Design and Modeling Concepts (1990)

ARTICLE · JANUARY 1990

2 AUTHORS:

Jerzy W Rozenblit
The University of Arizona

Zeigler Bernard Zeigler
The University of Arizona

puter-controlled rivet removal. The basic design is a tool turret configuration with a translational degree of freedom for initial alignment.

The deriveting operation requires careful centering and perpendicular alignment of the end-effector tool center over the rivet. The robot first places the end effector at a previously taught rivet coordinate (±0.25 in. of the actual rivet center is sufficiently accurate).

ROBOT

A Cincinnati-Milacron T3-776 robot was selected for use in the robotic deriveter system. The 776 is a 6-axis articulated electric robot with a load capacity of 150 lb (10 in. out from the tool mounting plate). The repeatability to any previously taught point is ±0.010 in. The 776 uses Milacron's Version 4 controller which provides controlled path motion, teaching in either rectangular or cylindrical coordinates, and the capability of interfacing to a host computer using serial RS-232 communications.

VISION SYSTEM

The vision system used for locating and identifying fasteners is an International Robomation/Intelligence P-256. This is a 68000-based system that has a hardware array processor for very fast convolution operations such as image filtering and gradient approximation. The P-256 is programmed in FORTH, an incrementally compiled language that allows interactive program development and fairly fast program execution time.

Several algorithms are used to find the center of the fastener. The rivet image is first edge-enhanced using Robert's cross operator.

The vision system also includes a classifier that analyzes the patterns on the surface of the fastener head to determine if the fastener is a float-head rivet, a jo-bolt, a Phillips-head screw, etc. This information is compared with the expected type of fastener as stored in the data base to provide a check on the data and the system operation.

COMPUTER AND INTERFACE

The Hewlett-Packard A-700 computer system provides the primary input for the RDS operator during both teach and automatic deriveting operations. An HP 2623A graphics terminal is used to provide the operator with a menu-based control program allowing selection of the desired operating modes, data bases, etc.

RDS SOFTWARE

The robotic deriveter system software for the HP A-700 is divided into three levels: lower-level device communication routines, mid-level control task routines, and upper-level operating modes of the system.

SYSTEM PERFORMANCE

Teach

The two-pass method of teaching approximate rivet locations by moving the robot to points defining rows of rivets works well. Due to the resolution of the robot and operator positioning error, the coordinates obtained during Pass I are accurate to about 0.2 in.

In practice, the operator can use a video monitor to achieve higher positioning accuracy, but the time required increases greatly. It is more efficient to let the operator bring the robot to a position where the rivet is visible in the field of view of the vision system camera and then allow automatic refinement of position and orientation. The time required for the operator to move the robot to a rivet location using the teach pendant, identify the coordinates as individual fastener locations or as points defining a row of rivets, and store the coordinates in robot controller memory is about 30 s per rivet.

The second pass of teach mode (the active refinement of rivet locations) involves moving the robot and nearly always requires several iterations of the Hough transform method to find the rivet. Thus, it is slower than positioning during normal deriveting and requires about 30 s per rivet.

Derivet

In the automatic derivet mode of operation, the information of the data base has been refined to an accuracy limited by the repeatability of the robot (0.02–0.30 in.). Thus, when the system moves to a new position, it is almost always located within about 0.030 in. of rivet center. In this case, the faster Fourier transform method is used to locate the circle center. The time required to locate and classify the image varies from 3 to 6 s, depending on the image quality and on how near center the camera was initially. In some cases, when the image is severely degraded, the Hough Circle-finder algorithm must be used; this requires an additional 15 s.

The time required for drilling varies from 3–5 s for small aluminum rivets to up to 20–25 s for very large steel jo-bolts. Rotating the turret and punching the rivet shank out of the hole requires an additional 4 s. No data on the eddy current inspection is available at this time.

