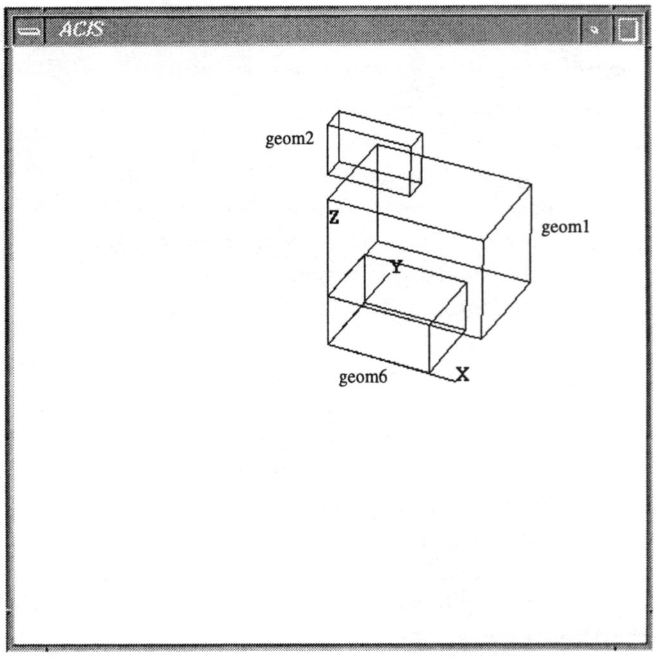
While the GEMCON system has not yet addressed issues related to the modelling of qualitatively described shapes via terms such as *small, large, narrow, broad, tall,* and *low*, they are certainly of great interest and related to the goal of the system. Martin (1994) has reviewed fuzzy set based shape description methods and examined their ability to meet the varying requirements of modelling inexact shapes in different domains such as design (in particular, the early stages), manufacturing as well as computer vision.

Yamaguchi et al. (1993) presented a *probabilistic solid modelling* scheme for representing and manipulating uncertain shapes to suit the early stages of geometric design. They have introduced the concept of a *probabilistic solid*, which can be seen as a fuzzy set of points of E^3 with a membership function associated. In this scheme, a probabilistic solid is created by applying a *distributing operation* to a deterministic solid (as found in conventional solid modelling sys-



(a) Configuration space of geom7 is enlarged to accommodate the change on its width.





(b) Upper most level of component structure of the configuration after geom7 is deleted.

(c) Corresponding graphic display of an instance of the configuration.

Fig. 21. The configuration before and after *geom7* is deleted.

tems). Two types of manipulations, *set operations* and *local modifications*, can then be applied to the probabilistic solid. Finally, a deterministic solid is obtained from the probabilistic solid by evaluating its boundary.

Table 3 presents a brief summary of the main features of the systems discussed above and the GEMCON system presented in this paper. Development of the GEMCON system is guided by the principles of minimum commitment and

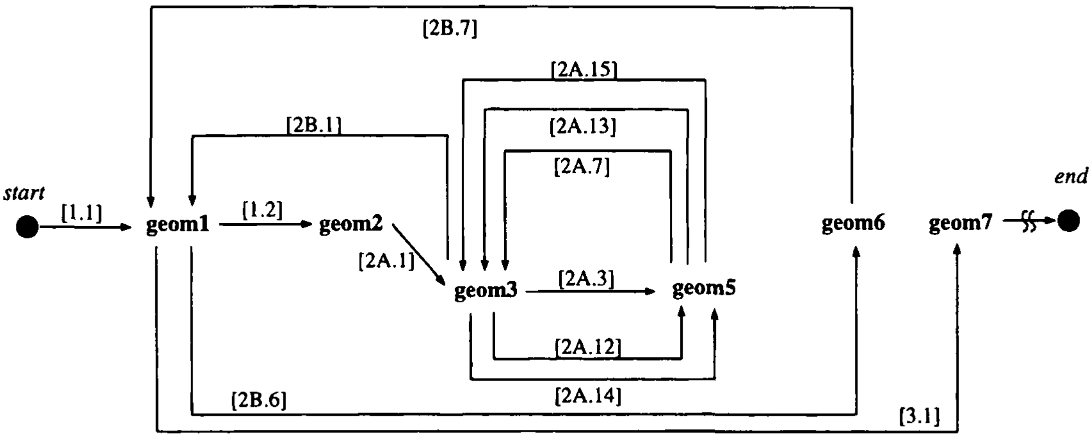


Fig. 22. The process of developing the example configuration.

incremental refinement. The minimum commitment principle is embodied in the novel interpretation in the system of geometric constraints (that define geometric configuration vaguely or precisely) as defining the solution space of the corresponding geometric configuration. The system enables an iterative, incremental, and nonsequential development process, which supports the minimum commitment-based modelling of configurations using simple geometric information. Finally, the system plays the role of an Intelligent Design Assistant and is not domain specific.

