A feasible approach to the integration of CAD and CAPP

Xionghui Zhou *, Yanjie Qiu, Guangru Hua, Huifeng Wang, Xueyu Ruan

National Die and Mold CAD Engineering Research Center, Shanghai Jiao Tong University, 1954 Hua Shan Road, Shanghai 200030, China

Received 17 November 2005; accepted 16 January 2007

Abstract

Although current CAD systems are declared to be feature-based, in fact, the so-called feature is just a modeling macro or menu name such as Protrusion, Revolution, Cutout, Block, etc., instead of a design feature or manufacturing feature in accordance with engineering practice. Consequently, product model data insufficiency and incompatibility between varieties of application systems are still the major barriers to system integration, especially the integration of design and process planning. This paper proposes a practical solution for a bi-directional integration of CAD and CAPP on the platform of commercial CAD systems. The key techniques such as feature recognition and conversion, feature parameter and constraint extraction, feature tree reconstruction, technical information processing, process planning, automatic process drawing marking and 3D material stock CAD model generating are discussed. And the extracted features and their related technical information and knowledge are encapsulated together with the geometry-oriented CAD model to form an integrated product information model to facilitate effective integration with the downstream activities. The integrated CAD/CAPP system is implemented on a commercial CAD package, UGS/SolidEdge. A case study and industry implementation illustrate the feasibility of the approach proposed.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Feature recognition and conversion; Feature tree reconstruction; Technical information processing; Process planning and optimization; Material stock generation; System integration

1. Introduction

Integrating CAD and CAM not only plays a key role in achieving digital manufacturing and computer integrated manufacturing, but also is vital to the competitiveness of mechanical manufacturing enterprises and their ability to respond rapidly to market changes, and one of the most critical links for the integration is the link between design and process planning activities [1].

The significance of CAD/CAPP integration arises from the fact that CAPP relies on the product model data provided by CAD to perform precise and consistent process planning for manufacturing. However, they tend to have different product data descriptions, i.e. CAD is usually geometry-based, whilst CAPP/CAM are feature-based and domain-dependent, which results in unsatisfactory practical implementation, or a common weakness of CAPP systems — they usually act as stand-alone functions and do not have a link with either CAD or CAM systems.

This problem could be solved by developing a feature-based CAD system to provide data directly to CAPP systems, but it inevitably imposes limitations on product design and/or modeling, and other downstream applications [2]. Consequently, although many commercial CAD systems are declared to be feature-based, in fact, the so-called feature is just a modeling macro or menu name, such as Protrusion, Revolution, Cutout, Block, etc., instead of a design feature or manufacturing feature in accordance with engineering practice. Even if features, parameters, or constraints are used in design, and CAD/CAM modules are from the same system or vendor and internally linked, consistent information of features is not communicated from CAD to CAM because vendors have treated them only as geometric modeling ‘macros’ [5,6].

Besides the feature information, which serves as the bridge to a high level integration between design, analysis, process planning and manufacturing, being difficult to recognize and extract from CAD models, another problem is that it is also difficult to embed non-geometric technological information, such as dimensional and geometric tolerance, surface roughness and hardness, that is necessary for CAPP, in current CAD models. At a glance, CAD models seem to incorporate these...
data as seen in the drawings; in fact, most, if not all, of these data are stored independently in a data file, and do not connect directly to the related features, or exist as real attributes of CAD models [4,5]. In other words, they are not represented in an integrated form to facilitate effective integration with the down-line activities of the product cycle [3], but are simply represented as technical notes for human interpretation [4].

Therefore, despite a lot of effort made in the past few decades to interlink design and process planning, sharing of design and manufacturing information still remains a bottleneck [1–10]. It has been reported that imperfect interoperability imposes at least $1 billion per year on the members of the U.S. automotive supply chain, and the majority of these costs are attributable to repairing or re-entering data files involved in the design and manufacture of automobiles, which are not usable for downstream applications [11].

Considering the fact that the developments in the CAPP area have not kept pace with those in CAD/CAM [12–14], and a general-purpose CAPP system is far from reality, to meet the increasing requirement for such a CAPP that is customized and bi-directionally integrated with current “feature-based” CAD systems that are widely used in industry, this paper proposes a practical solution for integrating design and process planning on the basis of commercial CAD systems. The proposed methodology, system architecture and function implementation are addressed in the following sections.

2. Related research work and review

As the bridge between CAD and CAM, Computer Aided Process Planning (CAPP) is used to interpret product/part design data in terms of features, analyze the shape, size, tolerance, location, orientation and relationship of various geometric features on a part, and translate them into manufacturing operation instructions in optimal process sequences, so as to convert the stock (raw material) into a finished part economically and competitively. Generally speaking, feature recognition and technical information processing, process planning, and system integration are hot topics involved.

2.1. Feature recognition and technical information processing

Feature recognition and/or conversion has been an active research topic and a critical component for the integration of CAD and CAM since the 1980s [1,2,4,6,15–26]. Syntactic pattern recognition, rule-based search, graph-based matching, volumetric decomposition, hint-based geometric reasoning, neural networks and genetic algorithm-based methods, etc., are feature recognition techniques available. Han et al. [21] reviewed the state of the art in manufacturing feature recognition at length with the focus on the three most active approaches: the graph-based approach [16–18], the hint-based approach [19–21], and the volumetric decomposition approach [22–24]. Recently, Lim et al. [25] and Sundararajan and Wright [26] emphasized volumetric feature recognition for solids with freeform surfaces. Li and Liu [27] discussed detailed feature recognition and decomposition by extracting form features and then selectively suppressing the uninteresting features to generate geometric models for CAE analysis. Joshi and Dutta [28] proposed a rule-based method for recognizing and simplifying features in freeform surface models for automatic finite element mesh generation.

