Research report 150

AN EXPERT SYSTEM FOR HOLLOW EXTRUSION DIE DESIGN

J R Thomas*
A G Cohn

(RR150)

This report is based upon a project to develop a prototype expert system to design hollow extrusion dies. Input to the system consisted of a cross sectional representation of the required extruded shape. Previous research has concentrated on critiquing existing designs for possible areas of manufacturing difficulty but until now no one has attempted to automate the design process itself. Problems in building such a system are centred on representing and performing the required spatial reasoning, and in producing a design which meet various conflicting design goals. Output from the prototype system consists of one or more designs which satisfied the constraints for the extrusion die. Further system development is required in order to identify the optimal design when more than one design is produced and to increasing the efficiency of the implementation.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

October 1989

*Department of Psychology

An Expert System for Hollow Extrusion Die Design

J. R. Thomas,* Department of Psychology,
and
A. G. Cohn, Department of Computer Science,
University of Warwick, Coventry, CV4 7AL, U.K.

ABSTRACT

This report is based upon a project to develop a prototype expert system to design hollow extrusion dies. Input to the system consisted of a cross sectional representation of the required extruded shape. Previous research has concentrated on critiquing existing designs for possible areas of manufacturing difficulty but until now no one has attempted to automate the design process itself. Problems in building such a system are centered on representing and performing the required spatial reasoning, and in producing a design which meet various conflicting design goals. Output from the prototype system consists of one or more designs which satisfied the constraints for the extrusion die. Further system development is required in order to identify the optimal design when more than one design is produced and to increasing the efficiency of the implementation.

* Current address is: ACTL, National Research Council of Canada, 6815 8th St. NE, Calgary, Alberta, Canada T2E 7H7.

INTRODUCTION

Although designing a die to extrude a solid shape in metal is a relatively simple process, die design for a shape with one or more voids is a more complex task. At the present time there are no textbooks covering hollow die design for metal extrusion and the design skill is something of an art rather than a science Most of today's hollow die designers have received no direct formal training but have learned the skill through experience or by traditional apprenticeship. In addition, knowledge of the subject has not reached a stage at which mathematical modeling of the process is feasible.

There have been various attempts to develop expert systems which are able to perform design tasks. Most of these have been in the area of electronic engineering [1, 2, 3, 4, 5], and computer systems [3, 6] design. Very few have looked at design of mechanical systems [7, 8] and perhaps one of the most interesting of those is the "Design to Product" project currently running under the UK Government's Alvey program (See the paper by L.D. Burrow in these proceedings).

In terms of die design, the work by Purnell & Males [9] is an attempt to provide a set of tools and resources to the designer of a solid section die for non-ferrous metals. As such it is not an autonomous expert design system, but it incorporates a number of basic rules to facilitate the generation of an extrusion die orifice. Die design using this system is an interactive process between the die designer and the system, in which the system performs most of the mathematical calculations needed in the development of the die design. One clear advantage of this system is its ability to produce a NC tape for a wire spark-erosion machine.

Hirschtick and Gossard [10], working with both hollow and solid dies, adopted a different approach and developed a design advisory system to identify geometric features of the extrusion which could cause manufacturing difficulties. The system is based upon production rules to encode the guidelines [11] and used the YAPS [12] rule interpreter. As such, this system provides a convenient tool for the die designer and is useful in providing an outline of the requirements of an autonomous die design system.

Technical Preliminaries

For solid shapes there is little difficulty in designing a die, with the possible exception of calculating the bearing lengths and angles (which can be used to control flow at different parts of the shape). For a hollow shape however, the situation is much more complex since the hollows must be created. The basic technique is to support the center(s) of the die face with "stalks" (called *mandrels*) which enter the die cavity and which are themselves supported by lateral constructions joining to the side of the die, called *bridges*. This is achieved by designing a *back die* or *body* . A back die consists of a number of bridges from the side of the container which support the mandrel(s) and in turn these create the hollow(s). The metal flow deforms around the bridges and through the *ports* between the bridges and then welds in the *weld chambers* beyond the bridges before deforming around the mandrel(s) to create the appropriate shape hollow.

