AI Magazine Volume 7 Number 4 (1986) (© AAAI)

Edward L. Fisher

An AI-Based Methodology for Factory Design

Introduction

Factory design is the specification of functional requirements for a new factory or the specification of functional changes to an existing factory. Factory design is essentially initiated upon formalization of a product or set of products that must be manufactured. Once designed, the factory is subjected to a continuous cycle of redesign that is only complete when the factory has served its useful life, which can include the manufacture of products not conceived during the original design.

The design of a factory and the implications of this design on the manufacture of goods typically involves millions of dollars in expenditures. Recent estimates are that 8 percent of the U.S. gross national product (GNP) can be attributed to new factory design and construction (Tompkins and White 1984). Additionally, a large annual investment is made in factories or portions thereof that require periodic redesign due to changes in product line, manufacturing processes, or equipment used to manufacture the products.

At its largest scope, factory design involves the functions specified in figure 1.

For new factory design, decisions regarding product analysis might already have been made. Very often, the redesign of an existing factory is in response to the introduction of a new product(s) or a significant change to an existing one. Needless to say, the type of product and the projected demand for that product has much to do with the sizing of the plant and its components and the general technology that is economical for manufacturing the product. A high-volume product benefits from hard automation, whereas a low-volume product is better suited to job shop manufacturing. Flexible automation, such as robots, also has its place and should be selected when volume and quality conditions permit. The design of the factory requires careful consideration of existing and near-future technologies as well as economic sequencing and balancing of operations. Figures 2*A*, *B*, and *C* illustrate examples of components that are involved in factory design. Shown respectively are a workstation, a work-in-process area, and material-handling equipment.

Conventional methods for designing factories can generally be categorized as the following:

- Manual, rule-of-thumb approaches
- Manual, quantitative approaches
- Manual and computerized mathematical models
- Computerized design models

Only the first of these is in common use, as indicated in a recent study conducted by Nicol and Hollier in Great Britain (1983). Formal techniques are particularly uncommon in redesign tasks. Reasons for infrequent use of formal techniques include (1) effort involved in learning how and when to use them, (2) the inaccessibility of these techniques, and (3) insufficient flexibility to solve a variety of design problems. These reasons essentially dictate a prescription for future formal design techniques that emphasizes the need to solve a variety of design problems in a friendly, high-level environment which determines when and how to use the available formal methods and aids the user in the interpretation of their results.

Knowledge-based systems (KSs) that perform or aid the factory design process seemingly support the increased use of formal methods by synthesizing them with the expert design logic needed to use, interpret, and explain their results. Perhaps the most important benefit that can be derived from the development of an intelligent factory design agent is the ability to create an electronic model of the factory for subsequent use by other KSs and problem solvers. This *virtual* factory would benefit, for example, redesign of a factory when a change in product line occurs because only change-

Abstract This article provides a discussion of factory design and an artificial intelligence (AI) approach to this problem. Major issues covered include knowledge acquisition and representation, design methodology, system architecture, and communication The facilities design expert system (FADES) developed by the author is presented and described to illustrate issues in factory design.

Edward L Fisher is an assistant professor of industrial engineering at North Carolina State University, Raleigh, North Carolina, 27695-7906 He serves on the faculty of the Integrated Manufacturing Systems Engineering Institute (IMSEI) and heads the artificial intelligence laboratory. Dr Fisher is also Chairman of Knowledge Systems Corporation, a knowledge engineering consulting firm based in the Research Triangle

related information would need to be collected due to the a priori existence of a factory model.

One issue of concern with regard to the development of an intelligent factory design system is the degree to which it would be used in practice. The immense amount of information required to design a factory and the infrequent nature of design and redesign might preclude the use of such a system. However, as indicated earlier, the most benefit from a virtual model of the factory might not present itself so much in the initial design and redesign as in its more frequent use as a foundation for daily decision making in such activities as scheduling, production planning and control, product and process design, simulation of projected demand scenarios, maintenance, and other vital functions. The virtual model created by an initial design could then be amortized over a variety of activities and hence be made cost effective. In essence, the model becomes an information base for real-time decision making, allowing dynamic reconfiguration of the manufacturing system in response to changing loads or equipment status. The reconfiguration can be virtual—that is, the material-handling system is employed to implement a new conceptual configuration—or real—that is, the physical placement of process components is altered to better meet current organizational objectives, demands, and process sequences.

The knowledge-based approach was sufficiently promising that a research effort was initiated in 1982 to formulate a methodology and construct a prototype system. Early results of this effort were presented in 1983 (Fisher, Nof, and Whinston 1983), and the first prototype, the FAcilities Design Expert System (FADES), was completed in 1984 and initially described by Fisher (1984) and Fisher and Nof (1984).

Review

The knowledge-based methodologies presented here for factory design have benefited from two general bodies of knowledge: (1) conventional facilities design methods and (2) artificial intelligence theory for design. *Facilities design* is a general term that encompasses the design of factories as well as hospitals, airports, retail stores, and other types of facilities. Several manual and computerized techniques, generic to the majority of facilities, have been developed to aid the design process. Examples of these techniques and the tools based on them are systematic layout planning (SLP) (Muther 1961), computerized relative allocation of facilities technique (CRAFT) (Armour, Buffa, and Vollmann 1964), computerized relationship layout planning (CORELAP) (Lee and Moore 1967), automated layout design program (ALDEP) (Seehof and Evans 1967), and computerized facilities design (COFAD) (Tompkins 1972). Overviews of current techniques and tools are given by Foulds (1983), Heisterberg (1978), and Filey (1985). A large number of the techniques and tools reviewed are concerned only with the layout of departments and workstations.

Product Analysis: What do we make?

Process Design: How is it made?

Flow Design and Analysis: How do materials, work-in-process, and finished product flow through the factory?

Specification and Selection of Manufacturing and Handling Components: What is used to make the product, and how is the product transported between work cells?

Establishment of Component Relationships: Where do manufacturing and handling components physically reside in the factory?

Analysis of Design and Modification: How well does it work, and how can it be made to work better?

Figure 1. Functions of Factory Design.

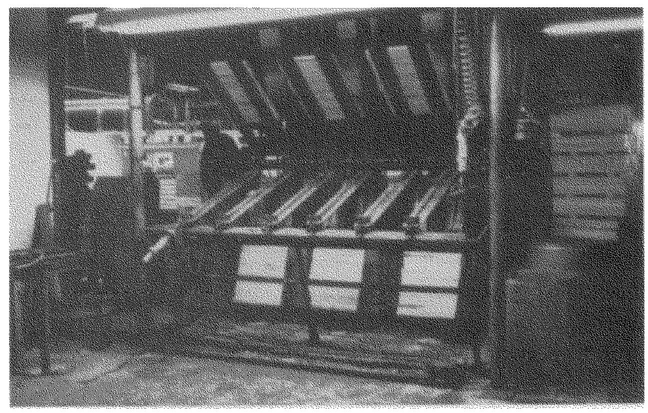

Photo courtesy of Dolly Madison Industries

Figure 2A. Workstation.

Layout techniques can be divided into two categories: construction and improvement. SLP, CORELAP, and ALDEP construct layouts by scoring alternative plans developed from an activity relationship chart (ARC). ARC contains values for the relative binding strength among various activities. The CRAFT program employs a quadratic assignment algorithm to selectively improve an initially specified layout by interchanging activities until no material-handling cost improvements can be made. COFAD was a significant departure from the earlier techniques because in addition to layout of process activities, it provided for the optimal selection of handling equipment.

One difficulty with conventional facilities design tools and techniques is that they tend to require a significant investment of user time for their use and are limited to several specific aspects of facilities design, as indicated earlier. Little progress has been made in computerizing those aspects of a human designer which are considered creative in nature. This is in part the result of the complexity of the design problem; the need for extensive amounts, and a synthesis, of information; and the lack of appropriate programming tools.

