Int J Adv Manuf Technol (2004) 23: 79–86
DOI 10.1007/s00170-003-1630-1

ORIGINAL ARTICLE

Y.-S. Ma · T. Tong

An object-oriented design tool for associative cooling channels in plastic-injection moulds

Received: 17 December 2002 / Accepted: 17 December 2002 / Published online: 17 October 2003

Abstract Due to the demand for short product development cycles, plastic injection mould designers are required to compress their design time and to accommodate more late changes. This paper describes an associative design approach embedded in a cooling channel module of a mould design software package. It gives a set of comprehensive object definitions for cooling circuits, and addresses balanced or unbalanced designs. CAD algorithms that have been developed are briefly explained. With this new approach, mould designers can easily propagate changes between mould plates or inserts and the cooling system without the need for tedious rework. Hence, this approach can significantly reduce the total design time and the impact of late changes.

Keywords Cooling circuit · Plastic mould design · CAD/CAM · Associative design · Design automation

1 Introduction

Currently, most CAD systems are unable to capture design intent completely and unambiguously. Rich design information cannot be fully represented in CAD models and late changes in the product development cycle cause a lot of rework. It has been acknowledged that CAD interoperability should cover integration with knowledge-based engineering (KBE) systems [1]. However, there is no mechanism to enable design intent information flow. Such an information gap is very obvious in plastic injection mould design as well. Mould designers are facing increasing pressure to reduce the design time, and yet are expected to assure mould quality.

Various CAD tools for designing plastic injection moulds have emerged since the early 1970s [2], most of which focused on moulding flow analysis and optimisation algorithms [3, 4, 5]. Recently, the design of mould sub-systems, such as core/cavity inserts [6, 7], runners [8, 9], gate locations [3, 4, 5] and cooling systems [10], have been the focus. For cooling system design, Wang et al. [11] suggested a strategy with three stages, initial design with one-dimensional approximation, two-dimensional design with optimisation, and three-dimensional design with cooling effect analysis. They have developed a program that uses 3D-boundary element methods to analyse 3D heat transfer. All the above-mentioned tools are only able to generate geometrical information. The representation and reuse of rich design information at different levels are not addressed.

Object-oriented (OO) software technology has been applied to meet the information representation gap in mould design [12]. Object definitions can provide a great deal of help in sorting out complicated entities, especially for part-independent parts and features. However, maintaining the relationships among geometrical entities and making them customisable is still not a trivial task. The CAD software development approach that can achieve persistent relationships among geometrical entities is referred to as the associative design approach. One way to build design intent and process knowledge into a CAD system is in the form of a process wizard, which is basically an application program coupled with a set of sequenced user interfaces (UIs) to guide users to complete certain interactions with the computer system. MouldWizard from EDS Inc. is one such process-based wizard [13]. This paper introduces the associative design approach applied in its cooling channel module. Market feedback shows that this concept can significantly reduce the gap between human knowledge and consistent computer representations.

The cooling system in a mould affects not only the quality of resultant moulding parts but also efficiency in

Y.-S. Ma (✉) · T. Tong
Design Research Centre,
School of Mechanical and Production Engineering,
Nanyang Technological University,
639798 Singapore, Singapore
E-mail: mysma@ntu.edu.sg

moulding production. In the current industry practice, it is common to use at least four major cooling circuits in a mould assembly. They are located on the cavity insert, the core insert, A-plate and B-plate [14]. Wang et al. [11] and Singh [15] recognised that the parameters in designing cooling system are numerous; design variables, such as locations, types of cooling channels, and 3D layout of circuits, are usually modified frequently to address late part design changes as well as mould design optimisation. The modification process is laborious and error prone because designers have to edit and update CAD models repetitively. Mok et al. [16] developed a cooling system that can automatically retrieve certain circuit patterns, such as straight or U types, but the association among geometrical entities is not discussed. An expert system for designing cooling systems was introduced by Kwon et al. [10]. The system consists of four levels: layout design, analysis, evaluation and decision-making. A decision-making module evaluates the redesign of cooling channels based on the rules stored in a knowledge base. However, there is no integration with a parametric CAD system.

In summary, an efficient and user-friendly cooling system design tool is highly sought; such a system can be expected to free mould designers from tedious geometrical updating and to keep design models consistent, so that the total mould design cycle time can be shortened. This paper presents a cooling channel design tool that provides substantial automation for cooling circuit generation with associative links among cooling holes and their drilling faces.