After feature recognition, the related technical information, such as dimensional and geometric tolerance, surface roughness and hardness, which indicates design intent and machining requirements, should be principally examined from the viewpoints of functionality and cost, and carefully treated in downstream activities, especially process planning. Unfortunately, most researchers have concentrated on geometric information extraction and conversion without tackling the importance of non-geometric feature information [5], and some commercial CAD systems also disregard this issue [4], or treat it in a different module and do not represent it in an integrated product information model, which results in human intervention at the first stage of CAPP being inevitable.

Prabhu et al. [3] reported a comprehensive method of heuristic search and natural language processing for automatic extraction of geometric and non-geometric information from ‘engineering drawing’. Shah et al. [29] presented a dimension graph model, where the whole dimension graph of a part can be separated into several sub-dimension graphs that control certain degrees of freedom and dimensions in certain directions of geometric entities. The method focused on how to model effectively the geometric dimension and tolerance (GD&T) of a part to capture the designer’s GD&T scheme on a feature-based design model, validate its completeness, and then transfer the GD&T to machining features extracted automatically by feature recognition methods. However it is a complex and difficult task to construct robustly such a GD &T model suitable for both CAD and CAPP. Considering the difference in representation between design and machining features, Gao et al. [5] proposed an algorithm flowchart for extracting and converting GD&T from design features, created by Pro/Engineer, to machining features, and two situations are discussed. In the situation of feature interactions, the algorithm was implemented to detect whether the design feature dimensions in the CAD database are consistent with the feature model, and then convert those virtual dimensions into real dimensions that correctly describe the machining features. For geometric tolerance conversion, the main task is to locate each tolerance of the component to the corresponding feature, and convert the virtual datum elements in the design model into recognizable elements for the machining application.

Nevertheless, despite two decades’ research, significant challenges to further research remain in all aspects of feature technology [2]. Currently, recognition of intersecting features, handling multiple interpretations of features that correspond to different ways to machine the part and therefore provide downstream applications with added flexibility, controlling computational complexity, feature constraint extraction, feature precedence tree reconstruction, and their practical application in industry are still open issues.
2.2. Process planning

Process planning activities, including process selection, operation sequencing, and resource selection, are knowledge intensive. The early research was mainly focused on variant process planning systems, which rely on standard plans summarized from previously manufactured parts and use group technology to retrieve process plans for similar parts. The second generation of CAPP systems employed semi-generative and generative CAPP approaches, which emulate the thinking of a human process planner to make processing decision semi-automatically or fully automatically by means of decision logic, mathematical formulas or algorithms, and expert systems according to the part and process information.

Currently, with the aim of producing products economically and competitively, the knowledge-based CAPP system has received much attention from researchers. Park [30] considered that a knowledge base should be not merely a set of rules, but a framework of process planning that can be controlled and customized using rules, and proposed a knowledge capturing methodology, in which four knowledge elements, facts, constraints, the way of thinking and rules for process planning, were derived from the model of process planning that was represented by a traditional three-phase modeling framework consisting of object model, functional model and dynamic model (decision logic and decision variables). Ramana [31] summarized the previous work and key issues related to data and knowledge modeling for product design and process planning.

Once knowledge bases are constructed, the corresponding artificial intelligence-based decision-making techniques, such as expert systems, rule-based reasoning, case-based reasoning, fuzzy logic, neural network, genetic algorithms, etc., could be used to explore the large solution space for a valid and optimal process plan under various constraints. Li et al. [32] developed a hybrid genetic algorithm and simulated an annealing approach to consider concurrently the processes of selecting machining resources, determining set-up plans, and sequencing operations for a prismatic part. Wong et al. [33] describes a fuzzy expert system and genetic algorithms for solving the process selection and sequencing problem under uncertainty. Many other papers also addressed the implementation of the automated operation planning and optimum operation sequencing, and tool selection algorithms [34,35].

Achieving automated process planning to reduce dependence on human judgment is the ultimate objective of CAPP. Though numerous studies have been reported, they are limited to academic discussion and prototype demonstration in principle, and it is not easy to find a commercial CAPP system applicable to complicated objects [9].

2.3. System integration

Product model data insufficiency and incompatibility between varieties of CAX software systems are the major barriers to system integration. To create the context where the information requirement and data flow for CAD/CAM system integration are defined, Feng et al. reviewed available process planning activity models developed in industry, academia and standard committees, including CAM-I, IMPPACT, STEP AP 213 and 224, and proposed a machining process planning activity model [7] and a manufacturing process information model [8], which specify functional components and information requirements for transforming raw material to finished parts. In order to have an accurate and faster decision-making support to downstream applications with some level of automation, the knowledge related to product development should be captured in its product model. Wang et al. [36] described an enhanced product representation model that consists of function data, control data, management data and reusable data, and developed a wrapper helping to transfer a traditional CAD model into a customized and reusable product model.

Zhang et al. [37] introduced a STEP-based product model data exchange framework for virtual enterprises, and many papers demonstrated the related data translation, such as one-to-one translator between an IGES file and a STEP AP203 file for data exchange of heterogeneous CAD systems [37], a STEP AP 203 file and an AP 209 file for CAD/CAE data exchange [37], and STEP AP 203 and AP 224 [4,38] and AP 238 (STEP-NC) files [10] for CAD, CAPP, CAM and CNC data communication, whilst Pratt et al. [39] described the progress of work aimed at extending the STEP standard to provide some important new capabilities such as allowing the transfer of procedural (construction history) feature-based CAD models with parameterization and constraints between different CAD/CAM systems.

