An object-oriented constraints-based system for concurrent product development

A. Gayretli, H.S. Abdalla

Department of Mechanical and Manufacturing Engineering, Faculty of Computer Sciences and Engineering, De Montfort University, The Gateway, Leicester LE1 9BH, UK

Received for publication 13 January 1999

Abstract

This research work aims to develop an intelligent constraint-based system that enables designers to consider at the early stages of the design process all activities associated with product’s life cycle. One of the most important aspects of these activities is the evaluation and optimisation of manufacturing processes that require various type of information from the different aspects of product’s life cycle. This research article discusses the development of a prototype system for manufacturing process optimisation using a combination of both mathematical methods and constraint-programming techniques. This approach enables designers to evaluate and optimise feasible manufacturing processes in a consistent manner as early as possible during the design process. This helps in avoiding unexpected design iterations that waste a great amount of time and effort, leading to longer lead-time. The development process has passed through the five major stages: Firstly, an intelligent constraint-based design system for concurrent product and process design has been developed. Secondly, a manufacturing process optimisation module has been constructed. Thirdly, the product features, processes, cost, time and constraints to be used for carrying out various design tasks has been represented in the format of constraints, frames, objects, and rules. Fourthly, the process optimisation and evaluation rules for the selection of feasible processes for complex features, and finally, the information management system that ensures consistency in information exchange and decision making activities have been developed.

Keywords: Feature-based design; Object-oriented programming; Concurrent engineering; Process optimisation; Cost estimation; Constraints; Knowledge-based systems

1. Introduction

Concurrent engineering as a philosophy aims to address the consideration of different life cycle issues of a product at early stages of the design process in order to analyse the factors affecting manufacturing processes. Recently, concurrent engineering has placed greater emphasis on the automation and optimisation of manufacturing processes due to its major effect on product cost. Huthwaite [1] stated that typical product cost includes 50% materials, 5% product development, 30% overheads and 15% labour. He also mentioned that almost 70% of total product cost is considered at the early stage of the design process. There are many constraints related to part features, feature-process relations, machine tools, cutting tools, cost and time in concurrent product development.

The other aspects of the product life cycle have an impact on process cost. Therefore, the constraints of the product’s life cycle issues also have to be involved in process evaluation and optimisation process to reach a cost-effective design in the early stages of the design phase. Representing these constraints in an efficient format is very important to evaluate and optimise the design effectively and to prevent users from being engaged in time-consuming iterations process. To achieve an effective management of those constraints, efficient and timely communication network system should be provided within different design and manufacturing areas. This requires the critical consideration of various tasks such as overall coordination, control, consistency, and data integrity to prevent costly design iterations. This can be achieved through the integration of different design areas through establishing LANs within the organisation in a consistent manner. Such integration should include a strategy for conflict resolution to avoid disagreements within the different activities or areas. Research work in
this area was investigated by several authors including Noble [2], Balasubramanyam and Norrie [3], Lander and Corkill [4] and O’Grady et al. [5].

There is no doubt that concurrent consideration of product and process design improves product quality and reduces the re-design work, which leads to shorter product lead-time. Research work has recently been devoted towards developing methods and tools for the estimation and optimisation of manufacturing costs [6–10]. Feature-based models that focus on machining form features such as hole, slot, flat surfaces, chamfer, cylinder and rectangular block have been investigated by Fen et al. [11] and Ou-Yang and Lin [12]. Activities such as drilling, milling, handling, set-ups, fixtures and tool changes for the component can be evaluated and optimised as the design progresses [3–17]. Taiber [18] has classified manufacturing cost optimisation of prismatic components as follows:

1. Tool cost.
3. Net machining time.
5. Number of tool changes.

This approach provides an effective cost calculation and optimisation of mechanical parts using milling and drilling processes. Another important factor to reduce product costs is the detailed analysis of manufacturability. Lee et al. [19], Abdalla and Knight [20], Hashemian and Gu [21] developed a number of methods, which enable designers, and manufacturing planners to address these detailed analysis. Existing approaches to process evaluation and optimisation generally have limitations on the feasibility evaluation and optimisation of particular geometric specifications, and available process combinations and capabilities. The possible use of multiple alternative machines and integration of manufacturability analysis with process evaluation and optimisation have not yet been fully addressed. Since a tremendous amount of information exchange and decision making activities are involved in the design process there is a need to develop an information management system capable of dealing with any inconsistency arising from different aspects of the product’s life cycle. The aim of this research is therefore to achieve such optimisation, integration and consistency.