1.1 Generic issues associated with capturing design intents

In the industry, cooling channels are usually designed in the form of cooling circuit, but represented as HOLE features using CAD tools. On the other hand, experienced mould designers found that solid cylinders are also commonly used instead to represent cooling channels. In the latter approach, when the design is finalised, all channels are united to form a cooling circuit. With such united circuits defined with the help of CAE analysis tools, the cooling effect can be evaluated [11]. These circuits are not converted into holes until the design has been finalised and is ready for CAM tool path generation. With this form of representation, a CAD system can display/draw cooling channels for visual inspection, without displaying detailed features of the core/cavity inserts and mould plates. Repositioning and modifying cylinders also require fewer steps than using HOLE features. It enables automatic checking for collisions between cooling channels and other mould features, such as cavities and ejecting-pin holes.

However, representing cooling channels in the form of solid cylinders has several problems. First, many steps are still required for a simple channel, such as creating a cylinder, chamfering the blind end in the case of a blind hole, and running through a series of dialogue boxes to position and orient it. Commonly, there are many channels in a cooling circuit, so creating them involves a lot of repetitive commands. When modifications are needed, cylinders have to be edited repetitively again. This situation is error-prone. Second, cooling channels are not self-identified. For automatic heat transfer analysis or collision checking, identifying cooling channels is particularly important. Third, they cannot provide orientation information for plugs, nozzles, or baffles to be inserted into cooling channels in a user-friendly drag-and-drop manner. Hence, mould designers have been frustrated with tedious steps.

1.2 Semantic definitions of a cooling system

An object-oriented software design approach can be applied to address the issues discussed in the above section. Defining a set of object types or classes that provides self-contained definitions of cooling systems and enables dynamic updating to validate the cooling system, is essential. In Fig. 1, the simplified semantic structure of a cooling system and its related component member types is shown. Each component type is defined as an object class.

A cooling channel is defined as a continuous straight hole that contains cooling liquid (water in most cases). It can be contained in a single mould component (plates or inserts), or it cuts across several. In this paper, "hole" is used to describe the geometrical shape of a cooling channel on a single mould component, however, its representation is not the same as traditional HOLE features (see the next section). An example of a cooling circuit is shown in Fig. 2. Holes 1–5 are cooling channels. A cooling circuit represents all the inter-connected cooling channels between an inlet and an outlet. Several cooling circuits form a cooling system. In Fig. 2, holes 1–5 jointly form a cooling circuit. A circuit can have several cooling channels with different orientations. These channels consist of cooling holes which are drilled from different faces of the mould plates or inserts. The face used to drill a cooling hole is called the penetrating face. Naturally, a cooling hole has one penetrating face and the hole-drilling vector always leaves from the penetrating face and points to the other end. Usually,

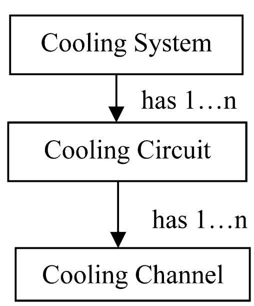

Fig. 1 Semantic structure of a cooling system

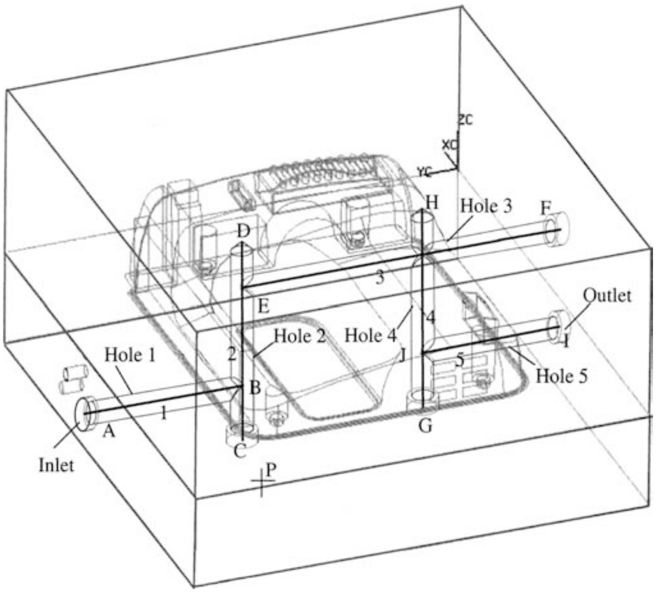

Fig. 2 An example of cooling circuit

cooling holes are perpendicular to the penetrating face. However, in order to cater to some special cases, this constraint is not imposed for the purposes of this article.

In practice, some cooling channels cut across multiple blocks; an example of this is shown in Fig. 3. It consists of several connected collinear cooling holes (hole 1, hole 2 and hole 3). Such channels are specially named collinear cooling channels.