The principal difficulties engendered by this technique are as follows. First, whenever the metal welds, a weld streak may be seen on the surface of an anodized product. Second, the bridges must be strong enough to support the mandrel(s) as there are many tons of pressure creating the metal flow. However increasing the bridge width to provide extra support will decrease the port area which will slow the extrusion speed. Alternatively, increasing the bridge depth will slow the extrusion because of increased drag. Weld streaks may be hidden by aligning them on corners, on details or inside slots of the extruded shape. However the ports must be balanced otherwise the shape may not be extruded cleanly and evenly as desired. It is possible to weld other than in the center of the bridge by chiselling the trailing edge of the bridge asymmetrically to induce the metal to flow to one side or the other. This technique does not always work as well in practice as might be expected in theory and it tends to shorten the life of the die.

Clearly whenever the shape is masked by a bridge, it may not receive an adequate amount of metal. Certain geometric features of a shape are particularly prone to such feed problems. For example re-entrant corners (corners which protrude back into the body of the form) of sufficient size are particularly prone to feed problems when masked by a bridge. Thus designs which have such feed problems are clearly to be avoided.

A shape may require multiple mandrels if it has several hollows. If the shape is relatively small compared to the container size, it may be necessary (for a number of reasons) to extrude several identical shapes simultaneously, such dies are known as multi-hole dies (not to be confused with multiple mandrel shapes).

Thus the major tasks to be faced when designing a hollow extrusion die are:

1) identify area(s) to be supported;
2) identify feed sensitive features;
3) identify possible weld sites;
4) locate mandrel(s) in container;
5) locate feed sensitive areas (e.g. re-entrant corners)
6) locate possible bridge intersects on mandrel(s) (weld sites);
7) decide how many bridges there should be;

8) locate bridge intersections on container rim (i.e. layout bridge axes);
9) decide on bridge width and depth;
10) construct ports;
11) evaluate and balance ports;
12) proportion feed channels;
13) set weld chamber widths and heights
14) allocate bearings.

Tasks 1-3 are relatively straight forward. Task 4 is almost always so, at least for single hole, single mandrel dies where the mandrel is invariably located centrally in the container (this involves calculating the centroid of the shape). Task 5 appears to be a relatively easy task, but is complicated in that some re-entrant corners are regarded by die design experts as being unlikely to cause a feed problems. Task 6 involves identifying points on visible surfaces which could hide weld streaks. Additionally of course, any non visible side could be used as a weld site. Task 7 is perhaps one of the most critical design decisions. In general it will not be possible to make a decision on this point without (partially) designing a number of dies and evaluating and comparing them. Task 8 is again relatively simple for two port designs since the bridges are always colinear and opposite each other with respect to the mandrel centroid. However, when there are more bridges the angle of the bridge axes is less determined. In fact, some considerable time was spent during the project in trying to elicit from designers the rules they used for four port designs. The remaining tasks are all fairly local processes which seem to be fairly straight forward though little time has been devoted to them during the pilot study. Task 11, evaluating and balancing ports is clearly the hardest of these tasks.

SYSTEM DEVELOPMENT

The Knowledge Acquisition Phase

In the initial phase, a number of experts were observed solving various typical design problems. Most experts were able to produce an acceptable design in under 1 hour and typically it took them somewhere in the region of 30 to 40 minutes to complete a design. It became evident that the experts would often start with different initial solutions to the design problem but that upon refinement of their solutions they would usually converge upon a single common design type.

The second phase involved a single articulate and enthusiastic expert die designer who was presented with a large collection of design problems and protocols of the solutions were recorded. Using these protocols and a laddering technique, the basic concepts used in the domain were elicited from the expert. Where these techniques indicated that there was incomplete or no information available from the expert on specific topics, the expert would consult with his colleagues. If it became clear that the information was unknown in the industry and that this information could improve the design process, then specific research projects were initiated to obtain the required information. As a direct by-product of this acquisition stage at least two successful and innovative design processes were developed.

