Journal of Intelligent Manufacturing, 16, 499–513, 2005

Development of an object-oriented blackboard model for stamping process planning in progressive die design

S. B. TOR[1,2], G. A. BRITTON[2] and W. Y. ZHANG[3,*]

[1]*Singapore-MIT Alliance, SMA-NTU Office, Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639 798*
[2]*School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639 798*
[3]*Institute of Artificial Intelligence, College of Computer Science, Zhejiang University Hangzhou, China, 310027*
E-mail: wyzhang@ntu.edu.sg

Received February 2004 and accepted November 2004

This paper reports preliminary work to investigate the suitability of using a blackboard framework as a problem-solving model for stamping process planning in progressive die design. The model is described at two levels: knowledge level and computational level. The knowledge level describes how the stamping process planning domain is represented in a blackboard architecture. The computational level describes how the blackboard architecture is modeled and implemented using object technology. A software prototype has been developed using CLIPS and C++ interfaced with Solid Edge CAD system. An example is presented to illustrate the feasibility and practicality of the proposed approach.

Keywords: Blackboard architecture, graphs, object-oriented, progressive die design, stamping process planning

1. Introduction

Progressive dies for producing sheet metal parts in mass production have been widely applied in various industries such as aerospace, electronics, machine tools, automobiles, and refrigeration. These dies can perform piercing, notching, cut-off, blanking, lancing, bending, shaving, drawing, embossing, coining, trimming, and other miscellaneous forming operations at a single setup. Hence a progressive die is generally very complex. Stamping process planning and die structure design are difficult and demanding tasks.

Stamping process planning starts with an unfolding of a model of stamped metal part to produce a flat pattern, followed by nesting the pattern to produce a blank layout. Next, stamping operations are planned and operations are assigned to die stations. The resulting plan is typically represented as a strip layout, which guides the subsequent die structure design. The productivity, accuracy, cost, and quality of a progressive die mainly depends on the strip layout, and hence a stamping process. However, stamping process planning still remains more of an art rather than a science. Historically, this activity is mainly carried out manually, based on designers' trial-and-error experience, skill and knowledge. This is in spite of recent advances in the field of artificial

---

[*] Author for Correspondence

intelligence (AI), which have achieved a lot of success in incorporating built-in intelligence and applying diverse knowledge to solving this kind of problem. The main difficulty is that existing knowledge-based systems for stamping process planning lack a proper architecture for organizing heterogeneous knowledge sources (KSs) in a cooperative decision making environment. This limits both their practicability and scalability.

To address the above issue, it is necessary to provide a cooperative problem solving strategy that can foster communication between diverse KSs, and accommodate different knowledge representation schemes within an integrated framework. A blackboard architecture is an effective AI technique for integrating diverse KSs. This paper presents our work on developing a blackboard architecture using object technology. The blackboard model is represented using unified modeling language (UML) version 1.5.

Our earlier work on case-based reasoning (CBR) for stamping process planning and die design (Tor *et al.*, 2003) is also integrated into the developed object-oriented blackboard model (OOBM) to search and reuse past design experience to solve new problems. For conciseness, the CBR methodology is not repeated in this paper. The limitation of a single CBR approach has been overcome in the OOBM, because other reasoning approaches are incorporated as well. A prototype system has been implemented using the object-oriented expert system shell C language integrated production system (CLIPS) (Giarratano and Riley, 1998), which is interfaced with a parametric- and feature-based CAD system, Solid Edge through C++.

Section 2 discusses related work in computer-aided stamping process planning, and different domain-specific applications of blackboard architecture. Section 3 describes the blackboard architecture for stamping process planning. Section 4 describes the implementation of the architecture using object technology by means of UML standard notation. Section 5 illustrates the approach using an example. The conclusion is given in Section 6.

2. Related work

Research in the computer-aided stamping process planning has been widely reported since 1970s. The advantages of automated process planning are productivity improvements, cost reductions and design automation.

From mid 1970s to mid 1980s, the first generation of CAD/CAM systems for progressive die design were developed (Fogg and Jaimeson, 1975; Nakaham *et al.*, 1978; Murakami *et al.*, 1980; Bergstrom *et al.*, 1988) though few of them are based on AI techniques. These early systems are characterized by basic computer graphics facilities, standardization of die components, and standardization of design procedures. They reduced design and drafting lead time. However, as these systems represent design know-how in the form of conventional procedural programming languages, only generation of the die part list and drafting of the assembly and part drawings are executed using computers. The designer still needs to decide most of the important decisions interactively, including strip and die layouts.

Since late 1980s, significant efforts have been made by worldwide researchers to integrate a wide variety of AI and traditional CAD approaches to develop dedicated progressive die design automation systems including strip layout design automation.

Knowledge engineering is a popular AI technique having been used in intelligent stamping process planning and die design system. For example, researchers at University of Massachusetts, USA have described a knowledge-based system for design of progressive stamping dies for a simple hinge part (Duffey and Sun, 1991). The system generates the flat pattern geometry and develops a strip layout automatically. Researchers at National University of Singapore have been developing an intelligent progressive die (IPD) design system since late 1980s. They used feature modeling and rule-based approach to realize automatic punch shape selection, strip layout development and 3-D die configuration (Cheok *et al.*, 1996; Cheok and Nee, 1998). Based on a feature-relationship tree that describes the stamped metal part and its topological information, model-based reasoning and spatial reasoning techniques have been employed to reason out certain stamping processes and guide the overall planning process to develop the strip layout automatically. Researchers at the Indian Institute of Technology have developed a computer-aided die design system, CADDS, for sheet-metal blanks (Prasad, 1994), based on heuristic rule-based reasoning and parametric programming techniques. The greatest advantage achieved by the system is the rapid generation of the most efficient strip layouts. Researchers at the University of Liverpool have worked on knowledge-based design automation for progressive piercing and blanking dies (Huang *et al.*, 1996; Ismail *et al.*, 1996). Their work is based on applying a coding technique to characterize the stamped part geometric features, which is subsequently used to generate the type and layout of the die punches, and then develop the strip layout automatically. Researchers at Huazhong University of Science and Technology, China, have developed an intelligent progressive die design system, HPRODIE (Li *et al.*, 2001). With feature mapping, rule-based reasoning and case-based reasoning techniques, most of design processes including strip layout design can be carried out automatically. Researchers at Pusan National University, Korea, have developed a compact computer-aided process planning (CAPP) system for progressive die design (Choi *et al.*, 1999). Based on production rules, the work is capable of carrying out an intelligent stamping process planning work with automatic development of blank layout, strip layout and die layout.

Researchers at the National Taiwan Institute of Technology have adopted various AI techniques including fuzzy reasoning, pattern recognition, rule-based reasoning, back-propagation neural network, genetic algorithms and petri net for the stamping process planning and design of progressive shearing cut and bending dies (Lin and Hsu, 1996; Lin and Chang, 1997; Lin and Deng, 2001).