In many cases, multi-impression design is used for the mould layout. There are two approaches to creating cooling circuits then: balanced and unbalanced. A cooling system is referred to as balanced if the same cooling circuit pattern is applied to every impression. Otherwise, the cooling system is unbalanced. Usually, if the mould is designed with a balanced multi-impression pattern [14], and the designer wishes to have an identical cooling circuit for each impression section, then the balanced approach is used. In this case, because each circuit is designed mainly to cover one impression; the cooling effect can be better controlled to satisfy heat-transfer requirements. This is especially recommended for complex moulding parts where the cooling effect can be optimised using simulation packages [11]. With this approach, a CAD function that is commonly required by mould designers is to reflect the changes in the cooling circuit pattern on the individual impressions.

On the other hand, the designer may want to treat the mould as a whole and design cooling circuits without considering the impression pattern; if this is the case he can apply the unbalanced approach.

1.3 Detailed representations

A detailed component structure of a cooling system is given in Fig. 4. A hole is represented with a straight line and an optional cylindrical solid. This straight line is called the cooling guideline for the hole. More precisely, a cooling guideline is a straight-line segment starting from the cooling-hole penetrating centre point to the hole's end centre point. In Fig. 2, AB is the cool guideline for hole 1, and CD is for hole 2. Guidelines contain hole-drilling vectors.

At both the start and end points of each cooling hole, the following types of hole-ends can be selected, as shown in Fig. 5: (1) Drill-through, (2) Counter-bored, (3) Blind without extension and (4) Blind with extension. Such geometrical feature information is represented as

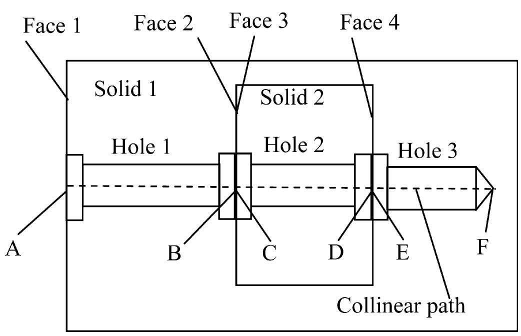

Fig. 3 A typical collinear cooling channel

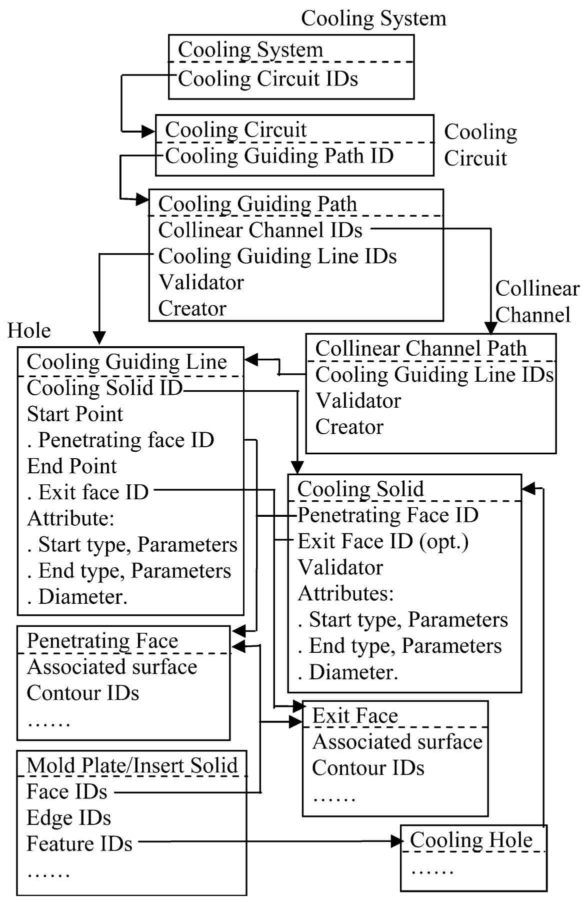

Fig. 4 Detailed component structures of a cooling system

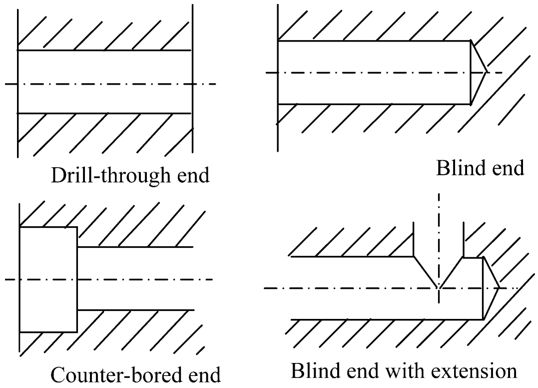

Fig. 5 Types of cooling cylinder ends

attributes attached to guidelines. The cylindrical solids can be generated anytime if it is needed based on the information stored with each guideline.