Architecture

From the information gained in the previous stage it was decided to adopt a mixed frame/rule based architecture for the system as this seemed to best fit the expert's representation of the problem domain. At the time of development there was no commercial product available with the required architectural features so we developed an in house representation system.

SYSTEM ARCHITECTURE

The architecture developed to support the Die Design Expert System can be briefly described as follows. There are five main modules: the geometric reasoner; inference engine; graphics system; parser and; the user interface. All the code is written in Franz LISP.

Geometric Reasoner

Input to the system consists of a line drawing specifying the shape to be extruded. At present this is entered as a series of line segments of a specified width and curvature (though non zero curvatures are not currently handled). colinear lines of different widths are entered separately. Each line segment is specified by two (x,y) coordinates and the width. In addition, each line segment may be tagged to indicate that its exterior surface is "visible" and thus weld lines should be avoided along its surface.

The geometric reasoner (also known as the shape preprocessor) converts the line drawing shape description into a structured and knowledge based abstraction. This information is stored in a set of structures that will be explicitly accessed by the inference system's rule-base.

Shape description input is 'simplistic' in that it is insufficient to design the die-plate, but it is sufficient to be able to design the underlying bridge structure. (The simplification currently made in the shape description is that slots are described by an enclosing rectangle rather than their precise geometry and details are described by a single point).

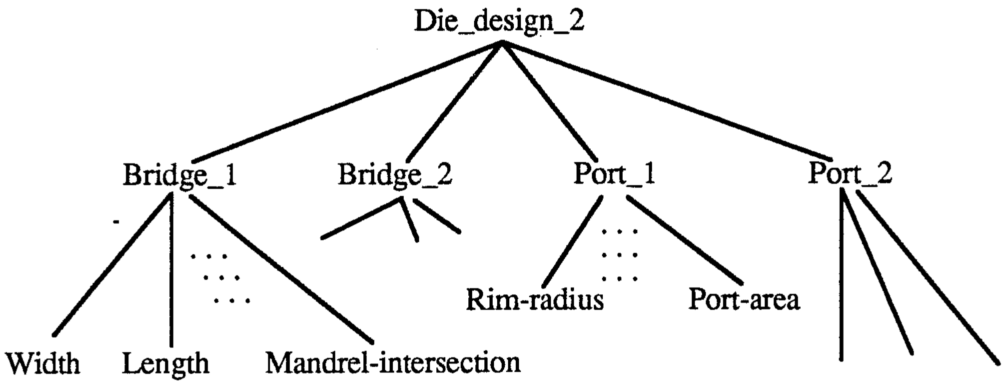
Pre-processing consists of finding the mandrels, and labeling all the webs, flanges, re-entrant corners etc. It will for instance identify and label those mandrel corners that are mathematically re-entrant, but will leave the inference engine to decide whether the re-entrant is significant (i.e. is likely to be a feed problem).

Inside/outside descriptions with respect to the mandrel are also computed. In addition, various numerical quantities are calculated including the center of a circumscribing circle for a set of points, line intersections, centroids and areas etc.

This is a fairly straight forward component of the system and little intelligence is required. Indeed conventional programming techniques were used to implement this section of the system.

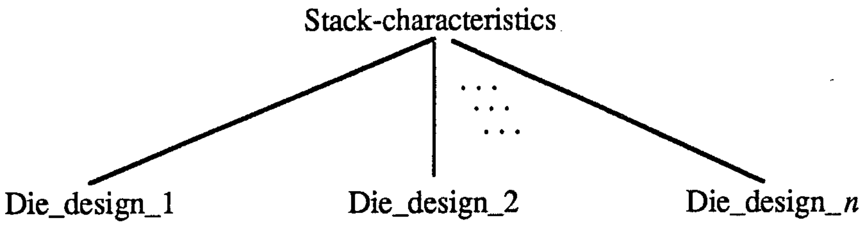
Knowledge Structures

The Expert system constructs a hierarchical knowledge base as it is operating, to store and retrieve the information that it has inferred so far. This knowledge base can be thought of as a set of nodes representing structures and sub-structures, arranged so that sub-structures occur as parts of their corresponding super-structures, e.g.



