Design for Assembly Evaluation of Orientation Difficulty Features

Jui-Te Yang, R. H. Sturges

EDRC 24-78-92

Design for Assembly Evaluation of Orientation Difficulty Features

R.H. Sturges
Assistant Professor

Jui-Te Yang
Research Assistant

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890
November 1991

Abstract

In support of the effort to bring downstream issues to the attention of the designer as parts take shape, an analysis system is being built to extract certain features relevant to assembly processes, such as the dimensions, shape, and symmetry of an object. These can be applied'to a model of the downstream process to evaluate the object's difficulty of handling and assemblability. In this paper, we will focus on the acquisition phase of the assembly process and employ a model of design for assembly (DFA) evaluation to quantify factors in this process. The capabilities of a non-homogeneous, non-manifold boundary representation geometric modeling system are used with an Index of Difficulty (ID) that represents the dexterity and time required to assemble a product. A series of algorithms based on the high-level abstractions of loop and link are developed to extract orientation difficulty feature data forming a DFA critic. Examples of the testing the robustness of the algorithms are given. Problems related to nearly symmetric outlines are also discussed.

1. Introduction

Traditional Design For Assembly (DFA) methods provide a viable avenue for cost reduction a process simplification [Boothroyd 1988]. Even if no change to the assembly process is indicated by such analyses, one can employ a DFA method to make a design better in other manufacturing respects [Rooks 1987]. Whether qualitative or quantitative, traditional DFA is carried out almost exclusively with a high degree of "manual interaction" among the design, the designer and the method. In other words, a high level of skill is needed to interpret the design with respect to the DFA procedure.

With increasing use of computer aided design (CAD) systems, designers may benefit from a tool that can automatically find some predictable difficulties related to assembly and give reasonable suggestions. With such a tool, a designer could more easily satisfy the requirements of function while taking into account of some of the factors pertaining to assembly.

In this paper, we briefly review a quantitative DFA method based on assemblability difficulty factors [Sturges & Wright 1989] and focus on the acquisition phase at the system level. The background and perspective of an Integrated Design for an Assembly Evaluation and Reasoning System (IDAERS) [Sturges & Kilani 1990] are discussed in section 2. Algorithms comprising the orientation difficulty evaluator, "Boss/Groove Critic," and how it works are discussed briefly in section

3. Examples illustrating its application areas are given in section 4, and a discussion of what we call the "near-symmetry problem" will be addressed in section 5.

2. Perspective of Automated DFA Evaluation

2.1 Background

The ability to integrate CAD and DFA rests on, at least, two important issues: The first is to evaluate the degree of difficulty associated with a given mechanical system and assembly (MSA) and refer this difficulty to total cost. Much prior work has been done in qualitative evaluations relative to assemblability (e.g., [Baum & Gabriele 1989], [Miles 1982], [Miyawaka & Toshirjo 1986].) In other research, a quantitative analyses based on empirical data [Boothroyd & Dewhurst 1988], and an Index of Difficulty (ID) [Sturges 1989] refer assembly difficulty to time and/or cost. The quantitative methods recognize a two-phase assembly process comprising acquistion and insertion activities. The ID approach attempts to measure the dexterity and the time required to assemble a part, and indicates ten significant measurable factors in assembly.

The second issue is feature extraction, linking a CAD part database to the assembly process. Syntactic grammar methods [Staley & Anderson 1983] use the formal definition of features in terms of a graph grammar and perform pattern recognition to recognize the features [Finger & Safier 1990]. The definition of a feature under this model becomes rapidly complex when dealing with 3D objects. Volumetric approaches use volume properties to extract features from solid models and apply them to certain primitives, forming a constructive solid geometry (CSG) tree representation. Unfortunately the representation of a parts by a CSG tree is not unique [Mantyla 1988J. Loop-based feature abstraction has recently been proposed (Gadh & Prinz 1991a], and to an extent has proven successful in recognizing features despite variations in topology and geometry, and combination complexity [Gadh & Prinz 1991b].

The above survey shows that we can use the higher-level abstraction "loop" to define some features which are applicable to the general problem, and we can evaluate the assemblability quantitatively by using the Index of Difficulty (ID). Our approach to assembly evaluation is based on these two concepts.

2.2 The EDAERS Model

An Integrated Design for Assembly Evaluation and Reasoning System (IDAERS) [Sturges & Kilani 1990] models ten significant measurable difficulty factors pertaining to assemblability. It categorizes them into 6 groups: 3 levels (component, system, and process) by 2 phases (acquisition and assembly.) These are shown schematically in Fig. 2.1. Quantitative evaluation of these factors is provided, in part, by an ID based on human motor capacity [Fitts 1954] (see equation (1) below). The architecture of a proposed IDAERS is shown in Fig. 2.2.

The development of the algorithms in support of this model is outlined in the function block diagram (FBD) of Fig. 2.3. An FBD, in contrast to a flow chart, shows how and why the elements of a design relate to themselves and the basic function being satisfied, or the larger problem being solved. From this illustration, we can also identify previous and current related work. For example: the capabilities of a non-homogeneous, non-manifold boundary representation geometric modeling system [Gursoz 1990] reside in the *"NOODLEZS"* function and ongoing research in component interactions [Wong & Sturges 1991] is identified. In the next sections, we present the results of our work on system ID evaluation, i.e., finding object orientation, features, and difficulty.

3. A Boss/Groove Critic

As will be seen from Figure 2.3, one of the ten significant measurable factors pertaining to assemblability is the orientation difficulty.. Since this factor derives from the bosses or grooves on a part, its evaluation function is termed the "Boss/Groove Critic."
To fully evaluate the assemblability of a part design with respect to acquistion, we need to quantify this factor as well as part size, handling distance, shape, and other handling conditions. Part size and handling distance are readily derived from a model of the part and the assembly workstation. Part shape is more subtle and will be discussed in section 5, below. The ID for orientation difficulty is based on the ratio of overall part size to the boss or groove dimensions of the part.

The dimensions of these features and the related part dimensions need to be extracted from the geometric model. To this end, several functions which constitute the Boss/Groove Critic were constructed. Section 3.1 describes the approach and the concepts behind the procedure. The major functional elements of the Boss/Groove Critic and a brief description of the algorithms are given in sections 3.2 and 3.3, respectively. The •'Boss/Groove Critic" is presently implemented on a SUN workstation, in the C language, and requires the NOODLES solid modelling system.

3. 1 Algorithm Development

Object features are usually the key elements used for performing assemblability and manufacturability evaluation. There are many ways to extract these features, such as using a CSG tree representation [Lee & Fu 1987], a graph grammar [Finger & Safier 1990], a syntactic rule [Staley & Anderson 1983], or topological data like faces, edges, loops, etc. [Henderson 1984]. However, none are applicable to the general case, take significant time due to the nature of graph matching, and are inflexible for small variations in a feature [Gadh & Prinz 1991a] [Mantyla 1988]. Therefore, we initially attempted to combine the nature of a CSG tree with pattern matching to form a new process which could deal with more cases, and do so faster.