Traditionally, cooling lines are also used to represent a cooling circuit [11], but they are separate entities from the containing solids, such as mould plates or inserts. One of the design ideas in this paper is that every guideline has start and end point that are associated with the corresponding penetrating and exiting faces, except for the end points of blind holes. Therefore, if these faces change their positions, the associated points can be derived dynamically and updated accordingly. In other words, cooling guidelines are always associated with their penetrating and exiting faces.

The cooling guidelines of all the holes within a cooling circuit are grouped as a guide path. In Fig. 2, five guidelines, AB, CD, EF, GH, and IJ, form a guide path. In this paper, as shown in Fig. 4, a guide path represents each cooling circuit completely while cooling guidelines can have certain attributes to describe the cooling-hole types, diameters, etc.

In fact, cooling cylindrical solids are generated only when needed for viewing, checking physical collision among different features/components, or creating detailed features on plates or inserts. These cooling solids can be deleted to simplify the display; as long as the guide paths are available, these cooling solids can be regenerated. At later stages, after confirming the cooling system design, geometrical holes may be still needed for CAM application or component structure detailing. They can be achieved by subtracting cooling solids from their corresponding plate/insert bodies.

A guide path is also used to maintain connectivity among its guidelines. To validate and verify this condition, a "validator" method is defined in the guide path class. The "collinear cooling channel" is the special object type that is created. From Fig. 4, it can be seen that a cooling circuit may contain such collinear cooling channels as well as simple channels. Each collinear channel can be represented by a group of guidelines called the collinear path. Obviously, its element guidelines must be connected from head to tail continuously along a straight line. In Fig. 3, AB, CD and EF form a collinear path and represent through hole 1 (with both counter-bored ends), through hole 2, and blind hole 3 respectively. It can be seen that within a cooling circuit, cooling elements are associated because they are validated instantly upon any change.

As shown in Fig. 4, the contents and representation of a circuit object change according to the context and user's choices, for example, a circuit can be displayed graphically as a set of inter-connected guidelines, or as a set of cylindrical solids. A cooling circuit is self-contained with geometrical and non-geometrical information in the form of rich attributes.

In summary, with this object structure design, cooling channels and their related mould plates or inserts can be automatically updated if some elements, such as penetrating faces or drilling-hole types are modified at later design stages. Since all the cooling channels are created in an associative approach, then the process knowledge, such as penetrating faces, drilling directions and continuity within a circuit can be embedded within the CAD model and stored persistently.

2 Implementation aspects

2.1 Embedded links and parameters

In a cooling design session with this module, guidelines are initially created through a graphical user interface. To associate the start and end points of every guideline with the penetrating and exiting faces, with the exception of the end points of blind holes, a "smart point" concept is used [13]. A "smart point" is "a point on the surface" associated with the face at the kernel database level. It keeps the persistent link with the corresponding face. Here, word "smart" represents the associative nature of an entity to another related entity. Since guidelines are created based on such smart end points, then the corresponding guidelines are also called smart lines. Each of them is connected to one (for blind holes) or two smart points (for through holes).

A cooling solid cylinder can be generated automatically along its smart guideline by sweeping a circular section profile [13]. For a blind hole, a cone end is added. For a cooling circuit, its cylindrical solids are then united as the solid representation. These geometrical features are represented with attributes attached to guidelines. Such related attributes include type of end (see Fig. 5), cooling hole diameter, and depth and diameter of the counter-bored portion, if applicable. They are used for cooling hole editing and cooling solid regeneration.

2.2 Functions and algorithms

The main functions that have been developed in this module to meet the requirements for cooling system design are listed here:

a. Addition of smart guidelines to form guide paths
b. Modification/repositioning of guidelines
c. Deleting of circuit guide paths
d. Creation of cooling solids
e. Modification of the cooling solids
f. Deletion of the cooling solids
g. Creation of balanced or unbalanced cooling designs for a multi-impression mould.

These functions are briefly described below.