It can be seen in the above example that the breadth of the hierarchy is composed of different descriptions at the same basic structural level whilst the depth of the hierarchy represents increasingly detailed levels of description.

Different branches of the hierarchy can also be used to store information about competing possibilities, e.g.



A task will in general be associated with a node in the knowledge base that is relevant to whatever it is doing, i.e. tasks are usually 'attached' to the node that will contain most of the facts that the task will infer. To give an example based upon the above knowledge base fragment, each of the die-design branches would have an associated task that would attempt to fill in the details of the die-design nodes. In this way tasks are 'focused' on those parts of the knowledge base on which they are intended to concentrate their work. Note however that whilst tasks are 'focused' as discussed above, they can still access the rest of the knowledge base if necessary.

Within the knowledge-base the nodes are created dynamically by the rules and so the knowledge base framework grows as the system continues to operate.

Rules

The rule base is the heart of the system. It is in rules that the design knowledge is encoded, including both how to construct designs and how to evaluate and compare them.

At present all rules are forward chaining rules. In a design task it seems natural to reason forwards from the initial data rather than work backwards from some hypothetical design goal. Observation of our experts during the knowledge elicitation phase of the project confirmed this view* .

Rules in this system may be considerably more complex and sophisticated than the traditional "IF ... THEN..." rule. For example it was found desirable, for expressiveness and efficiency reasons, to allow an ELSE component in the rule. However this then creates a problem. Variables in the premise cannot be used in the ELSE part as they will not be bound since the premise evaluated to false. The solution adopted was to have a two level premise component: (a) a test (tagged as GIVEN) which has to succeed for the rule to be applicable at all and then ; (b) a further test (tagged as IF) to choose between the THEN and the ELSE branch. Variables bound in the first premise component may thus be used in either branch of the conditional (while those bound in the second part may only be used in the THEN branch).

The general form of a rule in this system is:

```
(Rule-name
                [(AUTHOR  name)]
                [(DATE  date)]
                [(REASON  reason)]
                [ONE-SHOT]
                [LAZY | EAGER]
                [SYMMETRIC  var [var]+]
      GIVEN [(premise-clause-list)]
                 [IF (premise-clause-list)]
                 THEN (conclusion-clause-list)
                 [ELSE (conclusion-clause-list)])
```

Optional items are enclosed in square brackets. A '+' means 'one or more times'. The AUTHOR, DATE and REASON slots are there to provide simple housekeeping information. All of them are optional and are ignored by the system.

The ONE-SHOT and LAZY, EAGER options affect the way that the rule is used. ONE-SHOT declares that the rule is to be used only once within a given task-instance. Once the rule has successfully fired it will remove itself from that task-instance's rule agenda.

LAZY is the default mode of rule application and is mutually exclusive with the EAGER option. If the rule is declared to be EAGER then all the bindings that satisfy the rule at the time of evaluation will be used on that iteration of the rule cycle. In the LAZY mode of evaluation, only one valid substitution will be applied on any given iteration of the rule evaluation cycle.

A set of variables can be declared as being equivalent in terms of their bindings by using the SYMMETRICAL option. This feature is useful in order to stop two identical bridge designs being generated, one being a reflection of the other.

Premis clauses may test values in a particular frame, ask the user for a value, retrieve a set of values or perform arbitrary LISP computation (eg. to sort a previously returned set of values). Conclusion clauses usually conclude about the values of a particular slot in a particular frame, but may contain other actions such as printing a message. There is always a current frame (see the section on Tasks below) to which retrievals, tests and conclusions refer to bu default, but a special syntax allows this to be changed in any given premis or conclusion clause by specifying an optional explicit path.