Algorithms were developed utilizing rectangular and triangular *bounding boxes* to deal with a model with straight and inclined surfaces. Then, through performing recursive boolean operations on the model and its corresponding bounding box, a CSG tree was developed. Recognition of the elements in the CSG tree by pattern matching **and** recognizing the object's feature sizes was then carried out. This method was found not to be robust, because many exceptions appeared in the evaluation process [Kilani & Sturges 1991]. Since only the *dimensions* of each feature of an object needed to be recognized in order to evaluate its assemblability, and the only features which needed to be distinguished were straight lines, curves, and holes, then there appeared to be no need obtain detailed information about such features as a through hole, or a non-through hole. When calculating an ID, one only has to recognize if the analyzed portion is a curve or an edge and obtain both its dimension and the overall one in the related direction. In short, one needs to obtain the sizes of part features and their orientations, not the types of features in an object in an abstract sense.

Two other facets of feature extraction bear noting here: The first is that comparison between 2D and 3D models shows that the solid ones contain much more variation and associated information than the planar models. Thus, if one tries to deal with a boundary representation of a solid model, 2D abstractions will be much easier to deal with than 3D models. The second is that low level abstractions (edges, vertices, and faces) provide very detailed information but are inadequate to recognize features precisely and generally [Gadh & Prinz 1991b]. These observations led to the two main concepts behind our wurrent approach to a Boss/Groove Critic. They are:

J. *Use high level abstraction to define the needed features and to restrict the search domain to recognize the desired features.*

Because loop abstraction conveys more information than low level abstractions (edges, etc.) the former is used to meet the function mentioned above. This provides us with information about a full cycle of a convexity or a concavity and makes definitions of features among different models uniform. Other higher-level abstractions can also reduce the number of low level abstraction entities and restrict them to the domain we interested in (for example, a particular face in a model). The feature sizes of both the single part of an object and the whole object can be obtained by picking the "filtered" low level abstraction entities, and from them evaluate the ID. This approach avoids ambiguity and is more applicable to general cases.

2. Instead of dealing with a 3D model directly, try to abstract 3D models into 2D models by several methods, and then interpret the 2D models as representations of the 3D models.

Three 2D representations of the solid model are used in the Boss/Groove Critic: imprints, silhouettes, and slices provide different perspectives with respect to the 3D models. The Silhouette Method and Imprint Method are employed first, as the Slicing Method uses data from the other two to determine the minimum number of slices required in a specified direction. The results of the three methods are compared to cross check and produce the most important ID value and its cause [Yang & Sturges 1991].

In short, higher-level abstraction entities are used to reduce the complexity of obtaining feature sizes in representing the assemblability of an object by an ID. This overcomes difficulties in the sequential process of decomposing the object, recognizing the decomposed elements, finding elements which exist only in combination with other features, and then recognizing the sizes of those elements.

3.2 Algorithm Structure

The Boss/Groove Critic is described with the following structure and purposes:

J. *Extract 2D model faces from 3D original model solid*

Evaluate the ID for assemblability by using equation (1), which can quantifies the factors affecting the assembly process based on human motor capacity [Sturges & Wright 1989]:

$$ID\text{-}\log_2(s/w), \qquad (1)$$

where *s^f is the overall size and V is the feature size.

We observe that 2D analyses of a 3D model are sufficient if the views and methods are appropriate. By using the Slicing Method, Imprint Method, and Silhouette Method in parallel along XY, YZ, and ZX object planes respectively, we can construct an ID list sufficient to indicate 3D model characteristics. These methods are discussed in detail below.

2. *Find an appropriate part orientation, rotate the part back from a given arbitrary orientation, and then locate it on the fixed coordinate frame.*

In representing a 3D model through a number of 2D models, one employs projections, such as the Slicing Method. Since there are essentially an infinite number of orientations of the 3D model, choice of a sufficient number of "appropriate views'* needs to be bounded and defined. A rule called the "Maximum Projecting Area Rule," explained in section 3.3, is used here. It finds a limited set of appropriate orientations with respect to a fixed coordinate frame defined in the CAD modeler, and rotates the originally arbitrarily-oriented 3D model to each appropriate orientation with respect to the fixed coordinate frame. After each such reorientation, one uses the X, Y and Z directions to extract the information needed for equation (1). This implies that after reorientation, all features existing in the 3D model will be mapped onto 2D models in those directions. This procedure saves a lot of time and simplifies the analysis.

3. *Recognize holes in the model part and eliminate them from consideration.*

Holes in an object are difficult to detect and employ for orientation purposes [Boothroyd, et al. 19871, and an appropriate way to judge how they affect the assemblability of their corresponding object has not yet been determined. Therefore, their existence is recognized but their effects are neglected. This limitation in the method should be the subject of future work.

4. Recognize a curve in a model and project its arc length in two orthogonal directions of interest

Limited by the representation abilities of many current solid modeling tools, one cannot create a "rear curve. A curve is most often composed of multi-faced polygons, such as a octagon used to represent a circle. Its smoothness depends on the number of segments chosen. Moreover, the ID will be different if one interprets those segments as a single curve instead of several separated edges. Appropriate application of equation (1) to curves requires a projection of the curve onto the axis of overall size. Thus one needs to group the individual segments together to form a curve. The ID can then be calculated by dividing the reference basis by the sum of projections of each segment in a curve. Ambiguities of interpretation of a series of segments as a curve or a set of flat edges are resolved by referring to the default value for curves in the modelling system.

5. Find the dimensions of bosses/grooves and sizes of the bounding box with respect to a part.

From equation (1), the ID is the ratio of overall size to boss/groove size. Thus, to get the ID of an object, one must recognize the size of each feature in a given direction, such as the length of an edge or the projecting length of a curve, and the reference basis or so-called overall size in the same direction. This reference is generally given by the size of the bounding box of a part in that direction. From these data a list of ID's is created. The largest ID in the list is reported to the user along with the features which gave rise to it.

3. 3 Algorithms Comprising the Boss/Groove Critic

Here we briefly describe the algorithms of the seven major functions of the Boss/Groove Critic. Detailed descriptions are found in [Yang & Sturges 1991].

r

J. *Hole_Detector()*

Holes on a plane are defined by using a loop as Tor any loop on a certain plane, if any one of the edges of that loop is on the boundary of the plane, then it cannot be hole/ This process for finding a hole is repeated recursively until all faces are scrutinized.

2. Curve_Finder()

The definition of a curve is "For each edge in a loop of a 2D model, if it is a member of a series of adjacent edges inside the bounding box and its difference of angle and ratio of lengths between it and its adjacent edge are both within the tolerance of angle and

length, it may be part of a curve. For a valid curve, the number of adjacent edges in a series should be greater than or equal to two." Note that two parameters, curvejevel and curvejength, used to define the tolerance equations (2) and (3), need be set by the user in conformance with the modelling system defaults for representations of curves.

Tolerance angle = 180° - (360° / curvejevel) £ 0, **(2)**

Tolerance length = curvejength £ Max(A/B,B/A) , **(3)**

where 0, A and B are shown in Figure 3.1.

3. RightJTiewQ

The rule used here is "if an object, in a certain orientation, has maximum contacting areas relative to the bounding box whose faces are all parallel to the fixed reference frame, then at that orientation we can see most of features of the model from the X, Y and Z directions." This rule generates alternative views, especially for axisymmetric objects. These can be checked sequentially with no loss of generality.

4. ImprintO

To get the imprint of a model with respect to its corresponding bounding box, we need to do two things. First, we examine each facet of a model and pick the facets touching the bounding box. Second, we test these facets to see if they are directed toward positive or negative X,Y, or Z directions. These collection of facets in the X,Y, and Z directions are what we want, assuming they exist.