2.3 Creating and editing the smart guide path of a cooling circuit

To create the first guideline of a guide path, the user needs to select a face on an intended solid as the inlet penetrating (planar) face of the circuit (see Fig. 2). A plane equation can be extracted from the selected planar face. The initial start point A for the guide path on this face can be found based on the user's indication point; a smart point is then created. The default direction to generate the first cooling guideline is set to the reverse direction of the face to normal, and it is displayed on the graphic window. The user can interactively modify the initial point position and guideline direction with different submenus activated from the UI shown in Fig. 6. Then, the user can dynamically drag a cooling line or input a value of the length for the guideline of a blind hole, or choose another face to indicate the ending face for a throughhole. In the latter case, another smart point will be created at the end point of the guideline. After creating the first guideline, a sequence number, "1", is displayed near it.

To create the next guideline (see Fig. 2), a drilling vector is required. The user can indicate the bottom penetrating face at point P. Then, the next guideline direction is set to be in the reverse normal direction of the selected face. The vector's start point C is determined with reference to the previous guideline AB and the nearest point to the user's indicated point P. This is an embedded rule implemented in this work. To make the vector definition user-friendlier, many such context "rules" are applied to assist guideline creation. In this case, when defining the guideline CD, from the previous AB, it is extended automatically to find the drilling point C on the bottom face. A smart point is created at C on this face to associate the guideline. Again, sequence number "2" is displayed near the guideline. The user can also define the next guideline vector by selecting one working coordinate direction, +X, −X, +Y, −Y, +Z, or −Z, and then indicate a guideline start point. In the similar manner, a complete guide path can be defined. Upon confirming all the guidelines of the intended guide path, the continuity within the path is checked with the validator method (see Fig. 4). This guide path is then treated as a single entity. As expected, guidelines can be created or added to a guide path by CAD functions. Existing guidelines can also be removed easily.

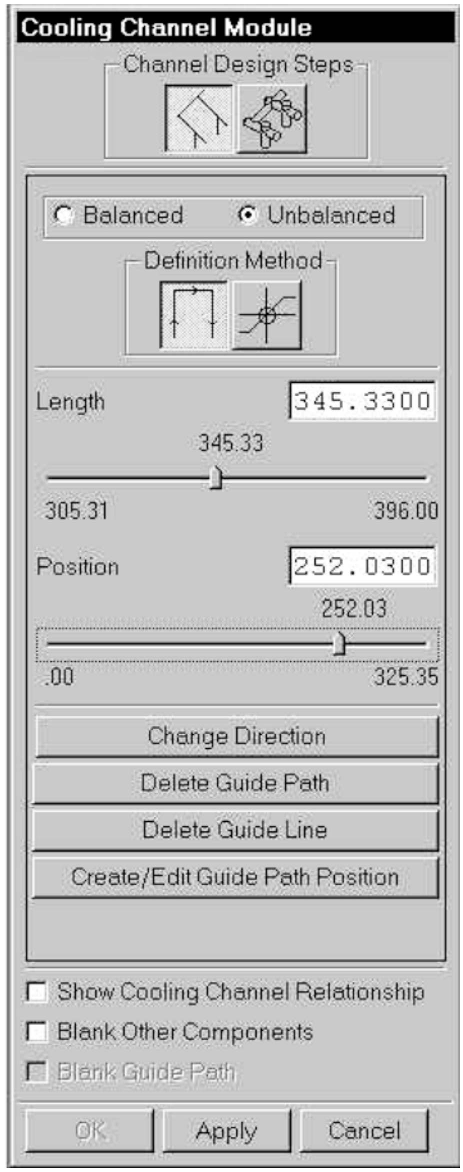

Fig. 6 UI for creating guiding lines

During the interactions to define guidelines, the user's input parameters and sequences are differentiated with the corresponding algorithm branches. For example, to create a simple blind hole, the user's selection sequences can be any of these three options: (a) just the penetrating face, (b) the penetrating face and then an existing perpendicular reference cooling hole, and (c) simply an existing cooling hole collinear to the intended one. Under each option, the user's selection sequences are distinguished; necessary adjustments to the intended cooling line are made to keep the guide path connected, and a friendly UI is designed. After a cooling guideline is selected, its properties, including its length, are displayed on the same UI as shown in Fig. 6. These can be changed and updated. In fact, when a guideline is selected, its guide path is also identified. This is because all the guidelines in a guide path are associated with continuity constraints. If the inlet point position of the guide path is moved, the whole path follows accordingly. A user can securely delete a guide path by selecting the relevant option from the editing UI.