2. Proposed approach to concurrent product and process design

The proposed model embodies a CAD solid modelling system, user interface, design representation, consistency manager, constraint-based system, process optimisation and manufacturability analysis, and various knowledge sources. The architecture of the simultaneous product and process design paradigm is shown in Fig. 1.

(1) Constraint-based system module: The constraint-based system is used to model and handle design requirements. The model uses constraints to model information about various life-cycle issues for effective use during the design process. It includes constraints and variables of different design areas and a constraint propagation module for design consistency. It also encompasses the design variables in the form of constraints. When a user assigns a value to a variable the constraints propagation checks the assigned value, a valid solution can only be reached if all constraints are satisfied. In the case of a constraint(s) violation a warning is given to the user, followed by some alternative suggestions.

Constraints collected from different knowledge sources such as experts can be formulated as rules, variables, values and domain. The proposed model covers most aspects of design and manufacturing constraints. These

![Fig. 1. The architecture of the concurrent product and process design paradigm.](image-url)
are process constraints, machine constraints, and material constraints, tooling constraints, part constraints, material handling constraints, and tolerance and surface finish constraints. The constraint-based system is also linked to a consistency manager and design representation module. The consistency manager ensures that no-violations exist in the system, and controls the constraints of design and manufacturing represented as rules, frames and variables in the knowledge-based system.

(2) **Consistency manager module:** The consistency manager is responsible for the management of the decision-making process and dealing with conflict situations, and justification of decisions made on design. It also detects conflicts, and gives warnings and explanations to the users, and finally applying a suitable strategy for solving conflicts in order to make sure that design consistency is in the constraint network and design output. The consistency manager also allows designers to take necessary actions for the problems and to monitor design violations via the user interface.

(3) **Design representation module:** The knowledge-based system toolkit KEE (Knowledge Engineering Environment developed by Intellicorp.) has been implemented to represent the design features. It is very important to build the design modelling in a systematic and well-organised way in order to allow an effective interaction between design and manufacturing constraints. Therefore, the organisation of knowledge has been considered as a hierarchical family structure consisting of objects, units, rules and/or methods with attributes inherited to all sub-classes. This structure allows designers to add new units and attributes.

(4) **Process optimisation and manufacturability analysis module:** The main functionality of this module is the analysis of the part and its features, and selection of processes, machines and tools. It then evaluates the selected processes, calculates time/cost values in order to determine whether the part is feasible. The processes and sequences are analysed and the total machining cost of the product including material, tooling, machining, overhead, and labour cost is calculated. If the product cost and machining time exceed the targeted one, then the user has to either seek advice from designer to make modifications. This process continues until a cost-effective product is designed. Further details of how the Process Selection and Optimisation module functioning is discussed in the subsequent sections.

(5) **User interface module:** An interactive user-friendly interface has been developed to allow users to customise the system easily and efficiently. The KEE features were implemented to create the user interface. Menus such as multiple-choice menu, pop-up menu are extensively used to get user-defined values in this research. Active images are also incorporated to the system to monitor constraint violations and value changes. The method actuators are included to activate methods in slots. The user interface enables users to also interact with the CAD solid modelling system (Pro/Engineer) to create 3-D solid models, add features, and alter feature attributes. The user communicates with the system by a mouse, a keyboard, a super panel including menus, active images, and method actuator, and lisp listener, KEE output window, and typescript window.