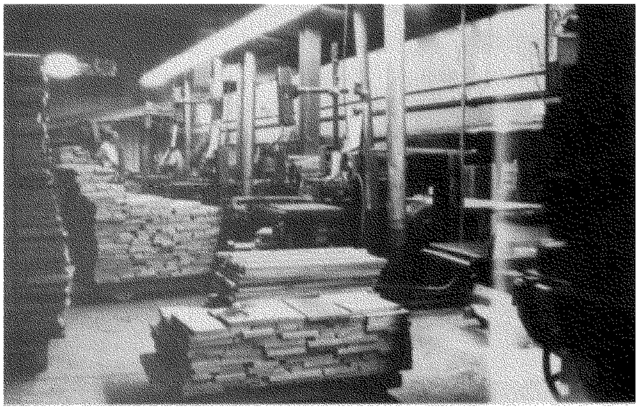

Photo courtesy of Dolly Madison Industries

Figure 2B Work-in-Process Area.

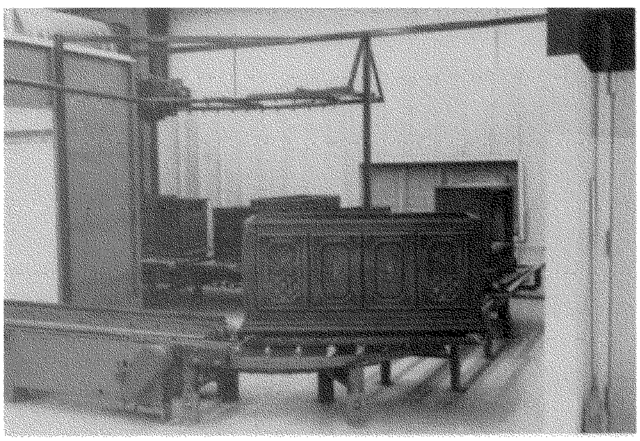

Photo courtesy of Thermwood Corporation

Figure 2C. Material Handling Equipment.

The study of design as an AI-based task is still in its infancy, with very few systems having been constructed. In AI terms, design can be thought of as the creative process of developing plans and specifying, selecting, and arranging components in order that the goals and objectives of the designer (and the employing organization) can best be achieved. Redesign is the process of satisfying additional constraints on an existing system with the aim of keeping the system intact where possible. This definition relates well to factory design. Mostow (1985) provides a good review of various AI-oriented efforts focused on modeling the design process as well as summarizing current research issues and innovative ideas for such approaches to design. Three interesting systems that relate to the work described here are discussed in the following paragraphs.

Digital's XCON system (an earlier version, R1, is described in McDermott [1982]) spatially configures its VAX and PDP computer systems. It is written in OPS and uses a foward-chaining inference mechanism. Features of XCON that are of interest to the factory design model include: (1) its resident knowledge of individual components and their relationships, (2) its embodiment of constraints to indicate which components can or must be associated with a system configuration, (3) its use of Match (see Hayes-Roth, et al. [1983], p.104) as a central problem-solving method to generate a single acceptable solution, and (4) its fixed partitioning of a large problem into several subproblems. Although the first feature is needed for factory design, it is not always directly applicable. XCON's individual component relationships are both known and basically unchanging for each application. In the factory design problem, relationships between components (departments or workstations) are often to be determined, changing with each application. Spatial configuration occurs after these relationships have been generated.

As in XCON, constraints can be used to limit search in factory design—for example, during relationship generation, the number of possible nexus queries (see discussion under Knowledge Acquisition) between two departments. Although the Match method of search is adequate in XCON, backtracking and a limited number of multiple layout solutions are actually desirable for factory design. This is partly due to the uncertain relationships among factory design components that must be dealt with. The fixed partitioning of subproblems employed by XCON is applicable to factory design, however, depending on actual problem definition, many steps in the design process might not be involved. Thus, although certain paradigms in XCON are attractive, other alternatives have been considered in our work to address some of the additional characteristics and needs of factory design.

Another configuration-oriented expert system is TALIB, developed by Kim and McDermott (1983). TALIB is an integrated circuit layout design assistant for NMOS cells, and is also written in OPS. TALIB is particularly interesting because it generates adjacency relationships among subcircuits and propagates distance relationships between signal terminals. Again, however, these relationships are defined for TALIB a priori as opposed to being generated. TALIB's use of a heuristic design structure to partition the problem into less complex subproblems at lower levels agrees with the approach employed in FADES for reducing the complexity of factory design.

A third interesting work is the AIR-CYL system for mechanical part design described by Brown and Chandrasekaran (1984). AIR-CYL, as its name implies, is used to design new air cylinders. To accomplish this task, it bases new design on exceptions to specifications of existing cylinder designs. For example, when a new specification arrives, AIR-CYL draws upon its knowledge of designs resulting from previous specifications to formulate a new design by applying design rules to the differences between the old and the new. The authors define the task in AIR-CYL as a Class 3 design problem, one of three classes they offer as a general taxonomy for knowledge-based design.

Class 3 in the authors' definition has both known knowledge sources and problem-solving strategies prior to design; Class 1 design is defined as not having either; Class 2 has known knowledge sources but no problem-solving strategies. Factory design is complex enough to contain elements of all three of the design classes defined; thus when developing a methodology for factory design, it is important to carefully narrow the scope of work to specific types of factory design problems and scenarios so that progress toward an ambitious goal can in fact be made. Rather than concentrate on a single decision class however, instances of each class type have been studied within an industry. This allows solution of industry problems to continue within the scope of the research effort.

Methodology

Although it is premature to build a comprehensive system that handles general factory design problems, much can be learned from the short-term development of methodology and prototype systems. The approach taken in our research was to develop a general methodology and to experiment with portions of the methodology in the FADES prototype. Where possible, specific problems are addressed and can be made available to the industrial community. One example is the material-handling equipment selection expert system (MATHES) program developed by Farber and Fisher (1985). FADES is thus used to address issues that impact long-range research and development priorities, and subcomponents address problems that are currently solvable with these methodologies. Initially, the emphasis in FADES was not on developing an entirely new reasoning methodology (in AI terms) but rather to build upon existing factory design software where appropriate and to integrate this into a knowledge-based environment.

This approach has provided the advantage of having working components on a faster time line, but it might also have caused us to delay the use of advanced methods. Nevertheless, our understanding of the problems has benefited, and because we have a solid base to work from, any new methodologies will build upon these experiences and the knowledge bases developed.

Issues investigated during the development of FADES included the following:

- Acquisition and representation of design knowledge
- Reasoning methods for design, including the integration of heuristic, algorithmic, and economic models in the decision process
- The process of dealing with uncertainty during design
- Architecture and organization of AI-based design systems
- The interfacing of design systems with external agents (the system user, database management systems [DBMSs], and application programs)

Approach

As indicated by Mostow (1985), there are two current models for the transformation of design specifications into implementations. These are (1)abstraction refinement that decomposes a specification S into components of implementation I and (2)a goal tree model which refines a goal into an executable transformation sequence, converting a specification S into an implementation I.

Factory design (at least new design) falls nicely into category 1 because it has been our experience that human factory designers tend to work this way, successively refining rough designs into finished designs. Each task is viewed at different levels of abstraction, with problem solving involving the refinement of an abstract solution space. The top level of this solution space is the factory, the next level departments, a third level work cells, a fourth workstations, and a fifth workstation components. In reality, refinement actually continues to additional detailed levels, including planned work methods for each component. In this way, the design proceeds from rough to finished design.

Of the factory design functions presented in figure 1, primary emphasis has been placed on numbers 4 and 5 to date, with emphasis on numbers 3 and 6 now increasing. Thus, it is currently assumed that the products to be manufactured are known and that their process sequence plans are available.

Within these functions, there are many pockets of expertise now held by human experts that must be acquired in order to develop a robust automated design aid. Examples of knowledge bases needed are (1) assembly technology assessment, (2) selection of material-handling equipment, (3) development of department-work cell-workstation spatial relationships, and (4) analysis of spatial design for optimal efficiency.

The actual work in FADES is performed by a number of knowledge modules, each responsible for specific aspects of the design problem. These modules can be viewed as specialists, as described by Brown and Chandrasekaran (1984), each working on one of the subtasks of the design. Communication takes place between a parent module and its offspring modules, with each module having the opportunity to select an action plan based on current problem characteristics. The design knowledge is generally invoked within a hierarchical structure, with the level of abstraction (Sacerdoti 1974) dependent on the stage of design, including whether the design desired is in fact rough or finished. Figure 3 illustrates several specialists that are needed in a factory design system.