2.4 Creating and editing cooling solids

After defining a guide path, cooling solids can be generated based on the attributes of the individual guidelines. Cooling solids are created only when the user needs them. As shown in Fig. 4, cooling channels can have different hole-types. These types can be represented as the start and the end features for the associated cooling solid. The UI for this purpose is shown in Fig. 7. Initially, the UI settings, such as start type, end type, hole diameter, and other parameters, are assigned with the default types and values preset in a UI configuration file. They are then updated based on the user's input. The values in this configuration file are always overwritten with the user's preferred values when he "accepts" the UI dialog box so that the UI settings can be updated when the user repeats the operation. The entries to different fields of the dialog box are also verified against preset conditions; for example, the value of the counter-bore diameter must be larger than the hole diameter. These checking functions are defined in a method for cooling solids called "validator" (see Fig. 4), which is invoked when the user clicks the "OK" button. If the input is not acceptable by the validator, modification is prompted with some error messages. Once these attributes are confirmed, cooling solids can be generated automatically with CAD API functions by clicking the "show cooling channel relationship" button on the UI.

Cooling solids can be deleted at any time, but the types and parameters still continue to be attached with the individual guidelines as attributes; hence cooling solids can be regenerated and edited anytime. However, if the user deletes the guide path together, then the cooling circuit is deleted completely. In more detail, solid generation algorithms are created for the following six hole types: simple blind, simple through, counter-bored blind, counter-bored at one end and through, counter-bored at the both ends and through, and finally, collinear cooling channels across multiple solids. Other algorithms for editing and deleting cooling channels are straightforward.

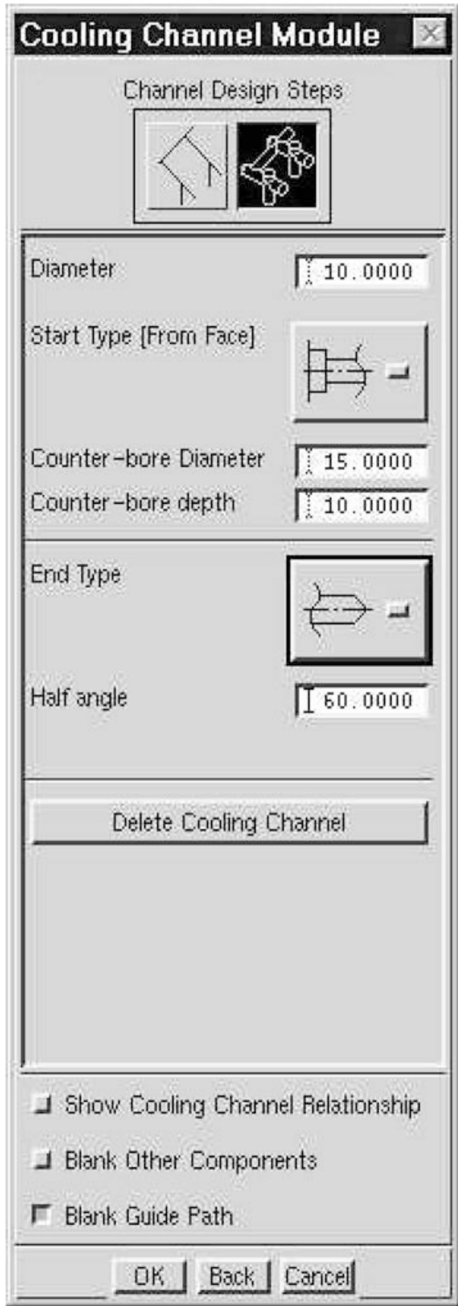

Fig. 7 UI for cooling solid attributes input

For the creation of a collinear cooling channel, association among collinear individual holes is achieved. Fig. 3 illustrates how they are associated. Assuming hole 1 (from left to right) is created via "Create a counter-bored through hole (both ends) by selecting two planar faces" start point A is then "tied" to face 1 and end point B is "tied" to face 2. Note face 1 and face 2 are part of solid 1. Any modification to these faces, such as offsetting them, will affect the depth of the hole.

Creating the middle hole 2 has more flexibility. The user can create it with either of the following two methods. In the first method, face 3 and face 4 (belonging to solid 2) can be selected as the references; hence, start point C and end point D are children of face 3 and face 4, respectively. Because this hole is supposed to be one part of the collinear channel, face 2, which is associated with the end point B of hole 1, is also associated with face 3. This is assured by the validator method of the collinear channel object. Hence, the first hole can slide along face 2 without upsetting the middle hole by creating two misaligned holes. In the second method, the first hole is used as the reference, then the parent of start point C is the end point of the first hole, point B. Due to the link, if the first hole is modified by sliding face 2, the middle hole will follow suit. Once point C is moved, face 3 will be updated as well. This smart association between the two holes creates embedded relations among multiple solids with a collinear channel. Similarly, the third blind hole from left to right in Fig. 3 can be created, and the collinear cooling channel consisting of three associated cooling holes is obtained.

