An Injection Moulding Strategist in an Information Model Environment

BY

Ronan John Vincent Lee

A Doctoral thesis
Submitted in Partial Fulfilment of the Requirements
for the Award of Doctor of Philosophy of the
Loughborough University of Technology
February 1996

An Injection Moulding Strategist in an Information Model Environment

BY

Ronan John Vincent Lee

A Doctoral thesis
Submitted in Partial Fulfilment of the Requirements for the Award of Doctor of Philosophy of the Loughborough University of Technology
February 1996

DECLARATION

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgments, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a higher degree.

ACKNOWLEDGEMENTS

I would like to acknowledge the assistance of the following people during the period of my Ph.D. research.

Firstly I thank Dr R.I.M. Young without whom I could never have completed this task. Bob has been an excellent supervisor and a friend and I am extremely grateful.

I would like to thank CarnaudMetalbox Technology for their financial support during my research and for their technical assistance with respect to injection moulding.

My sincere thanks go to Jim and Penny White for their part in providing me with a home and many good friends during my time as a subwarden in William Morris Hall.

I have to thank all my colleagues in GG101 for their friendly discussions which assisted my research progress and sometimes provided a welcome distraction. I wish all of these people success in the future. I would also like to thank all other people in the Department of Manufacturing Engineering who have assisted me during my time here.

Enormous thanks go to my parents who have provided me with immense support over the years. I will never be able to express my thanks to them or the extent of my appreciation.

Finally my thanks go to Sue for support, arguments, love and for enduring the disruption to her life brought about by my frequent lack of availability. Thank God for letting me get to the end of this project.

DEDICATION

To my FATHER and MOTHER for
their love and support.

Abstract.

The design of a product is a critical activity for any business, as it has a major influence on the quality and performance of the resulting product. Studies have shown that design can determine 80% of final production costs. In recent years the market place for manufactured goods has become increasingly global and competitive, and this has forced a recognition of the requirement for structural changes in the manufacturing industry of developed countries. Manufacturing companies have been forced to search for more effective and innovative approaches in product design and manufacturing in order to remain competitive. Design for manufacture (DFM) techniques have been developed in order to improve the quality and cost of new products. However, these techniques have reflected the traditional sequential approach to design and manufacture, providing support for individual areas of the product life cycle, eg design for assembly. The power of design for manufacture techniques can be increased when made an integral part of the concurrent engineering philosophy, which involves the simultaneous design of the product and the process to manufacture them.

The author has investigated software support tools to enable concurrent design for injection moulding, and the view is taken that the kernel for a manufacturing system can be defined as the source of product and manufacturing information which must be available for each application. The Product model provides a consistent source of product information as the design evolves, and the Manufacturing model captures information related to manufacturing resources and processes to support product realisation.

In order to explore data driven concurrent design for injection moulding, the functionality and structure of a set of injection moulding strategist applications has been investigated, in order to make use of the Manufacturing model and Product model to provide the designer with design for injection moulding information in the form of feedback advice on part

mouldability and mould design. The author has investigated the feature sets required to support interaction between the strategist applications and translation mechanisms between the features sets to enable design support strategies.

Investigating the provision of an injection moulding strategist, capable of providing support for concurrent product and mould design has led the author to explore the structure of i) a Manufacturing model representation to capture injection moulding process constraints, ii) the representation of the product and the mould in the product model to support the interaction of design and manufacturing applications within the injection moulding strategist, and iii) the representation of design intent in the form of functional constraints in a Functional model based on product ranges.

An experimental injection moulding strategist application has been implemented in an information modelling environment to demonstrate concurrent support for the design of injection moulded products and their moulds, using the Object DB object oriented software. A Manufacturing model based on a Booch and EXPRESS representation has been built in software form to provide a common source of information for a range of interacting strategist applications, supporting a link between the geometry of the product and that of the mould. Functional data has been captured in a Product Range Model based on a Booch and EXPRESS representation to support the linking of function and form. The object oriented methodology of Booch and the EXPRESS language have been shown to complement each other to provide a good prerequisite to the software implementation of an object oriented design support system.

This work has provided a contribution to a structured and extensible approach which should influence future CAE systems structure aiming to support concurrent engineering.

Table of contents.

8. EXAMPLE STRATEGIST SUPPORT FOR THE BUILD UP OF A PTPLUS PRODUCT DESIGN.

9. EXAMPLE STRATEGIST SUPPORT FOR THE BUILD UP OF A FLOWER POT PRODUCT DESIGN.

10. EXAMPLE STRATEGIST SUPPORT FOR THE BUILD UP OF A FLOWER POT PRODUCT AND MOULD DESIGN.

11. EXAMPLE STRATEGIST SUPPORT FOR THE BUILD UP OF A PTPLUS PRODUCT AND MOULD DESIGN.

Chapter 1.
Introduction.

The design of a product is a critical activity for any business, as it has a major influence on the quality and performance of the resulting product. Studies have shown that design can determine 80% of final production costs, Syan (1994). In recent years the market place for manufactured goods has become increasingly global and competitive, and this has forced a recognition of the requirement for structural changes in the manufacturing industry of developed countries. Manufacturing companies have been forced to search for more effective and innovative approaches in product design and manufacturing in order to remain competitive. The traditional sequential approach to design and manufacture, which leads to many iterations, has been unable to support reductions in cost and lead time as companies have been forced to keep reducing development times and sustain improvements in their products and their quality. Design for manufacture (DFM) techniques have been developed in order to improve the quality and cost of new products. However, design for manufacture techniques have reflected the traditional sequential approach to design and manufacture, providing support for individual areas of the product life cycle, eg design for assembly.

The power of design for manufacture techniques can be increased when made an integral part of the concurrent engineering philosophy. Concurrent engineering has been defined (Winner et al (1988)) as 'a systematic approach to the integrated, concurrent design of products and their processes, including manufacturing and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception, through disposal, including quality, cost schedule and user requirements'. The benefits of concurrent engineering are the reduction of the time to market, which has a strategic importance, and allows companies to increase their market share, and the reduction in the number of design changes and design interactions which reduce the overall product cost, and improve manufacturability and quality.

The author has investigated the application of concurrent engineering techniques to the injection moulding process. An interesting aspect of the injection moulded products is that Design for Injection Moulding (DFIM) involves not only the design of the product but also of the mould. Consequently one problem of the injection moulding process is the high start up time. In addition to prototyping of the product, prototyping of the mould is required. Errors in mould design can be extremely costly and time consuming. The complexity of the injection moulding process makes successful production start up on the first attempt a rarity. The successful implementation of concurrent engineering with respect to the design of injection moulded products can therefore provide the participating company with a significant competitive advantage.

Many software tools have been developed in order to aid design decisions. However these tools have tended to address only individual areas of the design cycle, and have not been built to support a range of design for manufacturing activities. In order to provide software support for concurrent engineering a requirement exists for software tools to support the designer through a range of design for manufacture activities as the design is progressing. In order to attain this kind of support for concurrent engineering, the evolving design must be monitored and evaluated continually.

In the context of the injection moulding process a concurrent design support tool must support concurrent product and mould design to achieve support for concurrency. ***The authors research is based upon the thesis that design support applications can be structured within an information modelling environment to provide concurrent support for the design of injection moulded products and their moulds.*** The group of design support applications which provide manufacturability support in Injection Moulded product design has been termed the ***injection moulding strategist or (IMS).***

To achieve this a consistent source of product and manufacturing data must be available. Capturing the representation of information to support the integration of design and manufacturing activities is called information modelling. Product modelling has been accepted as a mechanism to support integration between a range of design and manufacture functions. The Product model provides a basis for a more integrated design and manufacture system by providing a source and repository for all data relating to the product. Hence the Product model allows a range of design and manufacture functions to be integrated by interfacing each to a common data structure. To complete the data integrity that is required to support the design and manufacturing applications, there is a need to represent manufacturing process capabilities and characteristics as well as resources of the processes that may be used in manufacture. This method of data provision will ensure data integration as well as support the functional and manufacturing oriented applications so that their operations can be performed concurrently.

Given that a consistent source of product and manufacturing data can be provided, in the form of a Product and Manufacturing model, the opportunity exists for the development of software tools to provide concurrent support through a range of design and manufacturing activities. Such software tools must enable the interaction of Product model and Manufacturing model information to assess the product's design for manufacturability and provide feedback advice to the designer.