The action plan for a knowledge module consists of one or more tasks to be performed, each task consisting of a number of design steps. For example, layout is an example of a

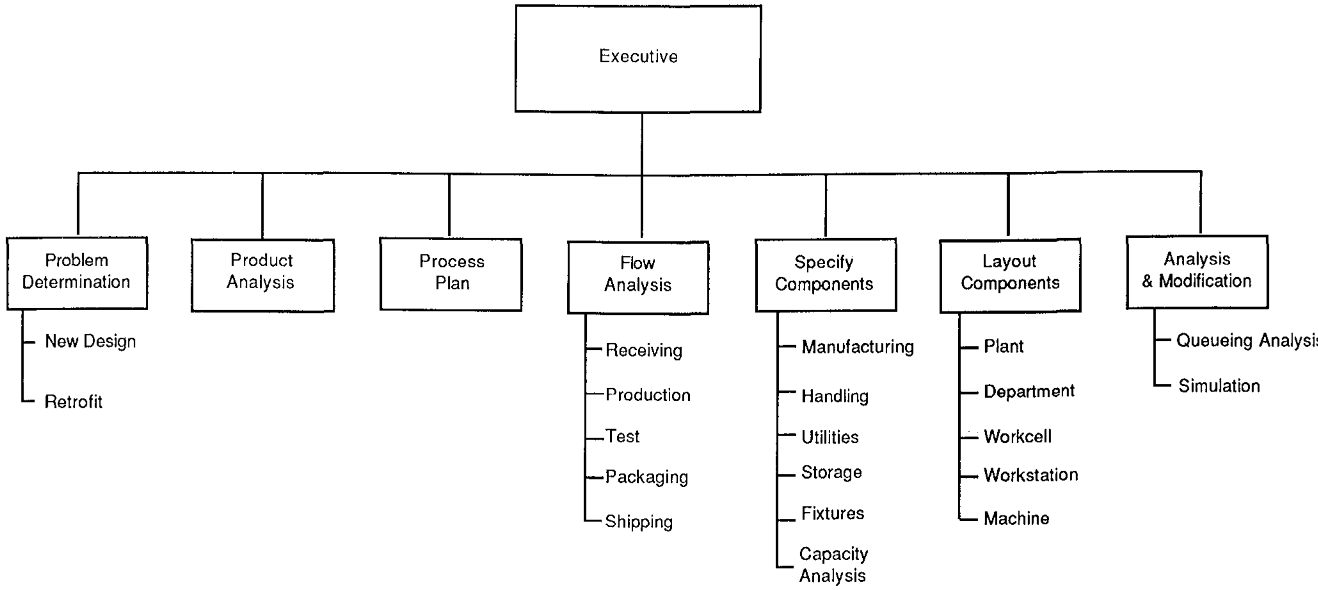

Figure 3. Hierarchy of Design Specialists

major knowledge module. Based on problem characteristics, the layout module would choose a sample action plan that might involve the following tasks:

- Determine product flow between all workstations
- Determine nonflow constraints and their effects on workstation interrelationships
- Combine flow and nonflow relationships
- Select layout generator routine
- Develop input data set for layout generator routine
- Invoke layout generator
- Analyze results of layout generator
- Select "best" current layout

Each of these tasks involves completing several design steps. For example, developing an input data set for a layout routine involves making decisions on the length-to-width ratio of the layout, whether any department should be fixed in a specified location, the relative scoring measure that should be used in scoring the layouts, and the number of layouts that should be attempted.

Knowledge Acquisition

Four kinds of factory design knowledge were identified during the acquisition process: (1) product-specific knowledge, (2) organizational knowledge, (3) industry-specific knowledge, and (4) generic knowledge.

Product-specific knowledge is indigenous to a particular type of product; for example, in the furniture industry, the knowledge used to lay out a chair factory differs from that used to lay out a table factory. *Organizational knowledge* is knowledge that differs among firms in the same industry, reflecting, for example, company policies. *Industry-specific knowledge* is common to all products within the same industry. *Generic knowledge* is design knowledge that is common to all products, organizations, and industries. In this work, the knowledge base contains four segments, one for each type of knowledge.

Initial knowledge base development was performed with two furniture manufacturers: Kimball International and Dolly Madison Industries. No formal tests were conducted with this knowledge other than qualitative measurements of the methodology's promise when compared to conventional design methods. The primary reason for working with these manufacturers was to determine major problems facing their designers, resource constraints, reasoning processes employed, and a general idea of the types of knowledge needed during the design process. Also, a large amount of declarative knowledge was obtained.

A large portion of the knowledge contained in the first version of FADES (summer 1984) was generic in nature, drawn primarily from reference materials, textbooks, and existing programs. Much of the work occurred in developing a framework for using the existing tools and integrating them within a knowledge-based environment.

During this period, it became obvious that the scope of factory design involved many experts, and often there was a choice among several within the same functional area. This situation is exacerbated because factory design takes place in all types of industries, many of which have different goals and objectives. As no one person could possibly be adept in more than a few industries, multiple experts were needed to capture as much of this diverse knowledge as possible.

After the effort involved in establishing the knowledge base for the first version of FADES, knowledge acquisition was actually suspended for approximately one year so that other portions of FADES, notably the DBMS interface and the user interface, could be improved. With progress made

Table 1. Attributes and Weighting Factors.

Attribute	Weighting Factor
Minimize material flow cost	5
Maximize shared personnel	4
Minimize information flow (paperwork) cost	3
Minimize employee exposure to noise	3
Minimize employee exposure to fumes	2
Minimize personnel movement	2

in the latter two areas, acquisition commenced again during the summer of 1985.

Over the past year, we have worked with three companies to acquire knowledge: Trion Corporation and Buehler Industries in the area of generating department spatial and functional relationships and Tompkins and Associates in the area of material-handling equipment selection.

To gather knowledge in the relationship generation area, a survey was designed to reveal how factory designers determine relationships in a given set of circumstances (Bray 1986). This survey contained two parts: (1) weighting attributes associated with factory design objectives and (2) assigning closeness values that reflect department spatial relationships.

Attribute Weights for Factory Design Objectives. The attribute weighting section of the knowledge-acquisition survey consisted of evidence used to determine interdepartmental relationships embodied in the objectives of factory design. (note that these same principles can be applied to relationships within departments, among workstations).

There is no single objective of factory design; rather, there are many, some conflicting. Five global factory design objectives were distinguished: (1) minimize material-handling costs, (2) minimize personnel flow costs, (3) minimize information flow costs, (4) maximize safety, and (5) satisfy utility constraints.

Each objective further consists of attributes (or reasons) that are the basis for relationships. Thus, relationships for any given facility can be adequately expressed based on the attributes of these objectives. It is recognized that the degree of emphasis placed on factory design objectives is dependent upon the overall policy of each individual facility. In order to obtain results (from the knowledge-acquisition survey) that could be compared across a group of experts, a weighted factor was supplied for each attribute. These weighted factors were determined from discussions with expert designers. The weighted factors, reflecting priority, range from 1 to 5: (1) do not consider, (2) place low priority, (3) place moderate priority, (4) place high priority, and (5) place highest priority.

A subset of the attributes of the global factory design objectives was chosen for use in the knowledge-acquisition survey. Listed in table 1 are selected attributes and their respective weighting factors. The attributes are of two types: nexus and constraint. *Nexus values* express evidence of closeness shared between two departments (for example, material flow, shared personnel). *Constraints* (for example, noise, fumes) reflect a need for some distance between departments because attributes of one department might adversely affect another department.

Closeness Value Assignment. The second portion of the knowledge-acquisition survey concerned the assignment of closeness values. Each of the factory design attributes can assume several values. For instance, material flow can be stratified as heavy, medium, or light flow, and exposure to noise can be intolerable, tolerable, or controlled.

During development of relationships, departments involved in an industrial case study are normally grouped together in pairs. Information about the attributes is given pertaining to the pairings in one of two ways: either as single pieces of evidence or as couples of two. Based on the information and the weighted factor attached to the corresponding attribute, the user is asked to assign a closeness value according to the given closeness relationship scale:

(1) side by side, (2) very close, (3) close, (4) relatively close, (5) closeness does not matter, and (6) far.