However, the intelligent systems described above are limited either to specific application domains or require considerable interactive input from experienced designers. In our previous work, a CBR methodology for stamping process planning and die design was developed (Tor *et al.*, 2003). The proposed retrieval strategy can narrow down the design search space efficiently and retrieve the most similar design case in a reasonable period of time. However, in stamping process planning, it is difficult to obtain enough cases to cover the whole problem space in the initial stage when the system is set up. CBR will fail to generate a strip layout solution if the number of cases is insufficient.

In this paper, another popular AI technique, blackboard architecture, is adopted to develop a blackboard-based stamping process planning system. In the last two decades, blackboard architecture has been successfully used in a wide variety of areas, such as speech recognition, signal processing, engineering design and process planning. Thompson and Lu (1989) used a blackboard architecture to represent design rationales in the form of design plans and design constraints and to establish the relationships between descriptions and design processes. The system provides a cooperative decision making environment that is suitable for concurrent product and process design. Srihari *et al.* (1994) developed a real-time CAPP system for printed circuit board (PCB) assembly by integrating multiple knowledge sources (KSs), including planning expert and dynamic information processing modules in the blackboard architecture. The integrated system generates process plans that can be implemented in real time. Chen *et al.* (2001) developed a concurrent, two-stage design evaluation system for product design, using a blackboard architecture. A qualitative evaluation is applied during the stage of searching for combinations of solution principles, then a quantitative evaluation is applied to provide information on performance, assemblability, manufacturability, and costs to facilitate design selection.

In the past few years, blackboard architecture has proven to be suitable for tooling design such as fixture design (Roy and Liao, 1998) and injection moulding design (Kwong *et al.*, 1997), though this kind of application is still in its infancy stage. However, we have not found in the literature any attempt to apply the blackboard architecture to stamping process planning for sheet metal parts. It has been mentioned in our earlier work (Britton *et al.*, 2004) that a blackboard architecture is well suited for constructive problem solving like process planning of stamping operations, where the problem space is large and knowledge from many different sources must be integrated to achieve a solution.

The discussion of the blackboard architecture that follows is carried out at two levels: knowledge level (Section 3) and computational level (Section 4). The

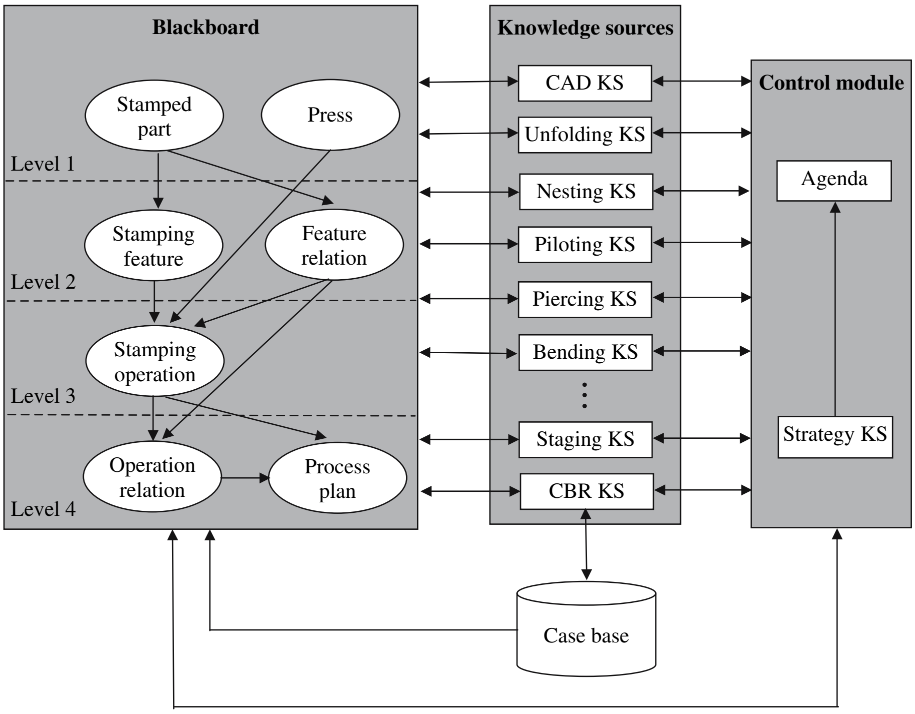


Fig. 1. Blackboard architecture for stamping process planning.

knowledge level discusses the conceptual model for representing the stamping process planning domain and the reasoning model. The computational level deals with the implementation of the knowledge level, in this case, using object technology.

3. Blackboard architecture for stamping process planning

Cooperative decision making for knowledge-based stamping process planning involves a variety of KSs such as unfolding knowledge to produce a flat pattern, nesting knowledge to produce a blank layout, various types of planning knowledge for different stamping operations like piloting, piercing, notching, cut-off, blanking, bending, etc., and staging knowledge to sequence the stamping operations. These KSs may be expressed in different representation schemes such as procedures, algorithms, reasoning methodologies, frames and rules. This justifies the use of a blackboard architecture because it can manage heterogeneous KSs effectively. The KSs interact through the blackboard to develop a solution incrementally.

The proposed blackboard architecture consists of three major components: the blackboard data structure, KSs and a control module (Fig. 1). The different components of the blackboard architecture are described in the following sub-sections.

3.1. *Blackboard data structure*

The blackboard is a globally accessible database, which contains the data and partial solutions and is shared by a number of independent KSs. The KSs contribute their partial solutions to the blackboard, which lead to a final solution incrementally. The blackboard is structured as a hierarchy of abstraction levels, which represent different aspects or stages of the solution process. Partial solutions are associated with each level and may be linked to information on other levels using algorithmic procedures or heuristic rules. Each level contains planning objects that are used to represent the solution space in an object-oriented manner.

Referring to Fig. 1, the planning solution is partitioned into four different object levels—(1)

input data including stamped part and press; (2) stamping features and feature relations; (3) stamping operations; and (4) operation relations and stamping process plan – each representing initial input or different partial solutions posted on the blackboard by the specialist KSs.

3.1.1. *Input data to the blackboard*

Input data to the blackboard mainly includes the part and press objects. The generic declaration of a part object includes the basic attributes such as part type, part dimensions, weight, surface treatments, blank thickness, blank material, annual production, blank dimensions, etc., and pointers to its constituent stamping features and feature relations that will be elaborated later on. The press object contains the attributes such as press type, press tonnage, bolster dimensions, bed open dimensions, shut height, number of strokes, etc. The press data are useful for determining the stamping operations that will be elaborated later on.

3.1.2. *Feature-based modeling to stamped metal parts*

Features are an effective way of linking two different viewpoints of the same design object or component, e.g., machining features provides a direct link between CAD and CAM (Shah and Mantyla, 1995). Similarly, stamping features of a stamped metal part enable stamping process planning tasks to be linked directly to the geometric model. Stamping features are information carriers that are used to model a stamped part with a set of design and manufacturing information including geometric and non-geometric attributes. Each of these stamping features can be manufactured with a specific stamping operation or a combination of stamping operations.