2.5 Dealing with balanced and unbalanced cooling circuits

In this paper, mould components are organised with an assembly tree structure, which is automatically created when the user initialises a new mould design project. The original plastic part is assigned under the top assembly part and is referred to as the product part (prod-part) (see Fig. 8). Impressions are stored under the product part as instantiated components with a layout pattern (core/cavity Inserts). A part that is specially designated for cooling solids is automatically created under the top assembly as well. It is called the cooling line (CL) part.

In order to address this balanced and unbalanced cooling circuit design issue, a waved entities concept [13] must first be introduced. This feature enables geometrical entities, such as solids, faces, lines, points, etc, to be referred associatively from different parts in an assembly tree. This is achieved by copying the entities from one part to another with persistent association. Those copied entities are referred to as waved entities. When a source entity is modified, its corresponding waved entities are automatically updated. The source entities are called prototype entities. Some possible waved faces in an assembly are shown in Fig. 9. Assume prototype face A is in component part 1, it can be waved to create an associative copy, face A_1 in its parent part (child to parent), or face A_2 in component part 2 (child to child). In an assembly-modelling environment, another concept that needs explanation is the work part [13], which is defined as the part where new entities are created. Hence, the user has to explicitly select a work part in order to create new entities in it.

In this paper, when creating balanced cooling circuits, the work part is set to the product part in Fig. 8. When the user selects a face in core/cavity inserts to create a

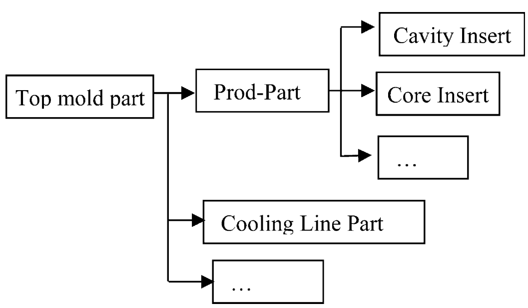

Fig. 8 Cooling line part in the mould assembly tree

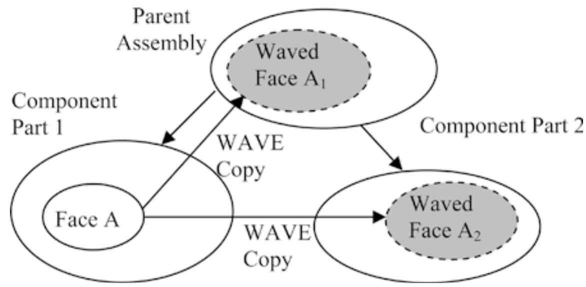

Fig. 9 Two examples of possible waved faces in an assembly

cooling guideline, a waved face (child to parent) is created in the product part. All the entities for cooling, including smart points, guide paths and cooling solids are created in this part as well. At the same time, the related waved guide paths and solids (child to child) are created in the cooling line part. Cooling entities are then copied according to the impression pattern. The resultant cooling system is automatically balanced for different impressions. In Fig. 10, a four-impression example with balanced cooling paths is illustrated.

When creating unbalanced cooling channels, the work part is set to the cooling line part (see Fig. 8). When the user selects a face from insert parts, a waved copy is created in the cooling line part (child to child). Then all related prototype cooling entities, such as smart points, guide paths and cooling solids, are created in the cooling line part. Thus the cooling entities in the cooling line part can be updated automatically if their reference faces in different inserts are changed. It can be appreciated that for both approaches, the assembly tree

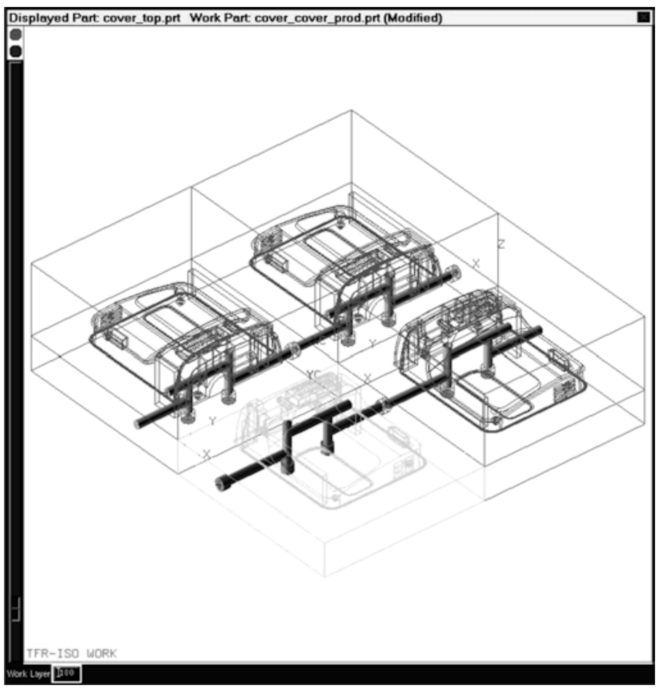

Fig. 10 An example of balanced cooling circuits

structure enables the design modification effort to be largely reduced.