The intent of the knowledge-acquisition survey was twofold. First, it was used to determine if, given a measure assigned to each attribute value, there exists some function f (of the attribute value and the weighted factor associated with its respective attribute) that produces closeness results which, when stratified on the closeness scale, model the reasoning process of human factory design experts. Second, the relationship knowledge base could be developed from the information acquired.

Whereas the acquisition of relationship knowledge such as that described was obtained from a survey plus personal interviews, there exist opportunities in factory design to elicit knowledge for a particular task from one expert. Such was the case for MATHES, where a knowledge base is employed to select from among 16 types of material-handling equipment given parameter values for a unit load to be moved. The parameters identified were (1) path (fixed or variable), (2) size, (3) volume, (4) distance, and (5) whether the unit is palletized. This knowledge base consists of approximately 200 rules and was acquired from Dr. James Tompkins, a recognized expert in this area, through intensive one-on-one sessions.

Future knowledge acquisition will continue to be a mixture of the earlier described methods, depending upon the type of expertise being elicited. Knowledge acquisition in FADES typically involves interviewing industrial factory designers to glean their methods and approach and specific heuristics they use for abstracting and pruning the solution space. It should be noted that knowledge normally has been different enough among designers that a specific designer's rules have been captured in a separate file for each expert (for example, *bob* or *frank*), and in some situations the FADES user has the option of selecting a designer based on known characteristics, for example, safety consciousness or cost orientation.

Knowledge Representation

The current version of FADES is written in Prolog (see Clocksin and Mellish [1984] for pedagogical material). Prolog employs a clausal form of first-order predicate logic for representation and uses a control mechanism that combines goal-directed depth-first search of an AND/OR tree with backtracking. The general problem-solving mechanism is a resolution refutation proof procedure as described by Nilsson (1980). A Prolog predicate consists of a predicate formula with an n-ary argument list; for example:

$pred_a(x_{a1}, x_{a2}, \ldots, x_{an})$

However, rather than focus on the specifics of a Prolog representation, the discussion here is presented at a higher level.

Declarative Knowledge. For the most part, the standard Prolog clausal logic method of knowledge representation has been adequate for initial versions of FADES; however, a representation method based in frames with inheritance appears beneficial (although not essential) for addressing complex design issues. Advantages of frame representation include its flexibility in lending itself to a variety of languages and tools (for example, object-oriented programming). Its optimization of data storage and its ability to make the most common access paths efficient by guiding the invocation of rules and eliciting relevant information from the user.

Most often, FADES knowledge was adequately represented and manipulated as object-attribute-value triplets in clausal logic form. Neither inheritance nor object-oriented methods have been explicitly used to date in FADES, although this is now changing with a concentrated effort on a user interface and simulation system (King and Fisher 1986). The use of a formal frame representation in FADES so far has been restricted to the area of component relationship generation.

Currently, there are four frame types used in FADES, all in conjunction with relationship generation during the layout phase: objective, component, attribute, and attribute-value frames. *Objective frames* capture the attributes associated with each of the five global factory design objectives; for example:

Objective:	minimize material-handling cost
Attributes:	material flow
	share equipment
	maintenance/repairs

The majority of the frame-based declarative knowledge is captured in component frames. *Component frames* consist of characteristics and constraints associated with each department or workstation; for example:

Department:	shear
Nexus:	share personnel
	material flow
	personnel flow
Constraint:	noise

The attribute and attribute-value frames handle the expression of underlying reasons associated with objective frames. Much of the other declarative knowledge in FADES, as indicated, is represented as simple attribute-value pairs or object-attribute-value triplets, all in predicate logic form. The exception to this is the representation of facts as Englishlike phrases in a higher-level shell embedded in the FADES program and sitting on top of the Prolog interpreter.

Procedural Knowledge. The representation method for procedural knowledge, encompassing each of the four types of knowledge in the knowledge base, has been IF-THEN rules. Sample knowledge for several categories is given in the following:

Workstation Selection

IF workstation technology is robotic
AND odd-angle insertion needed
AND force insertion is required
THEN select robot model-A cf 80

Handling Equipment Selection

IF path is variable
AND size is medium
AND volume is medium
AND distance is medium
THEN select equipment type Light-load Automated Guided Vehicle System (AGVS) cf 95

Capacity Planning:

IF quantity of this machine type is Q
AND total time needed for operations on this machine type is TT
AND total time available for operations on this machine type is AT
AND Q = TT/AT
AND the fraction of Q above the integer is less than 0.5
AND (
product demand is expected to increase by more than 20%
OR an additional product will be added within 6 months that has an operation performed by this machine type
)
THEN Round Q to next highest integer

Relationship Generation

IF material flow is medium and noise is tolerable
THEN shear and shipping/receiving are bound by the given Closeness Sum of 10 with certainty 80

Note: The combined closeness sum is a numerical measure of closeness derived from existing evidence. This sum can be either positive, representing closeness, or negative, representing distance.

Layout Application Program Selection

IF relationships have been generated
AND construction program required
AND single best layout desired
AND potential conflicts determined
THEN setup, execute, and interpret CORELAP

Layout Conflict

IF weight of machine to be placed is W
AND load limitation of candidate site is L
AND LL is (L − W)/L
AND LL < 0.1
THEN flag this assignment as undesirable

A sample logic flowchart for assignment algorithm selection appears in figure 4.

The actual internal representation of procedural knowledge in FADES varies between Englishlike syntax and Prolog, the former used when the high-level shell is being used for representation.

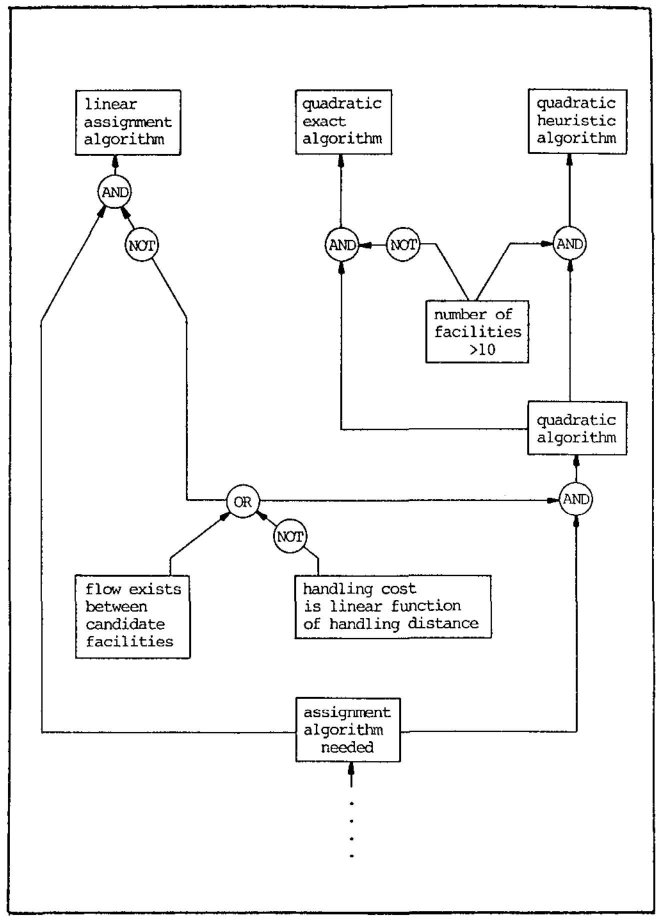

Figure 4 Sample Logic Used to Select Assignment Algorithms

Reasoning Techniques

From the FADES experience, we have determined that the design process benefits from both data-driven and goal-driven inference control, depending upon the genre of task being executed. The overall control is explicitly goal driven (in the manner of Prolog), with major subgoals being the design steps. Data-driven inference is employed in certain subgoals, such as problem determination and component relationship generation subtasks within the goal-driven mechanism, as illustrated in figure 5. Current design-state information is passed between subgoal predicates as variables. Such information could also be transferred without variable passing through assertions of fact into the internal Prolog database.