Using the hierarchical classification structure of general design features by Chen *et al.* (1991), a stamped metal part can be modeled with four categories of stamping features:

Primary features: flat, drawing, etc.
Positive secondary features: tab, curl, emboss, hem, bead, flange, etc.
Negative secondary features: hole, extrusion hole, profile, deform, slot, step, etc.
Connective secondary features: bend, blend, etc.

Both design and manufacturing information can be encapsulated in a stamping feature object from object-oriented perspective. For example, a *Hole* feature object contains the basic attributes such as feature type, feature ID, primary feature ID, position, orientation, depth, diameter, precision, roughness, etc.

Four critical types of relations among stamping features—*is-in*, *is-on*, *adjacent-to* and *precision-associated* are identified in this paper. The first three relation types—*is-in*, *is-on* and *adjacent-to*—are adopted from Chen *et al.*'s (1991) definition of relations among general design features. Within the specific domain of stamped metal parts, the *is-in* relation can be used to indicate the spatial interaction that arises when a negative stamping feature is in another stamping feature (e.g., primary feature). Similarly, the *is-on* relation can be used to indicate the spatial interaction that arises when a positive stamping feature is on another stamping feature (e.g., primary feature), and the *adjacent-to* relation can be used to indicate the spatial interaction that occurs when a connective stamping feature is adjacent to another stamping feature (e.g., primary feature). A new *precision-associated* relation type was introduced in our earlier work (Tor *et al.*, 2003) to represent design constraints that arise when a stamping feature does not directly connect to, but is associated with, another stamping feature by a toleranced dimension.

3.1.3. *Stamping operation objects mapped from stamping feature objects*

On the blackboard, the stamping operation objects are at a lower level than the stamping feature and feature relation objects. They are used to define the manufacturing process from metal strip to the formed metal part. Essentially, the stamping process planning task is to transform a set of stamping features and feature relations into a set of stamping operations, and describe the relations among these stamping operations. The generic declaration of a stamping operation object includes stamping operation type, geometric shape, geometric constraint, precision, roughness, relationship with stamping feature, and control parameter. Typical stamping operation objects include piercing, notching, cut-off, blanking, lancing, shaving, drawing, embossing, coining, and trimming. A stamping feature may be manufactured with a specific stamping operation (one-to-one mapping) or a combination of stamping operations (one-to-many mapping). Several stamping features may also be manufactured with a single stamping operation (many-to-one mapping).

3.1.4. *Graph-based stamping process plan*

After the mapping from stamping features to a set of stamping operations, the remaining process planning task is to assign stampings operation to die stations in an optimal sequence. A graph-based approach is used to arrange the stamping operation objects in a stamping process plan. The graph consists of a set of nodes that store information about the stamping operations, and a set of arcs that store information about the operation relations. Stamping operations are related to one another through two kinds of relationship, *cluster* or *precedence* relations. Cluster stamping operations are executed simultaneously and can be staged at the same die station. Sequential stamping operations are defined by precedence relations and so they are staged in adjacent die stations. *Cluster* and *precedence* relations are represented by dashed ellipses and directed solid lines respectively, as shown in Fig. 2. Note that stamping operations *C*and *D* are executed simultaneously, and are staged at the same die station, while stamping operation *A* precedes operation *C*, and is staged in a die station immediately prior to the one for the operation *C*.

The strip layout can be generated automatically using the graph-based stamping process plan.

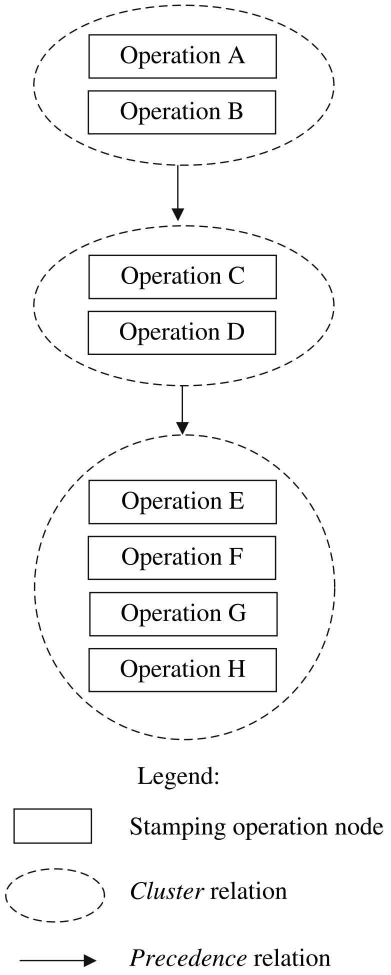


Fig. 2. Graph-based partial stamping process plan.

3.2. *Specialist knowledge sources (KSs)*

Referring to Fig. 1, the planning objects on the blackboard outlined above are not isolated data structures, but are interrelated to each other by a set of specialist KSs that resemble experts (by embodying the problem solving knowledge). These KSs are independent chunks of knowledge and do not communicate directly with each other. Instead, they participate in the problem solving process by contributing their partial solutions on the blackboard, or updating the contents of the blackboard.

The KSs related to stamping process planning include, but are not limited to, unfolding, nesting, piloting, piercing, bending and staging. Due to the modularity of blackboard architecture, it is convenient for end-users to expand the KS space in the system by integrating different methods of knowledge representation, such as procedures, algorithms, reasoning methodologies, frames and rules. A rule example in bending KS is shown below:

If a bend has a bending angle between 90° and *135°, then it needs a two-step bending operation.*

Owing to the modeling flexibility of blackboard architecture, our earlier work on CBR for stamping process planning and die design (Tor *et al.*, 2003) is also integrated into blackboard architecture to improve the productivity of stamping process planning by reusing past design experience stored as case objects in case base. A CBR KS was constructed for this purpose. New planning solutions from rule-based reasoning or CBR are saved in the case base for future usage. The CBR methodology is not discussed in this paper.

The blackboard architecture supports knowledge-based framework construction and cooperative problem solving process, but it doesn't support representation of the geometrical and topological information related to stamped metal parts, the intermediate flat pattern and blank layout, and the resulting strip layout. Hence, it was necessary to integrate the blackboard architecture with an existing CAD system. Solid Edge was chosen because of its parametric nature, its ability to enable the user to design with features, and its built-in functions that facilitate feature recognition. The CAD interface can be considered as a CAD expert (knowledge source) module, i.e., CAD KS.

3.3. *Agenda-based control module*

The specialist KSs respond opportunistically to the changes on the blackboard. Referring to Fig. 1, an agenda-based control module (Engelmore and Morgan, 1988) is used to monitor the changes on the blackboard and decides the actions to be taken next. The agenda keeps track of all the events on the blackboard, serves as a repository of specialist knowledge source activation records (KSARs) that can be selected for execution, and calculates the priority of execution for handling conflicts between KSs.