In order to explore the research thesis the objectives of the research can be stated as

- To explore the functionality and structure of an injection moulding strategist application to support the designer in concurrent design of a product and mould
- To understand and provide a representation of injection moulding process capabilities.
- To understand and provide an appropriate representation in a Product model to support integration between a range of interacting design support applications.

- To build an experimental system to explore the injection moulding strategist structure and the information model representations.

This chapter of the thesis sets the work in context for the reader. Chapter 2 provides a literature survey.

Chapter 3 presents the MOSES (Model Object Oriented Simultaneous Engineering System) research project, highlighting its relationship to the authors research. Also discussed in Chapter 3 are the formal methods used in the research and the experimental environment.

Chapter 4 defines the scope of the research in relation to the broad band of issues involved in computer support for injection moulded product design. Chapter 5 describes the authors investigation into the functionality of an injection moulding strategist and the resultant implied structure.

Chapter 6 describes the authors investigation into the feature sets required to support interaction between the strategist applications and Chapter 7 describes the authors investigation into translation mechanisms between the features sets to enable design support strategies.

Chapter 8 describes the authors investigation, based on the features sets identified in Chapter 6, into the structure of i) a Manufacturing model to support the operation of an injection moulding strategist, ii) a Functional model based on product ranges to support the linking of function and form, termed the Product Range Model, and iii) a Product model to represent all viewpoints in injection moulding design, suitable to support design analysis and enable interactions between strategist applications.

Chapter 9 describes the design of an experimental injection moulding strategist application to fulfil the requirements of supporting concurrent design for injection moulding. Chapter 10 explains the experimental work performed by the author to explore support for concurrent design for injection moulding by the injection moulding strategist, and its use of the representations in the Manufacturing model, Product Range Model and Product model.

Finally the research conclusions are provided in Chapter 11, along with recommendations for further work.

Chapter 2.

Literature survey.

2.1. INTRODUCTION.

This chapter surveys the literature of relevance to software support for product design. Section 2.2. discusses the aims of design for manufacture and how this relates to design for injection moulding. The philosophy and importance of concurrent engineering is discussed and software support for concurrency in design.

Section 2.3. discusses the use of features technology in design, and manufacture. The problems of feature based design are discussed and past research into features technology is reviewed.

Section 2.4. discusses the use of data models as an integrating mechanism in software support for design for manufacture. Previous work into Product modelling and Functional models is reviewed and recent work into the use of Manufacturing models.

Section 2.5. reviews previous work into software support for design for injection moulding, looking at general systems for component mouldability and undercut detection, and more specialised software to support gating and feeding system design and design of the mould

2.2. DESIGN FOR MANUFACTURE.

2.2.1. An overview of design for manufacture.

The design of a product is a critical activity for any business, as it has a major influence on the quality and performance of the resulting product. Studies have shown that design can determine 80% of final production costs, Dowlatshahi (1994), Corbet (1986). Design for manufacture (DFM) techniques have been developed in order to improve the quality and cost of new products. There are many descriptions of design for manufacture (DFM), but most refer to particular manufacturing processes, eg design for assembly, Dewhurst and Boothroyd (1987), or design for (injection moulding) mouldability, Cutkosky, Brown, Tenenbaum (1989), etc. However design for manufacture is the practice of designing a product for ease of manufacture, and is not limited to any particular manufacturing process being considered by the designer. Design for manufacture aims to design products for the particular manufacturing process being considered to provide minimum manufacturing cost and time, without compromising the functionality or quality of the product, Ellis (1993)b. Thus if the manufacturing process is machining, the design for manufacture process is design for machinability, or if it is assembly, the process is design for assemblability, etc.

Design for injection moulding (DFIM) differs from other DFM processes because more than one area of manufacturing expertise is required, ie tool engineers and process engineers, Ishii,Hornberger, Liou (1989). DFIM means that the product must be designed for mouldability, and to allow production tooling to reach the desired levels of performance, ie design for mould manufacture and production. As the injection moulding process itself has direct consequences for the performance of the product, which does not occur in the metal removal process, the DFIM process should also take account of the causal effects of the mould on the performance of the product and its mouldability, Huh, Kim (1989).

Conventionally the design process has considered the various design viewpoints sequentially and the manufacturing consequences of functional decisions have not been considered early enough in the design cycle. The functional definition of the product would be refined through interaction with process experts, to achieve a manufacturable product definition. In the traditional design process manufacturability of the product definition is therefore limited by the form of the initial product definition, which is the input to the design for manufacture process.

Increasing competition in the last two decades has meant that companies have been faced with rapid technological change, competitive pressures on quality and cost, and shorter time to market with additional new product features, Syan (1994). Sequential design techniques have no longer been adequate to provide competitive products at the cost and time required in the market place. Simultaneous Engineering (SE) or Concurrent Engineering (CE) seems to be the key to achieving and sustaining a competitive advantage, through the development of high quality functional products that are produced effectively through the synergy of integrated product and process design whilst also considering multiple life cycle factors, such as functionality, serviceability, marketability and recyclability, Al–Ashaab (1994).

Concurrent engineering has been defined by Winner, Pennel, Bertrand and Slusarczuk(1988) as a 'systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset to consider all elements of the product life cycle from concept through disposal, including quality, cost, schedule, and user requirements.'

Sohlenius (1992) describes Concurrent engineering as 'a way of work where the various engineering activities in the product and the production development processes are integrated and performed as much as possible in parallel rather than in sequence'.

It has been identified by Kimura and Suzuki (1989) that concurrent engineering is not just feedback from later activities in the design cycle, but also a means to identify freedom and interdependence among the various activities. Each process can know the effects of its own and other processes (mutual transparency).

In all of the above definitions the importance of addressing different aspects of concurrent engineering systematically are emphasised. In order to facilitate systematic approaches to design considerable research has been carried out into software support for design for manufacture, as described in section 2.2.2.

2.2.2. Software support of design for manufacture.

Historically DFM support software has reflected the sequential approach to design, eg Dewhurst and Boothroyd (1987). However the identification of concurrent engineering as a key technology for competitiveness in the future has led to extensive research into software support in this area. Most of this research has been carried out in acadamic institutions rather than industry, Molina, Al–Ashaab, Ellis, Young, Bell (1995).

Integration of software is a key area of research in which successful results will have a major significance to industrial performance, Corrigal, Lee, Young, Bell (1992), and one area of business particularly influenced by integration is that of design and manufacture. The true aim of DFM is to design a component 'right first time', and so the evolving design must be monitored and evaluated continually, Ellis (1993)b. DFM is a process oriented activity, and to bridge the (DFM) gap involves the simultaneous design of the product and the processes to manufacture them, Cutkosky, Tenenbaum, Muller (1988). Therefore the important task of an intelligent Computer Aided Design (CAD) system would be to make useful and relevant manufacturing knowledge available to the designer early in the design process, Dixon (1988). In such a case the product definition is produced as the output of the disciplines involved in the DFM analysis, rather than as an input to the process.

From the above, one of the important aspects of DFM support is that the design process is an interactive man–machine activity, where the system acts in an advisory role to the designer, who carries out the actual design work, Brown, Cutkosky, Tenenbaum (1991), Cutkosky, Tenenbaum (1990), Corbett, Woodward (1991). For example the design methodology of Corbett and Woodward (1991) was to provide the designer with the appropriate design knowledge as they are working on the product design, in order to alert them to any potential design problems. The information is given to the designer in such a way that he may either use it or discard it, but he may not ignore it. The designer can also ask for explanation of advice.

The designer therefore retains overall control, and the system acts in an advisory role, but with access to the appropriate information. Most of this information could be represented in standard form, for example, condition = insufficient draft, consequence = part sticks in die. These can be mapped onto a knowledge base. If <condition> then consequence. The knowledge therefore appears on the screen as a warning to the designer, who can either dismiss the screen warning, or ask for an explanation and make the appropriate modifications.

Cutkosky, Brown and Tenenbaum (1989) asserted that the designer has to consider too much detail at the start of the design process, and this therefore constrains the designer and limits the options for manufacturing engineers. The proposed solution is for software support to only specify those details that are important at each stage. It was asserted that the problem should be posed in terms of features and constraints relating to design problems. They therefore put forward the idea of using an on line catalogue of existing solutions, for example standard bolts etc, and the use of search in design (or design fragment) libraries, and associated process plans, to be modified for use.