A majority of papers addressed such a neutral file-based solution for the integration of CAD and CAM, where the CAD model of a part is exported via a STEP or IGES file from a certain commercial CAD system to an external feature recognition system, and the recognized features are used in conjunction with knowledge and/or AI-based methods to prepare a process plan for the part. And again the process plan is exported via STEP to CAM systems. However, this is a kind of one-way integration from CAD to CAPP, and further to NC code generation. Furthermore, its practicability is dependent on a mature feature recognizing system.

In contrast to most current academic literature and research reports which focus on theoretical study or developing a prototyping system, this work has the following aims:

- Develop a practical CAPP system that is bi-directionally integrated with commercial CAD.
- Research a mechanism for extracting, converting and encapsulating feature, knowledge, technical information and other related data into current geometry- or design feature-oriented CAD models to create an integrated product information model to facilitate effective integration with the downstream activities of the product lifecycle.
- Explore a method that hybridizes knowledge engineering, neural networks and genetic algorithms to obtain a global optimal process plan.
- Provide a tool for implementing process drawing marking and 3D material stock generation automatically.
3. The proposed methodology

3.1. System architecture

At present, both CAD and CAM system technologies are quite mature, and many commercial automation island systems are widely used in industry. The main problem is the lack of an appropriate CAPP system that could interlink CAD and CAM.

To meet the urgent requirements from industry for an integrated CAD/CAPP system, and overcome the problems such as low efficiency and error proneness of traditional feature recognition systems, this paper proposes a feasible and practical approach for integrating CAD and CAPP bi-directionally on the basis of a current “design-by-feature” commercial CAD system that is widely used in industry. Fig. 1 describes the integrated CAD/CAPP system architecture, where all the functions that are needed to interlink CAD and CAM but are not supported by current commercial systems are developed by using the APIs provided by the CAD platform and Visual Studio toolkit.

3.2. Information integration

As mentioned at Section 2, because the current CAD data model is mainly geometry- or design feature-oriented, product model data insufficiency and incompatibility are still the major obstacles to system integration, especially the integration of product design and process planning. To make up for the shortfall, this paper proposes an Integrated Product Information Model (IPIM), as illustrated in Fig. 2, where the STEP modeling method, feature technology and the traditional relational data modeling technique are used, and the IPIM data could be exported in a neutral file that can be imported by other application systems of the product lifecycle.

IPIM is a software representation of geometry, feature, knowledge, technology and other management data that describes a product throughout its lifecycle. Wang et al. [36] discussed how to define and enhance the function data, control data, management data and reusable data to construct a reusable product model on the basis of a geometric model. This paper focuses on how to recognize, convert and encapsulate feature data, and the related process data and knowledge that are necessary for information sharing and process integration.

3.3. Function model

The detailed function or activity model of the integrated CAD/CAPP system is shown in Fig. 3. First, the user finishes product design and CAD modeling (A1) according to customer requirements. Then, Functional Module A2 will translate the geometry-oriented CAD model into a feature-based CAD model by going through feature recognition and/or conversion, feature parameters and constraint extraction, and feature precedence tree reconstruction. And Module A3 is used to supplement or extract the necessary technical specifications such as location dimension and tolerance, surface roughness and hardness that may not be involved or recorded explicitly in the CAD model, to create a feature-based product model involving geometric and non-geometric feature information, from which the process plans will be created by Module A4. Both feature recognition and conversion (A2) and technical information processing (A3) are preconditions for process decision-making, and the key steps for transferring a geometry-oriented CAD model into the IPIM, and preserving the integrity and consistency of the feature model and product information, so that they could be shared by downstream applications. Finally, Module A5 will automatically mark the technical specifications and process plans on 2D engineering drawings, and create the corresponding 3D material stock geometry model, inter-process stock model and the removed allowance model.
Among these activities, Activity A1 could be supported by the CAD system itself, while Activities A2 to A5, which are not supported or provided by current commercial systems, should be developed and customized. Their implementation will be detailed in the following sections.

4. Feature recognition and conversion

4.1. Feature recognition and conversion algorithm

Feature carries a mass of engineering information, both geometric and non-geometric, and is the medium of information transmission among CAD, CAPP, and CAM. However, although many commercial CAD systems are declared to be feature-based, in fact, the so-called feature is just the modeling operator, menu name (such as Protrusion, Revolution, Sweep, Cutout, Copy, etc.) or a range of primitive volumes (typically Block, Cylinder, Cone, Sphere, etc.), instead of a real form feature or design and manufacturing feature. Such a modeling operation can create many different form features or manufacturing features. For example, a Cutout operation may create a hole, step, slot, pocket, round, chamfer, etc. In other words, if these features are created by Cutout operation, all of them will be recorded permanently as “Cutout feature” in the CAD model or database, even though their forms or dimensions changed by way of feature interaction or one feature modifying another with the design process going on. So, in order to transfer the exact engineering information
to downstream application systems to make correct decisions, these Cutout features should be recognized and converted into the corresponding hole, step, slot, pocket, round, chamfer, etc. respectively.

Therefore, regardless of the geometry- or feature-based CAD system, the identification of form or design features from CAD models and their further conversion into manufacturing features is a key step indispensable to the integration of CAD, CAPP and CAM. Because current CAD systems are able to record the modeling procedure or construction history that reflects the modeling method and the order in which a component model is originally constructed, this section presents a feature recognition and mapping approach for converting modeling features into the corresponding form features, to which the design intents and manufacturing intents will be attached.