The control module uses heuristic control rules as the strategy KS to set the above agenda, e.g., by defining the dynamic priorities of triggered KSARs at the particular point in different stamping process planning stages, and invoking execution of a KSAR with the highest priority. A rule example in strategy KS is given below.

If there is no solution in blackboard after executing CBR KS, then activate other planning KSs.

During cooperative problem solving process, the solution is built up a step at a time. The sequence of KS execution is dynamic and opportunistic rather than fixed and deterministic, depending on the changes on the blackboard enacted by the specialist KSs. The specialist KS execution may result in the modification to the blackboard, bringing the system back to the beginning loop. The blackboard control cycle repeats until an acceptable solution has been found or the system can't proceed further due to lack of knowledge or data.

4. Development of OOBM using UML

In the field of software engineering, the benefits of object-oriented approach to software design are already well known, and object-oriented modeling (OOM) (Rumbaugh *et al.*, 1991) is becoming more and more widely used. One major advantage of OOM in developing blackboard architecture is its support to software modularity, reusability and scalability. We refer to the OOM for blackboard architecture as "OOBM". We use the graphical notation based on the UML Version 1.5 to represent the classes of objects and relationships between them (e.g., inheritance, association, and aggregation).

4.1. *OOM of blackboard object*

Referring to Fig. 3, the blackboard object is responsible for the aggregation of generated planning objects and retrieved case objects. The *Blackboard* class contains the basic attributes including *Problem*, *Event* and *Solution*, and basic methods include *Reset* ( ), *Return_solution* ( ) and *Return_case* ( ). *Event* indicates the changes made to the blackboard by KSs. *Reset* ( ) is used to clean the blackboard. *Return_solution* ( ) returns the solved stamping process plan. *Return_case* ( ) is used to return the successful planning solutions and store them into the case base for future CBR purpose.

4.2. *OOM of planning objects*

The blackboard is structured as a hierarchy of abstraction levels, which represent different aspects or stages of the solution process. Each level contains

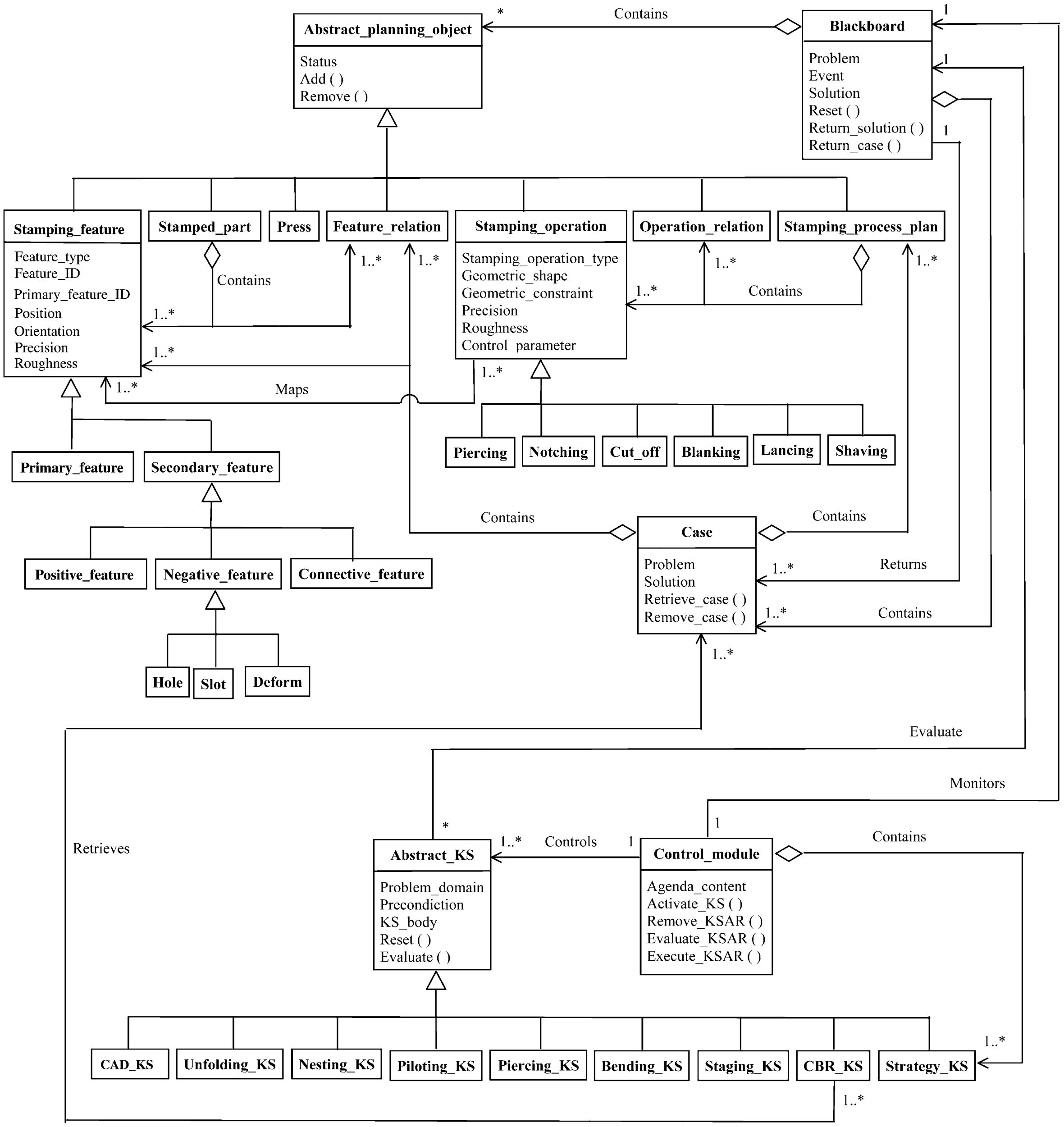


Fig. 3. OOBM developed using class diagram of UML.

planning objects such as stamped part, press, stamping features, feature relations, stamping operations, operation relations and stamping process plan that are used to represent the solution space.

Referring to Fig. 3, the class of abstract planning object can be represented as the *Abstract_planning_object* class with basic attributes, such as *Status*, and basic methods, such as *Add* ( ) and *Remove* ( ). *Status* indicates whether the planning object is currently invoked or not, i.e., serves as the context of invocation from KS. *Add* ( ) and *Remove* ( ) are respectively used to add or remove a planning object to or from the blackboard.

Classes of concrete planning objects such as *Stamped_part*, *Press*, *Stamping-feature*, *Feature _relation*, *Stamping_operation*, *Operation_relation* and *Stamping_process_plan* are defined as child classes of the *Abstract_planning_object* class.