5. *Silhouette!)*

The silhouette of an model is comprised of a set of three orthogonal projections for each appropriate orientation. To obtain the silhouette, we first distinguish the faces parallel to the fixed reference frame from the other faces. Then we project the slanted faces onto the planes parallel to the reference frame, and union all faces in the same direction.

6. SlicingO

We cut the model along three axes on the fixed reference frame at equal intervals. The sufficient number of slices, N, in a given direction depends on the ratio of the largest dimension (overall size) to the smallest one obtained from by the Silhouette Method and Imprint Method. If the ratio N is an integer, the number of slices (including faces on both ends) in a direction is N+l, otherwise, use Truncate (N)+2.

7. IDJEvaluatorO

To quantify the assemblability, compute the ID's in a assembly process according to equations (1) and/or (4). Equation (1) is applicable to all bosses and grooves and indicates the general ID. Equation (4) applies to plain figures which are nearly square, i.e., for the case that b and a are not equal, and it represents the potential difficulty caused by the nearly symmetric parts.

$$\mathbf{ID = \log_2 (Max(b,a)/ Abs(b-a))} \qquad \textbf{(4)}$$

where ^{f}b' and 'a* are the lengths of the 2D model in the X and Y directions, for example.

4. Case Studies

In this section, two examples are given to illustrate how the Boss/ Groove Critic works. One is a rectangle with a slot at the middle of one side, the other is a round-cornered cap, or binder. The ID based on the ratio of overall size to boss/groove size, aspect ratio, and the recognition of holes and curves will be discussed from three different viewpoints. Note that the two parameters curvejevel and curveJength need to be set in advance to discriminate a curve from a faceted or prismatic surface.

4.1 Case I: A Rectangle with a Slot on one side

4.1.1 Boss/Groove Features

The shape and dimensions of this model are illustrated in Figs. 4.1, 4.2 and 4.3. Here the curvejevel and curvejength are set to 10 and 1.5 respectively. The pertinent features of this model are the slot and the similarity of dimensions in Y and Z directions.

4.1.2 Imprint Method

The imprint code reported the following:

1. there is no hole nor curve found in the model;

2. the largest value of ID due to aspect ratio is 2.3219 (log2 (5/(5-4))).

3. the largest ID values are 1.0 (log2 (4/4), log2 (5/5), log2 (8/8)), 1.3219 (log2 (5/2)), and 2.0 (log2 (8/2)).

This data shows the difficulty due to the slot was identified, and that the difficulty caused by the similarity of dimensions in the Y and Z directions may result in a more serious problem in the assembly process than the slot does. In other words, it is easy to identify the orientation of the part by the slot, but more difficult by relying on only the outside dimensions.

4.1.3 Silhouette Method

The silhouette code reported the identical results as the imprint, as would be expected for this simple part.

4.1.4 Slicing Method

Here the number of slices in X, Y, and Z directions were computed to be 5 (8/2 +1), 4 (Trunc(5/2)+2), and 2 (4/4+1) respectively, because the smallest dimensions recognized in the Silhouette Method and Imprint Method are 2, 2, and 4 with respect to the X, Y, and Z axes. Its ID's shows the first three largest values as 2.0, 1.3219, and 1.0, plus the ID value from the aspect ratio as 2.3219. They are all consistent with the results shown above. In addition, it reported information about the *difference* of certain dimensions. For example, it reported an ID of 0.737 (Iog2 (5/3)), where 3 is the thickness of the material at the bottom of the groove in the Y direction, or, in other words, the difference of the height of boss and the depth of groove in Y direction. Since the ID was much smaller than the others, it can be safely neglected in this case.

4.2 Case II: A Round Binder

4.2.1 Boss/Groove Features

This model exhibits three-fold symmetry with respect to its X, Y, and Z directions (shown in Figs. 4.4, 4.5 and 4.6, respectively), and each one of its round surfaces is constituted by a 24-faced cylindrical segment. Here the curvejevel and curvejength are set to be 20 and 1.3 respectively. The pertinent features of this part are the rounded surfaces and the narrow slots.

4.2.2 Imprint Method

The imprint code reported the following:

1. there is no hole found in the model;

2. one curve was identified in each of the 3 directions, and the projections of the curve's length are all 12 in each plane. The other directions do not contain curves;

3. the ID value due to aspect ratio was not reported since the part dimensions in all directions are the same.

4. the largest ID values are 1.0 (Iog2 (24/12)), and 2.585 (Iog2 (24/4)), each value appearing three times;

5. the ID with respect to the curves are both 1.0 (Iog2 (24/12)) in two projection directions.

This data shows the difficulty due to the slot was identified, and the curves were appropriately recognized and evaluated. These ID's indicate that the handling problems of this part are most possible to appear due the slot, not the rounded surfaces.

4.2.2 Silhouette Method

For this part, the silhouette code only reports that the largest ID is 1.0, since the shadows of this part in three directions do not represent its slot features.

4.2.3 Slicing Method

For this part, the smallest dimensions are all 4 in each of three directions. Actually, it is symmetric with respect to the three axes. Slicing reported that the largest ID's are 1.0 and 2.585, and that the projections of the curved lengths are all 12 in each plane. It also found different lengths of curves from the conjunction of two 24-faced cylindrical segments. This effect can be expected to result in high ID's since the features appear subtle.

4.3 Summary

Generally speaking, all features existing in the model are recognized through these three methods, and the results obtained form three viewpoints are consistent with respect to the whole. Although one method like Silhouette Method sometimes can not identify all features, other two methods always can always complement the results to make the data complete. The Slicing Method will sometimes provide much more detailed information about the inside of the model than is needed for part acquistion, and may report higher ID values than would actually be experienced in practice. There is a need to devise a "filter" to eliminate some unnecessary information generated by the Slicing Method.

5. Discussion

Assemblability factors and higher-level abstractions were found to lead to the design of an effective "Boss/Groove Critic" which operates on geometric models. Currently, the Boss/Groove Critic can deal with many problems where a hole is negligible due to its insignificance in part handling, and where factors beyond the acquisition phase at system level are ignored. The Boss/Groove Critic can provide users with the first three highest Index of Difficulty (ID) values.

Algorithms related to symmetry are necessary to make the process more robust with respect to certain subtle classes of part outlines. Currently we employ aspect ratio detection to deal with a simple case of near-symmetry. There are many examples in the real world which will result in a large change in ID due to an apparently slight asymmetry. This near-symmetry problem is discussed with examples in [Kilani & Sturges 1991].

In future work, one needs to consider the symmetry of an object to deal with combining the effects of symmetry and feature sizes. In addition, the concept of "orientation tolerance* needs to be explored: for example, an object like a pine cone can have many faces, and their normal directions are different but similar. The "appropriate" way to orient such a part would be based on the tendency of their normal directions. With a suitable tolerance, one could then summarize several main directions and find the appropriate viewing orientation.

With an expansion of working algorithms in the assembly evaluation domain, more recommendations could be obtained automatically, such as how to make a design more symmetric. A larger knowledge base could be built based on this information, and a more intelligent assembly advisor can then be constructed.

(Word count: 4227)

References

Baum, J.P. and Gabriele, G.A., 1989. "Design for Assembly of Aerospace Structures. A Qualitative Interactive Approach," *SME Technical Paper* (series) MS89-158.