Kahaner and Lu (1993) identified and classified the enabling technogies for concurrent engineering as below:

1. Decision support systems – Tools that support decisions made throughout the product life cycle by providing multi criteria evaluations, simulation or feedback.

2. Virtual teaming support systems – Multi media based tools that enable cooperative working and support team interactions.

3. Concurrent product life cycle systems – Integrated information modelling systems that are capable of capturing complete life cycle information needs.

4. Time responsive intelligent information systems – The systems that support the above by coordinating access to, and the gathering of, information in a timely manner considering user needs.

2.2.3. Knowledge based systems.

Knowledge based systems are a special class of programs that purport to perform, or to assist humans in performing specified intellectual tasks. Such a system contains and uses explicit knowledge – that is, it contains and uses knowledge that is separate from but accessed by its problem solving algorithm, Dixon (1995).

A review paper on 'Computer Aided Support for Simultaneous Engineering' was carried out by some members of the MOSES group in Molina et al (1995). Some of the important aspects of the research reviewed in the paper are detailed below. Molina et al (1995) classified applications for concurrent engineering into 'stand alone simultaneous engineering applications' and 'integrated frameworks'. Stand alone applications are considered as the first generation of concurrent engineering applications. These applications are aimed at specific tasks , but they do not support team-work. Integrated frameworks are considered as the second generation of concurrent engineering applications, in which the support of concurrent engineering team work to achieve product realisation is enhanced through the use of information models and integrated engineering applications. These research efforts have been illustrated in Table 2.1. The Figure has been modified by the author to include researching not identified by the review team but described later in this section..

Kusiak and Larson (1995) and Vujosevic and Kusiak (1991) are developing a framework for concurrent design. The system intends to allow parallel generation of process plans and schedules during feature based part modelling and to assist the designer in the evaluation of a design from different perspectives.

<table>
<tr><th rowspan="3">STAND–ALONE SIMULTANEOUS ENGINEERING APPLICATIONS</th><th colspan="4">INTEGRATED FRAMEWORKS TO SUPPORT SIMULTANEOS ENGINEERING TEAM</th></tr>
<tr><th rowspan="2">INFORMATION MODELS</th><th colspan="2">INTEGRATION ENVIRONMENT</th><th rowspan="2">INTEGRATED ENGINEERING APPLICATIONS</th></tr>
<tr><th>CAE TOOLS INTEGRATION</th><th>COMPUTER SUPPORTED COOPERATIVE WORK (CSCW)</th></tr>
<tr><td>Ishii et al. (LCDL)</td><td colspan="4">Cutkosky et al. (Stanford University)</td></tr>
<tr><td>Kusiak et al. (ISL)</td><td colspan="4">Wu et al. (CCAD)</td></tr>
<tr><td>Rehg et al. (EDRC)</td><td colspan="4">Lu et al. (KBESRL)</td></tr>
<tr><td>Gadh et al (EDRC)</td><td colspan="4">DARPA INITIATIVE IN CONCURRENT ENGINEERING (CERC)</td></tr>
<tr><td>Finger et al. (EDRC)</td><td colspan="2">Kimura (University of Tokyo)</td><td></td><td>Kimura (University of Tokyo)</td></tr>
<tr><td>Bowen et al. (CSD)</td><td colspan="4">ESPRIT Projects</td></tr>
<tr><td>O'Grady et al. (LISIDM)</td><td colspan="2">MOSES (LUT and Leeds)</td><td></td><td>MOSES (LUT and Leeds)</td></tr>
<tr><td>Hambaba et al. (SIT)</td><td colspan="2">Urban et al (Arizona State University)</td><td></td><td></td></tr>
</table>

LCDL – Life Cycle Design Lab, Ohio State University
ISL – Intelligent Systems Laboratory, University of Iowa
CCAD – Center for Computer Aided Design, University of Iowa
KBESRL – Knowledge–Based Engineering Systems Research Laboratory, University of Illinois at Urbana–Champaign
CERC – Concurrent Engineering Research Center

CSD – Computer Science Department, North Caroline State University
LISDM – Laboratory of Intelligent Systems In Design and Manufacturing, North Carolina State University
EDRC – Engineering Design Research Center, Carniege–Mellon University
SIT – Intelligent System Laboratory, Stevens Institute of Technology
LUT – Loughborough University of Technology

Table 2.1	SIMULTANEOUS ENGINEERING RESEARCH	LUT–SE Research Group

Ishii (1991), (1992) and Ishii and Mukherjee (1992) focussed on Design Compatibility analysis (DCA) in order to achieve concurrent design. Research has addressed the need for a knowledge based program that encompasses knowledge of the various life cycle elements, eg functionality, manufacturability etc. A computerised model for design review was developed which utilised DCA. A candidate design is simultaneously evaluated from multiple viewpoints in order to model a 'round table' meeting, where design experts evaluate a design from their perspective viewpoints. The identification of the form, and the development of a systematic method for the application of this knowledge to build a computer model for design aid are the main research objectives. The core of this research is a design representation based on semantic nets, using object oriented representation to capture the life cycle knowledge.

Reng et al (1988) have developed the CASE (Computer Aided Simultaneous Engineering) system to aid in mechanical design, specifically for the problem of manual window regulator design. The CASE system is a framework where different program modules interact to solve the design problem. The program modules are classified as: design agents, design critics and design translators. The CASE software was written in Lucid Common Lisp with Portable Common LOOPS, and uses the vega solid modeller developed in C and a Finite Element software implemented in FORTRAN, Sapossneck et al (1989).

Gadh et al (1989), (1991), and Gadh and Prinz (1995) use a knowledge based approach that handles features and their interactions. The particular focus is on the recognition of features. The system consists of several experts pertaining to specific elements in the product life cycle eg manufacturability. The main interest is providing a manufacturability critic of product created by net shape manufacturing processes, eg injection moulding, casting, extrusion etc. The knowledge in the expert is in the form of rules, based on the part features, their interactions, parameters, material used for manufacture and process conditions. One of the experimental systems is PIMES (Plastic Injection Moulding Expert System) which uses the extracted features from a CAD systems B–rep solid modeller using a feature graph grammar approach. Also the ManuFEATURE system has been developed to recognise complex shape features using the Differential Depth Filter to assess the manufacturability of the moulded part.

Finger et al (1992) are developing software to enable a designer to consider concurrently the interactions and trade offs among different and even conflicting requirements. The system is called 'Design Fusion' and is based on a blackboard architecture that uses a 'heterarchical' control structure. The system architecture provides an interactive environment that facilitates group problem solving between the designer and knowledge based systems. The blackboard is used to provide a multi level, shared, dynamic, domain neutral representation of the design based on a combination of technologies, ie geometric representation using B rep and CSG, feature representation (graph grammar) and constraint networks.The focus is on representing the geometry, features and constraints associated with a design. Concurrent design is supported by enabling the simultaneous consideration of constraints from different life cycle elements.

Bowen and Bahler (1991), (1992), (1993) demonstrated that object based constraint networks are a suitable basis on which to build a language for implementing concurrent engineering applications. A language has been implemented named Galileo3, in which a program is a declarative specification of an object based network and which allows a constraint to be an arbitrary sentence in first order free logic. The use of constraint networks in concurrent engineering applications is attractive because of the capability of such networks to propagate information in any direction. This characteristic allows an application program to disseminate the restrictions triggered by decisions made about different phases of the life cycle.

Young, Greef and O'Grady (1991) developed a constraint based network language called SPARK to implement a wide variety of concurrent engineering applications. SPARK uses frame based inheritance and is built upon an implementation of first order predicate logic. The predicates allow the representation of a collection of constraints which are interconnected via shared variables that have to be satisfied. Constraints satisfaction is made possible by a combination of user inputs and values automatically determined by the system.