Besides all the attributes inherited from the *Abstract_planning_object* class, *Stamping_feature* class includes new attributes such as *Feature_type*, *Feature_ID*, *Primary_feature_ID*, *Position*, *Orientation*, *Precision* and *Roughness*. Using Chen *et al.*'s (1991) hierarchical classification of feature objects, the *Stamping_feature* is further developed into lower level classes such as *Primary_feature*, *Secondary_feature*, *Negative_secondary_feature*, *Hole*, *Slot*, *Flange*, *Flat*, *Bend*, etc.

*Stamping_operation*class includes new attributes such as *Stamping_operation_type*, *Geometric_shape*, *Geometric_constraint*, *Precision*, *Roughness*, and *Control_parameter*. The *Stamping_operation* class can be further developed into lower level classes including *Piercing*, *Notching*, *Cut_off*, *Blanking*, *Lancing*, *Shaving*, *Drawing*, *Embossing*, *Coining*, and *Trimming*. The class declaration of other concrete planning objects is similar to that of *Stamping_feature* and *Stamping_operation*, and is omitted here. Note that the *Stamped_part* class is the aggregation of both *Stamping_feature* class and *Feature_relation* class. *Stamping_process_plan* class is the aggregation of both *Stamping_operation* class and *Operation_relation* class. *Stamping_operation* class associates with *Stamping_feature* class through "Maps" association.

4.3. *OOM of case objects*

The proposed blackboard architecture supports CBR by means of embedded CBR KS. Successful stamping process planning solutions are stored in the case base in an object-oriented manner. To facilitate case indexing, each case object includes problem information represented by stamping features and feature relations (Tor *et al.*, 2003). Referring to Figure 3, a *Case* class is the aggregation of classes of *Stamping_feature*, *Feature_relation* and *Stamping_process_plan*. The *Case* class contains the basic attributes such as *problem* and *solution*, and basic methods such as *Retrieve_case* ( ) and *Remove_case* ( ). *Retrieve_case* ( ) is used to retrieve previous successful planning case to the blackboard. *Remove case* ( ) is used to remove an useless case from the case base. Note that *Blackboard* class associates with *Case* class through "Returns" association.

4.4. *OOM of KS object*

KSs participate in the problem solving process by contributing their partial solutions on the blackboard, or updating the contents of the blackboard. Referring to Fig. 3, the abstract KS objects are represented as *Abstract_KS* class with basic attributes, e.g., *Problem_domain*, *Precondition* and *KS_body*. *Problem_domain* indicates the domain of KS application. Each KS object may be expressed in a different representation scheme, e.g., procedures, algorithms, reasoning methodologies, frames or rules in its KS body. However, no matter what representation scheme is used in its KS body, each KS object has a *Precondition* attribute that is defined in a standard manner. The attribute determines the situations in which the KS contributes to the problem solving process. Thus the body sections of KSs can be programmed using different techniques yet still be integrated into the blackboard architecture through the standard precondition interface. As external functional modules, the body sections of KSs are called by the main blackboard program to return their partial solutions in a standard manner to the blackboard platform.

The KS is activated if its precondition is satisfied. Instead of immediately executing the activated KS, the control module creates a knowledge source activation record (KSAR) and places it into the agenda pending execution of the KS body. The *Abstract_KS* class also contains the basic methods such as *Reset* ( ) and *Evaluate* ( ). *Reset* ( ) is used to restart the KS, and *Evaluate* ( ) is used to evaluate the state of the blackboard. Note that the *Abstract_KS* class associates with *Blackboard* class through "Evaluate" association. The *Abstract_KS* class can be further developed into lower level concrete KS classes including *CAD_KS*, *Unfolding_KS*, *Nesting_KS*, *Piloting_KS*, *Piercing_KS*, *Bending_KS*, *Staging_KS*, *CBR_KS* and *Strategy_KS*. Note that the *CBR_KS* class associates with *Case* class through "Retrieves" association.

4.5. *OOM of control module*

An agenda-based control module (Engelmore and Morgan, 1988) is used to monitor the changes on the blackboard and decides the actions to be taken next. Referring to Fig. 3, the class of *Control_module* object contains *Strategy_KS* class

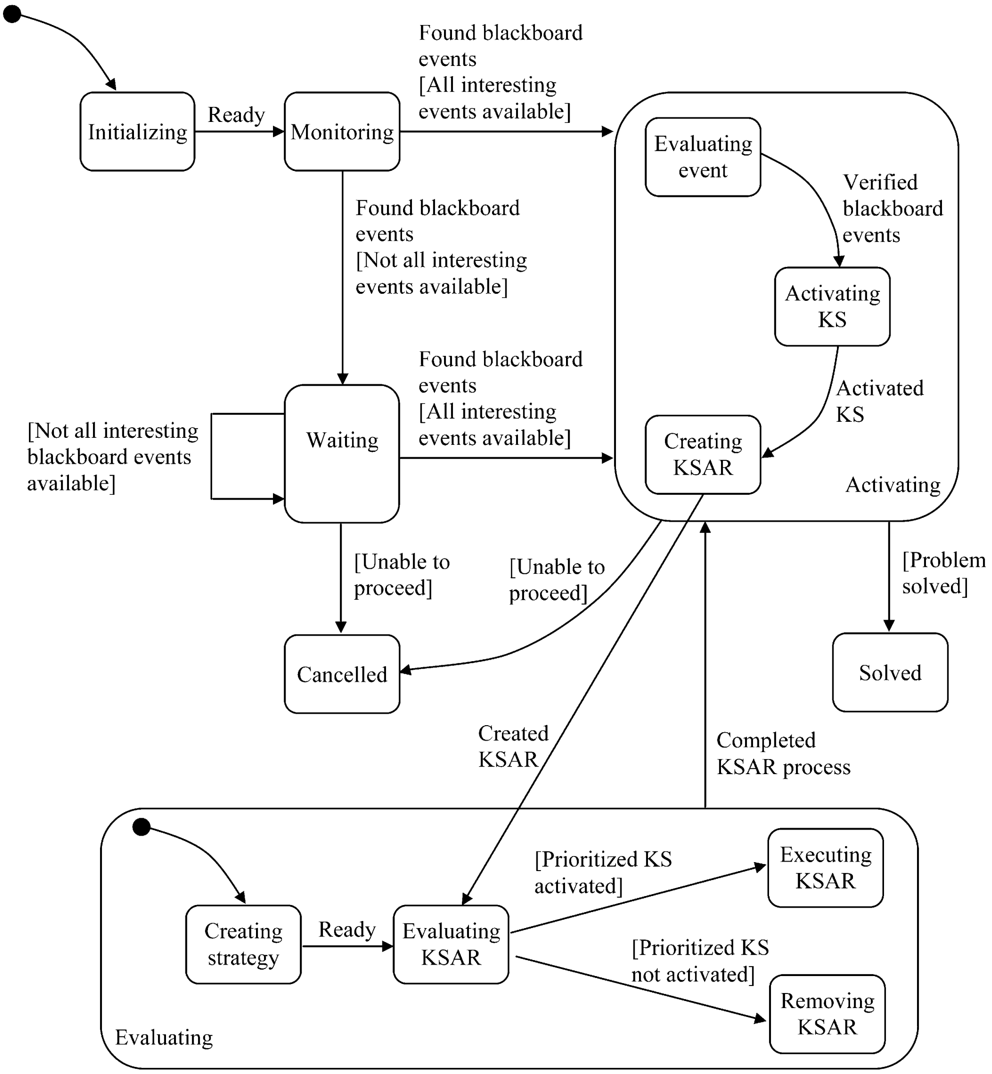


Fig. 4. UML state diagram of *Control_module* class.