Boothroyd, G., Poli, C.R.. & Murch, L.E., 1987. *Handbook of Feeding and Orienting Techniques for Small Parts,* Department of Mechanical Engineering, University of Massachusetts, Amherst, MA 01003.

Boothroyd, G 1988. "Making It Simple: Design For Assembly," *Mechanical Engineering,* Vol. 110, No. 2, pp. 28-31.

Boothroyd. G. and Dewhurst, P. (1988), *Design for Assembly Handbook,* University of Massachusetts, Amherst, MA.

Gadh, R. & Prinz. F.B., 1991a. "Abstraction of Manufacturing Features From Design," *Proc. of Winter Annual Meeting,* ASME, Atlanta, GA, Dec. 2-5.

Gadh, R. & Prinz, F.B., 1991b. "Shape Feature Abstraction in Knowledge-Based Analysis of Manufactured Products," *Proc. IEEE Conf. on AI Application,* Miami, Florida. Feb. 24-28, pp &.

Finger, S. and Safier, S.A., 1990. "Representation and Recognition Features in Mechanical Designs," *Second International Conference on Design Theory and Methodology,* Chicago, Sep. 16-19, pp &.

E. L. Gursoz, 1990. "A User's Guide To NOODLES's Geometric Modeling System (version 7)," Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA 15213.

Henderson, M.R. and Anderson, D.C., 1984. "Computer Recognition and Extraction of Form Features: A CAD/CAM Link," *Computers in Industry,* pp. 329-339.

Kilani, M.I. and Sturges, R.H., 1991. "Detection and Evaluation of Orientation Features for CAD Part Models," EDRC Report No. 24-66-91, also to appear in *Journal of Engineering Design.*

Lee, Y.C. and Fu, K.S., 1987. "Machine Understanding of CSG: Extraction and Unification of Manufacturing Features," *IEEE CG\&A,* January, pp. 20-32.

Mantyla, M., 1988. *An Introduction to Solid Modeling,* Computer Science Press, Rockville, MD.

Miles, L.D., 1982. *Techniques of Value Analysis and Engineering,* 2nd ed., McGraw-Hill.

Miyawaka, S. and Toshirjo, O.f 1986. The Hitachi Assemblability Evaluation Method (AEM)," *International Conference on Product Design for Assembly,* Newport, RIt April 15-17.

Rooks, B., 1987. "Danish Innovation Focus on Design For Assembly," *Assembly Automation,* February, pp. 11-13.

Staley, S.M., Henderson, M.R. and Anderson, D.C. 1983. "Using Syntatic Pattern Recognition To Extract Feature Information From a Solid Geometric Data Base," *Computers in Mechanical Engineering,* September, pp. 61-66.

Sturges, R.H., and Kilani, M.I., 1990. Toward an Integrated Design for Assembly Evaluation and Reasoning System" EDRC Report No. 24-67-90, also to appear in *Journal of Computer Aided Design.*

Sturges, R.H., 1989. "A Quantification of Manual Dexterity: The Design for Assembly Calculator," *Journal of Robotics and Computer Integrated Manufacturing,* Vol. 6, No. 3, pp. 237-252.

Sturges, R.H. and Wright P.K., 1989. "A Quantification of Dexterity," *Journal of Robotics and Computer Integrated Manufacturing,* Vol. 6, No. 1. pp. 3-14.

Wong. J. H.-W., and Sturges, R.H., 1990. "An Extension of Design for Assembly Methods for Large and Heavy Parts" EDRC Tech. Report 24-60-90.

Yang, J.T. and Sturges, R.H., 1991. "DFA Evaluation of Orientation Difficulty Features," EDRC Tech. Report, No. 24-78-92, Carnegie Mellon University.

	Acquisition phase: Parts are brought from the feeding point to the assembly point.	Assembly phase: Parts are joined.
Component Level Factors: Factors pertaining to the individual component independent of the overall system.	• Part Size: Difficulty in handling small or thin parts. • Part Shape: Difficulty in rotating the part about different axes to align for assembly.	• Stability: Difficulty represented by parts which require restraint or extra manipulation te.g., parts that need to be held down temporarily or flexible parts).
System Level Factors: Factors dependent on the interaction between components.	• Boss or Groove Size / Feature Size: Difficulty in determining the correct orientation for the part to be acquired.	• Clearance: Difficulty that arises from the relative clearance between mating parts. • Direction: Difficulty arising from having to move the part in various insertion directions during assembly.
Process Level Factors: Factors dependent on the process employed to assemble the mechanical system.	• Handling Distance: Difficulty associated with bringing the part from the feeding point to the assembly point.	• Fastening Method: Difficulty presented by the method used to fix the part for assembly. • Assembly Path: Difficulty associated with the path the component follow during assembly process.

Figure 2.1 Assemblablity difficulty factors,
organized with respect to associated phase and level.
(Source: Sturges, R.H., and Kilani, M.I., 199.0.)