Hambaba et al (1992) are developing a hybrid system that aims to automate the design of injection moulded parts and their moulds by using a combination of conventional and neural network approaches. The research integrates these two techniques in a concurrent engineering framework within an object oriented environment. In this system the plastic part features (slab, boss, rib etc) are represented as objects and are geometrically linked to each other. The system architecture consists of a tightly coupled Intelligent Design Model which is itself a rule based system containing general design rules and neural network models to produce the fuzzy parameters associated with objects. A graphical user interface is provided to enable the user to draw a skeletal design of the plastic part using part features. Missing part features are calculated by the intelligent design model. The design model follows the principle of responsibility driven design, where the responsibility is given to the most influential feature of the plastic part. This influential feature is designed first and then the system designs the remaining features. After the design cycle is completed, the features do a self check to determine if all their specification is met. If not the design process is reiterated until an acceptable design is reached or no further changes can be made. This is then considered a failure due to over specification. This work is to be extended to automate the whole process of designing injection moulded parts, including material selection, cost analysis and mould design.

Mantyla (1993), Mantyla et al (1994) and Laakko and Mantyla (1993) are investigating the construction of Open Architecture Concurrent Engineering frameworks. In their view, the existence of product modelling systems should be utilised to provide analysis and simulation tools to aid decisions during the design process. This should be associated with the manufacturing system modelling to achieve a practical concurrent engineering system which connects different views of product life cycle activities. The emphasis is that significant progress toward concurrent engineering is only possible if manufacturing systems models specifically intended for supporting the configuration and operation of concurrent engineering systems can be utilised. This configuration is possible on the basis of a modular collection of concurrent engineering system components and various concurrency enabling techniques.

Cutcosky, Brown and Tenenbaum (1989) (1991) developed an experimental computational framework for concurrent engineering called Next Cut. The system implements the concurrent product and process design methodology that they believe is the best way to achieve design for manufacturability. They considered that the essence of computer aided concurrent engineering is to address manufacturing implications as the design evolves and to give the designer immediate feedback about the manufacturing implications of design decisions. To do this the system had to maintain dependencies between designs and process plans and be able to incrementally modify these in response to design changes. They suggest that process design should be performed incrementally as the product design is being produced. Next cut consists of models and modules which exchange information through a central model which is a knowledge representation, in object oriented format, of the manufacturing environment. Modules are pieces of software consisting of agents (specialised domain modules) and editors (intelligent graphical tools). The system, which relies heavily on specialized modellers and a specialized planner, aims to create a virtual design team in which users, acting through editors are interchangeable.

Next cut supports only single user operation; however, the software has subsequently been incorporated into PACT, Cutcosky et al (1993). PACT is a truly distributed system and is intended to support team based design. PACT is an agent based architecture system. The system consists of four sub systems, NVisage (a distributed knowledge based integration environment), DME (Device Modelling Environment, a model formulation and simulation environment), Next cut and Designworld (a digital electronics design simulation, assembly and testing system). The PACT architecture is an extension to the Open Distributed Processes (ODP) architecture. The architecture aims at facilitating information propagation and collaborative problem solving among software tools in a concurrent engineering environment.

Wu et al (1992) used product modelling as a basis for realising concurrent engineering of mechanical systems by integrating CAE and CAM applications via a shared product definition using a global–local data model scheme. The Product Model is based on PDES/STEP which provides geometric representation (form features, geometry, precision features) and material properties. This PDES/STEP based product definition provides most of the data needed for dynamic analysis, structural analysis, machining process planning, and assembly process planning. Future work is intended to investigate the coordination and interpretation of design changes between CAE and CAM applications. Wu and Choong (1992) have developed a CAE framework based on the DICE architecture, Londono et al (1990), to model engineering processes and capture engineering knowledge for mechanical system design and analysis using an object oriented approach.

Lu et al (1993) focussed on the concept of "Knowledge Processing Technology" and its impact on concurrent engineering applications. The main issue of this research is the development of intelligent computer tools for cooperative team support to improve group productivity. It has been identified that cooperation at the knowledge level is needed to fully support concurrent engineering. The cooperation at the knowledge level requires the exchange and sharing of data and knowledge, and two communication modes (one way batch and multi way interactive). Therefore the research has concentrated on investigating new software technologies and computer tools to enhance and achieve a cooperative team approach at the knowledge level. The challenges that have been identified to address these issues have been categorised as follows:

1. Integration of complementary engineering expertise: support knowledge sharing and expertise integration during product development (eg support multiple data/knowledge representations, integration with CAD and database tools, etc).

2. Cooperation between multiple competing perspectives: effective management of multiple competing perspectives (eg management conflicts, decision histories and rationale, provide comprehensible explanations, etc).

3. Communication of upstream and downstream concerns: enabling and promoting the communication of decisions at early stages of the product development (eg allow early evaluation of decisions, support the least commitment approach in decision making, etc).

4. Coordination of group problem solving activities: support group productivity by means of group interaction of engineering teams with different expertise and geographical locations (eg use of homogeneous and heterogeneous tools, allow centralised or distributed interactions, etc).

Several tools have been implemented for different concurrent engineering tasks and a detailed description can be found in Lu (1992).

The Concurrent Engineering Research Centre (CERC) was established as part of the DARPA Initiative in Concurrent Engineering (DICE). Research at CERC has showed that a computer assisted environment to support concurrent engineering practices requires five generic services:

1. A shared information model formed by a series of models of product, process and organisation. This shared model has been named PPO (Product, Process and Organisation) model, Kinstrey et al (1990).

2. A networked multi media communication environment, Srinivas et al, (1992).

3. Team coordination services that ensure common focus and consistency among people

working in parallel, Nichols (1992).

4. Tool and framework integration to provide a standardised collection of facilities for integrating and exploiting application tools, Kannan et al (1990).

5. Management of design history to support continuous improvement based on the rationale of past decisions and best practices.

The main result is the development of the concurrent engineering testbed (DICE architecture) which demonstrates the capabilities and benefits of the DICE technologies. A "parameter–to–part" system centred on a shared model of the product, the process and the organisation has been implemented together with General Electric Aircraft Engines (GEAE) with focus on the redesign of hollow airfoil fan blades, Kamar (1992). Similar scenarios have been implemented in the domain of printed circuit boards, and tubular and sheet components for heat exchangers, CERC (1993).

Kimura (1991), (1993)a,b, (1994) in the University of Tokyo has taken a model based approach for developing manufacturing system software for Product realisation. The focus of the research is to effectively model all the necessary product behaviour and associated manufacturing processes by computer as precisely as possible, and to predict potential problems for product functionality and manufacturability before making a real system. This is a form of virtual manufacturing. Concurrent engineering is adopted to organise engineering activities because a virtual manufacturing environment requires information whenever and wherever it is necessary. The Product model, Manufacturing Resource model and Physical model are computer executable object models used to represent physically realisable objects. Human activities to manipulate these objects are represented by activity models.

Some research not identified in the MOSES review was that of Urban, Shah and Rogers

(1993). The authors present an architecture for engineering data management. In particular a framework is presented that views an engineering design environment as an integrated, heterogeneous system. This integrated view of data makes use of data exchange standards in PDES/STEP, presenting a framework for the representation of meta data that describes an object oriented view of data and supports an object oriented query language for the retrieval of data from multiple sources. The approach of Urban Shah and Rogers (1993) to data integration is that of defining the logical relationships between the sub parts of a design to provide a view of how all parts of the design are related. An integrated design environment is used to give the designer the impression that all relevant design data is contained in one large database. More importantly it assists the designer in understanding how different aspects of the design fit together to create a composite design, thus providing a basis for the expression of design constraints.

Ishii (1995) describes a hierarchical semantic network called LINKER for the representation of the layout structure of a design. LINKER is comprised of components and sub–assemblies (nodes), and the relationships between the nodes. LINKER allows the designer to evaluate a design for various stages in the life cycle and evaluation methodologies have been identified for analysis of assembly, service and product retirement. The above methods led to a PC based program to support a designer in evaluating a layout design for life cycle costs.

Dixon (1995) describes a knowledge based system for the design of small parts called Dominic. The system uses a guided iteration methodology which has four stages:

a. Formulating the problem.

b. Generating alternative solutions.

c. Evaluating alternatives, and if none is acceptable

d. Redesigning, guided by the results of the evaluation.

This methodology is employed repeatedly as the design proceeds from the concept to the detail design.