### 2.1. The proposed model for process selection and optimisation

The proposed model composed of a form feature database; designer requirements, manufacturing processes and constraints; an evaluation and optimisation module, and a user interface as shown in Fig. 2. The designer communicates with each module via the user interface and his/her requirements (i.e. process time and cost; tool cost; set-up time and cost) can be entered to the system as a set of constraints. For example, he/she can specify to the system that process cost, time and tool cost are not exceeding predefined values. The form feature database includes various types of form features, of which a part is composed. The system retrieves manufacturing form features and parameters from the feature database to choose the feasible processes. The processes and constraints module contains manufacturing information such as feature type; material; length and diameter ratio; cutting tool specifications; process availability; machines; accessibility; process sequences; tolerance; surface finishes; optimum cutting parameters, cost and time. In addition, it includes representation of processes and constraints to evaluate and optimise the manufacturing processes of a part. Based on the manufacturing constraints, the system analyses the features of the part, then selects the feasible processes, and calculates process’s time and cost. For example, if a form feature of a part with tight tolerance, special surface finish, and complex shape can be first machined by a machining centre with a special cutting tool. Then a grinding or reaming process is used to meet the required tolerance and surface finish. Finally,
the system using an algorithm that evaluates selected processes subject to some criteria provided from the design specification, and calculates total processes time and cost of each form feature. If all process combinations are found unacceptable, the system will initiate a dialogue with the user concerning possible modifications on the design. This process will be repeated until a set of process combinations is reached.

2.1.1. Feature representation

A feature is defined as a generic shape carrying product information, which may aid design or communication between manufacturing and design, or between other engineering tasks such as assembly, manufacturing and maintenance. The features should be represented explicitly in a form that matches manufacturing knowledge. To generate a detailed process plan it is necessary to analyse the form features directly related to manufacturing under consideration.

In this analysis manufacturing form features are the key to generate and optimise process plans, since manufacturing features provide designer and process planners with a natural way of communication, and simplify process planning by avoiding unnecessary considerations of unlimited ways to manufacture a feature. Therefore, feature-based representation approach has been utilised to represent design in greater detail than it can be used by a designer, process and assembly planners or an expert system, which accomplishes these activities. The research is on defining manufacturing form features derivable from topological and geometrical description of the part in order to achieve a cost-effective process planning. For example, a slot can be defined by its parameters such as name, diameter, depth, locations, tolerance, process and surface finish. These parameters are used to select the manufacturing processes, set-ups, and tooling. The most common form features and 3-D solid model of an engine head are represented in Figs. 3 and 4.

The proposed model contains knowledge about form features and their manufacturing processes. It also, contains rules that manipulate the behaviour of the feature and process data in structured way and effective format to obtain a feasible solution. Information about manufacturing environment capabilities (i.e. maximum length, diameter, tolerance, surface finish, and tools) is included in the rules. A hole object-oriented is represented in the model as shown below.

(hole
   with
   (name: through-hole)
   (diameter: 10 mm)
   (depth: 30 mm)
   (depth/diameter ratio: 3)
   (dimensional tolerance: ± 0.0001)
   (x-distance: 100 mm)
   (y-distance: 34 mm)
   (first process: drilling)
   (second process: grinding)
   (…) )

2.1.2. Representation of design and manufacturing knowledge

Various knowledge representation techniques used in this research are described below.

2.1.2.1. Constraints. Information from the different domains related to the product life cycle, including design and manufacturing requirements were formulated in the system as sets of constraints. For instance, from the process planner’s point of view, the topological attributes of a specific form feature were represented as constraints, similarly the product targeted cost is another constraint. Constraints are in general a very effective way of holding variables associated with design and manufacture in a slot or a rule class. In this research, design and manufacturing requirements are formulated as a set of constraints. Design and manufacturing variables are stored in the slot of a unit. Their values are kept within specific constraints defined by the user. Each object in the system has its own variables and working constraints. For example, in the production rules shown below manufacturing variables of a form feature is defined as diameter, depth, lower tolerance, upper tolerance and surface finish. The lower tolerance variable has its own constraint;

> = ?lower tolerance 0.025 mm which means that the
lower.tolerance must be equal or bigger than the limit 0.025 mm). An example of how manufacturing requirements are formulated as constraints in a rule class is shown below. The flowchart of process selection for a blind hole is shown in Fig. 5.