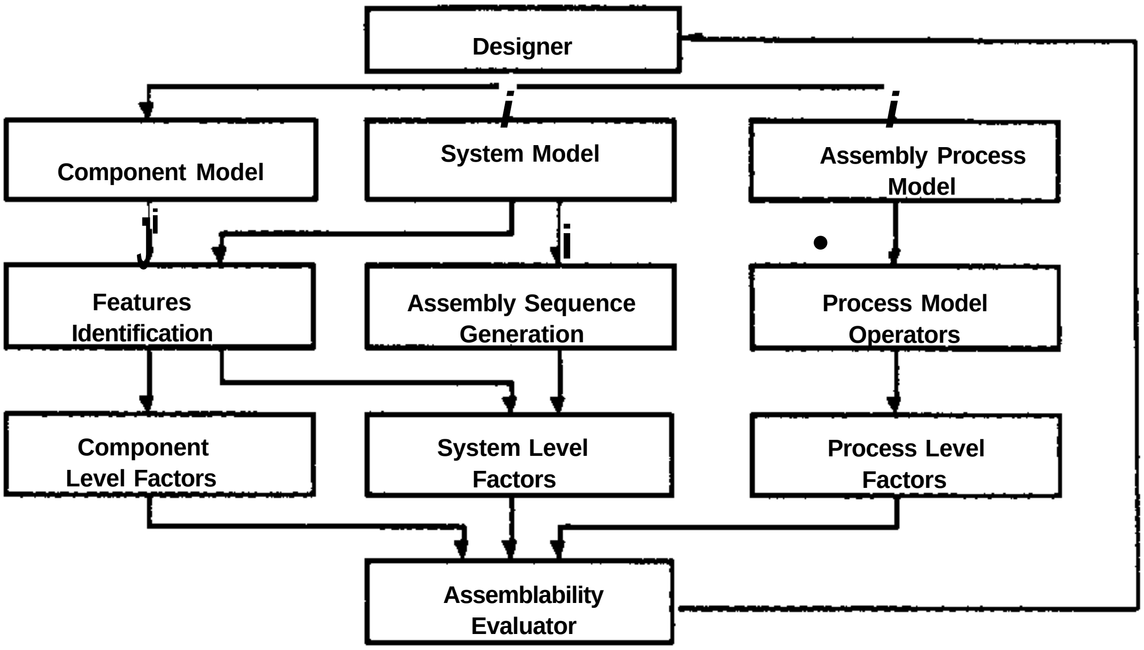
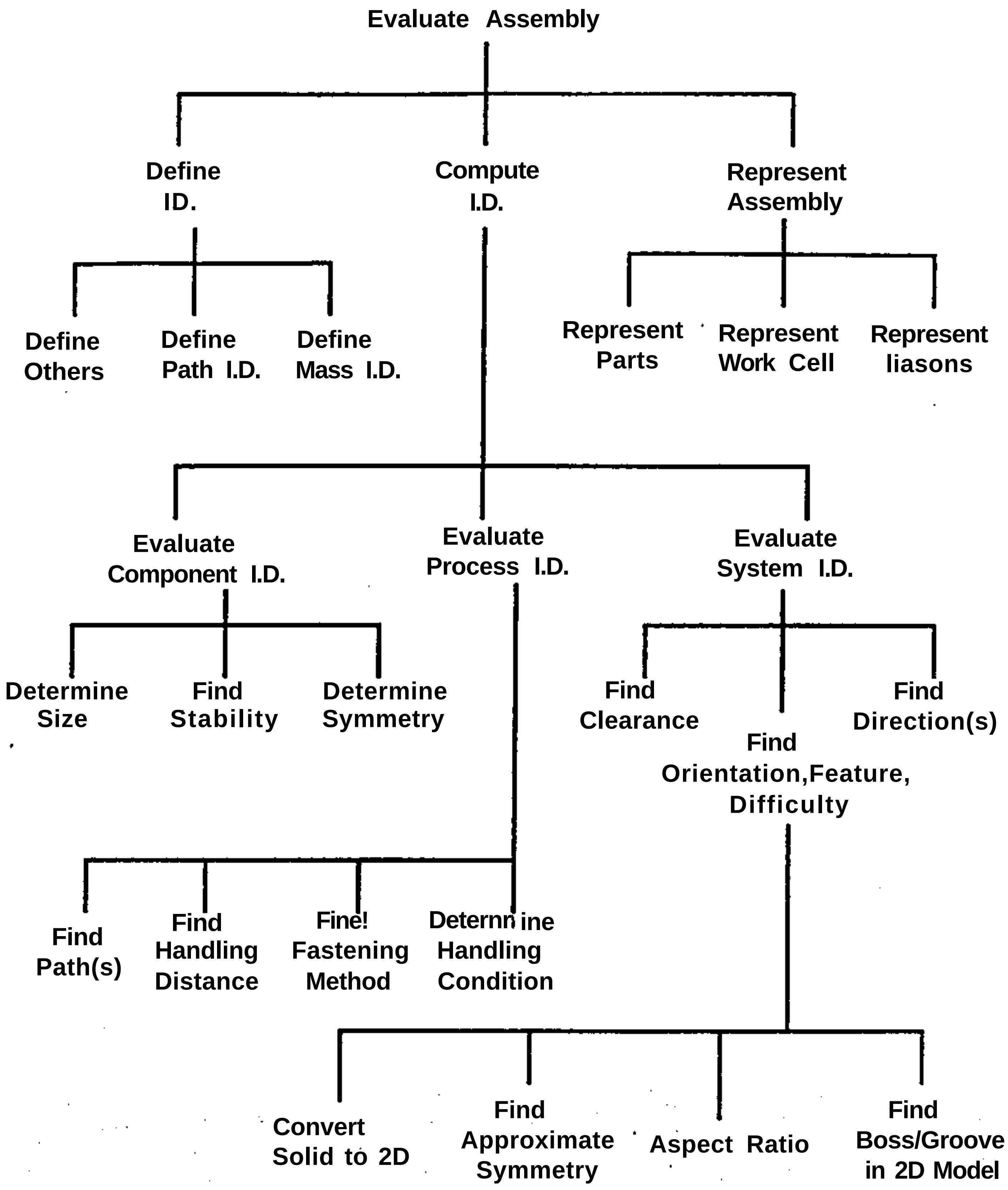


Figure 2.2 An Architecture for IDAERS
(Source: Sturges, R.H., and Kilani, M.I., 1990.)