6.3. Limitations and further research issues

The capacity of the GEMCON system in its present form has a number of limitations.

- It can model only primitive shapes. Thus, a user is unable to refine a simple, preliminary shape of an object incrementally into a more complex or detailed one as design progresses.
- Cubical GCS and orthogonal bounding boxes of objects are used in reasoning about the bounds of cubical UR of objects. This means that, for objects with shapes other than cuboids, a) for a *contained-by* relation, URs derived are only an approximation, b) only loosely arranged geometric configurations can be developed.
- The UR bounds of an object specified via spatial relations are generated with respect to the corresponding reference objects and configuration space. It does not consider the presence of other geometric objects in the same configuration which are not related to the object spatially. Consideration of the presence of these surrounding objects in reasoning about the URs would require the system to be able to represent and reason about URs with at least polyhedron shapes (see Fig. 23 for example).

Consequently, we propose the following research directions to further the work reported in this paper.

- Enrichment and extension of the modelling language developed in the current system to satisfy the need for incremental modelling and refinement of geometric configurations from vague to completely and precisely defined models.
- Development of the representation structure and reasoning mechanism for supporting such a modelling language.
- Development of a facilitating user interface for presentation and rapid input of vague information.

7. SUMMARY

To overcome the inability of traditional CAD systems to model vaguely specified early geometric concepts, this paper has presented a prototype system that is being developed for the rapid capture and interactive development of vague geometric models. The geometry of an object is characterized by meaningful geometric design parameters. High-level constraints are used for describing the geometry of an object, and are satisfied to obtain the values of the design parameters. These values are kept explicitly and passed, when necessary, to the underlying geometric modelling system to construct boundary models. The system considers rapid and qualitative spatial configuration as a significant element of early geometric design, where a designer investigates the structural or topological organization of a product without committing to unnecessary details. A set of basic location constraints have been incorporated to support this task. The system supports a designer in establishing geometric configurations using geometric constraints such as primitive inequality, range, or equality size constraints and spatial relations or point positions. The novelty here lies in the way the constraints are interpreted in the system. Instead of rep-

Table 3. *Related works—summary of features*

Approach	Relevant Works: Minimum Commitment: Adopted?	Embodiment	Modelling/Working Process	User-System Relation	Computational Approach to Geometry: Representation: Object Levels	Shape	Size	Orientation	Location	Reasoning	Dimensionality	Domain
GEMCON	yes	interpretation of constraints: size range, location UR	nonsequential/ flexible, incremental refinement, iterative modification	interactive IDA based	yes	primitives	ISC: inequality range equality	rotational: in four directions	spatial relations, point position, datum point	constraint based	3D	generic
Chambon and Tollenaere (1987)	no	N.A.	sequential	expert system (automated)	?	fixed solids (cylinders, parallelpipeds)	?	four directions	object center, exact/interval value for the coordinates of the center, symbolic relations	rule-based, numerical checking	3D	mechanical components (hydraulic manifold blocks)
LOOS (Flemming et al., 1989) ABLOOS (Coyne, 1991)	no	N.A.	sequential expansion of patial solution, pruning of less promising candidates	automated	no yes	orthogonal rectangles (corner points)	lower + upper bounds of corner point coordinates	no	four spatial relations	rule-based, hierarchical generate-and-test, search	2D	generic
Mäntylä et al. (1989)	yes	relaxed feature models, relaxation groups	⟵ relaxed feature models, relaxation groups, feature-based geometric modelling ⟶								3D	manufacturing process planning
Nielson (1991)	yes	abstract feature-forms, value ranges?	iterative/incremental modification	interactive	yes	⟵ feature-based + abstract feature-forms + design intent ⟶					3D	thin-wall component design
ALADIN (Farinacci et al., 1992)	yes	value ranges	sequential	expert system automated	⟵ N.A. ⟶							material design
WRIGHT (Baykan & Fox, 1992)	yes	variable ranges, spatial constraints than specific location	?	problem definition by user, automated solution finding by system	no?	rectangles (four boundary lines and their coordinates)	algebraic equations & inequalities on dimensions	four directions	over 30 algebraic relations on lines constraining their locations	constraint satisfaction, heuristic search	2D	spatial layout in architectural context
Hei-Or et al. (1994)	yes	soft constraints	⟵ parametric/variational modeller + soft constraints + solving by kalman filter ⟶								2D	generic
Yamaguchi et al. (1993)	no	N.A.	as traditional geometric modelling systems	interactive	?	⟵ probabilistic solid modelling (probabilistic solids + set operations + local modifications) ⟶					3D	generic

Notation: ?—unknown; N.A.—not applicable.