(Blind_hole_rule_1)
(if (? what is in blind.holes)
  (the diameters of ? what is ? diameters)
  (the depth of ? what is ? depth)
  (the lower.tolerance of ? what is ? lower.tolerance)

  (the upper.tolerance of ? what is ? upper.tolerance)
  (the surface_finish of ? what is ? surface_finish)

  (lisp (≥ ? lower.tolerance 0.025)
     (lisp (≥ ? upper.tolerance 0.15)
       (lisp (≥ surface_finish 1.6)
        (....)
        then
        (lisp (format t "****The possible process for ~d is drilling, end-milling and edm..****" ? what))
        (the volume of ? what is (lisp (*(*(/14)/(π))

Fig. 5. The flowchart of process selection for a blind hole.
(* ? diameters ? diameters) ? depth)))
(the.first.process.selection of ? what is ok)
(the.possible.process._1 of ? what is drilling/counter-boring)
(the.possible.process._2 of ? what is milling)
(the.possible.process._3 of ? what is edm)
(...))))

2.1.2. Frames. A frame is described as a structure for storing interconnected information about a design and an object. It is a very effective means of knowledge representation of stereotypical objects. A frame consists of a name and a number of slots. The value of slots can be numeric (12, 24), logical (yes or no), procedural (methods) and symbolic (steel). The frames also allow designers to attach images and active values to any slots in order to monitor value changes. By using facets, values of slots can be controlled. An example of how product design specification is represented as frames is shown below.

Super class: Product Specification
Sub-classes: Part, Manufacturing Cell Capacity, Available Machine Tools
Properties: (part type

(value ((lambda (self)
(with-keio (setq choice (prompt-use 'choice-multiple choices '(rotational non-rotational)
:prompt "Please select one: "
:few-choices-mode 'menu))))))
(inheritance method)
(valueclass (# [Unit: method keedatatypes]))
(default nil)
(activeimage (# [Unit:windowpane-availability-of-machine.constraints. manufacture007])
. unique.values) (cardinality:min (1))
(cardinality:max (1))
(length —))

2.1.2.3. Production rules. Production rules represent knowledge as rule of the forms: IF premises THEN conclusion. In the proposed model, rules are used as values of unit attributes (slots). Therefore, they can be manipulated and inherited from higher classes to sub-classes. The involvement of rules in slots enables a complex rules system to be linked to different sets of rules associated with different units. They are also combined with methods, which are LISP functions stored as a value of a slot. In prototype-testing-rules, shown below total manufacturing cost calculation is carried out by a LISP function. The UNITMSG sends a message to the PART unit. The message is sent to the TOTAL-COST method, which performs the action for calculation of the total process cost. The calculated value becomes the value of the TOTAL—MANUFACTURING—COST slot. The production rules are used to establish design and manufacturing constraints as rules. A sample of how the production rules can be used for prototype testing is shown below.

(prototype-testing-rule-1
(if (the.total_m_cost—control of part is ?
total_m_cost—control)
(? total_m_cost—control = ok)
then
(the.total_manufacturing_cost of part is
(lisp (unitmsg 'part' total_cost))))
(prototype-testing-rule-2
(if (the.total_manufacturing_cost of part is ?
total_manufacturing_cost)
(the.target_manufacturing_cost of part is ?
target_manufacturing_cost)
(lisp (? total_manufacturing_cost ? target_manufacturing_cost))
then
(lisp (format t " ~ % The estimated manufacturing cost of
cost of part is ~ D $ and the
manuf. cost is ~ D $.. ~%")
?
?)
(prototype-testing-rule-3
(if (more rules))))