1) Feature definition and representation

To clearly describe the feature recognition and mechanism of conversion between different feature sets, the following features are defined:

Form feature \( F_F \) represents a specified geometric shape that is composed of a set of finite surfaces with certain adjacent and constraint relationships. The mathematical representation of the form feature can be generally defined as:

\[
F_F = (f_1 \cup f_2 \cup \cdots \cup f_j) \cup (A_1 \cup A_2 \cup \cdots \cup A_n) \\
\cup (C_1 \cup C_2 \cup \cdots \cup C_m)
\]

where \( f_i \) denotes the composing surface set of form features; \( A_j \) represents a set of adjacent relationships between the composing surfaces, \( A_j \in A, A = \{A_{co}, A_{cv}\} \); \( A_{co}, A_{cv} \) represents the adjacent relationship of two surfaces which are linked together through the concave edge and convex edge, respectively; \( C_k \) denotes a set of constraint relationships between surfaces of the features, \( C_k \in C, C = \{C_{perp}, C_{par}, C_{ang}, C_{coaxis}, C_{coface}, C_{pattern}\} \). Both \( F_F \) and \( F_D \) can be represented uniquely. For example, the form feature “Rectangular Slot”, as shown in Fig. 4, can be defined as:

\[
\text{RectSlot} = (f_1 \cup f_2 \cup f_3) \cup (A_{co}) \cup (C_{perp} \cup C_{par}) \\
A_{co}((f_1, f_2), (f_2, f_3)) \\
C_{par}(f_1, f_3), C_{perp}((f_1, f_2), (f_2, f_3)).
\]

The definition could be used as the predefined template for recognizing and extracting all “Rectangular Slot” instances from CAD models, and the perpendicular constraints can distinguish it from other slots such as the dovetail slot and V-slot.

Using a mathematical description of the feature, machining features can be deduced and formed by the Set Operation, and the difficult problem of feature interaction can be described mathematically and converted in theory [40].

Manufacturing operation feature \( F_O \) is a modeling operator or menu name of a CAD system, such as Cutout, Extrusion, Sweep, etc., which is used to create a certain form feature. It can be defined as a set of Modeling operator, Operation direction and Geometry object.

Design feature \( F_D \) is a form feature that can represent design intents and be used to construct the shape of a component for realizing certain functions, where design intents include precision attributes (e.g., geometric dimension and tolerances, surface roughness and hardness), functions (such as centerline, symmetric axis, datum, localizer, etc.) and material.

Manufacturing feature \( F_M \) is a form feature removed from the material stock and the related machining intents including machining method, tool access direction, and attributes (dimensions and tolerances, allowance, surface roughness, etc.) to create a design feature or the finished part up to design intents and economically, where material stock can be obtained by offsetting the part model with machining allowances, or determined by the minimal encapsulation volume of a cubic block containing the designed part.

2) Feature mapping mechanism

The feature definition above implies the relationship and mapping mechanism between \( F_F, F_O, F_D \) and \( F_M \). Both \( F_D \) and \( F_M \) are \( F_F \). Form features can be classified into two types: additive volume features or protrusion features, such as Boss, and subtractive volume features or depression features, such as hole, slot, etc. Similarly, \( F_O \) which is used to create the corresponding type of \( F_F \) can also be identified as an additive volume operation, such as Protrusion, or a subtractive volume operation, such as Cutout. For a subtractive volume form feature, \( F_M = F_D \), and for an additive volume form feature, \( F_M \) = Material stock – \( F_D \). Therefore, the main task of feature recognition is converting \( F_O \) into the corresponding \( F_F \).

3) Algorithm overview

1) Get the \( F_O \) one by one from the CAD model object using APIs provided by the CAD platform.
2) Extract the geometry entities of the \( F_O \), including its 2D sketch or profile (if it exists), and get the surface set of the created feature.

Identify surface adjacent relationships among the surface set by crossing the outside normal vectors of every two adjacent surfaces and dotting with the directional vector of the common edge to get the angle (constraint) between adjacent surfaces. If the angle is less than 180°, two surfaces are adjacent through a concave edge. Otherwise, two surfaces are adjacent through a convex edge [38]. In the case of a freeform surface, the normal vector at the neighborhood near
the common edge can be used to determine the concave or convex attribute, and the neighborhood can be obtained along two intersection curves that are created by intersecting a plane normal to the common edge with two adjacent surfaces [25].

Detect constraints between surfaces that do not join together directly by vector operation on outside normal vectors of such two surfaces; if their normal vectors are opposite, it means there exists a parallel constraint between these two surfaces.

Analyze the adjacent character of the surface set, detect all subsets of surfaces that are joined together, and determine available kinds of form features \((F_F)\) created by the \(F_O\) by comparing each subset of surfaces and the related constraints with the predefined form feature templates. If more than one feature is created, record their twin relationship for application feature mapping.

Calculate feature parameters and extract the form feature instances, including calculating the distance between two parallel surfaces, inheriting parameters from \(F_O\) and the parent feature to which the form feature is attached, and deriving the feasible tool access directions from the axis direction, modeling the operation direction of \(F_O\) and/or outside normal directions of the corresponding surfaces in the surface set which are not sheltered from other faces of the geometry model.

4.2. Manufacturing feature recognition and reconstruction

Manufacturing features represent the volumes to be removed from the material stock of a part, and can be classified into surface features and form features. Surface machining features refer to any surfaces that have removal volumes, from the stock, but do not belong to any recognized form feature of a part.