The backtracking feature of Prolog has been a mixed blessing. During system technology selection, invocation of the technology selection knowledge module can result in a list of two appropriate technologies. The design can continue with the "best" choice, and if subpar results are obtained, it can backtrack to try again with the "next-best" choice. The general backtracking mechanism works for selected design tasks, but a modified version of the backtracking algorithm might eventually be needed if Prolog continues as the primary implementation language for FADES.

Abstraction was incorporated into the FADES problem-solving process by separating decision making into levels reflecting extent of detail. This concept is illustrated for component specification and layout in figure 6. Only portions of this abstraction methodology are actually implemented in the current version of FADES.

Particularly in the relationship generation task, methods are needed to limit the search space of all possible attribute combinations. A method similar to constraint bounding of the solution space (such as used by Fox [1983] for job-shop scheduling) is used to limit the possible queries during relationship generation. After an introduction to the way in which relationship uncertainty is addressed, the general inference procedure for developing component relationships is presented.

Combining Evidence. When evidence is combined in FADES, especially in goal-directed modules such as MATHES, the method of combining evidence is generally a direct application of the MYCIN calculus. However, for relationship generation, a thorough study of several methods of combining evidence was conducted, so that the way in which human designers form relationships could better be modeled. The study results indicated that both a closeness

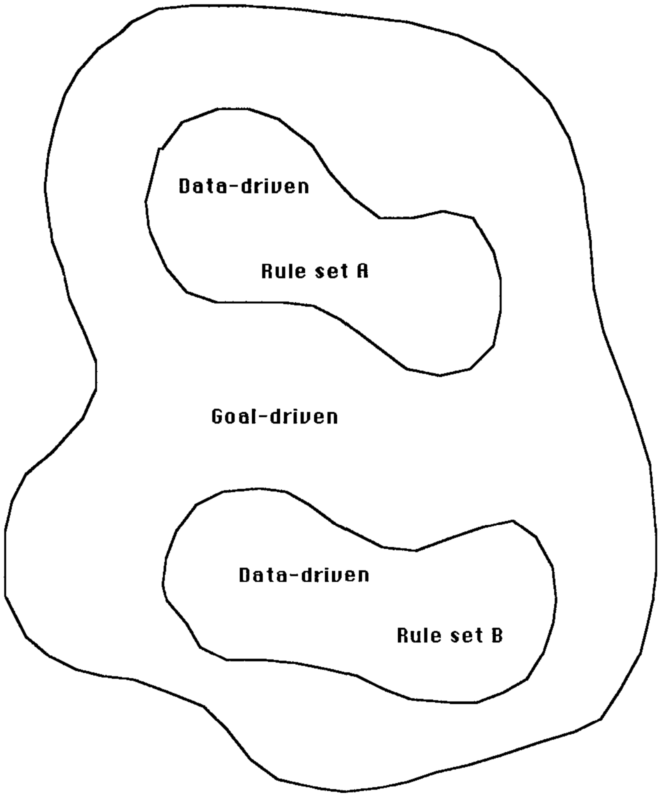

Figure 5. Mixed Inference Mechanisms Used in FADES.

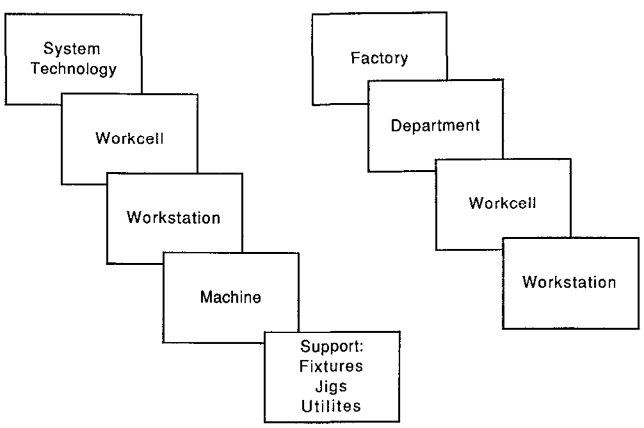

Figure 6. Abstraction Levels for Two Design Modules, Specification and Layout.

sum and a measure of the strength of belief in the closeness sum of two components is involved in relationship generation.

I now discuss the function f (mentioned earlier) of an attribute value and the weighted factor of its respective attribute (remember that attributes consist of objectives) that produces a closeness result, also referred to as a closeness sum.

Let τ be the set of factory design objectives whose elements are denoted by $t_k \epsilon \tau$ for objective k. Let S_j be the grouping of relevant attributes for $t_k \epsilon \tau$. Let W_j be the weight of attribute j, where $W_j \epsilon \{\pm 1, 2, 3, 4, 5\}$. The W_j for objectives that denote distance rather than closeness are taken as negative numbers (for example, the W_j for noise would be -3). Given $A_{j,i}$, the set of literal attribute values for each attribute j, $A_{j,i}$ represents attribute measurement i for attribute j. Let B_j be defined as the set of numeric values for attribute j, where B_j and A_j share cardinality (that is, there are corresponding numeric values between the two). There exists some $B_{j,i}$, a numeric value associated with attribute measure i for attribute j. Using an interval scaling technique, $B_{j,i} \epsilon \{.2, .5, 1\}$, where $B_{j,i} = 1$ represents most important $B_{j,i} = .5$ average importance, and $B_{j,i} = .2$ least important, the following attribute vaues exist:

t_k : minimize material handling cost

$A_{j,i}$: material flow value i
A_{11} = heavy flow
A_{12} = medium flow
A_{13} = light flow
$B_{11} = 1$
$B_{12} = .5$
$B_{13} = .2$

Note: For computation purposes, $B_i = \{2,5,10\}$.

There exists a value, $C_{m,n,j}$, a closeness sum, which represents a measure of evidence of a closeness relationship between departments M and N such that

$$C_{m,n,j} = W_j B_j \quad (1)$$

where

$B_j = B_{j,i}$ when $A_j = A_{j,i}$.

This closeness sum, $C_{m,n,j}$ is intended to be a measure of a single piece of evidence. A method was needed for combining multiple pieces of evidence into a single value. Several approaches that model human reasoning were explored.

In all, five methods of combining evidence, ranging from averaging to adding to a combination of the two, were considered for implementation (Bray 1986). Each of the five methods was tested on a subset of all of the possible combinations of pairs of evidence taken from the attributes in the knowledge-acquisition survey, such that the same subset was used for each. This subset consists of nineteen paired pieces of evidence. Results from each method were compared to the relationships given by an expert for the same pairings. Although 100 percent accuracy was not achieved, the model of evidential strength provided the best results given the hypothesis that one method was indeed better than the other four.

The model of evidential strength is a method of combining evidence employed in MYCIN (Shortliffe 1976) that incorporates both the averaging and adding techniques. Letting $L = \{1, 2, 3, \ldots\}$ be a number assigned to each piece of evidence $C_{m,n,j}$, to be combined, two pieces of evidence

$C_{m,n,1}$ and $C_{m,n,2}$ exist, and $S_{m,n}$ represents the combined closeness sum. Hence, for single pieces of evidence, three cases are considered:

Case I : $C_{m,n,1}$ and $C_{m,n,2} \geq 0$,

$$S_{m,n} = C_{m,n,1} + C_{m,n,2} - \frac{(C_{m,n,1}C_{m,n,2})}{100} \quad (2)$$

Case II : $C_{m,n,1}$ and $C_{m,n,2} < 0$,

$$S_{m,n} = C_{m,n,1} + C_{m,n,2} + \frac{(C_{m,n,1}C_{m,n,2})}{100} \quad (3)$$

Case III : $C_{m,n,1}$ or $C_{m,n,2} < 0$,

$$S_{m,n} = \frac{C_{m,n,1} + C_{m,n,2}}{1 - \frac{(min\{|C_{m,n,1}|,|C_{m,n,2}|\}}{100}} \quad (4)$$

For three pieces of evidence, the combined closeness sum, $S_{m,n}$, is computed for the first two pieces of evidence. Now, the $S_{m,n}$ is simply a single piece of evidence that is in turn combined with the third or remaining piece of evidence. This basic process of combining evidence in pairs is repeated for more than three pieces of evidence.