Other features of the rule language included a macro facility and control annotations to affect the way and times at which the rule is fired.

Tasks

Rather than having a single huge, homogeneous rule set, the architecture allows rules to be grouped according to tasks. This yields both efficiency gains and increased flexibility of control. For example different tasks can be associated with fleshing out different die designs and such tasks can be suspended or resumed independently as required.

---

* Interestingly, it would seem that this was not always the case, it used to be the practice of most die designers to proceed by modifying one of a small number of 'standard' dies to solve a particular problem. The reasoning of some die designers still seems to follow this methodology as they try to reason backwards from a small number of prototypical designs, trying to see which one fits the shape best. However, advances in machining have made this method obsolete as almost any design can be machined today.

Therefore tasks provide a mechanism for structuring the rule base and reducing the search space, because only rules in active tasks are considered at any one time. In addition, they effectively reduce a rule's size because some rule preconditions are made implicit in the task preconditions.

Task syntax is expressed in a similar format to that of a rule and consists of:

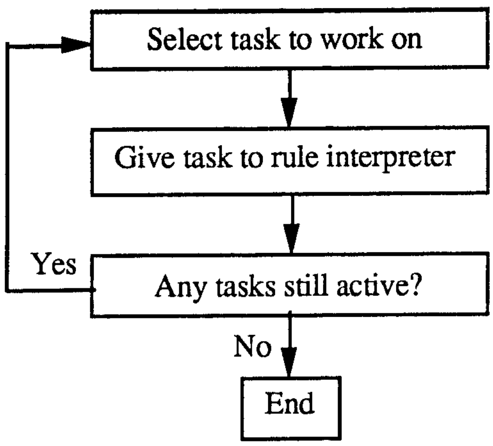
```
(Task_name
        [(PRECONDITIONS   preconditions)]
        [ON_ENTRY_ACTIONS on-entry-actions)]
        [(SUCCEED_CONDITIONS   succeed-conditions)]
        [(SUCCEED_ACTIONS   succeed-actions)]
        [(SUSPEND_CONDITIONS   suspend-conditions)]
        [(UNSUSPEND_CONDITIONS   unsuspend-conditions)]
        [(FORWARD_RULES   forward-rule-agenda)]
        [(BACKWARD_RULES*   backward-rule-agenda)]
        [(RULE-PRIORITY rule-priority-list)])
```

Among the tasks in the current version of the system are tasks which a) find potential feed problems; b) find potential weld locations; c) propose bridge/mandrel intersections for two, three and four port designs; d) identify actual feed problems with particular partially instantiated designs, and; e) which refine two, three and four post designs.

Inference Engine

The inference engine is the second major module in the system. Its basic mechanism can be thought of as comprising three parts; (a) a task interpreter; (b) rule interpreter and; (c) a set of task control demons.

In the system the task interpreter is a loop whose basic operation can be described by the following flowchart:



There is a global agenda of tasks that are currently active, whilst each task instance has its own agenda of usable rules.

The rule interpreter takes a task and selects a rule from the task's own agenda of rules that it has available to it. This is done on the basis of priority and usability. Each task gives certain rules higher priorities than certain other rules, and rules of a higher priority will be checked for usability before rules of lower priority.

Two criteria must be met for a rule to be usable, the premise must have a consistent set of variables bindings (i.e. that make the premise true), and that set of variable bindings must not be equivalent to a set that has previously been used by this rule in the context of this task (cf. refractoriness in OPS5).

Task demons constituted the third section of the inference engine. The task control demons are activated whenever an entry is made into the database that is referenced by either a task's precondition, succeed condition, suspend condition or unsuspend condition. They check to see whether the condition is fulfilled and perform any associated actions. The associated actions fall into two categories, implicit and explicit.