2.3 FEATURE BASED DESIGN.

2.3.1. Origin of features.

A CAD modeller can do nothing more than produce visual graphic shapes of an object. However, intelligent design for manufacture CAE systems must reason about the topology and geometry of designed artifacts. Therefore a sufficient representation of geometry is a key issue, Cunningham, Dixon (1988). Since reasoning is in terms of features, representation should also be in terms of features. Geometric features are a very important part of product generation, and typically four categories of feature knowledge are required, Gadh, Prinz (1992): (a) feature parameters eg boss diameter, rib thickness etc, (b) feature relations such as distance and orientation between two features, (c) feature interactions, eg one feature rests on another, (d) topological entities on a feature (faces, edges, vertices).

The generation of the product definition involves not only geometry and topology, but also tolerances and dimensions of the product, plus manufacturing process details and specification. Features to enable adequate reasoning for a DFM CAE system are therefore required to enable representation of this information in addition to the geometry and topology of the product. Features are defined by Cunningham and Dixon (1988) as a 'geometrical form or entity, used for reasoning in one or more design activities, for example, fit, function, manufacture, analysis interfacing, inspectability etc.' Libardi, Dixon and Simmons (1986) defined a feature as 'any geometric form or entity i) Whose presence or dimensions are relevant to one or more CIM functions, or ii) whose availability to designers as a primitive facilitates the design process.' Wierda (1991) defines a feature as 'a partial form or product characteristic that has semantic meaning in the design, process planning, manufacturing, cost engineering or other engineering disciplines'. The third definition most closely reflects the requirement to represent information other than geometry and topology by the acknowledgement that a feature can be a 'product characteristic'.

The features derived for one design activity, for example functionality, will not be the same set as those for another such as manufacturability, ie each design process–activity pair (ie design process in order to perform a given activity) has its own set of features, Cunningham and Dixon (1988). The different types of features used in different design domains stem from the differing points of view associated with those different domains Wierda (1991). The knowledge in the literature about a given process–activity pair is usually in the form of rules and guidelines (heuristics). These heuristics give rules and conditions which should be imposed on the components topology or geometry in a given process, to perform a given activity, Cunningham and Dixon (1988). Features generally have attributes according to type eg 'wall' has the attribute 'thickness'. Heuristics can place restrictions on attribute values. This can assist the designer who must consider the fundamentals of the intended process as it is designed, for example an injection moulded part should not be designed without a die parting line direction in mind. Features therefore originate from the reasoning processes used in the various design and manufacturing activities, Cunningham and Dixon (1988).

2.3.2. Use of features in design and manufacture.

Features are useful in design because they match the level of abstraction, with respect to the design of the product, to the level at which the designer thinks, Wierda (1991). In addition features can serve as units for the storage of data. From a designers point of view therefore it is advantageous to design with features, since he can define an object in terms that match his conception. In addition with respect to process planning, the planner primarily thinks of a part in terms of features, form elements to make, tolerance and surface qualities to obtain. Therefore all generative computer aided process planning (CAPP) systems are based upon feature descriptions of parts. A limitation of conventional CAE systems is the ability (or lack of) to capture designers intent, Dixon (1988). There is no knowledge of why something was done, and therefore no basis for managing constraints later when intention modifications are made. It is considered by Dixon that the use of a features representation will allow intelligent CAE systems to capture this design intent.

Feature based design, involves the use of either feature recognition techniques or the use of 'design with' features. The 'design with' features approach allows the designer to use a library of predefined feature primitives to build up the product. Each feature primitive can be associated with functional or manufacturability constraints regarding the main parameters of the feature, and predefined template plans for CAPP to be joined with other plans once the product is fully defined. One argument for the 'design with' features approach is that the designer can be restricted to using only manufacturable feature primitives. Examples of 'design with' features approaches are, Farris and Knight (1992), Duan Zhou, Lia (1993), Young and Bell (1993), Weirda (1991), Cutcosky, Tenenbaum, Muller (1988), Latif, Boyd, Hannam (1993), Medland and Mullinuex (1993), Torbenau and Lianchum Mo (1993) in machining design, Libardi, Dixon, Simmons (1986) in extrusion design, Luby, Dixon, Simmons (1986), Ishii and Miller (1992) in casting design, Dong, De Vries, Wozny (1991) in fixture design, and Rho, Sheen, Lee (1990), Cunningham and Dixon (1988), Ishii and Hornberger (1991), Irani, Kim, Dixon (1989)(1990), Huh and Kim (1989), in injection moulding design.

Feature recognition involves two stages, one is the decomposition of the product and the other is feature recognition. In order to use feature recognition, the product geometry must be generated before the design can be analysed for manufacturability (or anything else). Feature recognition is a complex procedure and has been shown to be a difficult programming task. Not only is there the problem of recognising individual feature types, but also the appropriate decomposition of the part model.

According to Joshi and Chang (1990) the development of a feature recogniser entails the following problems:

i) Choosing a representation scheme for the features that is suitable for automatic recognition.

ii) Unique definition of the features based on the presentation scheme.

iii) Inference procedure capable of recognition in a complete and consistent manner.

Several different techniques have been developed for recognising features in both 2D and 3D CAD representations:

Syntactic pattern recognition, eg Choi (1982), Choi et al (1984), Jared (1986), (1991) represents a picture by using semantic primitives. A set of grammars consisting of some rewrite rules defines a particular pattern. A parser is then used to apply the grammar to the picture. If the syntax of a picture language agrees with the grammar, the picture can be classified as belonging to the particular pattern class. This is very similar to natural and formal language processing, in which a sentence can be analysed to see if it is grammatically correct. Syntactic pattern recognition is most suitable for two dimensional pattern recognition, Joshi and Chang (1990).

State transition or automata, eg Iwata et al (1980)(1986)(1992), Milaric (1985), is very similar to syntactic pattern recognition. In a system described by Iwata et al (1980), part geometry is described using the SWEEP operators, and/or the UNION of swept volumes. The generating surface is described by ordered pattern primitives, together with technological information. Features are recognised using a state transition diagram where, instead of using grammars and primitives, the relationship between adjacent primitives is used.

Decomposition, eg Armstrong et al (1984), Fridshall (1988), Kim (1992), partitions a design model into several smaller volumes. To be usable the decomposed smaller volumes need to be manufacturing or design features in order that they can be used for process planning. A recognition step is required after the decomposition step to find the semantics of the features.

Expert systems approaches, eg Fagan (1986), Joshi et al (1988), Bond and Chang (1988), are based on the logic used by a human in detecting features. This approach attempts to capture the notion of a feature into some form of rules or logic. Rules are represented in the form of IF–THEN. Rules are created for each feature that needs to be recognised.

Pande and Prabhu (1990) describe an expert system for automatic identification of machines surfaces on symmetrical rotational components. The OPSPLUS expert system shell was used to represent the procedural knowledge to reason and extract the internal and external part features and their dimensions in order to select form tools for machining. All rules are expressed as 'condition–action' pairs.

Lee and Fu (1987) used a CSG approach. Feature recognition systems have mostly favoured the use of boundary representations of solids. Features have been viewed as collections of faces exhibiting certain relationships, and a boundary representation explicitly defines the faces, and makes extraction of topological relationship easy, Joshi and Chang (1990). However Lee and Fu (1987) used a CSG representation for the extraction and unification of manufacturing features. Features considered where chamfer, round, bore neck, created by the union or subtraction of cylindrical primitives. In order to circumvent the problems associated with CSG (redundant primitives and operations) the procedure is based on the use of a CSG tree.

Joshi and Chang (1990) developed a graph based approach for feature recognition. In this approach the boundary representation is transformed into an Attributed Adjacency Graph

(AAG). An AAG is a graph with attributes assigned to each arc. Each face of the part is a node and each edge or face adjacency is shown as an arc. The problem is recognising features graphs from the complete graph.

Case and Gao (1993) carried out a review of features technology. They considered that although features recognition allows the design of parts using existing conventional CAD systems that have sophisticated geometric modelling facilities, there are severe problems. In particular only a limited number of features can be recognised from the solid model, the pattern matching process is complicated, especially for 3D complex objects, and the definition of the features is not precise (the same geometry may be converted into different features by different processing algorithms). They observed that the design by features approach can eliminate the need for feature recognition and gives a unique, predefined feature list with which designer may construct their parts. Also design intent is maintained, whereas it is destroyed during decomposition for recognition techniques. However it is considered present systems still impose limitations on the designer: the design library is finite, and the feature operations, such as add, delete and edit are frequently limited.