The vision system software has been able to find rivet centers accurately. On the basis of several hundred drilled rivets, it has been found that mechanical considerations introduce the bulk of drill location error. Center location by the vision system has an error of 0.001–0.003 in. Under well-controlled circumstances (drilling similar types of rivets at the same drill feed rate and in nearly the same location with respect to the robot), drilling is performed with a corresponding error. With a required drill depth on the order of 3/16 in., angular errors in drill alignment are not particularly critical. In the course of using the ultrasonic sensor to examine the region around the rivet, the normal to the surface is found to about 1° accuracy, leading to a displacement on the order of 0.00327 in.

By far the largest cause of drill-positioning error is deflection of the fixtured wing and of the robot arm itself when drill force is applied. Errors of 0.03–0.05 in. can occur for a drilling force of 150 lb applied along the wrist axis. This deflection is due primarily to deformation in the elbow joint bearing and is greatest when the loading force is applied at right angles to the horizontal projection of the arm. To eliminate this effect, a system has been designed to apply a preload to the wing before adjusting the end-effector motions to center the image seen by the vision system. In this way, forces on the robot will be nearly constant and very little deflection will occur during drilling.

BIBLIOGRAPHY

E. Franke, D. Michalsky and D. McFalls, "Location and Identification of Rivets by Machine Vision," *SME Vision '85 Conference Proceedings,* Society of Manufacturing Engineers, Dearborn, Mich., 1985.

DESIGN AND MODELING CONCEPTS

JERZY W. ROZENBLIT
BERNARD P. ZEIGLER
The University of Arizona
Tucson, Arizona

MULTI-OBJECTIVE SYSTEM MODELING

The ability to increase the decision-making capabilities in different environments is related to the scope of our possible intervention into the operation of the systems being modeled. Zeigler classifies the levels of intervention into three broad categories: management, control, and design. Management type of intervention connotes determining policies whose interpretation and execution is then delegated to subordinate levels. Control intervention is an action deterministically related to policy. In contrast, design represents the greatest scope of intervention in that the designer either creates the "real system" or augments, modifies, or replaces a part of existing reality (see also ASSEMBLY, ROBOTIC, DESIGN FOR).

The effects of interventions are uncertain due to the existence of uncontrollable parts in the system and it becomes necessary to encode knowledge about such parts in models of the system. Models represent abstractions of the reality whose primary function is to capture the structural and behavioral relationships in the system. These relationships would be difficult to observe were the models not available.

Thus, the modeling methodology should be an inherent component in computer-aided decision systems in management, control, and design. The tools and activities prescribed by this methodology enable the decision makers to evaluate (based on the analysis of the models' simulation) the effects of interventions before they are actually carried out. The "best," in terms of performance measures related to the system under evaluation, intervention alternatives are chosen and finally deployed in the real system. The choice of performance measures reflects the objective the decision maker (be it an economist, a designer of a power plant, or a technician supervising a chemical process) attempts to achieve. The nature of the objectives orients and drives the modeling and simulation processes. Envision a collection of partial models, each reflecting a specific objective. This implies that the objectives orient the model building process by helping to demarcate the system boundaries and determine the model components of relevance. The fundamental formal concept supporting these activities is the *system entity structure*. The entity structure enables the modeler to encompass the boundaries and decompositions conceived for the system.

The role of the objectives is equally important in the process of specifying the experimentation aspects for the models that have been perceived for the real system. The key concept in this process is that of *experimental frame* ie, the specification of circumstances under which a model (or the real system) is to be observed and experimented with. The experimental frame definition reflects the objectives of modeling by subjecting the model to input stimuli (which in fact represent potential interventions), observing reactions of the model by collecting output data, and controlling the experimentation by placing relevant constraints on values of the designated model state variables. The data collected from such experiments serve as a means of evaluating the effects of intended interventions.

Generation of meaningful experimental conditions is not a trivial task and requires that the modeler understand the nature of the objectives and their interactions. A frame, similarly to a model, may reflect a single or a complex set of goals.