Figure 2.3 IDAERS Function Block Diagram
(Read "How?" from top to bottom; read "Why?" from bottom to top.)



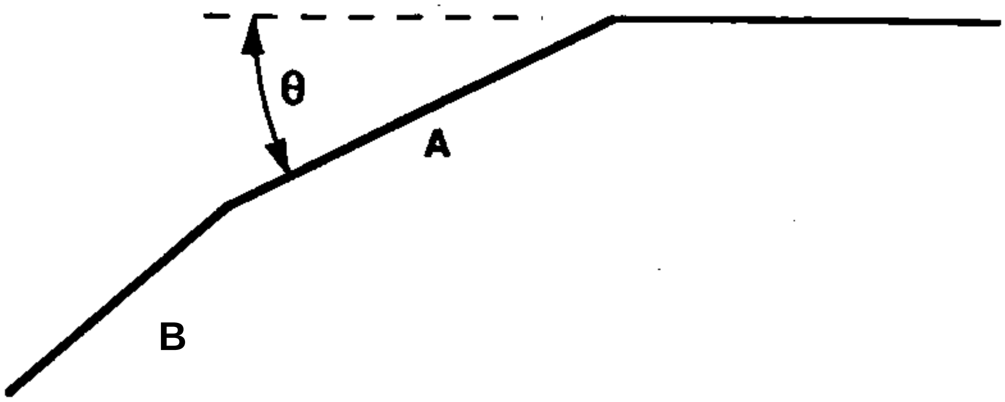
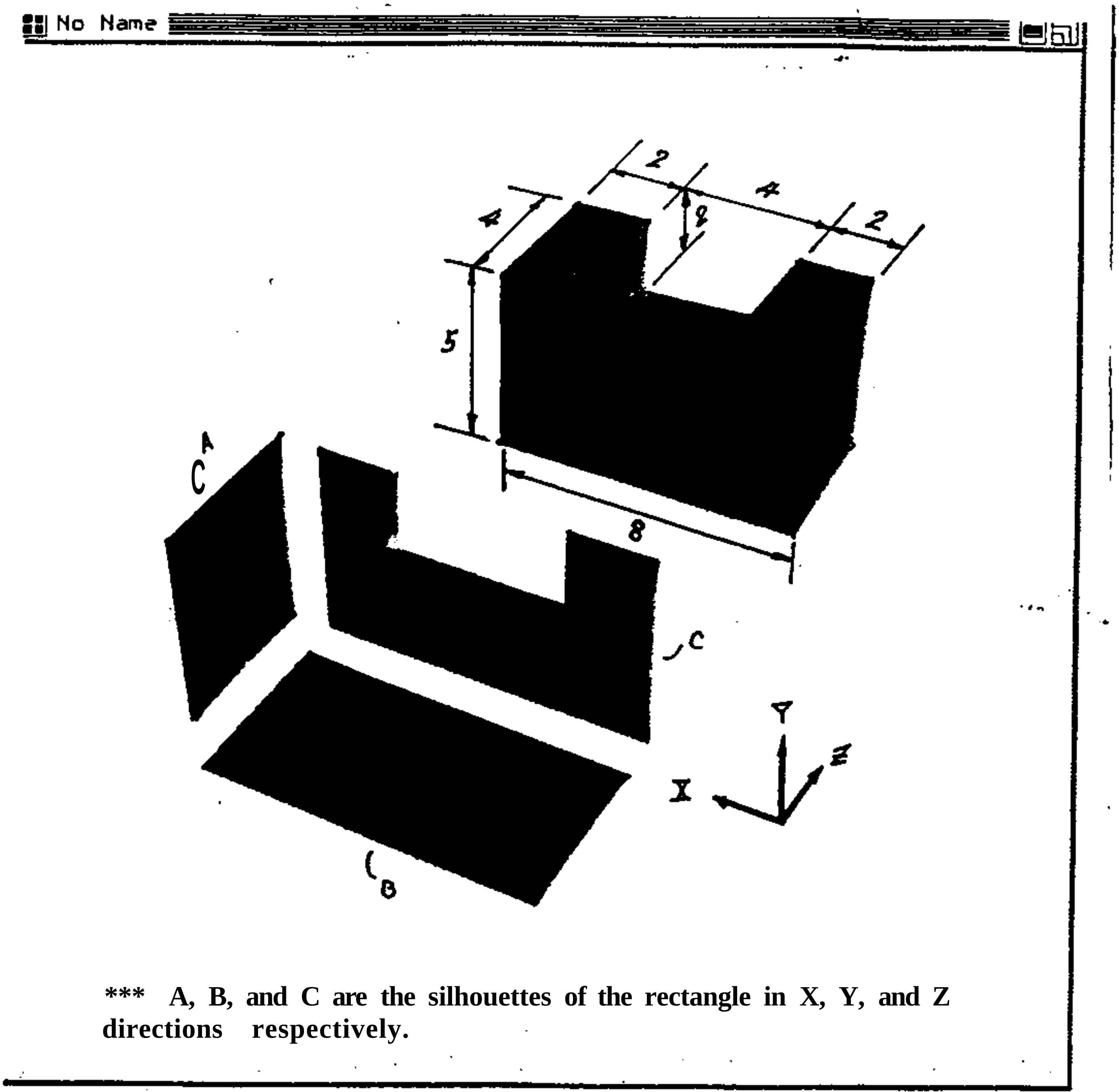