Mill, Salmon and Pedley (1993) identified the following truism, In order to recognise a feature one must first have a definition of that feature. If such a definition exists then there is no reason why the designer could not have made use of it in the first instance.

It was observed by Wierda (1991) that at present most programs have to use manual feature identification to create feature libraries, or programs for feature recognition. As feature recognition programs have been shown to be difficult to write, the preferred approach is the use of 'design with' features, ie a set of predefined feature primitives. However 'design with' features are also not without complications:

Two questions have to be asked with respect to 'design with' features,

i) Will the designer be allowed to create his own feature types?

ii) What level of complexity should the features have?

With regard to the first question, from a designers point of view, high level predefined features will speed up his work and allow him to work at his own level of abstraction. However a big problem is that a predefined finite set of features could limit his design possibilities and thereby constrain his way of working. High level features are in themselves an advantage to the designer and would simplify the product model, Wierda (1991), but it would be difficult to create a predefined list of high level features from the point of view of feature derivation. Therefore probably only the most complex forms would be predefined, and the designer would have freedom to develop new complex features from a fairly complete set of lower level types.

From the viewpoint of process planning, features play an important part in CAPP and feature based models are a better starting point than the geometry models. However according to Wierda (1991), feature based models do not solve all the problems:

i) CAPP can handle predefined features, but it will be difficult to handle user defined types, ie those features created by the designer from lower entity types or less complex features. From this viewpoint, the freedom to create user defined features is undesirable.

ii) CAPP usually involves two stages:

1. Generate process plans for all features.

2. Combine these plans to give the optimum sequence for overall optimisation.

Alternative template plans for predefined feature types can be used for the first step. The complexity of the second step depends on the number of features and the number of relations between them. Thus the development of CAPP will be much easier if the level of abstraction of the features is higher, with more known cases and less new combinations to consider.

It is considered by Cunningham and Dixon (1988) that to avoid the difficulty of writing programs to 'extract' the features from the finished object, the product should be defined using 'design with' features to give a primary representation, where a monitor system should ensure that all operations requested by the designer to build the object are allowable and understandable by the system. In this way the manufacturability, functionality etc of the object can be examined at any stage of completion, unlike when using feature extraction, where it is only possible when the system can detect recognisable features upon which to base any analysis. In addition the possibility of monitoring every operation requested by the designer allows secondary representations to be created from the primary. Secondary representations convert the primary to that representation required by the various design and process planning functions.

Domazet (1992) suggested that a potential solution to the problem of different features for different design problems, eg functionality, manufacturability etc, is to define a library of design features, containing only those that can be manufactured, inspected etc on a specific facility. The problems with the approach include the restriction to individual manufacturing or inspection facilities etc, plus the problem of derivation of an appropriate set of features.

Allada and Anand (1995) carried out a review of feature technology and concluded that the problems which have eluded researchers and remain yet to be completely formalised are

i) feature interactions resulting in non standard features.

ii) feature relations.

Allada and Anand assert that a widely held view among experts is that the features based system architecture should be a blend of a design by features approach and automatic feature recognition

Gadh and Prinz (1995) presented an approach to resolving the problem of feature interactions based on recognising feature types using higher level entities known as loops. This approach enabled the recognition of feature types with shared faces. The limitations of the system are that it cannot cope with features that gradually blend into one another, ie with no sharp corners between them. Also recognition is limited to parts with no fillet or corner radii.

Due to the intractability of feature recognition and the inflexibility of 'design with' features some hybrid approaches have been advocated. Pratt (1993) suggested that a designer could indicate to create a feature of a given type. The system would prompt for the appropriate dimensional information and then instantiate a feature of the required type with the required dimensions. The feature is initially a separate entity until the designer indicates the required position and orientation on the main part model. The system would then position the feature and unite it with the overall model. This approach would avoid the necessity for model decomposition, and allow each feature to be validated for manufacturability etc before being joined to the model. However the product has still to be constructed using relatively inflexible predefined feature primitives.

Another suggested hybrid approach, Lee (1992) was to allow the designer to create his own geometry (feature or group of features) which could be recognised before being joined to the main model. This would negate the problem of removing individual features from a completed model and allow the design process to become more concurrent compared with the conventional feature recognition approach. Different design considerations may wish to recognise that geometry at different times, depending on how much geometry has been created and when it becomes recognisable as a feature or group of features for that viewpoint. In this way the power of 'design with' features can be combined with the flexibility of feature recognition, whereby the designer is not unduly constrained regarding the geometry he uses to build up the model.

Zu, De Pennington and Saia (1993) combined the methodology of design with features and feature recognition as the underlying philosophy of their research. A graph grammar approach is presented for feature representation and transformation. Geometric constraints are used to define features symbolically and to define relative position and orientation of features, which are integrated with geometric models. A graph grammar is generated for representing and manipulating features and geometric constraints. Feature reasoning is then achieved by incorporating graph grammar parsing with knowledge based techniques.

De Martino et al (1994) present a hybrid approach based on an intermediate model, which is shape feature based and provides a communication link between design by features and feature recognition. The system provides the design with the possibility of generating the product feature–based description using both features and geometry primitives which are subsequently used to create a feature based model. The system also provides the possibility of creating application specific feature taxonomies to map feature based descriptions between different application contexts. Conversion mechanisms map a geometric model onto the intermediate model, and from this representation to a context dependant feature based model, and vice versa.

Laakko and Mantyla (1993) have implemented a feature modelling system based on a hybrid feature recognition and design by feature system. The designer has the flexibility to use either of two approaches while designing the Product model. The system is based on a feature recogniser that provides 'incremental feature recognition' which allows changes to a geometric model to be recognised as new or modified features while preserving previously recognised features that remain unchanged in the geometric model.

2.3.3. Features for injection moulding.

Hanada and Leifer (1989) aimed to implement a knowledge based system to represent injection moulded parts and mould design knowledge. They established a process–function analysis method for derivation of the appropriate features for the feature based model. It was considered that for a well structured design system, the following issues should be addressed, i) what information is required at the design stage, ii) how that information is communicated between design stages, iii) what the user requires of the injection moulding. Their model consisted of several design stages, the first of which, the skeleton stage, extracted the user requirements as specifications. At the following stage, the product design stage, features would then be added with regard to functionality and the injection moulding process. In the mould design stage, the mould manufacturing process would be considered. Between the stages the authors approach was to have a 'design translator' which would act upon the designed objects. This design translator had the knowledge to transform the features for one design domain into those for another.

The process–function method used to develop the required features consisted of five stages:

i) Divide the basic function, ie make an injection moulded product, into sub functions.

ii) Ideas should be generated to the lowest level of sub functions, ie where no more sub division of the basic function is possible.

iii) After the ideas have been generated these can be evaluated using a filter. Non appropriate ideas are eliminated using evaluation by constraints, either user defined, physical or manufacturing. This stage should leave the user with a selection of feasible sub function ideas, which when combined can be used to generate the product. There may be more than one possible combination of ideas to do this.

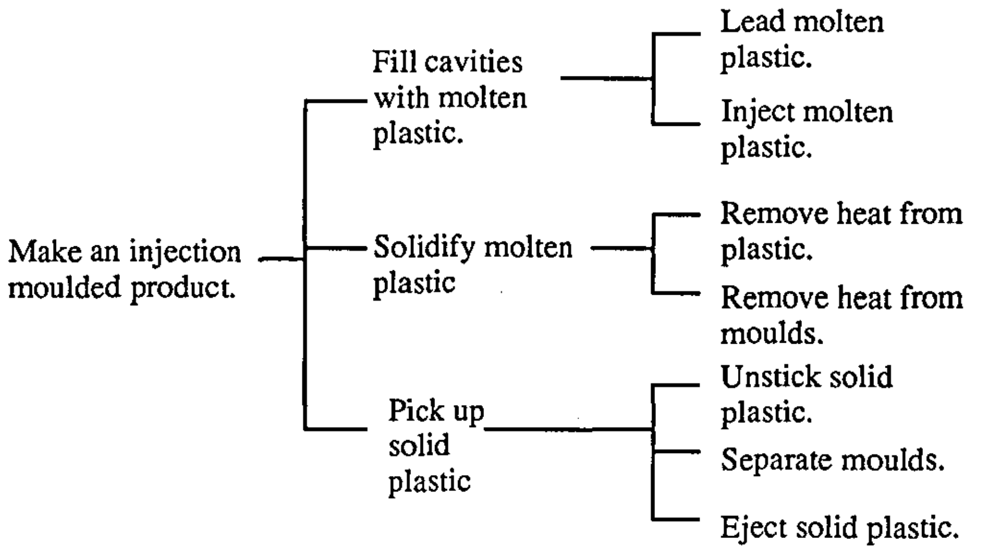


Figure 2.1. Injection moulding function tree.

iv) After selecting the best ideas, these would then be combined to give the entire product, (reverse of stage i)).

v) Finally the different combinations (if more than one) are evaluated using another filter, where the criteria are performance and cost. If performance is the same for two then cost is the priority.