This article discusses and recognizes the multiplicity of objectives, models, and experimental frames as a *sine qua non* condition of the modern, advanced modeling methodologies. The focus is specifically on the issues concerning the system design, understood here as the use of modeling techniques to procure and evaluate a model of the system being designed. In the ensuing section, the synergism between simulation modeling and system design is underscored.

System Design and Modeling, Synergies

The design aspect in decision-making offers the widest scope of intervention in that the designer develops a model from which a new system will be created. As opposed to system analysis, where the model is derived from an existing, real system, in system design the model comes first as a set of "blueprints" from which the system will be built, implemented, or deployed. The blueprints might take several forms; they could be simple verbal informal descriptions, a set of equations, or a complex computer program. The goal of such defined system design is to study models of designs before they are actually implemented and physically realized.

Recall that the primary goal is to locate the system design within the modeling framework. An attempt is made to provide a systematic methodology for a design process supported by adequate formal structures and leading toward future computerization. As depicted in Figure 1, system design is brought into the multifaceted framework with design process being supported by the modeling and simulation techniques in the manner described below.

Modeling is a creative act of individuals using the basic problem-solving techniques, building conceptual models based on the knowledge and perception of reality, requirements, and objectives of the modeling project. The models are design blueprints.

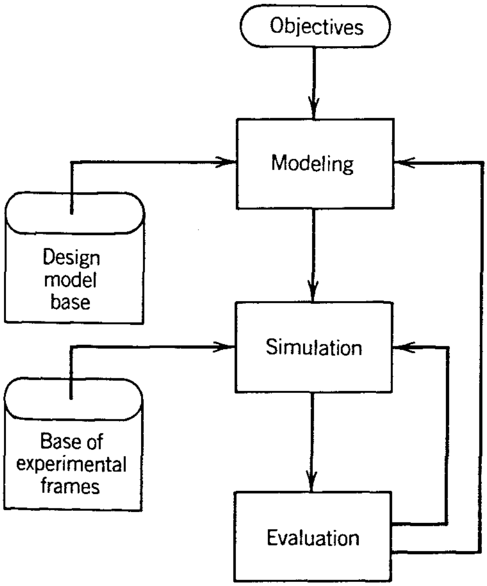

Figure 1. Design in the multifaceted modeling context.

FORMAL FRAMEWORK FOR MODEL-BASED SYSTEM DESIGN

This section provides the necessary formal background for the multifaceted system design introduced earlier.

The System Entity Structure

To represent a family of design configurations appropriately, a structure is needed that embodies knowledge about the following relationships: decomposition, taxonomy, and coupling. The *decomposition* scheme allows a representation in which an object (component of a system being designed) is decomposed into components. The structure should be able to operate on and communicate about the decomposition scheme.

Taxonomy is a representation for the kinds of variants that are possible for an object, ie, how they can be categorized and subclassified.

The third kind of knowledge to represent is that of *coupling constraints* on the possible ways in which components identified in decompositions can be coupled together.

A formal object that embodies these three basic relationships is called the *system entity structure*. The system entity structure is based on a treelike graph encompassing the system boundaries and decompositions that have been conceived for the system. An *entity* signifies a conceptual part of the system that has been identified as a component in one or more decompositions. Each such decomposition is called an *aspect*. Thus, entities and aspects should be thought of as components and decompositions, respectively. The system entity structure organizes possibilities for a variety of system decompositions and model constructions.

The entity/aspect distinction can be interpreted as follows: an entity represents an object of the system being designed, which either can be independently identified or is postulated as a component in some decomposition of the system. An aspect represents one decomposition, out of many possible, of an entity. The entities of an aspect represent disjoint components of a decomposition induced by the aspect. The aspects of an entity do not necessarily represent disjoint decompositions.