A second version of the knowledge-acquisition survey (the first version was described in Knowledge Acquisition) was developed and distributed to local industries. This version contained thirty-three paired pieces of evidence. Thirteen surveys were returned. These thirteen sets of data were tested using truth scores and simple linear regression. Results of this test again proved the model of evidential strength to be the best of the five methods for combining closeness relationship evidence.

Also associated with each piece of evidence is a strength of belief, which is a measure of truth or validity with respect to the evidence. The following three methods for measuring strength of belief were examined: (1) Bayesian probability, 2) the Dempster-Shafer theory of combining evidence, and 3) the use of certainty factors. For this particular application, certainty factors were found to be the best method of representing strength of belief.

Procedure for Relationship Generation. Given these methods of addressing uncertainty, the following is a procedure that describes how relationships are actually generated, thus giving a glimpse at one reasoning approach used in a FADES task. The six-step process involved in determining relationships between departments in FADES is:

1. Select a pair of departments.
2. Match common characteristics, and combine with any existing constraints to produce a query list.
3. Make queries concerning the nexus and constraint values contained in the query list.
4. Search the rule base or fact list for a match to the responses given in step 3.
5. Combine closeness sums and certainty factors associated with matching rules or facts.
6. Assign a relationship rating and a combined certainty factor to the department pairing.

Steps 1–6 are repeated until relationships are determined between all possible combinations of paired departments.

For each FADES task, the reasoning process is individually considered where it is felt that vanilla data-driven and goal-driven mechanisms might not provide optimal results.

External Communication

External communication involves the development of interfacing capability between FADES and its working environment. There are three important external interfaces that a factory design expert system requires for success. These interfaces are (1) external computerized problem solvers, (2) external databases, and (3) the human designer. The first FADES prototype included the interface of Prolog design logic with external conventional language, for example, Fortran programs (executable versions). The progress in this area is satisfactory for current capabilities and is not currently being enhanced except for incorporating additional external programs into FADES. A significant portion of our current research effort at North Carolina State University (NCSU) is focused on the database and user interfaces. Both are essential building blocks for further FADES development.

Work on a database interface is attempting to connect FADES to multiple heterogeneous commercial DBMSs using their query and primitive languages.

A detailed discussion of this work can be found in Bhatt, Fisher, and Rasdorf (1985). The purpose of this interface is to promote the retrieval of information needed during the design process from locations in one or more external databases, possibly of different structure, for example, hierarchical, network, or relational. In original work at Purdue (Fisher 1984), the Prolog-based FADES model was interfaced with the query language of a network DBMS, MDBS III. This work was expanded by interfacing FADES to two other DBMSs, both relational. These are relational information manager (RIM) and INFORMIX. Simple multiplexing capabilities of this interface have been demonstrated that allow the selection of the appropriate DBMS based on the kind and location of data requested by FADES. Experiments performed with the interface include a separate interface program written in the C language and an interface implemented directly in Prolog as a procedural attachment.

In work on the human interface, we are attempting to develop a multimodal communications link between FADES and the designer. For example, a current difficulty that FADES faces is the extraction of a design problem description from the human designer. Alternative modes to keyboard entry, such as voice, touch, and graphic icons, are being investigated for purposes of communicating concep-

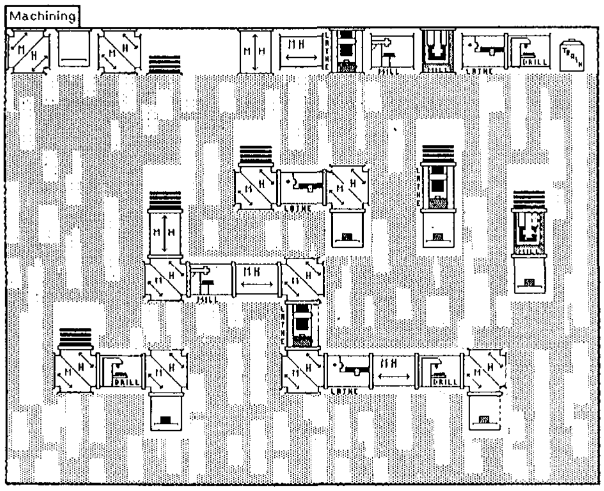

Figure 7. Iconic Menu of Components Used for Experimentation with User Interface.

Figure 8. Multimodal User Interface Experiment Incorporating Keyboard, Voice Input and Output, Graphics Output and Touch Input.

tual design knowledge from the human designer to FADES and vice versa. A sample screen experiment (shown in figure 7), implemented in Smalltalk on a Tektronix 4404, includes selection of graphic component icons with a mouse device. In this way, a user can override FADES to produce an initial layout. Figure 8 illustrates the integration of icons for factory components, a touch screen for locating components, voiced commands for entering component identifiers, and traditional keyboard input. A voice synthesizer highlights decisions made and reinforces decisions made by the system as the session progresses. Additional detail regarding this interface can be found in Fisher and Joost (1985). Success in these endeavors is resulting in more active interaction between FADES and the human designer, which promotes enhanced designer creativity and performance.

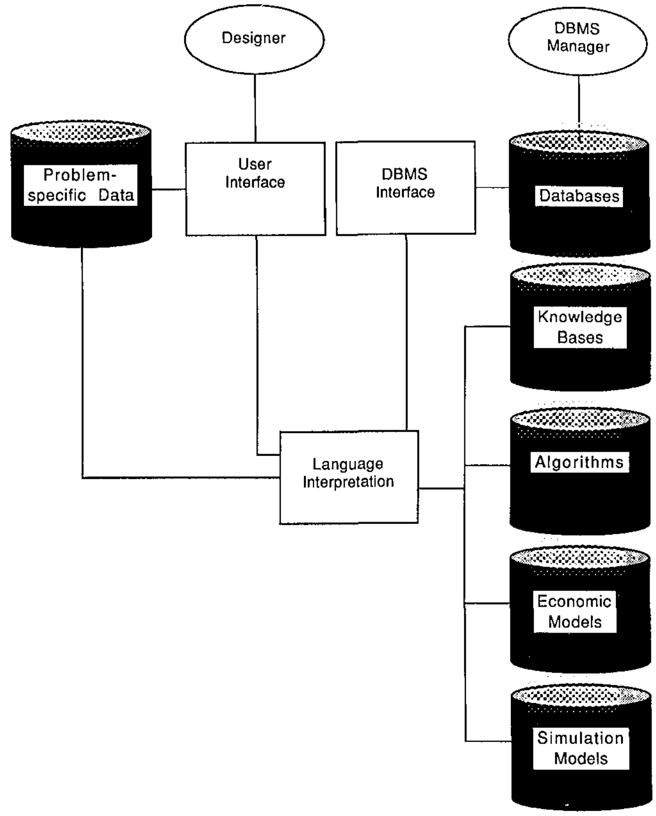

Figure 9. Current Organization of FADES.

Status of FADES

FADES has been employed primarily in validating the knowledge-based approach to factory design and in performing experimentation with functional needs in the design process. The current status of the model is presented in the following paragraphs. The most recent reference manual availiable is Kirks and Fisher(1985).

System Organization

The current organization of FADES is shown in Figure 9. The FADES model consists of the following:

- A Prolog interpreter
- Several knowledge bases containing design knowledge modules and support programs
- DBMSs that manage data needed in the design process, such as product demands and machine specifications
- A DBMS interface that allows FADES to have access to data in the DBMSs
- A collection of algorithms used in design
- Economic models for analysis of design opportunities
- A user interface that allows FADES to interact with the human designer
- A simulation model that is currently under development for analysis of generated designs

- Working memory for storage of task-specific design knowledge

Major knowledge modules in FADES as of summer 1986 are illustrated in Figure 10. The three top-level modules (largest blocks) aid the designer in design problem definition, facilities selection and specification, and layout. Surrounding the major modules are specialist modules that can perform specific tasks identified by the parent modules.

The performance of FADES to date for the design problems tested has been reasonable on small numbers of components. Reasonable means that the user feels comfortable with the response times that range from seconds to over a minute during some segments of analysis. Only one large industrial problem, a layout of existing components, has been tested. The data and resulting layout for this problem are given in Figures 11 *A, B, and C.* Time for complete layout of this facility, having 29 departments, was on the order of a half hour. The complete interactive session is closer to three times this. This time, however, is quite acceptable given the nature of the task and the benefits. Additional detail in the design would only lengthen these times.