(1) Feature recognition and parameter extraction

Fig. 5 shows a typical case study for feature recognition and reconstruction, where, except for the basic feature of the component, and Feature 1 and Feature 3, all form features are created by using the same modeling operator \((F_O)\). ExtrudeCutout, and naturally all the corresponding feature instances are identified as “ExtrudeCutout” in the CAD model or database. Obviously, this so-called ExtrudeCutout feature information cannot be accepted or shared by downstream application systems such as CAPP and CAM. Through feature recognition, all ExtrudeCutout modeling features are correctly recognized as the corresponding form features including through hole, blind hole, rectangle slot, dovetail slot, keyway, pocket, step, blind step, round, chamfer, and depression freeform feature, and further converted into manufacturing features (surface features and form features), and the feature precedence tree indicating the feature attachment relationship is reconstructed very well.
Taking Feature 9 in Fig. 5 as an example, it could be identified as a modeling feature, Extrude Cutout, together with its four faces, \( f_1, f_2, f_3 \) and \( f_4 \), from the CAD model. By analyzing its geometry objects following the algorithm procedures above, it could be figured out that face pairs \((f_1, f_2)\), \((f_3, f_2)\), \((f_4, f_2)\), \((f_1, f_4)\), and \((f_4, f_3)\) are connected by a concave edge and they are perpendicular, while face pair \((f_1, f_3)\) are parallel. These attributes are matched with pre-defined form feature “Blind Slot”, so ExtrudeCutout is recognized as a Blind Slot. Its width is the distance between two parallel side face pair members \( f_1 \) and \( f_3 \). The height is the distance between its base face \( f_2 \) and the base face of Step 25. The length is the distance between its end face \( f_4 \) and Basic Surface 3. These feature parameters will be displayed automatically when processing technological information of the feature. The feasible Tool Access Directions (TAD) are opposite to the outside normal of its base face \( f_2 \) or end face \( f_4 \), and the recommended TAD is opposite to the outside normal of the base face \( f_2 \). Similarly, by using the normal vector in the neighborhood near the common edge on freeform surfaces, Feature 3, a depression freeform feature, and Feature 4, a keyway feature, can also be recognized.

Because the feature recognition and mapping approach presented in the paper is converting a geometric modeling feature into a corresponding form feature, in theory, no matter how complex the part is, it is built by a series of simple modeling operators each of which may create an individual or several basic form features. As shown in Fig. 5, the basic form features commonly used, such as Hole, Blind Hole, Counter Bore, Counter Sink, Slot (Rectangle Slot), V-Slot, Dovetail Slot, T-Slot, Blind Slot (Open Pocket), Step, Closed-end Step, Blind Step, Pocket, Boss, Round, Chamfer, Keyway, Pattern, and so on, could be handled by the system.

(2) Feature interactions and consistency preservation

Feature interactions have a wide range of effects on a component model because they not only change the predefined feature geometric form but also alter their attributes. According to the investigation on current feature-based modeling processes and systems, it was found that the inconsistency between the final component model and the information available in the CAD database is another key issue required to be solved for downstream applications.

In this method, because the high level information retrieved from a CAD system is utilized, the interacting and interrelating relationships between features can be handled, so problems such as the low efficiency and error proneness of traditional feature recognition systems are overcome, and the validation and integrity of features can be checked and preserved. For example, Feature 1 in Fig. 5 is initially created and used as Blind Hole 2. Then with other features being added to the model, such as Feature 13, which corresponds to Extrude Cutout 25 in the CAD database, and Step 25 in the recognized feature tree, because of feature interactions, the original blind hole has been modified into a through hole as identified by ThroughHole2 in the reconstructed feature relationship tree, but it is identified permanently as Blind Hole 2 in the CAD database. Similarly, the height of Feature 9 is also changed, instead of the original parameter indicated in the CAD database.

Furthermore, even if a feature is divided into several sub-features because of feature interactions, such as Basic Surface 4, which is divided into three surface features by a rectangle slot and a dovetail slot, these features could be treated as the same manufacturing feature by using coface constraints or referring to its original modeling feature. This is very important for process planning to arrange these sub-features to be machined at the same time.

(3) Feature tree reconstruction

CAD modeling usually starts from a simple geometry entity represented the basic shape of a component, such as a block for prismatic parts, or a cylinder for rotational parts, and then to which other features are successively added to construct the component. But the CAD system usually records such a modeling procedure in time order linearly, as shown by the original modeling features and construction history tree in Fig. 5, instead of the feature precedence tree that denotes the feature attachment relationship. Such a feature precedence tree plays an important role in process sequencing, so it should be reconstructed before process planning.

In this paper, the surfaces of the initial simple geometric entity of a component are defined as the basic surface features which serve as the original datum or target surfaces (parent features) of other attached features. By searching the reference or datum surface of a form feature, we can get its parent feature, and similarly by searching the internal loops on surfaces of a feature, we can get its child features. In this way, the feature precedence tree is reconstructed, as shown by the recognized features and reconstructed feature relationship tree in Fig. 5. For example, on the base face \( f_2 \) of Blind Slot, there is a circular internal loop, which implies the existence of another child feature, Blind_hole. So in the reconstructed feature precedence relationship tree, Blind_hole should be located at the next layer of Blind Slot.