An entity may have several specializations. Each specialization may in turn have several entities. The original entity is called a general type relative to the entities belonging to a specialization, which are called special types. Since each such entity may have several specializations, a *hierarchical structure* results which is called a *taxonomy*.

Hierarchical decomposition is in many ways analogous to the specialization hierarchy just discussed. The alternation property now requires alternation of aspects and entities. An aspect is a mode of decomposition for an entity just as a specialization is mode of classification for it. There may be several ways of decomposing an object, just as there may be several ways of classifying it. Formally, aspects and specializations are quite alike in their behavior. They each alternate with entities but cannot be hung from each other. A special type of decomposition called a *multiple* decomposition facilitates flexible representation of multiple entities whose number is in a system may vary. (Throughout the illustrations, the multiple decomposition aspect is denoted by triple vertical bars and specializations by double vertical bars.)

To express the coupling constraints, the following procedure is employed: apply the mapping to remove the specializations to obtain an entity structure containing only entities and aspects. Now imagine that models are synthesized by working down the entity structure, selecting a single aspect for each entity and zero or more entities for each aspect. Such a process is called *pruning* of the entity structure (see Entity Structure and Experimental Frame-based Design Model Development). A pruning procedure is also defined as one that operates directly on entity structures with specializations. The coupling constraints must then be associated with aspects, since they represent the decompositions chosen when pruning. Moreover, a constraint must be associated with an aspect that contains all the entities involved in that constraint. What is more, this aspect should be minimal in the sense that there be no other aspect that lies below it in the entity structure which also encompasses all the entities involved in the constraint.

The Experimental Frame Definition

The system entity concept facilitates the representation of design structures. A means for expressing the dynamics of the design models is also needed. Since the design framework is objectives-driven, the experimental frame concept is used as the other underlying object in system design. The role of experimental frames will be twofold. First, the frame will represent the behavioral aspects of the design objectives and facilitate retrieval of entity structures that conform to those objectives. Secondly, the experimental frame will serve as a means of evaluating the design models with respect to given performance measures.

The conceptual basis for a methodology of model construction in which the objectives of modeling play the key and formally recognized role (therefore called *objectives-driven methodology*) was laid down by Zeigler.

The basic process in this methodology is that of defining an experimental frame, ie, a set of circumstances under which a model or real system is to be observed and experimented with. This process comprises the following steps. The purposes (objectives) for which the simulation study is undertaken lead to asking specific questions about the system to be simulated. This in turn requires that appropriate variables be defined so that a modeler can answer these questions. Ultimately, such a choice of variables is reflected in experimental frames that also express constraints on the trajectories of the chosen variables. The constraints on observations and control of an experiment should be in agreement with the modeling objectives. A choice of relevant variables constitutes the first important stage of experimental frame specification. The next step is to categorize the variables into input, output, and run control and place constraints on the time segments of these variables.

CHARACTERIZATION OF THE MULTILEVEL, MULTIPHASE SYSTEM DESIGN

System design concepts are found in many disciplines. The paradigms of each discipline underlie the methods for design representation and methodology. In systems theory, the dominant framework is the mathematical representation. In systems methodology, the methods are adopted from operations modeling; in philosophy, the models of thinking play an important role.

Common Traits in System Design Methodologies

Reviewing most of the conventional design methodologies leads to the following scheme of reasoning:

1. State the problem.
2. Identify goals and objectives.
3. Generate alternative solutions.
4. Develop a model.
5. Evaluate the alternatives.
6. Implement the results.