---

* At the present time there are no backward chaining rules in the system and this slot is currently ignored.

Explicit conditions are those given by the writer of the task description in the relevant action slot. Such actions may be to assert some fact into the database, to call some function or procedure, to focus the task instance on some part of the database, or maybe to perform no explicit action.

Implicit actions deal with creating new tasks, linking tasks into the relevant part of the database hierarchy (unless over-written by the task writer), asserting that a task has started or finished, and adding or removing the task from the active or suspended task agendas.

The Parser

Translation from the external syntax of tasks and rules into the internal form and performing consistency checking is carried out by the parser. It takes three sets of data files as input and from them produces four parsed data files.
The three files that it accepts as input are the database structure declarations, the task descriptions and the rule-base. Output from the parser consists of (a) a parsed database declaration; (b) a parsed task description knowledge base; (c) a parsed rule-base and; (d) a file that sets the relevant global variables that contain the indexing information needed to trigger the instantiation, termination and suspension of tasks.

Having the parser as a separate unit allows the inference engine to work using data structures tailored to the application whilst allowing the knowledge base to be written in a more straightforward and readable manner. Incorporation of the parser mechanism within the inference engine structure so that it can interpret the 'raw' knowledge base would be feasible but there would be corresponding additional computational overheads when the inference engine is running.

Graphics System

A very rudimentary graphics output facility has been constructed in order that designs can be displayed. Extensive use is made of the SUNCORE graphics system (SUNCORE is an extension of the CORE graphics standard implemented on the SUN range of workstations) in the implementation of the graphics facilities.

Within the graphics system the system can draw the input shape on the workstation screen, label the parts of the shape as computed by the shape preprocessor and display the bridge layout associated with any particular design. In addition, the output can be printed for a hard copy of the design. A future enhancement of this system would also provide a facility to output NC tape for the production of the designed die.

User Interface

The user interface would ultimately be a very important part of the system, providing explanations and justifications of the design. At present the user interface is very rudimentary. The graphics system already referred to could be viewed as part of it. In addition there is a simple interface to the history of rule firings. To date, the LISP debugging and trace package has been able to provide much of the interface function required, albeit in a somewhat unfriendly form.

System Operation

In this system a design is built up in stages, starting with a very abstract design which merely specifies the intersections of the bridges on the mandrel. Successive refinements specify the intersections of the bridges on the die container circumference, their cross-sections, the radiussing of the ports, mandrel undercuts and so forth. At any one stage there may be any number of designs being developed within the system which have for one reason or another been temporarily, or permanently, suspended.

As a result of the system architecture it is possible for the system to posit one or more design solutions for any one input shape. When multiple designs are proposed by the system there is at present no mechanism for placing them in any order of preference.

CONCLUSIONS

Capabilities and Limitations of the Prototype System

At the present stage of development, the system is able to produce designs for single-hole dies. The designs produced by the system include consideration of two, three and four port solutions. There are some limitations in the types of structure that the present system is able to work with, such as basic straight line shaped cross-sections. In addition, the system is restricted to non-remote mandrel designs.

In designing and implementing the prototype system we have concentrated on the development of the bridge structure and the associated ports. At present the system doesn't have a sufficiently detailed representation of the shape to design the die-plate but the implementation of this should be a relatively simple problem.

The system can identify likely problem areas (e.g.. feed, surface quality etc) with the shape to be extruded and problems with the designs which the system has suggested in its initial design phase. One advantage of the present system structure is that it provides the capability for the system to produce designs based upon several 'competing' knowledge sources. So it is possible for the system to have knowledge sources based upon 'die-designer X' and 'die-designer Y' and using both these knowledge sources the system will produce designs based upon each style of die design.