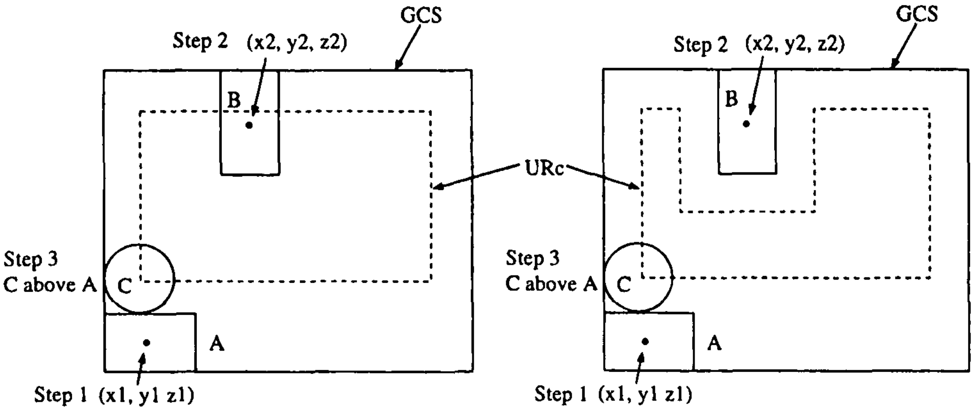


Fig. 23. Consideration of surrounding objects as well as the reference objects requires representation of and reasoning about at least polyhedron URs.

resenting only distinct solution points that satisfy the constraints, the system reasons about, maintains and represents the configuration solution space, that is, the location uncertain regions and size ranges, defined by these constraints. Furthermore, the system is designed to support the incremental definition of a configuration model using fragments of information that become available as the design progresses. A configuration does not need to be fully and uniquely constrained before it can be modelled by the system. In other words, constraints can be introduced into a configuration incrementally as a design progresses. Finally, use of the system in developing a configuration does not require the user to follow a specific, fixed, or predefined sequence. Namely, it supports a nonsequential process of geometric configuration design. Thus, the system supports a minimum commitment, incremental, and iterative approach to geometric configuration design.

ACKNOWLEDGMENTS

The research reported in this paper was carried out within the CAD Centre, University of Strathclyde, U.K. The authors acknowledge the support of EPSRC, U.K. (Grant No. GR/J11409), and thank their colleague, David Stevenson, for providing Figure 1.

REFERENCES

Asimow, M. (1962). *Introduction to design.* Prentice-Hall, Inc., Englewood Cliffs, NJ.

Baykan, C.A., & Fox, M.S. (1992). WRIGHT: A constraint based spatial layout system. In *AI in Engineering Design*, (Tong, C. and Sriram, D., Eds.), Vol. I, pp. 395–432. Academic Press, Inc., New York.

Chambon, R., & Tollenaere, M. (1987). An expert system for objects placing in three-dimensional space. In *Knowledge Based Expert System in Engineering: Planning and Design*, (Sriram, D. and Adey, R.A., Eds.), pp. 449–459. Southampton, U.K.: Computational Mechanics Publications.

Coyne, R.F. (1991). *ABLOOS: An Evolving Hierarchical Design Framework. EDRC 02-15-91.*

Dym, C.L. (1994). *Engineering design: A synthesis of views.* Cambridge University Press, New York.

Farinacci, M., Hulthage, I., Rychener, M., & Fox, M. (1992). ALADIN: An innovative materials design system. In *Artificial Intelligence in Engineering Design* (Tong, C. and Sriram, D., Eds.), Vol. II, pp. 215–262. Academic Press, Inc., New York.

Flemming, U., Coyne, R.F., Glavin, T., Hung, H., & Rychener, M.D. (1989). A *generative expert system for the design of building layouts.* Technical Report EDRC48-15-89.

Gross, M.D. (1996). The electronic cocktail napkin—A computational environment for working with design diagrams. *Design Studies 17(1)*, 53–69.

Guan, X., & MacCallum, K.J. (1995). Modelling of vague and precise geometric information for supporting the entire design process. In *Knowledge Intensive CAD-1*, (Tomiyama, T., Mäntylä & Finger, S., Eds.), Preprints of the 1st IFIP WG5.2 Workshop, pp. 301–319.

Guan, X., & MacCallum, K.J. (1996). Adopting a minimum commitment principle for computer aided geometric design systems. In *Artificial Intelligence in Design '96*, (Gero, J.S. and Sudweeks, F., Eds.), pp. 623–639, Kluwer Academic Publishers, New York.