The methods to address each of the above aspects of design depend on the discipline and often vary in the degree that they are found in most of the approaches. The most important traits are

1. *Objectives-driven nature of design.* Everything in the designed system is considered and evaluated in relation to the purposes of the project.
2. *Hierarchical nature of design.* Structures of systems being designed are represented in multilevel hierarchies that express decompositions of the system into subsystems.
3. *Design as decision making.* Design is concerned with actions to be taken in the future. Thus, incorrect decisions in the early stages of design may impede all subsequent actions.
4. *Iterative nature of design.* It is widely recognized that the design process should be iterative in that the designer should be able to return to earlier phases of design from any stage of the process. Analogies are often made here with cybernetics and control theory where iteration is continued until a desired equilibrium point is reached.
5. *Optimization in design.* Design seeks optimization of the whole system with respect to its objectives. Careful consideration must be given if attempts are made to optimize subsystems separately. Appropriate coordination methods must then be used to attain the overall objective.

Orthogonal System Design

In the context of this article the term system design will denote the use of modeling and simulation techniques to build and evaluate models of the system being designed.

The design process is considered a series of successive refinements comprising two types of activities. The first type concerns the specification of design levels in a hierarchical manner. The design levels are successive refinements of the decomposition of the system under consideration. The first, and thus the most abstract level, is defined by the behavioral description of the system. Subsequently, the next levels are defined by decomposing the system into subsystems (modules, components) and applying decompositions to such subsystems until the resulting components are not further decomposable. The atomic system components are represented at the lowest level of the design hierarchy. At each level, the specialization of components into different categories is allowed for. This facilitates the representation of design alternatives.

The second type of design activity is concerned with "horizontal" actions associated with design levels. Such actions include requirements specification, system functional description, development of design models, experimentation by simulation, evaluation of results, and choice of the best design solution.

The design should proceed along both axes of the above characterization. The designer must be able to structure the designs, explore alternative structures, and derive complete specifications and models at any level. Transitions between design levels must be possible and easy to perform.

Such an orthogonal specification of the design process is often called a design matrix. The design process is represented in Figure 2.

ENTITY STRUCTURE AND EXPERIMENTAL FRAME-BASED DESIGN MODEL DEVELOPMENT

This section presents a framework that operationalizes the system entity structure and

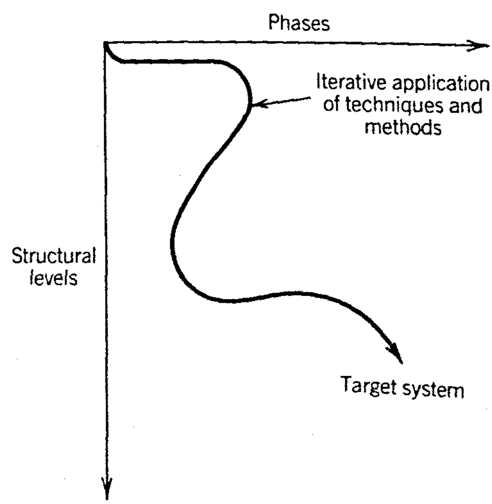

Figure 2. Orthogonal system design.

the experimental frame concept into a systematic design framework. First, the choice of the system entity structure as the underlying object in our design methodology is justified.

Recall that the entity structure represents the following relations:

Decomposition hierarchy of the system being designed. This enables the direct representation of the design levels as discussed earlier.

Taxonomy. This relationship (captured by the specialization aspects) facilitates the classification of design components and constitutes a means of expressing various design alternatives.

Coupling constraints on the possible ways in which components identified in decompositions/specializations can be coupled. This enables the use of the system entity structure as a basis for the hierarchical design model construction.

Again, it is emphasized that the entity structure is the basic means for organizing the family of possible design configurations (structures, architectures). It also serves as a skeleton for the hierarchical design model construction.

It is equally important to understand the role of the experimental frame concept in the design process. The role of frames is twofold. First, an experimental frame is a means of representing the performance measures associated with the behavioral aspects of design objectives. Subsequently, the frame is used in a simulation experiment performed to evaluate the merits of a design model. There is, however, a second important role of the frame concept. A generic form of an experimental frame is employed in the design framework to delimit the design model space given by the system entity structure.

Generic Experimental Frames

A generic experimental frame type represents a general class from which experimental frame specifications can be derived. A generic frame is defined by unqualified generic variable types that correspond to the objectives for which the design study is undertaken.