We are led to understand that approximately 80-90% of the single-hole production dies that are currently designed would in theory be covered by the prototype system, even with its existing limitations. As a prototype, the current system takes considerably longer than a human would to produce a satisfactory design. In this feasibility stage attention has been concentrated upon demonstrating that a design system is possible rather than on considerations of efficiency. Re-coding the system in a more efficient form and using a more powerful workstation is likely to result in a system performance considerably closer to that of a human expert. As more is learned about the design process it may be possible to further improve the efficiency of the system.

During the two years of the feasibility phase of the project there have been a number of changes within the die design technology. A number of these have been as a direct result of the interaction between this project and the work of our expert. In addition it has been possible to identify a number of areas where there is a need to promote further basic research in order to provide parametric data for inclusion within the expert system.

During interactions with the experts it was interesting to discover that a number of the 'rules' that they were using in developing their designs were in fact inaccurate and were often based upon 'old' technological constraints which no longer held. For example, although experts are aware that there have been changes in the standard of the die material, they were still using 'rules' which were based upon producing dies at the time when the quality, and therefore strength etc., of the die material was more variable. In general it is clear that die designers are often operating in what is effectively a more cautious mode and as a result producing less efficient dies than could ideally be the case. With the growing awareness of this information there have been some changes in the die design process.

Further Developments

This project has clearly demonstrated the feasibility of a die design expert system. The next stage is the development of a production prototype which would be able to operate within a production plant. This would require the development of an output capability to produce either a full line drawing or the production of an NC tape. The input capability could also be extended to except a CAD system output.

Acknowledgements

We gratefully acknowledge the financial support provided to this project by ALCAN International, Warwick University for providing our time and support, and to the various die designers who have so willing given of their time and expertise. In particular we owe our thanks to Mr G. D. Footring for his enthusiasm and hard work on the project and to Mr I. H. Rowsell who has given us so much of his knowledge and time with such enthusiasm.

References

[1] Sussman, G. "Electrical Design: A problem for AI research". Proc. Int'l Joint Conf. AI, pp. 894-900, 1977.

[2] McDermott, D. "Circuit design as problem-solving". In J. C Latombe (ed.), AI and Pattern Recognition in CAD, Amsterdam: North-Holland, pp. 227-252, 1978.

[3] Grinberg, M. "A Knowledge-Based design system for digital electronics". Proc. AAAI First Ann. Nat'l Conf. AI, pp. 283, 1980.

[4] Mitchell, T., Steinberg, L., & Shulman, J. "A Knowledge-based approach to design". IEEE Trans. Pattern Analysis and Machine Intelligence, 7(5), pp. 502-510, 1985.

[5] Subrahmanyam, P.A. "Synapse: An expert system for VLSI design". Computer, Vol. 19, No. 7, pp. 78-89 (July 1986).

[6] Polit, S. "R1 and beyond: AI technology transfer at DEC". The AI Magazine, Vol. 5, No. 4, pp. 76-78 (Winter, 1985).

[7] Mittal, S., Dym, L.D. and Morjaria, M. "PRIDE: An expert system for the design of paper handling systems". Computer, Vol. 19, No. 7, pp. 102-114 (July 1986).

[8] Brown, D.C. and Chandrasekaran, B. "Knowledge and control for a mechanical design expert system". Computer, Vol. 19, No. 7, pp. 92-100 (July 1986).

[9] Purnell, C. & Males, D. "Extrusion Die Design by Computer". Light Metal Age, Vol. 38, No's. 3 & 4, pp.12-15, 1980.

[10] Hirschtick, J.K. & Gossard, D.C.. "Geometric reasoning for design advisory systems". Computers in Engineering 1986, Vol. 1, pp.263-270, The American Society of Mechanical Engineers, New York, N.Y. 1986.

[11] Aluminum Association. "A guide to aluminum extrusions". Technical Report, The Aluminum Association, Inc., 818 Connecticut Avenue, N.W., Washington, D.C. 20006, (May, 1979).

[12] Allen, E. "YAPS: Yet Another Production System". Technical Report TR-1146, University of Maryland AI Group, (February, 1982).