(4) Technical information processing

After feature recognition, some relevant technical information that may not be involved or recorded explicitly in the geometry-oriented CAD model should be handled interactively. Technological information can be classified into two types depending on whether it is self-referenced or needs a cross-reference. To the former belong surface roughness, hardness, straightness, flatness, cylindricality, and so on. Dimensional tolerance, parallelism, concentricity, perpendicularity, angularity, etc. are typical examples of the latter. The self-referenced tolerance can be treated as attributes of a feature or geometry entity. For instance, the surface roughness and hardness could be stored simply as attributes of a feature. In contrast, the cross-referenced tolerance implies a character between two geometric entities. For example, an entity couple for linear dimensional tolerance may be face to face, face to edge, face to vertex, edge to vertex, or vertex to vertex. Since most CAD modelers do not provide a data structure for incorporating tolerance information, an especially designed data structure should be provided to record the datum entity ID, target entity ID, dimension and its upper and lower tolerance, and other attributes.
Fig. 6 depicts the interface and procedure for assigning dimensional tolerances, surface roughness and hardness to Feature Blind_Slot_23. The user can select any feature whose technical information should be processed by picking the corresponding feature on the reconstructed feature tree or on the part model with a mouse. The selected feature will be highlighted and the technical information processing dialog box will be activated. At this time, all feature attributes obtained through feature recognition, such as the length, width and height of Blind_Slot_23, will be shown on the interface automatically, and the feature parameters are also marked on the 3D part model for users to make sure which parameter should be processed. The user is just required to pick the corresponding reference entities and select or input values that need to be supplemented by following the navigator. For form dimensional tolerance of a feature, its referenced entities are recognized automatically. If location dimensional tolerance needs to be attached, the user is required to pick a target entity and a datum entity, for example, side face f1 and Basic surface 1 to locate Feature Blind_Slot_23. Once the cross-referenced entities are specified, the dimension parameter will be calculated and marked on the part model. Meanwhile, these dimensional parameters can be modified and the parameterized modeling function will rebuild the part model. And some other technical requirements can also be added; for example, the user can specify the functional description of a feature—for example, a locating hole, and whether the hole should be reserved or removed from the material stock model.

All the recognized features and related attributes will be well organized in IPIM, that could be shared by downstream application systems, and in the case where a feature parameter or technical datum is detected as unreasonable through process analysis, once confirmed, it could be modified directly to reach a two-way integration.

5. Process planning

Once manufacturing features and the related technical information are obtained, the following key steps are process selection and operation sequencing. In this section, a method that hybridizes knowledge engineering, a neural network and a genetic algorithm (GA) is used to seek a global optimal process plan.

5.1. Process decision-making

Generally speaking, there are alternative machining methods that can be used to machine a feature to the required technical specification. It is usually difficult to decide which one should be selected. However, if we consider such factors as batch size of a product and enterprise resources, the priority of these machining methods might be different.

In this paper, a back-propagation Artificial Neural Network (ANN) model is built and trained to make decisions on these alternative machining methods and their selection probabilities. The training samples, i.e. the available machining methods and their probabilities of being selected for machining a feature instance, are graded by process experts according to certain conditions such as the dimension, tolerance and surface roughness of the feature, and the batch size, material, stock type, heat treatment and structure form of the component. If the final machining method of a feature is selected, its previous
Fig. 7. ANN and backward reasoning-based process decision-making procedure.

Fig. 8. Flowchart of the GA-based approach for process planning and optimization.

Datum precedence: Datum features used to locate other features should be machined first.

Feature precedence: Parent features on the feature precedence tree of a part should be machined before their child features.

Process precedence and concentration: Some common process precedence constraints include roughing – semi-finishing – finishing, drilling – boring – reaming for hole making, turning – grooving – chamfering before thread cutting, and thin-wall precedence occurred when the distance between features was very small. On the other hand, the operations for pattern features or other features with the same TAD should be arranged together. Coface and coaxis features should be machined at the same time, etc.

For example, a feasible process plan for hole making could be described as: (hole starting (spot drilling) – core making (twist drilling/spade drilling/end drilling/gun drilling) – hole improving (reaming/boring/precision boring) – hole finishing (grinding/honing)).

For the detailed constraint adjustment algorithm please refer to the literature [41].

Then, at the inner loop, a steady-state genetic algorithm, which can converge to a global optimum [42], is used for operation sequencing optimization. The fitness function is the total weighted costs of the machine costs, tool costs, machine change costs, set-up costs and tool change costs, and the individual representation, fitness evaluation, crossover and mutation algorithm are detailed in Reference [32].

After each inner loop, the GA can obtain an optimal process plan candidate based on the selected machining schemes and corresponding operation precedence constraints. By comparing the machining costs of those optimal process plan candidates obtained after all outer and inner iterations where all kinds of
different machining schema available for each feature could be generated and tested for operation sequencing optimization, the process plan candidate with the minimum cost will be the global optimal process plan.

Figs. 9–11 show a case study of process optimization for a mold slide part by different machining schema selection strategies. Fig. 10 shows the five best solutions for process optimization obtained by using the HPS method. These five solutions have the same machining schema with the highest selection probability of each feature but different process sequencing, and their minimal cost is 5639. Fig. 11 shows the five best solutions obtained by using the RWS method. These five solutions correspond to different machining schemes for each feature and different process sequencing, and their minimal cost is 5253. From Figs. 10 and 11, we can draw the conclusion that the combination of feature machining schemes with the highest selection probabilities (i.e. the best machining process plans for every feature) might not result in a global optimal process plan. We should synthesize the selection of the machining scheme and the optimization of operations.
The optimal process plans for the slider part