Design objectives are associated with performance indexes that allow for a final judgment of the design models.

The Entity Structure-based Generation of Design Model Structures

Due to the multiplicity of aspects and specializations, the design entity structure offers a spectrum of design alternatives. The procedures that limit the set of design configurations by extracting only those substructures that conform to the design objectives are presented now. Called pruning, this extraction process is based on the following scheme. Assume that an entity structure has been transformed into a structure with no specializations. Then, imagine that the structure is traversed by selecting a single aspect for each entity and zero or more entities for each aspect. All selected entities carry their attributes with them. Also, the coupling constraint of the selected aspect is attached to the entity to which this aspect belongs.

The above process results in *decomposition trees* that represent hierarchical decompositions of design models into components (*design model structures*). The process that extracts the model structures from the design entity structure is called *pruning*.

DESIGN MODEL SYNTHESIS

The pruning process restricts the space of possibilities for selection of components and couplings that can be used to realize the system being designed. Thus, it is assumed that the design can now be equivalent to the synthesis of a design model based on the pruned structures and the structural constraints imposed by the project requirements.

- Restrict the design domain by pruning the design entity structure in respective generic observation frames.
- Examine the resulting substructures and their constraints. Try to convert as many constraint relations as possible

into the active from, ie, into rules that can satisfy them. For those that cannot be converted into such rules, write rules that will test them for satisfaction.

- Write additional rules and modify existing ones to coordinate the actions of the rules (done in conjunction with the selected conflict resolution strategy).

ENVIRONMENT FOR INTEGRATED, MODEL-BASED SYSTEM DESIGN

It has been our contention thoroughout the foregoing sections that the system entity structure and the concept of generic frame type constitute the knowledge that can support automatic construction of design models and experimental frames. To explain the argument, the following architecture for an expert system design environment is proposed. As illustrated in Figure 3, the data base of design objectives is one of the major components in the system. It must be well understood that the design objectives drive three fundamental processes in the methodology: first, the retrieval and/or construction of the design entity structure. (Naturally, the designer desires to obtain a family of design representations for a given set of objectives.) Secondly, the objectives serve as a basis for the specification of generic observation frames. Finally, the structural aspects of design generate a set of rules for the design model synthesis.

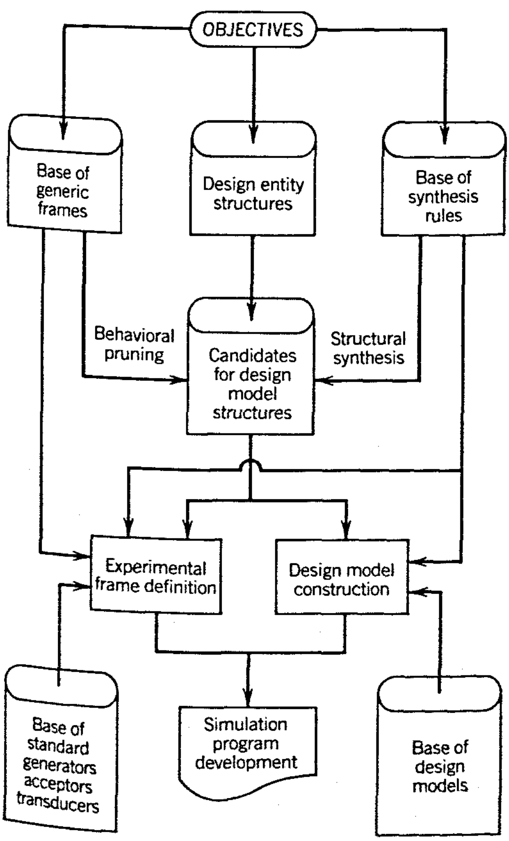

Figure 3. Integrated environment for system design support.