3 Future integration with expert systems

Clearly, the functionality of this module can be expanded further. Due to its object-oriented design, it is highly possible to integrate this module with an expert system that can incorporate design rules for cooling channels. Some of these logical rules were discussed in [10, 11, 15]. The authors believe that this should be a future research direction.

4 Conclusion

This paper presented an associative design approach embedded in a cooling channel design tool. The emphasis was put on the unique guide path and cooling channel solid representations, and on the geometrical association between cooling channels and mould plates or inserts. Compared to the approaches used in [10, 11, 15], this approach has the advantage that the mould designer can accommodate modifications easily throughout the design life cycle. Rich information, including cooling hole drilling direction, orientation and connections among circuit members, is embedded associatively in CAD models. Such information can support interactions at higher level of knowledge rules related to circuit patterns, the closest distance from the moulding surfaces, or collision checking. This approach has enabled effectiveness and efficiency in mould design application.

Acknowledgement This paper is intended to report a research approach only. The authors acknowledge that the research work presented in this paper was mainly carried out when the first author was working in Singapore Institute of Manufacturing Technology (SIMT). A project team of SIMT implemented the software product. R&D engineers from EDS Inc at Cypress, CA, USA, provided close technical support. Unigraphics (UG) and MouldWizard are registered trademarks of EDS Inc.

References

1. CASA/SME, Virtual enterprise integration: creating a sustainable manufacturing life cycle. In: Technology Trend Report 2000,www.sme.org/casa
2. Ong SK, Prombanpong S, Lee KS (1995) An object-oriented approach to computer-aided design of a plastic injection mould. J Int Manuf 6:1–10
3. Irani RK, Kim BH, Dixon JR (1989) Automated injection mould gating design system. Proceedings of the 47th SPE Annual Technical Conference and Exhibition (ANTEC'89), New York, pp 1265–1269
4. Irani RK, Kim BH, Dixon JR (1990) Automated design of the feed system of injection mould. In: Proceedings of manufacturing international '90, part 4, Boston, pp 45–51
5. Irani RK, Kim BH, Dixon JR (1989) Integrating CAE, features and iterative redesign to automate the design of injection moulds. In: Proceedings of the 1989 ASME international computers in engineering conference and exposition, Anaheim, pp 27–33
6. Kishinami T, Kikuchi T, Kanai S, Saito K (1983) Development of interactive mould cavity CAD/CAM system. CIRP Ann 32(11):345–349
7. Fu MW, Fuh JYH, Nee AYC (2001) Core and cavity generation method in injection mould design. Int J Prod 39(1):121–138
8. Salloum G, Charland D, Brouwers X (1986) PLASFEED—an interactive design software of feeding system for injection mould using solid modelling. In: Proceedings of the 44th SPE annual technical conference and exhibition (ANTEC '86), Boston, pp 1412–1418
9. Jong FR, Wang KK (1990) Automatic and optimal design of runner system in injection moulding based on flow simulation. In: Proceedings of the 48^{t}h SPE annual technical conference and exhibition (ANTEC '90), Dallas, pp 385–389
10. Kwon TH, Weeks PA (1988) Expert system aid for intelligent moulding cooling system design. In: Proceedings of the ASME 1988 computers in engineering conference, San Francisco, 1:281–286
11. Wang KK, Wang VW (1987) Computer-aided mould design and manufacturing. In: Isayev AI (ed) Injection and compression moulding fundamentals, Marcel Dekker, pp 607–669
12. Britton GA, Ma Y-S, Tor SB (1999) Object technology development and unigraphics. In: Proceedings of unigraphics user group 1999 spring conference: managing design evolution, New Beach, CA
13. EDS Inc. (1999) User guide for unigraphics v18 (user functions). EDS software documentation
14. Pye RG (1983) Injection mould design, 3rd edn. Longman Group
15. Singh KJ (1987) Design of mould cooling system. In: AI Isayev (ed) Injection and compression moulding fundamentals, Marcel Dekker, pp 567–605
16. Mok CK, Chin KS, Ho JKL (2001) An interactive knowledge-based CAD system for mould design in injection moulding processes. Int J Adv Manuf Technol 17:7–38