2.1.2.4. Object-oriented programming. This approach provides a very effective and efficient way for organising design and manufacturing objects such as machine tools, cutting tool features, material features, and machining elements into various classes. The classes are represented in hierarchies. A class is made up of a class name and several attributes associated with a number of values. Sub-division of classes is made until all components of the class are determined. An instance or a member of a class can be added to a sub-class to represent a specific and available process in the manufacturing cell. The members have their special characteristics such as power, weight, capacity, and tolerance. A specific value defined in a higher class is inherited to the lowest class members. When new knowledge is added the objects in the knowledge base increases.

2.1.3. Process selection and optimisation

The main goal in process planning is to produce products in accordance with the specification to achieve the highest possible quality. However, economic considerations are also very important. Producing an optimal process plan is very desirable and then improving it with respect to given criteria. Process optimisation can be carried out at several detail levels. At the highest level, it is necessary to choose processes, machine tools, etc., and also sequencing of operations. At a more detailed level, there is the optimisation of cutting parameters such as feed-rate, cutting speed and depth of cut. In addition,
estimation of process time and cost of processes, tools and set-ups are included in this level.

2.1.3.1. The process selection. Process selection necessitates the consideration of a number of criteria for the appropriate process selection for each form feature. These include:
1. Feature type: through hole, blind hole, and slot.
3. Length and diameter ratio: 8/1 for twist drilling, 5/1 for boring, and 20/1 for EDM.
4. Availability: YES/NO from user.
5. Tolerance: broaching $0.025$ and boring $0.04$.
6. Surface finishes: drilling $1.6-6.3$ and reaming $0.8-3.2$.

How these criteria are included in rules for process selection is shown in Section 2.1.2.1 for process selection of a blind hole. Further details of the process selection procedure are available from the authors on request.

2.1.3.2. The process evaluation. The material of the part is important for selecting the machining parameters such as tool material, cutting speed, feed rate, and tool diameter. The Machining Data Handbook [22] can be used for obtaining these machining parameters. The AM Cost Estimator/Cost Estimation Database [23] has been used for cost estimation of direct labour cost, time, material cost, operations, parts, and manufacturing technologies. In addition, it is utilised to determine parameters of machining tools such as set-up times, time to stop and start machines, cutting tools, and fixtures. The proposed model for selection and evaluation of manufacturing processes is shown in (Fig. 6).

Cost optimisation is defined as minimising the total process time and cost which are subject to process variables such as cutting speed, feed-rate, cutting force, power, and surface finish constraints [24]. The model uses a table including manufacturing form features of a component, selected possible processes for each feature, and criteria for optimisation of the processes for the part. This enables users to define constraints for processes of each feature based on their input about process time, cost, and tool cost. Table 1 shows a detail description of the part features, possible processes and constraints. Process time and cost are calculated by the system. A number of processes for each feature are represented in the table. For example, there are three possible processes for feature_1. They are process_1, process_3 and process_n. The aims of creating the table are to generate possible processes for each feature and evaluate combination of all processes for the component by assigning only one process to each feature in turn.

Fig. 6. The optimisation approach.
Table 1  
Process-feature table for optimisation of processes for the part

<table>
<thead>
<tr>
<th>Processes</th>
<th>Features</th>
<th>Feature_1</th>
<th>Feature_2</th>
<th>Feature_3</th>
<th>Feature_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process_1</td>
<td></td>
<td>T_{11}</td>
<td>...</td>
<td>T_{1n}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C_{11}</td>
<td>...</td>
<td>C_{1n}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TC_{11}</td>
<td>...</td>
<td>TC_{1n}</td>
<td></td>
</tr>
<tr>
<td>Process_2</td>
<td></td>
<td>...</td>
<td>...</td>
<td>T_{22}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>T_{23}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>TC_{23}</td>
<td></td>
</tr>
<tr>
<td>Process_3</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Process_4</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Process_m</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

This methodology will enable users to evaluate hundreds of process combinations for a part. However, this process can be very time-consuming when there are hundreds of processes for a feature. Therefore, our approach is to find feasible processes in a reasonable time. To do this, the processes, which are not feasible, have to be excluded. The rule-based algorithm has been developed to solve this problem as presented in the Section 2.1.5. In addition, maximum allowable cost and time can be defined for candidate processes of a feature.