The ultimate purpose of the system depicted in Figure 3 is to analyze and integrate the relationships concerning the objectives specification base, the generic observation frame base, and the design entity structure. Such an integration should result in the formulation of design models and simulation experiments for a problem at hand.

The behavioral aspects of the design objectives are expressed in terms of generic observation frames. Pruning the design entity structure in corresponding observation frames results in substructures conforming to the behavioral objectives.

The substructures are then tested for satisfaction of synthesis rules that are derived from the design structural constraints as presented in Design Model Synthesis. Both behavioral and structural pruning applied to the design entity structure should result in design structures that are candidates for hierarchical model construction. The term candidates implies that some checks for consistency and admissibility (in the sense of conformance to the objectives) should be performed at this stage. If the candidate is inadmissible or no candidates can be obtained by pruning, the process should be reiterated with possible user intervention. The kinds of interventions suggested are modifications or retrieval of the new system entity structure, enhancement of the generic experimental frame, or modification of synthesis rules. The system should construct design models for the structures generated as a result of behavioral and structural synthesis employing the multifacetted model construction methodology presented earlier.

The design models should be evaluated through extensive simulation studies in experimental frames induced by the generic frame types.

SUMMARY

In presenting the model-based system design methodology, the main focus is on two major aspects of the design process, the *design model development* and *specification of experimental circumstances* for design simulation.

Based on the formal concepts of the system entity structure and experimental frame, a framework for objectives-driven design model generation has been developed. In this framework, the behavioral aspects of design objectives are reflected in the *generic frame types*, which are prestructures for the experimental frames. The generic experimental frame types serve as a basic means of extracting model structures conforming to the behavioral objectives from the design entity structure. Effective *pruning procedures* have been developed to perform this task. The procedures have been further refined to extract model composition trees from the design entity structures in which specialization relations occur.

The structural aspects of the design objectives represented by a set of constraints have been shown to drive the process called *model synthesis* effectively. A canonical *production rule* scheme has been given for generating model synthesis rules.

These concepts are of a propositional nature. It is stressed that an attempt has been made to lay a foundation on which a design process can be based. A computer-aided expert design environment that internally represents the entity structures and generic frames and has means for dynamically manipulating these structures has been envisoned. Implementation of such a package, in all its generality, may be a long way off. However, efforts are under way to advance the design methodologies further.

BIBLIOGRAPHY

B. P. Zeigler, *Multifacetted Modeling and Discrete Event Simulation,* Academic Press, Inc., New York, 1984.

B. P. Zeigler, "Knowledge Representation from Newton to Minsky and Beyond," *Applied Artificial Intelligence* **1,** 87–107, (1987).

J. W. Rozenblit, "A Conceptual Basis for Model-Based System Design," Doctoral Dissertation, Wayne State University, Detroit, Mich., 1985.

B. P. Zeigler, "Structures for Model Based Simulation Systems," in T. I. Ören, B. P. Zeigler, and M. S. Elzas, eds., *Simulation and Model-Based Methodology: An Integrative View,* Springer-Verlag, New York, 1984.

DESIRABILITY OF ROBOTS

Anil Mital
University of Cincinnati
Cincinnati, Ohio

INTRODUCTION

The need to strive for higher productivity is perpetual. Reduction in production costs and greater competition in the international market are two main concerns behind the move of many manufacturers to automate their existing facilities. New technologies, such as FMS, CAD/CAM, and Robotics, are rapidly being implemented in medium and high volume manufacturing. Robots, which are the key supportive elements in automated factories and stand-alone manufacturing cells, are dominating such functions as welding, painting, and loading/unloading.

When first developed, robots were expected to replace workers only in hazardous environments. However, robots have also begun replacing workers in monotonous, highly repetitive, and unstimulating tasks. Existing statistics on robot growth indicate a tremendous increase in the U.S. robot population. With improved capabilities and lower costs, the market could expand to as many as 200,000 units per year. By 1990, the world