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rough shaping basic surface 1</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>23</td>
<td>Tapping thread 10</td>
<td>M1</td>
<td>C5</td>
<td>−z</td>
</tr>
<tr>
<td>2</td>
<td>Semi-finish shaping basic surface 1</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>24</td>
<td>Rough grinding countersinking 8</td>
<td>M13</td>
<td>C27</td>
<td>+x</td>
</tr>
<tr>
<td>3</td>
<td>Rough shaping basic surface 3</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
<td>25</td>
<td>Rough grinding countersinking 7</td>
<td>M13</td>
<td>C27</td>
<td>+x</td>
</tr>
<tr>
<td>4</td>
<td>Semi-finish shaping basic surface 3</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
<td>26</td>
<td>Rough shaping step 15</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
</tr>
<tr>
<td>5</td>
<td>Rough shaping basic surface 5</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>27</td>
<td>Semi-finish shaping step 15</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
</tr>
<tr>
<td>6</td>
<td>Semi-finish shaping basic surface 5</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>28</td>
<td>Rough shaping basic surface 4</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
</tr>
<tr>
<td>7</td>
<td>Rough shaping basic surface 2</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>29</td>
<td>Semi-finish shaping basic surface 4</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
</tr>
<tr>
<td>8</td>
<td>Finish milling basic surface 1</td>
<td>M11</td>
<td>C11</td>
<td>+z</td>
<td>30</td>
<td>Rough shaping step 13</td>
<td>M11</td>
<td>C11</td>
<td>−x</td>
</tr>
<tr>
<td>9</td>
<td>Semi-finish shaping basic surface 2</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>31</td>
<td>Semi-finish shaping step 13</td>
<td>M3</td>
<td>C17</td>
<td>+y</td>
</tr>
<tr>
<td>10</td>
<td>Rough shaping basic surface 6</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>32</td>
<td>Finish milling step 13</td>
<td>M11</td>
<td>C11</td>
<td>−x</td>
</tr>
<tr>
<td>11</td>
<td>Semi-finish shaping basic surface 6</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>33</td>
<td>Finish milling basic surface 1</td>
<td>M11</td>
<td>C11</td>
<td>−y</td>
</tr>
<tr>
<td>12</td>
<td>Rough shaping step 16</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>34</td>
<td>Finish milling blind closed pocket 17</td>
<td>M11</td>
<td>C35</td>
<td>−y</td>
</tr>
<tr>
<td>13</td>
<td>Semi-finish shaping step 16</td>
<td>M3</td>
<td>C17</td>
<td>+x</td>
<td>35</td>
<td>Finish milling blind closed pocket 17</td>
<td>M11</td>
<td>C35</td>
<td>−y</td>
</tr>
<tr>
<td>14</td>
<td>Drilling thread 9</td>
<td>M1</td>
<td>C37</td>
<td>+y</td>
<td>36</td>
<td>Rough milling step 14</td>
<td>M11</td>
<td>C11</td>
<td>−z</td>
</tr>
<tr>
<td>15</td>
<td>Rough reaming thread 9</td>
<td>M1</td>
<td>C36</td>
<td>+y</td>
<td>37</td>
<td>Semi-finish milling step 4</td>
<td>M11</td>
<td>C11</td>
<td>−z</td>
</tr>
<tr>
<td>16</td>
<td>Tapping thread 9</td>
<td>M1</td>
<td>C5</td>
<td>+y</td>
<td>38</td>
<td>Finish milling step 15</td>
<td>M11</td>
<td>C11</td>
<td>+x</td>
</tr>
<tr>
<td>17</td>
<td>Drilling countersinking 7</td>
<td>M1</td>
<td>C37</td>
<td>+x</td>
<td>39</td>
<td>Semi-finish boring inclined hole 11</td>
<td>M15</td>
<td>C32</td>
<td>−b</td>
</tr>
<tr>
<td>18</td>
<td>Drilling countersinking 8</td>
<td>M1</td>
<td>C37</td>
<td>+x</td>
<td>40</td>
<td>Finish reaming inclined hole 11</td>
<td>M11</td>
<td>C36</td>
<td>−b</td>
</tr>
<tr>
<td>19</td>
<td>Drilling inclined hole 11</td>
<td>M1</td>
<td>C37</td>
<td>−b</td>
<td>41</td>
<td>Drilling thread 12</td>
<td>M1</td>
<td>C37</td>
<td>−a</td>
</tr>
<tr>
<td>20</td>
<td>Finish enlarging inclined hole 11</td>
<td>M1</td>
<td>C37</td>
<td>−b</td>
<td>42</td>
<td>Rough reaming thread 12</td>
<td>M1</td>
<td>C36</td>
<td>−a</td>
</tr>
<tr>
<td>21</td>
<td>Drilling thread 10</td>
<td>M1</td>
<td>C37</td>
<td>−z</td>
<td>43</td>
<td>Tapping thread 12</td>
<td>M1</td>
<td>C5</td>
<td>−a</td>
</tr>
<tr>
<td>22</td>
<td>Rough reaming thread 10</td>
<td>M1</td>
<td>C36</td>
<td>−z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1

Table 2

Available machines and their cost index

<table>
<thead>
<tr>
<th>Machine No.</th>
<th>Machine types</th>
<th>MCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Drilling press 1</td>
<td>10</td>
</tr>
<tr>
<td>M3</td>
<td>Planer</td>
<td>15</td>
</tr>
<tr>
<td>M7</td>
<td>CNC-3-axis vertical milling machine</td>
<td>80</td>
</tr>
<tr>
<td>M8</td>
<td>3-axis vertical milling machine</td>
<td>60</td>
</tr>
<tr>
<td>M9</td>
<td>Vertical milling machine 1</td>
<td>50</td>
</tr>
<tr>
<td>M10</td>
<td>Planer milling machine</td>
<td>55</td>
</tr>
<tr>
<td>M11</td>
<td>Vertical milling machine 2</td>
<td>40</td>
</tr>
<tr>
<td>M12</td>
<td>Circular grinder</td>
<td>30</td>
</tr>
<tr>
<td>M13</td>
<td>Internal grinder</td>
<td>40</td>
</tr>
<tr>
<td>M14</td>
<td>Face grinder</td>
<td>50</td>
</tr>
<tr>
<td>M15</td>
<td>Horizontal borer</td>
<td>60</td>
</tr>
<tr>
<td>M20</td>
<td>5-axis interlock machining center</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2