This gives flexibility to control total process cost and time of the part or a single form feature. Then the set of process combinations or possible processes for a single feature are evaluated by comparing process cost and time, and the other variables against user requirements. Any combination, which satisfies user requirements, is considered to be feasible. One of the feasible process combinations is selected as a solution or further evaluation can be continued. The user carries out modification of constraints if there was no process combination to satisfy their requirements. Only one process combination can be reached by changing constraints if needed. In addition, explanations are given by the system via user interface.

2.1.3.3. **Process time and cost estimation.** Process time is calculated by using standard formulas. Since our approach is based on feature-based cost estimation and optimisation of manufacturing processes, it is considered to use the following formulas to estimate process time:

\[
\text{Process time} = \frac{\text{Form feature volume}}{\text{Material removal rate}}. \tag{1}
\]

Form feature volumes are calculated using standard formulas. Material removal rate differs from one process to another depending on tool diameter and type, type of material, and cutting parameters (Table 2). Some of the manufacturing processes and material removal rate are shown in Table 2.

In order to reduce process time, material removal rate (MRR) should be maximised subject to

\[
T_i = T_{i\text{max}} \quad \text{(tool life)},
\]

\[
D_i = D_{i\text{max}} \quad \text{(tool diameter)},
\]

\[
f = f_{\text{max}} \quad \text{(feed rate)},
\]

\[
V = V_{\text{max}} \quad \text{(speed)},
\]

\[
W = W_{\text{max}} \quad \text{(depth of cut)}.
\]

This needs cutting parameters such as cutting tool diameter, feed-rate, spindle speed and depth of cut to have maximum values. In addition, selection of cutting tools needs to be taken into account. The consideration of selected material attributes such as hardness, machinability and electrical conductivity has a major effect on cutting parameters.

Having chosen suitable cutting tools that meets the material requirements MRR for any processes can be estimated by using the related formulas. Then the process’s time for manufacturing form features of the part is calculated. Process cost estimation will be carried out using the estimated process time. Productive Hour Cost (PHC) of various processes have been provided from the cost estimation database to estimate process costs.
Using a PHC value, total process cost is calculated as follows:

$$\text{Total process cost} = \text{Lot} - \text{time} \times \text{PHC}. \quad (2)$$

Lot-time is based on the quantity of part or a form feature. Some of the PHC value of manufacturing processes including set-up costs are listed in Table 3.

From the above figures total cost can be formulated as follows:

$$\text{Total cost} = \text{Material Cost} + \sum [(\text{Lot} - \text{time} \times \text{PHC})$$

$$+ \text{Tool Cost} + \text{Setup Cost}]. \quad (3)$$

Set-up times for various machine tools included in the machining handbook are used to estimate set-up costs to reach a more accurate cost estimation.

### 2.1.4. Overall design consistency in the system

While the system carries out various tasks such as material selection, manufacturability analysis, process selection and optimisation, huge amount of information is accessed and shared in the knowledge base. New information is also added to the knowledge base, which will be used to execute such tasks. This information should be represented in a format that different design analysis can be done by using the same information. It can be seen from Fig. 7, that several agents which represent the life cycle aspects of the product have their own tasks, and a common knowledge base.

Each agent needs to interact one with another to accomplish its own task. Any value entered to the knowledge base is propagated by the constraint-based system to check if the value violates any constraints.

Fig. 8 shows details of how agents share information while ensuring consistent information flow in the system. An agent retrieves design input from the knowledge base to carry out its task. If this input is not within constraints of the agent, the consistency agent detects violation, and sends a message to a method slot, which carries a function of LISP responsible for conflict resolution. The method generates alternatives from which a design agent has to choose.