because the control module applies the strategy KS to set the agenda that stores KSARs. The *Control_module* class is defined to contain the attribute such as *Agenda_content*, and methods such as *Activate_KS* ( ), *Remove_KSAR* ( ), *Evaluate_KSAR* ( ) and *Execute_KSAR* ( ). *Agenda_content* indicates the stored KSARs that can be selected for execution. *Activate_KS* ( ) is used to activate a KS and places the created KSAR into the agenda. *Remove_KSAR* ( ) is used to remove a KSAR from the agenda. *Evaluate_KSAR* ( ) and *Activate_KSAR* ( ) are respectively used to evaluate the dynamic priorities of triggered KSARs, and to invoke execution of a KSAR with the highest priority. Note that the *Control_module* class associates with *Blackboard* class through "Monitors" association, and associates with *Abstract_KS* class through "Controls" association.

The blackboard serves as a central communication and paradigm integration medium used as a repository for a global data, partial solutions or diverse KSs, while the control module serves as a dynamic agent responsible for interacting among the KSs that operate upon the blackboard. As such, a UML state diagram shown in Fig. 4 is well suited for capturing the dynamic behavior of the *Control_module* class. Here we

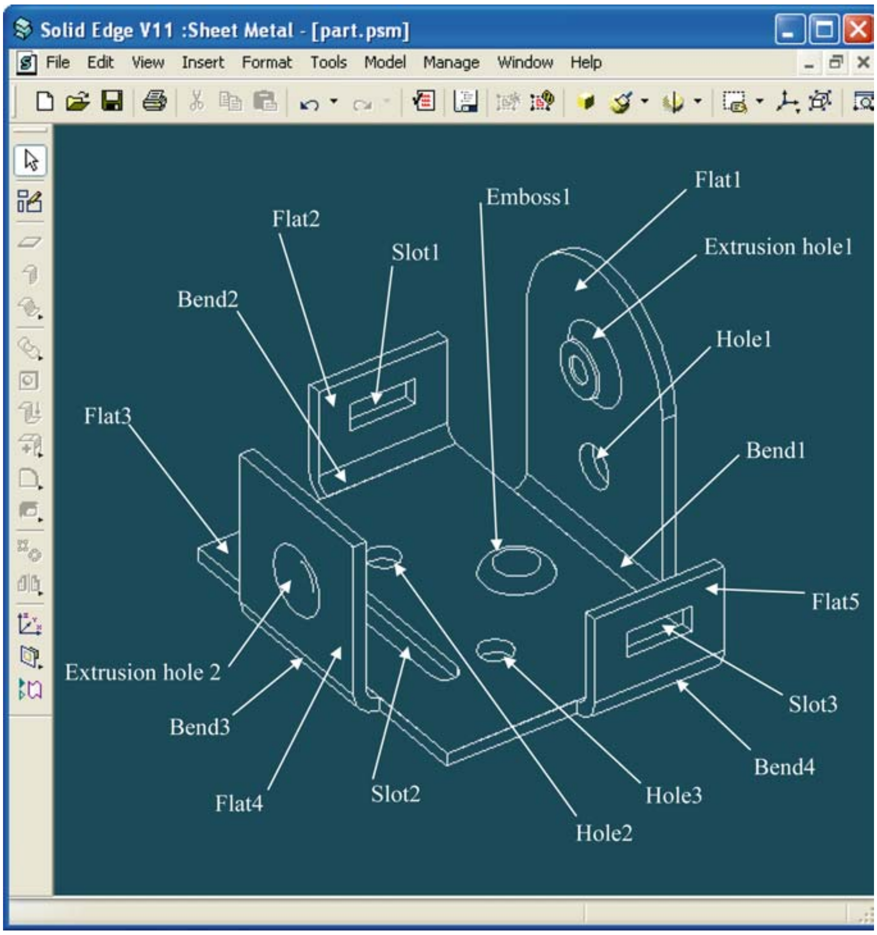


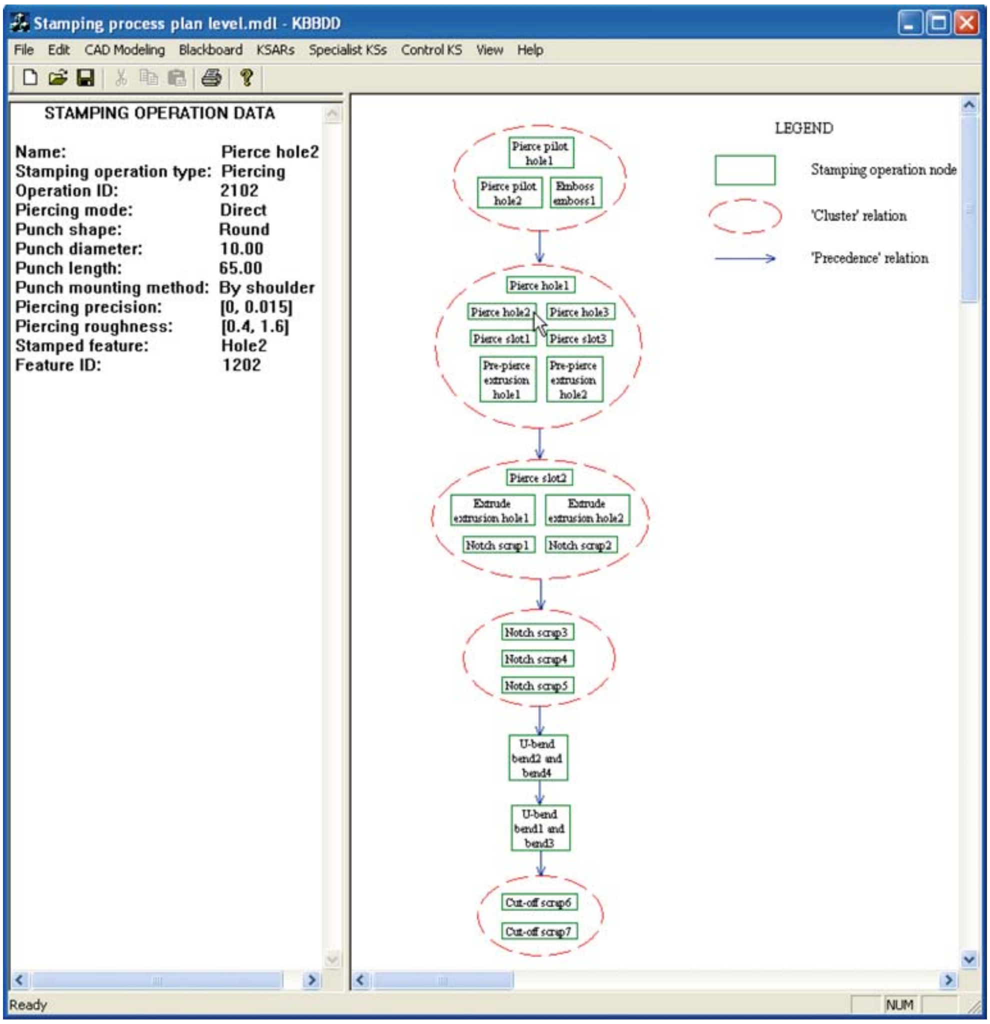
Fig. 5. Three-D feature model of a sample stamped metal part.