Available tools and their cost index

<table>
<thead>
<tr>
<th>Tool No.</th>
<th>Tool types</th>
<th>TCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Rasper</td>
<td>2</td>
</tr>
<tr>
<td>C5</td>
<td>Screw tap</td>
<td>10</td>
</tr>
<tr>
<td>C11</td>
<td>End mill</td>
<td>25</td>
</tr>
<tr>
<td>C15</td>
<td>Expanding drill</td>
<td>28</td>
</tr>
<tr>
<td>C17</td>
<td>Planer tool</td>
<td>20</td>
</tr>
<tr>
<td>C19</td>
<td>Special mill</td>
<td>35</td>
</tr>
<tr>
<td>C27</td>
<td>Grinding wheel 1</td>
<td>10</td>
</tr>
<tr>
<td>C28</td>
<td>Grinding wheel 2</td>
<td>15</td>
</tr>
<tr>
<td>C35</td>
<td>Ball-end mill</td>
<td>52</td>
</tr>
<tr>
<td>C36</td>
<td>Reamer</td>
<td>15</td>
</tr>
<tr>
<td>C37</td>
<td>Drill 1</td>
<td>8</td>
</tr>
</tbody>
</table>

6. Stock model and process drawing generation

6.1. Stock model generating

Mechanical parts are commonly manufactured using multiple manufacturing processes. Primary processes such as casting are usually used to provide the material stock from which the final product is machined. So, how to generate a reasonable 3D geometric model of the starting workpiece is a crucial and practical step. Unfortunately, besides simply selecting the minimal enveloping volume of a cubic block containing the part as a stock, other related research work is rarely reported. Kim and Wang [43] proposed a machining feature recognition method for cast-then-machined parts. In their work, the starting workpiece is generated by interactively identifying the faces on a finished part to be made by casting followed by machining, then offsetting these faces by a uniform machining thickness to obtain cast faces, i.e. the material faces.

In this paper, after process decision-making, the machining allowance for each machining operation is summed up to get the total material allowance of a machining feature, and then, the material stock model (including the in-process stock model) of the part is created automatically by removing those small features or feature details specified during technical information processing from the part model, followed by adding the corresponding material allowance to each machining.
feature surface. This function is very useful especially when the material stock is ordered from suppliers, which can ensure enough machining allowance for the manufacturing of a product or part, and prevent unauthorized disclosure of confidential product information.

Of course, because the neighboring relationship among surfaces on a part model differs in thousands or ten of thousands of ways, different material allowance adding operation strategies such as offsetting, thickening, extruding should be used, and sometimes it may be difficult to create a reasonable material model for a certain part shape or machining feature fully automatically, and some interactive operations could be require guiding the process or editing and modifying the created material model.

6.2. Process drawing generating

A process drawing is a 2D engineering drawing marked with process information such as “to-be-machined features or surfaces” and the related surface roughness, machining method and sequences, etc., which indicates where or which feature should be machined, how to machine it, and what the technical requirements are. Process drawings are usually required for archiving and machine operator’s reference. In this integrated system, the technical information, usually the surface roughness, machine methods and sequences, of a feature could be marked automatically on the related geometric entity that indicates the feature in the engineering drawing. Since it is difficult to locate each ‘to-be-marked symbol’ at the most appropriate position, once a process drawing is created, all marked items can be relocated by picking and dragging interactively. If the process plan is modified on the product model, the corresponding process drawing will be updated automatically.

6.3. Other aided functions

Other aided functions include Preference or experience parameter setting, for example, feature dimension parameter setting for hole, step, slot, and other small features or feature details such as round and chamfer located at convex edges that should be filled or removed when creating a material stock model; Standard process knowledge and template customizing for each type of machining feature or pattern feature; Technical parameter statistics and checking, including summing up machining feature type, number, machining area and removed material volume, sorting machining features (surfaces) by machining method or technical parameter values, and displaying surfaces with different colors according to their process attributes, such as blue color for material surfaces, other colors for machining surfaces with different roughness values; and Neutral file output of product model data including process plans together with geometry, feature and technical specifications. Fig. 12 shows a case study from industry.

7. Conclusions

The basic issues towards the integration of design and manufacturing engineering have been discussed systemically. In contrast to most current academic literature and research reports which focus on theoretical study or developing a prototyping system, this paper proposed a practical approach to a total integration of CAD and CAPP based on commercial systems with the focus on key techniques such as feature parameters and constraints extraction, feature precedence tree reconstruction, technical information processing, automatic process marking and 3D material stock geometric model generation. The integrated system is implemented on a commercial CAD platform, SolidEdge. This platform software was specified by industrial partners; it does not imply that
the platform is necessarily the best available for the intended purpose, and it could be any other commercial CAD system that is used in industry.

The implemented system not only encapsulates design and machining intents in the traditional geometry-oriented CAD model well, to support CAD/CAPP/CAM integration effectively, but also makes the complex and burdensome process planning, process drawing marking and material stock generation processes automated. A case study and industry implementation illustrate the feasibility of the approach and algorithms proposed. The authors believe that the proposed methodology will contribute to filling the gap, that current CAD/CAM systems do not support CAPP, and be highly beneficial to industry.

Further development work will aim to implement system integration with CAM and subdivide complex product or material stock models in such a way that the manufacture of individual subdivided parts and subsequent assembly is easier than the whole.

References