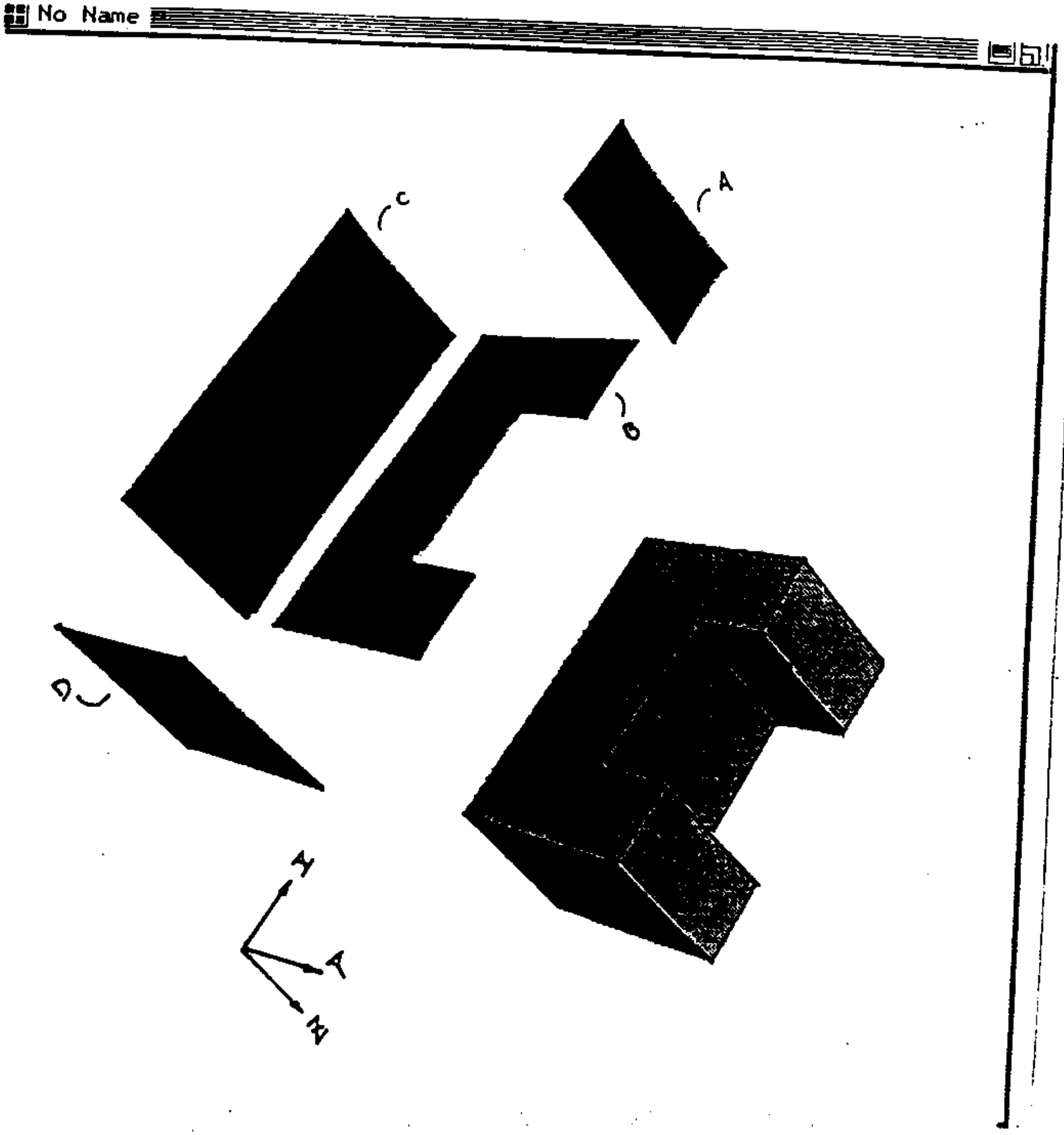
Figure 3.1 Tolerances for Curves

Figure 4.1 The Silhouette Method for Case I



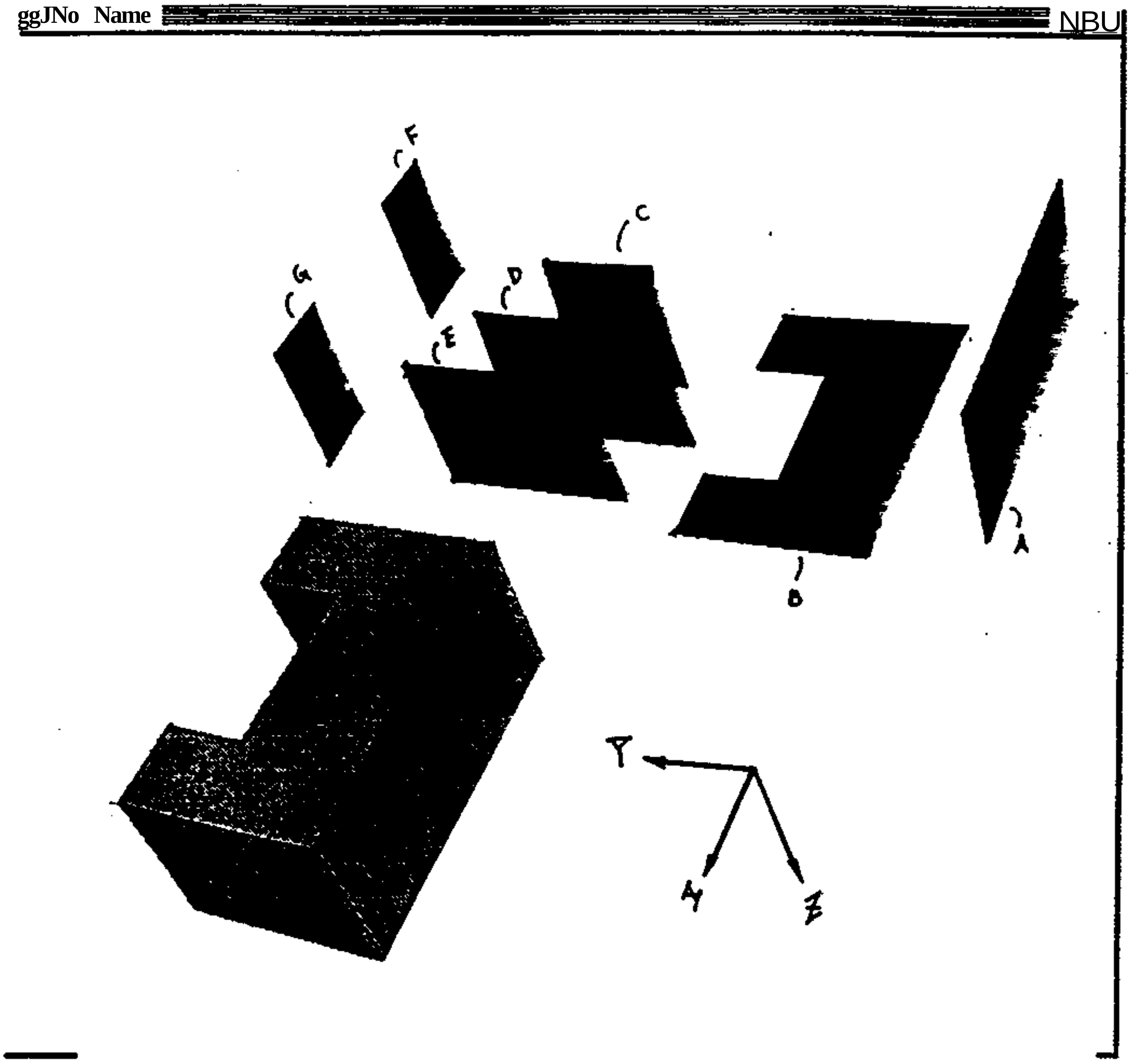
*** A, B, and C are the silhouettes of the rectangle in X, Y, and Z directions respectively.

Figure 4.2 The Imprint Method for Case I



*** *A* and C are the imprints of the rectangle in Y direction;
*** B is the imprint of the rectangle in Z direction;
*** D is the imprint of the rectangle in X direction.

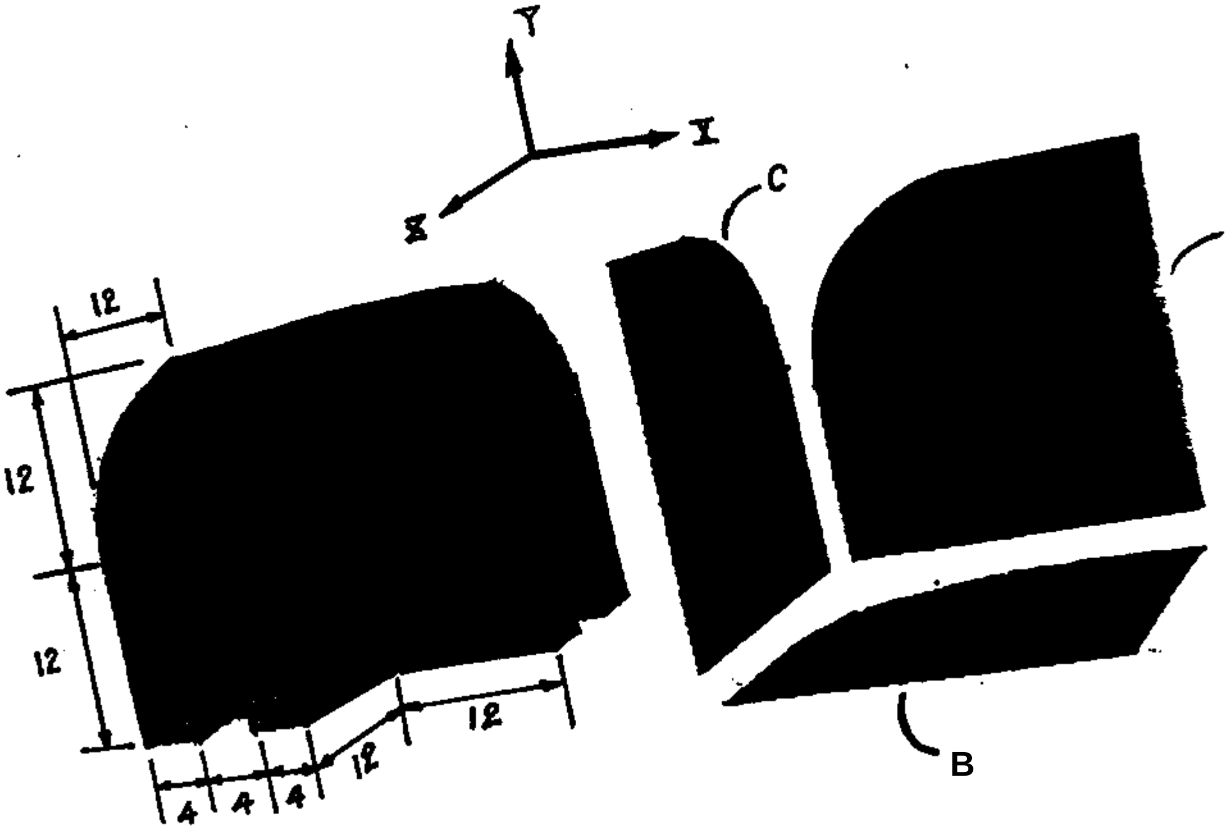
Figure 4.3 The Slicing Method for Case I



*** A, G, and F arc the slices of the rectangle in Y direction;
*** B is the slice of the rectangle in Z direction;
*** C, D, and E are the slices of the rectangle in X direction.