The length of the session for the industrial example indicated was expected to be a power curve function of the number of components and the attributes that relate them. Through experimentation with smaller numbers of components, a central processing unit (CPU) time function in the exponential family was in fact determined, on the order of $t = 2e^{002x}$ for component relationship generation.

Insertion into Field Test

Although no formal field tests have been conducted, several industries and experts have been involved in the acquisition and evaluation of design knowledge. No plans exist to insert FADES into a field test any time soon; however, as indicated earlier, subcomponents such as MATHES have been tested on industrial situations, and this system is available for use. The next available subcomponents will be in the layout generation area. Field testing of FADES will be accelerated when an adequate user interface is developed, but much remains to be completed in this area.

Current Research Focus

Research into AI methods for factory design is continuing. Two areas are emphasized in our current work: (1) enhancement and redesign of the knowledge base, reasoning processes, and analysis methods where needed and (2) external communication.

In the first category, three efforts are ongoing. The first deals with better ways of establishing relationship bindings between components to be located and subsequent generation of their location under constraints. Some of the results of this effort were discussed in this article. The second is establishing complete knowledge bases for selected subtasks. An example is MATHES, originally developed with the aid of a

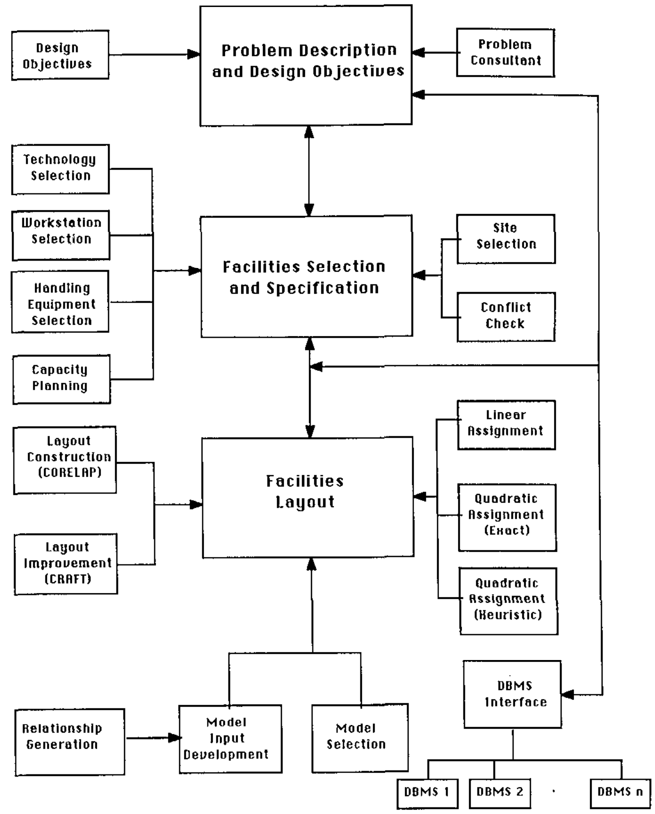

Figure 10 Major Knowledge Modules in FADES.

microcomputer knowledge engineering shell (Texas Instruments Personal Consultant) and now being subsumed by FADES. This effort is the first of several that will enlarge and enhance the knowledge base of FADES. The third category is the investigation of object-oriented programming techniques for factory design (King and Fisher (1986), especially in light of the particularly good match of object-oriented methods with factory simulation. Simulation is a very important adjunct to a factory design system, allowing the examination of the resulting design for various operating scenarios and subsequent design modification.

In the area of external communication, work is continuing on a user interface incorporating the multimodal and graphic icon ideas and enhancements to the commercial DBMS interface.

Future Direction

From our work with FADES, it seems that the following items make up an ideal factory design system:

- Design Library
- Browser
- User Interface
- DBMS Interface
- Design and Analysis Knowledge Sources
- Algorithm Library

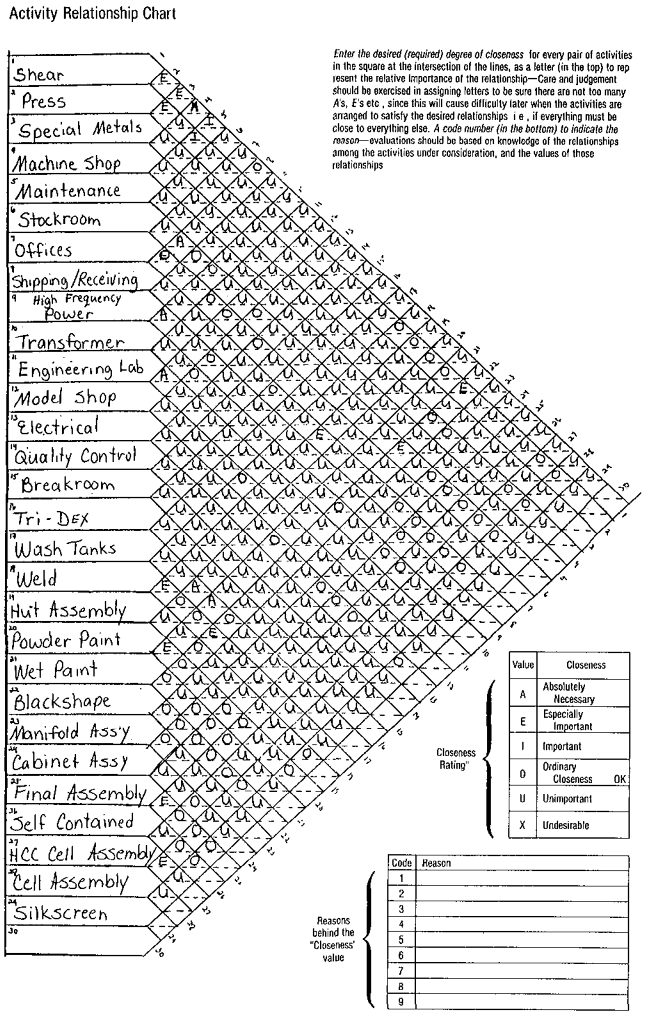

Figure 11A. Departmental Relationships (Generated).

11.	Shear	26.	Tri-Dex
12.	Press	27.	Wash Tanks
13.	Special Metals	28.	Weld
14.	Machine Shop	29.	Hut Assembly
15.	Maintenance	30.	Powder Paint
16.	Stockroom	31.	Wet Paint
17.	Offices	32.	Blackshape
18.	Shipping / Receiving	33.	Manifold Assembly
19.	High Frequency Power	34.	Cabinet Assembly
20.	Transformer	35.	Final Assembly
21.	Engineering Lab	36.	Self Contained
22.	Model Shop	37.	HCC Cell Assembly
23.	Electrical	38.	Cell Assembly
24.	Quality Control	39.	Silkscreen
25.	Breakroom		

Figure 11B. Department Layout Key.