The new input, which satisfies the agent’s constraints, is stored in the knowledge base to be used by other
agents if required. Conflicts are immediately reported to the user through the user interface. As an example, a manufacturing capacity agent that possesses constraints of the production capacity checks production quantity defined by the user. If the capacity is not within the limit the system informs user and proposes alternative suggestions for sorting this problem. The production quantity can be used in the system as a consistent value if it satisfies the requirements of the manufacturing capacity agent. This system provides consistent information flow between different tasks of product’s life cycle by avoiding conflicts that can cause costly delays in the design process.

2.1.5. The process optimisation scenario

Manufacturing processes should be optimised after a detailed manufacturability analysis of manufacturing cell capability, and feature types, dimensions and tolerances subject to production quantity, available processes and their constraints. The rule-based algorithm for optimisation of manufacturing processes has been presented below. It consists of two main steps; process selection and the optimisation. Steps 1–4 is the selection of feasible processes for each form feature based on material, lot-size, tolerance, surface finish, and feature type. The next step contains suitable cutting tool selection, machine tools selection, selection of optimum cutting parameters, calculations of process variables such as material removal rate (MRR), lot time, set-up cost, and tool cost. Also, the comparison of selected processes based on the process variables is carried out. Fig. 9 shows the various steps and tasks involved in the process optimisation in detail.

1. Select a material from the database
2. Get lot-size of part or features
3. Get a form feature from the part
4. Select feasible processes for the feature satisfying requirements of the part (tolerance, surface finish, feature type, etc.)

5. Take one of the possible processes
6. Select the biggest possible diameter, shortest length, and available and cheapest cutting tools for the selected process
7. Select available machine tools and fixtures
8. Select optimum cutting parameters; depth of cut, cutting speed, feed-rate and cooling conditions
9. Calculate feature volume, MRR, lot-time, tool cost for the lot-size, set-up cost for the process
10. Calculate total cost and time of the process
11. If there are possible processes left to be analysed
12. Go to 5 else go to 13
13. Compare the possible processes for each form feature with each other and eliminate the processes which has higher tool, process and set-up cost and time value than others
14. If there is only one process for the form feature consider it to be the most feasible process else ask user to select one process from the list
15. If there are form features left to be analysed
16. Go to 3 else 17

Fig. 10. Sample of the system results.
17. Calculate final process cost of the part
18. Inform the user
19. End

The system has been evaluated through the design of a cylinder head shown in Fig. 4. The results showed that drilling and end milling are cost-effective processes for the two features, as shown in Fig. 10.

3. Conclusions

A prototype system for the evaluation and optimisation of manufacturing processes has been demonstrated in this research article. The proposed system is composed of form features database; designer requirements, manufacturing processes and constraints; an evaluation and optimisation module, and a user interface. This system assists designers in designing products concurrently, selecting manufacturing processes, and evaluating and optimising those processes in terms of various product life-cycle requirements and user constraints by avoiding inconsistencies. It also provides real-time cost estimation and feasible process plans, and conflict resolution using constraint-based system. The cost model and rule-based algorithm for estimation and optimisation of manufacturing processes have been discussed. This work is part of an on-going research programmes aims to develop a comprehensive system that covers all activities necessary for performing concurrent product and process development. As initial results obtained from the system are promising, using this system could lead to a significant reduction in the product cost and lead-time.

The major contributions of this research are summarised as follows:

1. A new methodology for integration of manufacturability analysis with process optimisation to provide designer with a complete result of these two tasks in the early design stage.
2. Development of knowledge base containing various product's life-cycle requirements.
3. Development of a rule-based algorithm for evaluation of form features and selected processes subject to the given constraints such as production volume, lot-time, tool cost, and process cost and time.
4. Effective information management avoiding conflict situation.
5. This methodology is a useful tool that can be extended to cover other activities of concurrent product and process development.

References