The Harlequin Group Limited. (1994). *Lisp works user's guide.*

Hel Or, Y., Rappoport, A., & Werman, M. (1994). Relaxed parametric design with probabilistic constraints. *Computer-Aided Design 26(6)*, 426–434.

Heintze, N., Jaffar, J., Michaylov, S., Stuckey, P., & Yap, R. (1991). *The CLP(R) programmer's manual*, version 1.1., IBM, Thomas J. Watson Research Centre.

Light, R., & Gossard, D. (1982). Modification of geometric models through variational geometry. *Computer-Aided Design 14(4)*, 209–214.

MacCallum, K.J., Duffy, A.H.B., & Green, S. (1985). An intelligent concept design assistant. *Proc. of the IFIP WG5.2 Working Conference on Design Theory for CAD*, 301–317.

Mäntylä, M., Opas, J., & Puhakka, J. (1989). Generative process planning of prismatic parts of feature relaxation. *Proc. ASME Design Automation Conf.*, 49–60.

Martin, R.R. (1994). Modelling inexact shapes with fuzzy sets. In *Set-theoretic Solid Modelling: Techniques and Applications.* Information Geometers Ltd.

Moore, R.E. (1979). *Methods and applications of interval analysis.* SIAM, Philadelphia.

Nielsen, E.H. (1991). *Designing mechanical components with features: Representing the form and intent of in-progress design for automated modification and evaluation.* PhD Dissertation. Department of Mechanical Engineering, University of Massachusetts.

Requicha, A.A.G. (1980). Representations for rigid solids: Theory, method, and systems. *Comput. Surveys 12(4)*, 437–464.

Retz-Schmidt, G. (1988). Various views on spatial prepositions. *AI Magazine 9(2)*, 95–105.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). *Object-oriented modelling and design.* Prentice-Hall International Editions, Englewood Cliffs, NJ.

Solano, L., & Brunet, P. (1994). Constructive constraint-based model for parametric CAD systems. *Computer-Aided Design 26(8)*, 614–621.

Spatial Technology Inc. (1992). *ACIS: Interface guide*, Boulder, CO.

Weld, D.S. (1994). An introduction to least commitment planning. *AI Magazine 15(4)*, 27–61.
Yamaguchi, Y., Nakamura, H., & Kimura, F. (1993). Probabilistic solid modelling: A new approach for handling uncertain shapes. In *Geometric Modelling for Product Realisation* (Wilson, P.R., Wozny, M.J. and Pratt, M.J., Eds.), pp. 95–108. Elsevier Science Publishers B.V., Amsterdam.

Xiaohong Guan is a senior consultant at Industrial Systems and Control Ltd, Glasgow, U.K. Before joining the company, she worked as a research assistant at the CAD Centre, University of Strathclyde. She obtained a B.Sc. in Radio Electronics Science from University of Sichuan, China, in 1984, an M.Sc. in Information Engineering from Xidian University, China, in 1987 and a Ph.D. from University of Strathclyde, U.K. in 1994. She is a member of AAAI and IEEE.

Ken MacCallum obtained his first degree in Naval Architecture from the University of Glasgow, proceeding to postgraduate study in Imperial College, University of London where he obtained a Ph.D. for research into the application of computer graphics to free-form surface design. After 3 years with a software company, he joined the University of Strathclyde, establishing the CAD Centre in 1985 as a research and postgraduate center. He is currently the Head of Design, Manufacture, and Engineering Management in the Faculty of Engineering at the University of Strathclyde. Professor Ken MacCallum's main area of research has been the application of Artificial Intelligence to Engineering Design. He has led projects concerned with intelligent design modelling, data exchange, computer-based design coordination, and computer-aided learning. He is editor of the International Journal on Artificial Intelligence in Engineering, is a member of IFIP WG5.2, and has been on the Technical Programme Committees of a large number of Conferences and Workshops concerned with computer aided design.

Alex Duffy completed a shipwright designer/draughtsman apprenticeship and a further 2 years in the shipbuilding industry before going to the University of Strathclyde to obtain his degree in Naval Architecture and a Ph.D. on knowledge-based computer support for conceptual engineering design. He is presently a Senior Lecturer and Director of the Computer Aided Design Centre, University of Strathclyde. He lectures in engineering design, computer-aided design (CAD), expert systems, knowledge-based techniques in engineering, and databases. His main research interests have been the application of knowledge based techniques in early stage design, product and product knowledge modelling, machine learning techniques and past design utilization, and design coordination. He is the leader of a European (EU) Basic Research thematic network subgroup working in Design Coordination, is on the advisory board of various international conferences and editorial board of Research in Engineering Design Jrnl., chairs the International Engineering Design Debate, and is a member of IFIP Technical Committee Working Group 5.8.