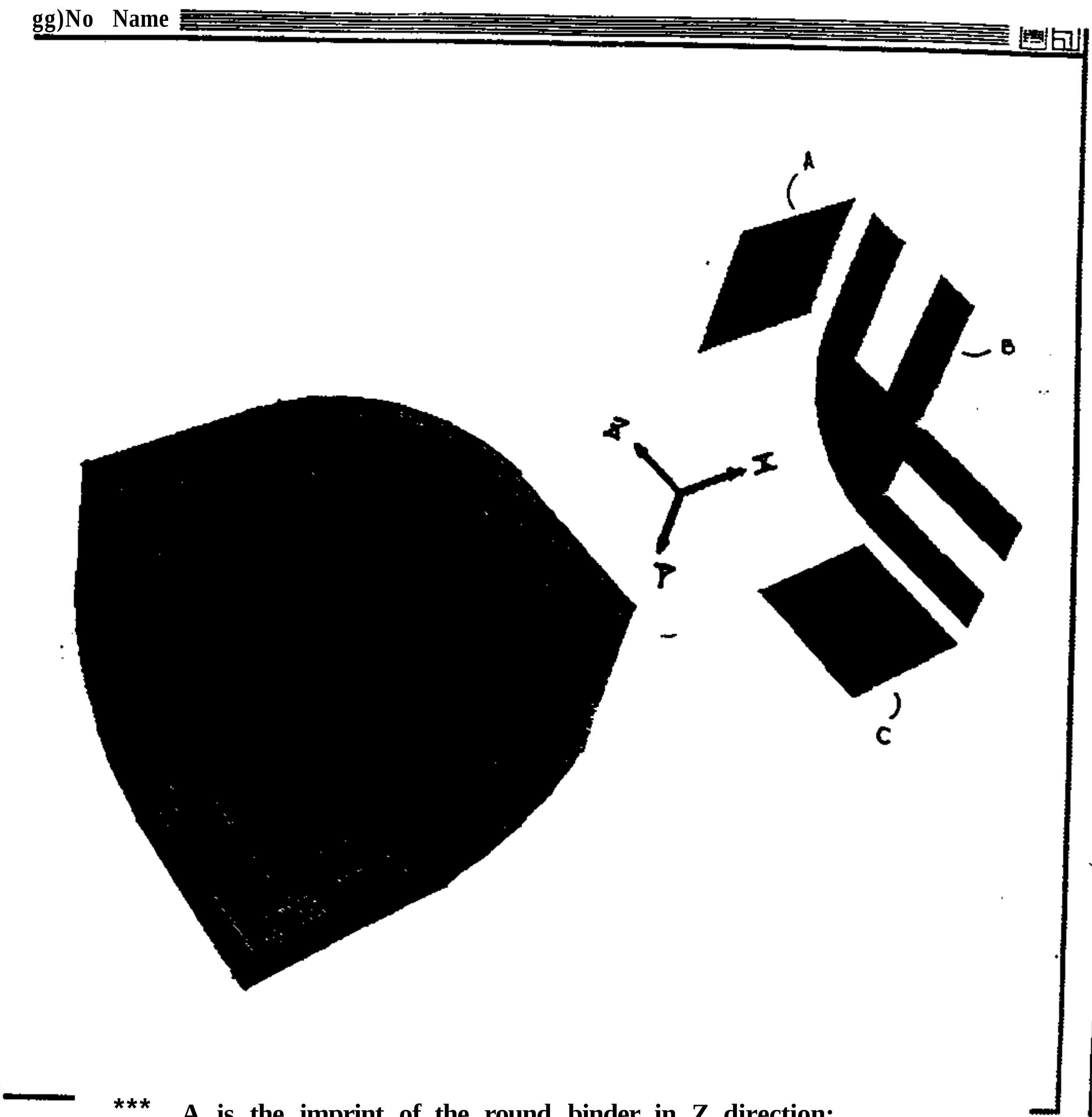
Figure 4.4 The Silhouette Method for Case II

SS| No Name HSU



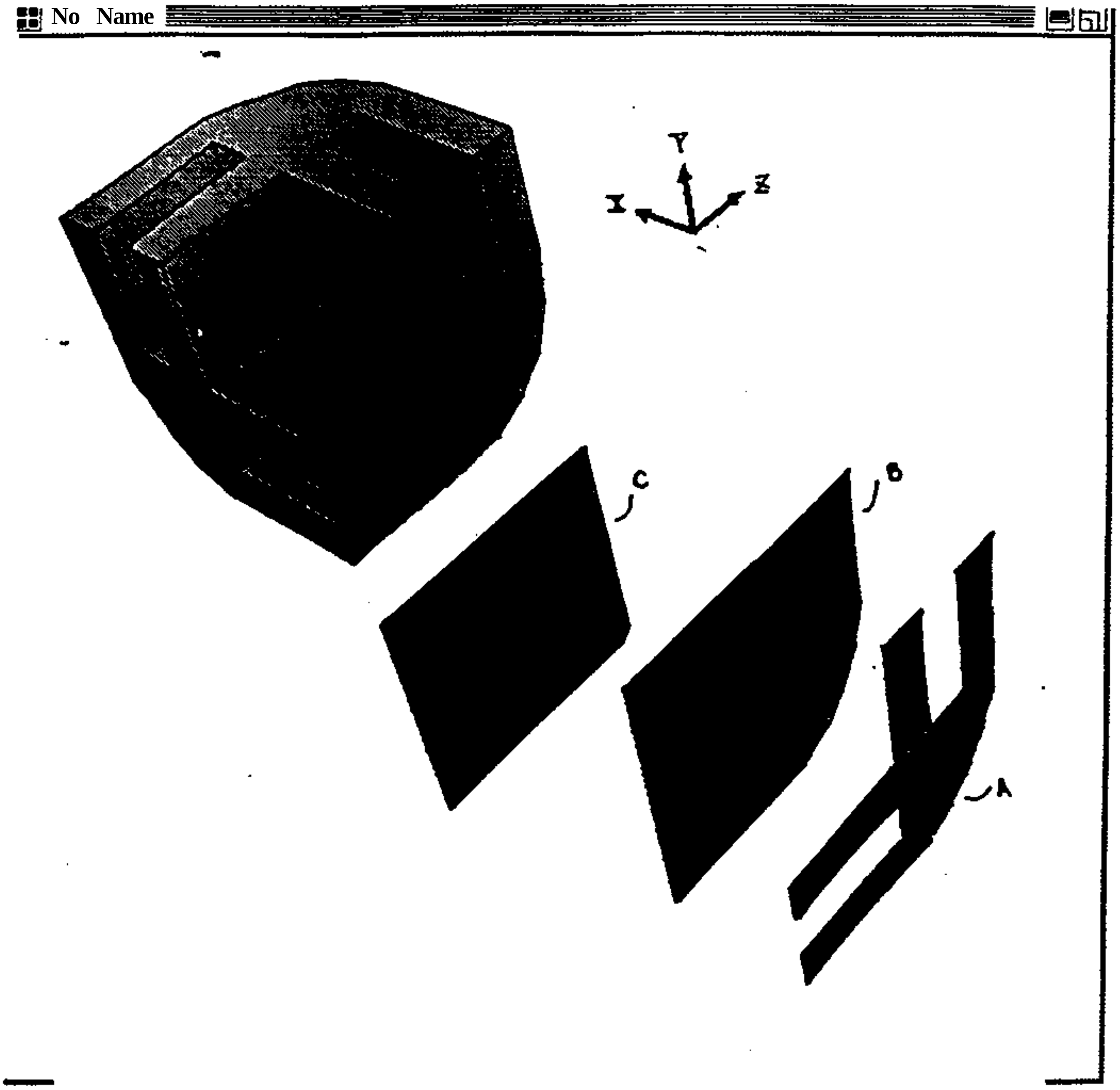
*** A, B, and C are the silhouettes of the round binder in Z, Y, and X directions respectively.

Figure 4.5 The Imprint Method for Case II



*** A is the imprint of the round binder in Z direction;
*** B is the imprint of the round binder in X direction;
*** C is the imprint of the round binder in Y direction.

Figure 4.6 The Slicing Method for Case II



*** A, B. and C are the slices of the round binder in X direction;