It was considered by Hanada and Leifer that by applying the above process–function analysis method that it was possible to systematically derive the features and create the knowledge base for product/process design. As an example Figure 2.1 shows the injection moulding process tree. The bottom elemental sub functions would then be implemented by some concrete design solutions, for example 'unstick plastic from mould' would require one to consider draft angle in the mould, the finish on the mould surface and/or adding a slipping agent to the plastic.

The design stages are all linked to one another. They have a hierarchical relationship, and the properties of the skeleton features are inherited by the product form features. The skeleton features describe the users specification completely. For example, the outline feature includes outside shapes specified by the user. The product form features are SLAB, BOSS, HOLE, RIB, JOINT and RELATION. Using these features the part model is created in the knowledge based environment. SLAB means a thin wall of the part, and it has a close relation to the OUTLINE feature in the skeleton stage.

Rho, Sheen and Lee (1990) attempted to identify the machining features associated with the components of mould dies and thence to implement the related information into a data structure to be used for general purpose planning functions, including estimation of machining time. Machining features were classified into three areas i) header information, ii) functional features, and iii) cavity core features. Header features were attributes for management such as assembly hierarchy, due date etc, and were thus used to retrieve information for one mould die. Functional features included runners, gates, cooling passages, pockets and holes. These five functional features were analysed and classified according to attributes of shape, size, orientation and machining accuracy. For cavity/core features the main cavity shape was divided and classed as either 3D, 2.5D or cylindrical shapes in order to estimate the rough machining time. The fine cut was estimated later using the process planner. The 3D and 2.5D shapes could be expressed as 'pockets'.

Kang, Park and Hong (1992), and Kang and Kim (1990) describe a CIM system for mould design and manufacture used in a mould manufacturing company. Using this system the basic product geometry is built up to the final product by adding secondary geometry such as ribs etc and defining the parameters of these secondary features to the system. Based on this geometry, the geometry and feature data for the mould is generated. The system generates the features from the fact that most mould components are standardised except core and cavity parts. Two types of feature are defined, functional features and atomic features. The standard components which can be generated parametrically are transformed to functional features, and the remainder to atomic features. The atomic features are primitives such as pockets, holes, slots, steps, grooves, islands, surfaces etc. Twenty seven different atomic features are defined and used. The parameters of atomic features are type, position and orientation, and size.

Ishii and Hornberger (1991) linked process review modules to an AI program called DAISIE. One of the modules was to evaluate injection moulded parts. This system is described in more detail in section 2.5.1. The features used in the system were walls, which were added by the designer to build up the main geometry, and a set of predefined shapes often found in injection moulded products, such as ribs and bosses, but also standard clips and slots for click fitting panels etc. These features were all selected from a predefined menu.

Dixon (1988) defined 'design with' features sets for many processes including extrusion, forging, stamping, casting, assembly and injection moulding. Three sets of features were defined for injection moulding, for manufacturability analysis, for tool and die design and for cost analysis. Feature types can be classified into five types, *primitives, intersections, add–ons, macros* and *whole forms. Primitive* features are primary building blocks, eg walls. These can be combined using intersections. *Add–ons* are features that provide some local function and are usually added to a primitive, eg bosses, *Intersections* specify the way that primitive and add–on features meet, eg 2D corner. *Whole form* features apply to an entire product, or at least a significant region, eg hollowness. *Macros* are pre–specified combinations of primitives, eg boxes.

Huh and Kim (1989) describe an expert system to support the design of injection moulded parts called RIBBER. The system allows the designer to build up the main geometry using wall features, and allows the addition of ribs, bosses and gate features by selecting primitives from a menu. The system uses empirical formulae to consider the mouldability of the designed part geometry as it is built up.

Two authors described feature based systems for the design of cast components, many of the characteristics of which could be translated to injection moulding:

Luby, Dixon and Simmons (1986) describe a system for the feature based design of castings. The design method was to pick from a screen menu of macro features such as slab, corner, L brackets etc, and co–features such as bosses, holes and ribs etc. It was considered that in order to support design for manufacture, the geometrical database must represent explicitly features in terms which correspond to the available manufacturing knowledge, and also must represent connectivity among those features. The system, called CASPER took its design rule checks from the casting design handbook, and the 'design rules slot' of each feature contained these tests.

Corbett and Woodward (1991) describe a 'design with' features system for die casting. The appropriate features were chosen by referring to a knowledge handbook. The system user selects a feature from the screen menu and enters suitable parametric attributes to fully define it. Boolean cut and paste operations are used to build the final representation. The component feature information is interfaced to an analysis system which accesses a knowledge base containing design for die cast rules. Features used for building component geometry were plates, walls, surfaces, bosses, holes, ribs, webs and ejector points.

2.4. INFORMATION SUPPORT SYSTEMS.

2.4.1. Use of data models to support design for manufacture.

Simultaneous engineering has been facilitated within many organisations by the establishment of multi–disciplinary teams to support new product development, Ellis, Young, Bell (1993)a. The teams use tools such as computer aided design, finite element analysis and process planning packages to support areas of design and manufacture. These software tools work well in isolation but the transfer of cross functional information between them, in a way that ensures data integrity, is difficult, McKay and Bloor (1991). According to Ellis et al (1993)a, understanding the information and information structures required in simultaneous engineering is the key to resolving this problem, and hence will provide a major step forward in the development of software tools to benefit product design and manufacture. There is a considerable amount of research being undertaken by the STEP/PDES community, PDES/STEP (1988), into developing a mechanism capable of describing product data throughout the life of a product and which is also independent from any particular computer system. STEP intends to enable effective integration by ensuring that the range of software applications within a CAE system address a consistent source of product data.

It is asserted by Ellis, Young, Bell (1993)b, that the true aim of DFM is to design a component 'right first time' and so the evolving design must be monitored and evaluated continually. For this to be achievable a consistent source of both product and manufacturing information must be available. Data model based CAE systems can benefit DFM by providing the most up to date and consistent product, process and resource data to both software tools and members of the simultaneous engineering team. The prerequisite to a system to provide early manufacturing knowledge is the development of a database that serves the information needs of the design and manufacture spectrum, ie not only design but also manufacturing evaluation, Libardi, Dixon,Simmons (1986). Data model Computer Aided Engineering (CAE) systems can facilitate the simultaneous design of the product and process by providing an interactive knowledge base that relates up to date product data to process constraints.

For really achieving concurrency of engineering activities to produce innovative products, it is considered by Kimura and Suzuki (1989) that a new framework or infrastructure is required for engineering information processing. It is asserted that instead of conventional CAD systems, a more powerful information representation is required which is understandable to both product designers and manufacturing engineers. The central concept for such a representation is *Product modelling*, Kimura and Suzuki (1989). Product modelling is defined by Kimura (1993)b, (1994) as a 'modelling framework which can capture and represent all necessary information through the whole life cycle of our products, from initial product planning until maintenance'. The approach of Kimura to product modelling is described later in this section, and his approach to manufacturing modelling in the next section.

Product modelling is being accepted as a mechanism by which a central source of product data can be captured to provide an integrating mechanism between a range of design and manufacture functions, Corrigal, Lee, Young, Bell (1992). With respect to the integration of design and manufacture, the benefits of shorter lead time and higher quality can be achieved through the introduction of advanced CAD systems. However it is considered by Corrigal et al (1992) that even greater benefits can come from making more use of the product data which is both generated and required by design and manufacture functions. Corrigal et al (1992) described the use of the product model environment as an integrating mechanism between a range of design and manufacture functions. The Product model concept is shown to provide a basis for a more integrated design and manufacture system by providing source and repository for all data relating to the product. Hence the Product model allows a range of design and manufacture functions to be integrated, by interfacing each to a common data structure or product data model.