see that a control module may be in one of seven major states: *Initializing*, *Monitoring*, *Waiting*, *Activating, Evaluating*, *Cancelled* and *Solved*. While initializing, the control module transits to *Monitoring* state to monitor the blackboard for a (set of) blackboard event(s) in which the corresponding KS is interested. If relevant blackboard events are not available to activate a KS, the control module transits to the *Waiting* state. In the *Activating* state, the control module naturally transits from the sub-state *Evaluating event* to *Activating KS* and eventually to *Creating KSAR*. Then the control module transits to the *Evaluating* state, wherein it is first in *Creating Strategy* sub-state by activating the strategy KS, then transits to *Evaluating KSAR* sub-state, and finally transits to *Executing KSAR* sub-state or *Removing KSAR* sub-state.

The control module unconditionally transits to *Cancelled* if it can not proceed, and to *Solved* if the problem is eventually solved.

5. An illustrative example

A typical stamped metal part modeled in Solid Edge CAD system (Fig. 5) is taken as an example to demonstrate the application of OOBM. The system starts with the retrieval of required geometrical information from the part CAD model, and user input of other technical information (e.g., part weight, surface treatments, blank material, annual production, press type, press tonnage, bolster dimensions, bed open dimensions, shut height, etc.) to produce the first level of



STAMPING OPERATION DATA

Name:	Pierce hole2
Stamping operation type:	Piercing
Operation ID:	2102
Piercing mode:	Direct
Punch shape:	Round
Punch diameter:	10.00
Punch length:	65.00
Punch mounting method:	By shoulder
Piercing precision:	[0, 0.015]
Piercing roughness:	[0.4, 1.6]
Stamped feature:	Hole2
Feature ID:	1202

Fig. 6. Stamping process plan level of the blackboard for the sample stamped metal part.

abstraction on the blackboard (at the stamped part level).

Then the CAD KS (CAD API functions) analyzes the geometrical and technological information of the part and press objects, and extracts stamping feature and feature relation objects to form the second level of abstraction on the blackboard (at the stamping feature level).

Assume that there is no solution after execution of CBR KS because there are insufficient case objects stored in the case base. Then the system opportunistically consults with other planning KSs (for different stamping operations) to transform the stamping features to a set of stamping operations to form the third level of abstraction on the blackboard (at the stamping operation level).

After further consulting with the staging KS, the stamping operations can be sequenced through a graph-based stamping process plan that forms the fourth level of abstraction on the blackboard (at the stamping process plan level) (Fig. 6). In this user interface, the right hand window shows the graph-based stamping process plan, in which different stamping operations are either grouped at the same station or sequentially at different stations. The figure shows 10 *Piercing* operations, 2 *Bending* operations, 1 *Embossing* operation, 2 *Extruding* operation, 5 *Notching* operations, and 2 *Cut-off* operations. The left hand window shows detailed information about a selected stamping operation.

Figure 7 shows the corresponding 2-D strip layout solution generated by the computer, which is stored in the case base for future CBR (Tor *et al.*, 2003). Of course, the user can always override the computer-generated strip layout by modifying the default solution with interactive tools residing in the CAD system.

6. Conclusion

This paper has presented an OOBM for stamping process planning. The key features of domain ontology and reasoning strategy of the blackboard architecture were discussed through a knowledge level description in Section 3. This domain-specific application of a blackboard architecture is new. The application of blackboard-based systems in stamping process planning for sheet metal parts has not been reported in the recent literature. In this application domain, the blackboard approach reduces the obstacles that exist in the conventional architecture of knowledge-based systems, which have difficulty in accommodating different knowledge representation schemes within an integrated framework, and are incapable of managing heterogeneous KSs effectively. It combines the strengths of several AI techniques such as CBR and rule-based reasoning to generate cooperative solutions and overcome the limitations of any single approach.s

Section 4 described the implementation of the blackboard architecture using object technology through a computational level description. The general features of this implementation is not new (Booch, 1994, Chapter 11). However, the authors have adapted Booch's (1994) method to suit the domain-specific requirements. In addition, the separation of knowledge level from computational level of the blackboard framework enables rich descriptions of domain knowledge that are independent of the implementation technology. The domain-specific knowledge content and the technical procedures used in a specific context (i.e. within a given company) will determine the final implemented form of the blackboard.

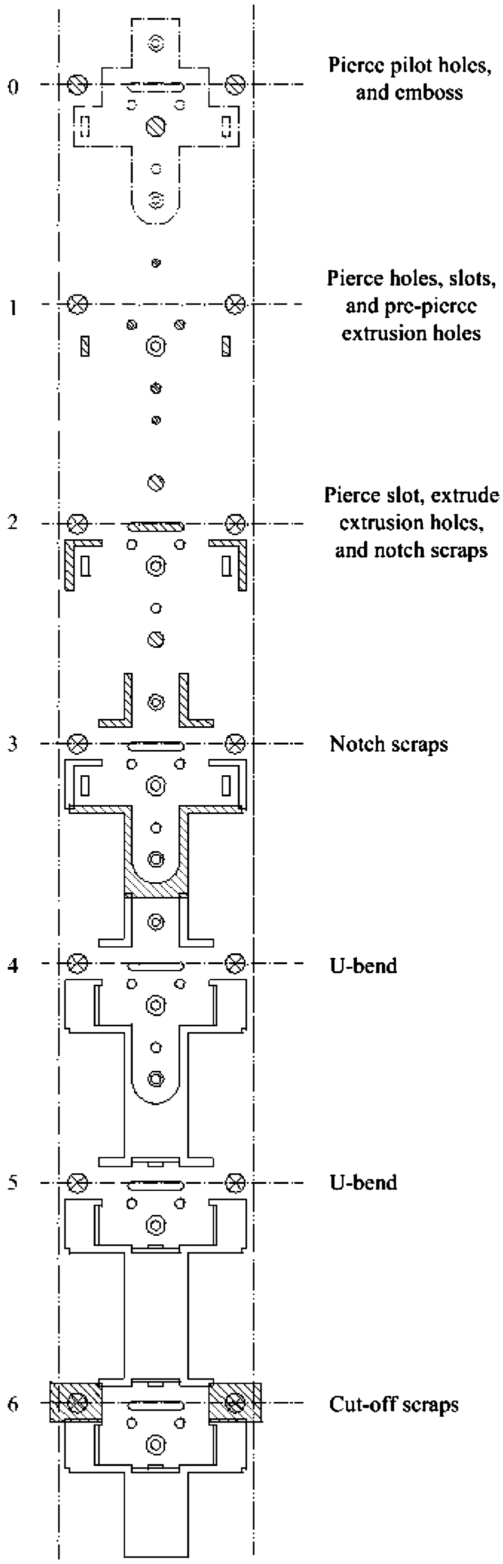


Fig. 7. Two-D strip layout solution for the sample stamped metal part (Tor *et al.*, 2003).