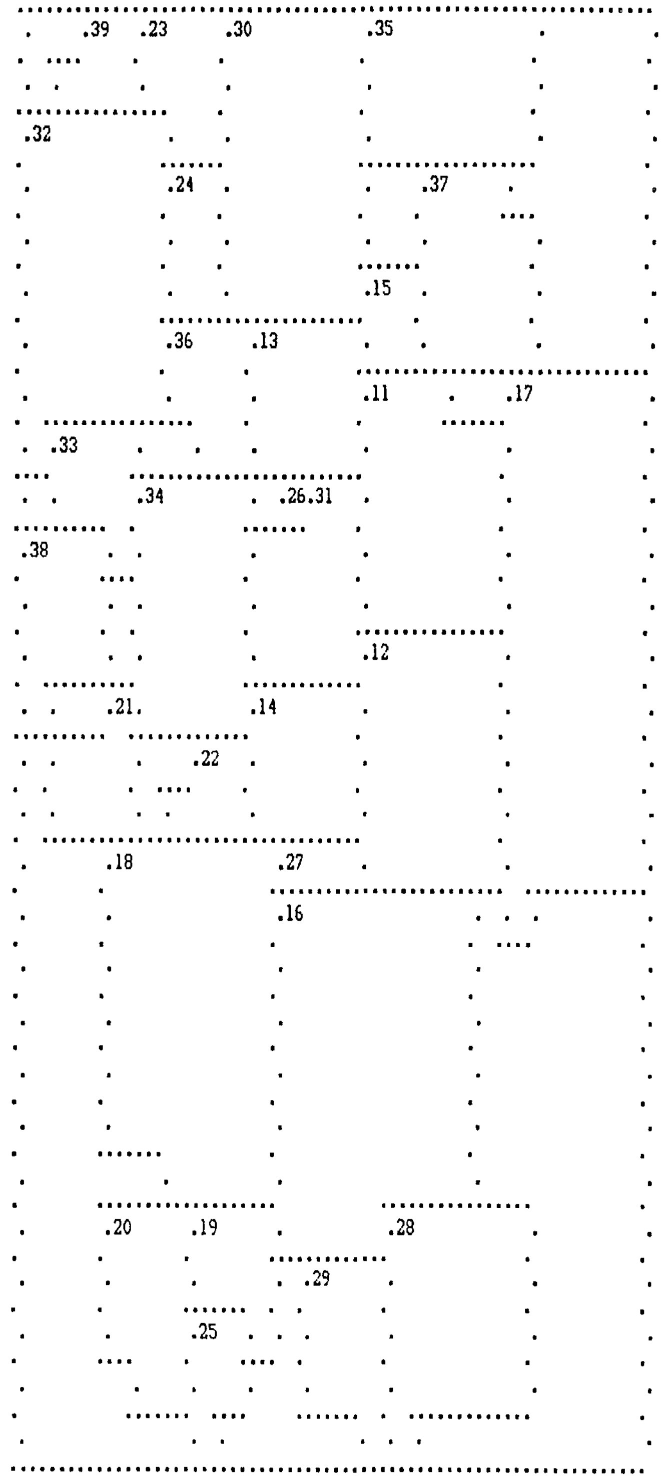

Figure 11C. Resulting Layout.

- Layout Generators
- Communication Medium (for example, a blackboard)
- Simulation-Modeling Capability

Many of these components exist in some state in FADES; however others, particularly the library and browser, are not implemented. Such a library of previous designs and partial designs, accessible by a browser, could eliminate much data gathering and provide needed templates for subsequent designs.

Summary

It has been my attempt in this article to provide a snapshot of an AI-based methodology being researched for the solution of factory design problems. To this end, the FADES system was described to support the contention that such a methodology is feasible. Should the reader desire further details on FADES or the methodology, several additional sources are listed in the references.

Acknowledgments

Work described here has been performed over several years and, during this period, has been partially supported by grants from the Integrated Manufacturing Systems Engineering Institute at North Carolina State University, the Computer Integrated Design Manufacturing and Automation Center at Purdue University, Digital Equipment Corporation, Tektronix, Texas Instruments, and Trion Corporation. Dolly Madison Industries and Kimball International have been most helpful in providing needed data for our work. I wish to thank Dr. Shimon Y. Nof of Purdue University for his contributions to the conceptualization of FADES. Research assistants (past and current) involved in these efforts are Sam Adams, Rajesh Bhatt, Anna Bray, John Dilley, Carole English, Jeremy Farber, Paul Guidry, David Hesselberth, Christina King, David Kirks, Brian Monahan, Patrick Murray, Doug Ramers, and Kamil Yavuz.

References

Armour, G. C.; Buffa, E S ; and Vollmann, T. E 1964 Allocating Facilities with Craft. *Harvard Business Review* 42:136–158

Bhatt, R V.; Fisher, E L.; and Rasdorf W. J. 1985. Infoımation Retrieval Architectures for Expert System/DBMS Communication In Proceedings of the 1985 Fall Industrial Engineeıing Conference 315–320 Atlanta, Georgia: Institute of Industrial Engineers

Bray, A K 1986 Rule-Based Relationship Generation Master's thesis, Dept of Industrial Engineering, North Carolina State Univ

Brown, D C., and Chandrasekaran, B 1984 Expert Systems for a Class of Mechanical Design Activity In Proceedings of Inteınational Federation of Information Processing Society WG5 2 Working Conference on Knowledge Engineering in Computer-Aided Design, Amsteıdam: North Holland.

Clocksin, W F., and Mellish, C. S. 1981 *Programming in Prolog, Second Edition.* New York: Springer-Verlag.

Farber, J , and Fisher, E L. 1985 MATHES: Material-Handling Equipment Selection Expert System, Technical Report, NCSU-IE-85-17, Dept. of Industrial Engineering, North Carolina State Univ.

Filey, R. D. 1985 Three Emerging Computeı Technologies Boost Value of, Respect for, Facilities Function *Industıial Engineeıing* 17(5): 27–39

Fisher, E L 1984. Knowledge-Based Facilities Design. Ph D diss. School of Industıial Engineering, Purdue Univ.

Fisher, E. L., and Joost, M G 1985. A Multimodal Strategy for Human-Machine Communication In Proceedings of the 1985 Fall Industrial Engineering Conference, 440–447 Atlanta, Georgia: Institute of Industrial Engineers

Fisher, E L , and Nof, S. Y. 1984. FADES: Knowledge-Based Facility Design. In Proceedings of the 1984 Annual International Industrial Engineering Confeıence, 74–82 Atlanta, Georgia: Institute of Industıial Engineers

Fisher, E L ; Nof, S. Y ; and Whinston, A B 1983. An Expert System Approach to Facilities Design of Computerized Manufacturing. Paper presented at 1983 Spring Meeting of ORSA/TIMS

Foulds, L. R 1983 Techniques for Facility Layout *Management Science* 29(12): 1414–1426

Fox, M S 1983. Constraint-Directed Search: A Case Study of Job-Shop Scheduling Ph D diss Dept of Computer Science, Carnegie-Mellon Univ

Hayes-Roth, F ; Waterman, D ; and Lenat, D , eds. 1983. Building Expert Systems Reading, Mass : Addison-Wesley

Heisterberg, R. J 1978. New Tools for Computer-Aided Facilities Planning and Design In Proceedings of the 1978 Annual Inteınational Industrial Engineeıing Conference, 107–120. Atlanta, Georgia: Institute of Industrial Engineers

Kim, J , and McDeımott J. 1983 TALIB: An IC Layout Design Assistant In Proceedings of the Third National Conference on Artificial Intelligence, 197–201 Menlo Park, Calif : American Association for Artificial Intelligence

King, C U , and Risheı, E L 1986. Object-Oriented Modeling, Simulation and Evaluation of Factories. To appear in Proceedings of the 1986 Fall Industrial Engineering Conference Atlanta, Georgia: Institute of Industıial Engineeıs

Kiıks, D. J., and Fisher, E L 1985. FADES Reference Manual, version 1 1, Technical Report, NCSU-IE-85-10, Dept of Industrial Engineering, North Carolina State Univ.

Lee, R C , and Moore, J M 1967. CORELAP—Computerized Relationship Layout Planning. *Jouınal of Industıial Engineering* 18(3): 147–159

McDermott, J. 1982 R1: A Rule-Based Configurer of Computer Systems *Artificial Intelligence* 18:39–88

Mostow, J 1985 Toward Betteı Models of the Design Process *AI Magazine* 6(1): 44–57.

Mutheı, R 1961 Systematic Layout Planning New York: McGraw Hill

Nicol, L. M , and Hollier, R. H. 1983 Plant Layout in Practice *Material Flow* 1:177–188.

Nilsson, N J 1980 *Principles of Artificial Intelligence* Palo Alto, Calif.: Tioga.

Sacerdoti, E D. 1974 Planning in a Hieraıchy of Abstraction Spaces *Artificial Intelligence* 5(2): 115–135.

Seehof, J. M and Evans, W O 1967 Automated Layout Design Program *Jouınal of Industrial Engineering* 18(12): 690–695

Shortliffe, E. H. 1976 *Computer-Based Medical Consultation*: MYCIN New York: American Elsevier

Tompkins, J A. 1972 Computerized Facilities Design. Ph.D diss., School of Industrial Engineering, Purdue Univ

Tompkins, J. A., and White, J A 1984 *Facilities Planning* New York: Wiley