A software prototype has been developed using CLIPS and C++ interfaced with Solid Edge CAD system. A brief example was presented to demonstrate the feasibility and practicality of the proposed approach.

Our future research is aimed at developing a concurrent engineering environment for progressive die design using OOBM, and extending the inferential capability of the system by incorporating graph theoretic algorithms to solve particular aspects of the design, e.g., graph colouring algorithms for clustering.

References

Bergstrom, K., Kivivuori, S., Osenius, S. and Korhonen, A. (1988) Computer aided design of progressive die. In Chersot, J. L. and Ohate, E. (eds), *Modelling of Metal Forming Processes*, Kluwer Academic Publishers, New York, pp. 155–162.

Booch, G. (1994) *Object-Oriented Analysis and Design*, Benjamin/Cummings, Redwood City, Calif.

Britton, G. A., Tor, S. B. and Zhang, W. Y. (2004) Techniques in knowledge-based expert systems for the design of engineering systems. In Leondes, C. T. (ed.), *Intelligent Knowledge-Based Systems*, Vol. 3, Kluwer Academic Publishers, New York, pp. 3–54.

Chen, C. -H., Occena, L. G. and Fok, S. C. (2001) CONDENSE: a concurrent design evaluation system for product design. *International Journal of Production Research*, **39** (3), 413–433.

Chen, Y. M., Miller, R. A. and Vemuri, K. R. (1991) A framework for feature based part modeling. In *Computers in Engineering: Proceedings of the ASME International Computers in Engineering Conference and Exhibition*, Vol. 1, Santa Clara, California, pp. 357–365.

Cheok, B. T. and Nee, A. Y. C. (1998) Configuration of progressive dies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **12**, 405–418.

Cheok, B. T., Foong, K. Y. and Nee, A. Y. C. (1996) An intelligent planning aid for the design of progressive dies. *Proceedings of Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture*, **210**(B1), 25–35.

Choi, J. C., Kim, B. M. and Kim, C. (1999) An automated progressive process planning and die design and working system for blanking or piercing and bending of a sheet metal product. *International Journal of Advanced Manufacturing Technology*, **15**, 485–497.

Duffey, M. R. and Sun, Q. (1991) Knowledge-based design of progressive stamping dies. In Venkatesh, V. C. and McGeough, J. A. (eds), *Computer-Aided Production Engineering*, Elsevier Science Publishers B. V., Amsterdam, pp. 241–247.

Engelmore, R. and Morgan, T. (1988) *Blackboard System*, Addison-Wesley, Reading, Massachusetts.

Fogg, B. and Jaimeson, B. (1975) The influencing factors in optimizing press tool die layouts and a solution using computer aids. *CIRP Annals*, **24**, 429–434.

Giarratano, J. and Riley, G. (1998) *Expert Systems: Principles and Programming*, 3rd edn., PWS, Boston.

Huang, K., Ismail, H. S. and Hon, K. K. B. (1996) Automated design of progressive dies. *Proceedings of Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture*, **210**(B4), 367–376.

Ismail, H. S., Chen, S. T. and Hon, K. K. B. (1996) Feature-based design of progressive press tools. *International Journal of Machine Tools and Manufacture*, **36**(3), 367–378.

Kwong, C. K., Smith, G. F. and Lau, W. S. (1997) A blackboard-based approach to concurrent process design of injection moulding. *Journal of Materials Processing Technology*, **70**, 258–263.

Li, C. Y., Li, J. J., Wen, J. Y. and Xiao, X. Z. (2001) HPRODIE: using feature modeling and feature mapping to speed up progressive die design. *International Journal of Production Research*, **39**(18), 4133–4151.

Lin, Z. C. and Chang, Y. C. (1997) Sash sheet metal development and the application of a back-propagation neural network in the calculation of the developed length. *Journal of Materials Processing Technology*, **69**, 95–105.

Lin, Z. C. and Deng, C. H. (2001) Application of petri net in the planning of a shearing cut and bending progressive die workstation. *International Journal of Materials and Product Technology*, **10**, 579–591.

Lin, Z. C. and Hsu, C. Y. (1996) An investigation of an expert system for shearing cut progressive die design. *International Journal of Advanced Manufacturing Technology*, **11**, 1–11.

Murakami, H., Shirai, K., Yamada, O. and Isoda, K. (1980) A CAD system for progressive dies. In *Proceedings of 21th Machine Tool Design and Research Conference*, Macmillan, London, pp. 587–591.

Nakaham, S., Toshio, K., Tamura, K., Asuke, F., Soda, C. and Nakamura, T. (1978) Computer aided progressive die design. In *Proceedings of 19th Machine Tool Design and Research Conference*, Macmillan, London, pp. 1–12.

Prasad, Y. K. D. V. (1994) A set of heuristic algorithms for optimal nesting of two-dimensional irregularly shaped sheet-metal blanks. *Computers in Industry*, **24**, 55–70.

Roy, U. and Liao, J.-M. (1998) Application of a blackboard framework to a cooperative fixture design system. *Computers in Industry*, **37**, 67–81.

Rumbaugh, J. B., Premerlani, M., Eddy, W. and Lorensen, F. W. (1991) *Object-Oriented Modeling and Design*, Englewood Cliffs, Prentice-Hall, NJ.

Shah, J. J. and Mantyla, M. (1995) *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*, John Wiley, New York.

Srihari, K., Cecil, J. A. and Emerson, C. R. (1994) A blackboard-based process planning system for the surface mount manufacture of PCBs. *International Journal of Advanced Manufacturing Technology*, **9**, 188–194.

Thompson, J. B. and Lu, S. C.-Y. (1989) Representing and using design rationale in concurrent product and process design. *Concurrent Product and Process Design*, ASME Winter Annual Meeting, pp. 109–115.

Tor, S. B., Britton, G. A. and Zhang, W. Y. (2003) Indexing and retrieval in metal stamping die design using case-based reasoning. *Journal of Computing and Information Science in Engineering*, ASME, **3**(4), 353–362.