It was identified by Corrigal et al (1992) that component data models must capture data relating to i) specification of component, ii) how the component was defined (ie engineering drawings, process plan etc), iii) actual components that have been made, ie quality information such as component measurements against the specification. It was identified that the issues involved in the provision of a product data model relate to the design functions that must be performed, their interactions and the data structures that can capture their requirements.

Linberg (1992),(1993) viewed DFM as a flow of data whereby the customer specification is transformed into manufacturing documentation. Linberg concentrated on the role of the product data model (PDM) structure in the flow of technical data. A general CAE system architecture is suggested founded upon a set of independent but cooperating subsystems, called functional systems (FS's). Each has a common architecture and communication with each other, and data handled by the system would be much the same as in the formal company documentation system. Documents would be tied to one level of the PDM by a parts list. The PDM would describe the consisting parts, their attributes and their relations, and the key elements are the 'part' and 'occurrence' entities, which model the parts themselves and any occurrences within the product structure, which could be more than once. Linberg notes that the general requirement is to view the product from different design perspectives and represent each perspective in a separate structure. It is hoped therefore to use the concept of FS's so that the original part structure can be taken apart and reassembled to get a new structure, which is appropriate for each perspective.

It is considered by Kjellberg and Schmekel (1992) that the product development chain must have allowance for many different data models to exist in parallel to allow product analysis from different design perspectives. These must be able to be updated, interrogated, and maintain consistent relations with each other. The PDM is thus a collection of different models with consistent relations between them. Kjellberg and Schmekels product representation is based on structures of functions, physical principles and solution principles. The structure of the functions specifies the product functionality, the structure of the physical principles specifies the physical behaviour of the product, and the solution principles specifies the technical solution which has the specified behaviour and fulfils the specified functionality. The set of relations between the different product models can be defined between engineering concepts, for example lines, vertices, edges etc, in the different models, representing equality, similarity, dependency or inconsistency between concepts. It is considered very important to handle dependency relations between models effectively. These relations would normally be expressed as a constraint which must be obeyed by all specified dependences, and therefore changes to one model must not be permitted without modifying the connections to other models, otherwise a conflict might occur.

Winjard, Carlberg, Kjellberg (1992) proposed a Product model structure divided into two parts, with a well developed interface between the two. One part is the 'geometric model' which contains an explicit description of shape. The other part which they have called the 'technological model' contained descriptions of all application specific terms, the relationships between them and the functions applicable to them. The geometric model is made up of a series of form feature entities which group together topological entities. Each feature element has an association into the technological model. This provides the form feature with a link to the functional requirements of the product, or to a particular requirement to be manufactured using a particular process etc. Thus the technological elements in the technological model represent the other characteristics of a feature than the explicit shape.

Winjard et al (1992) considered that there was a requirement in the Product model for a least three levels. The first level in their model was the 'meta class' level, describing a general solid model with form feature entities, a general technological model consisting of technological elements, attributes, relations and functions, and the coupling between the geometric and technological model parts. Then there is a 'class' level defining the different classes of technological elements and form feature entities. The lowest level is the 'instance' level, where instances of class descriptions are created and related, and the actual geometric model is also created. Winjard et al (1992) consider that the above is the basic strategy followed in all schemas being defined in STEP, which is first to define the application independent basic resource models, and then develop separate application protocols utilising a specialisation of these basic resource models.

Mantyla et al (1994) put forward a Product model structure consisting of feature instances and data representing the topological relationships between them. The topological relations amongst the feature instances are captured in a tree structure. Explicit surface frames are used to represent topological relationships. For example for a hole on a bottom face, an explicit data structure associates the hole with the specific face.

Product modelling research was carried out at Leeds and Loughborough University called

Information Support Systems for design and manufacture (ISS). The research was funded to investigate Product modelling environments and their benefits for the support of the integration of automated design and manufacturing applications, Corrigal et al (1992). Corrigal et al (1992) asserted the requirement for a tool to formalise the product data model and suggested the EXPRESS language to describe data structures.

In Kimuras approach (1993)b, (1994), the Product model is 'a framework which can capture all the necessary product information through the whole life cycle of our products. It represents target products, their materials and intermediate products. tools machines, and any other manufacturing resources and environmental objects'. In Kimuras definition of concurrency, (above section 2.1.1.), he identified the importance of processes being 'mutually transparent'. It was considered that the Product model plays a 'central role' for realising mutual transparency among respective activities.

In all of the above work the Product model is a passive database, where design support applications carry out operations on objects in the database. Domazet et al (1995) put forward an approach called Active Data Driven Design (ADDD), which considers data objects in a database as active rather than passive, capable of reacting to events that are of significance to the product status. Using this methodology events raised in the Product model are used to trigger automatic design changes or applications participating in a collabourative or concurrent product development process. If a Product model contains elements (active data) that can specify the generation of an appropriate signal whenever some database event is detected, then it can be used as a valuable piece of information for controlling the product development process. Instead of only using process models, this feedback can also be used for choosing the next step in the product development process. Therefore a designer or design agent modification to the Product model can trigger further automatic changes to the Product model or can activate other design agents before re-activation of the former design agent and/or further designer changes.

2.4.2. Use of functional models to support product design.

Kimura and Suzuki examined the issue of the representation of 'design intent' within their Product modelling research, Kimura and Suzuki (1989), (1995). According to Kimura and Suzuki, most research in the area of Product modelling has dealt with the process of how to generate such product information from the initial requirements for products. It is considered that for achieving efficient product design, design intent is one of the most important areas of information, and is considered to play a fundamental role in concurrency. Design intent is defined as the designer expression of their original objectives of the design, so that manufacturing engineers can well understand the process of product design and attain good manufacturability design without invalidating the performance of products. Design intent is considered in terms of a series of logical constraints and a formal framework is introduced to deal with design intent. Design considerations could be converted into the form of geometric constraints, and by solving the geometric constraints, it was possible to determine which are satisfactory for design requirements. ''Transparency' of this functional design process for other areas such as manufacture, via the Product model, enabled the manufacturing consequences of functional decisions to be considered, thereby allowing concurrency in design.

Chandrasekaran et al (1993) considered the use of a functional representation to capture how a device works or is intended to work. Specifically to capture design rationale, which is defined as 'the body of information that explicitly records the design activity and the reasons for making choices'. A top down approach is used in the representation, in that the overall function is described first and the behaviour of each component is described in the context of this function.

Schmekel (1989) identified that if a product is specified in a formal specification, then the functional specifications can be described in 'functional models', and be analysed and simulated for the required functionality of the product. Smeckel investigated the representation of functional models in terms of functional objects, such that functional requirements of products can be modelled. A functional objective is defined as 'a symbolic representation of a functional description'. It describes the functional requirement of a product such that a structure of functional objects describe a functional model of the product. Smekel asserted that a product specification describes the functionality of the intended product plus the constraints on the product from what is known about the manufacturing methods, price etc, which direct the development of the design. Some of the product specifications deal with functional requirements on a product and constraints on the functional requirements. This information can be used to compose a functional description of the product, ie a functional model.

Schulte and Weber (1993) considered that in order to develop future intelligent CAD systems, the relationship between function and shape had to be clarified because functional information is necessary for services other than just handling geometric data (eg value analysis, automatic classification of engineering design etc). They considered that the term function is used in design methodology in such a way that it often stays on a too abstract level for such applications. Thus an additional class of more concrete functions (technical functions) has to be introduced. In addition intelligent CAD systems have to process functional information together with geometrical data. Therefore an extensive understanding of the relationship between function and shape is absolutely necessary. Schult and Weber describe two approaches to identify the relationship between function and shape so that this can be modelled in a design support system. They advocate features based design in intelligent CAD systems, and consider that features for functional design must be a combination of function and form. Therefore it is considered that the functional feature object must have the following components:

i) A feature must be mappable to a generic shape, ie it has a syntax. Thus one component of a functional feature is a specific shape element.

ii) A feature must have a specific meaning within the engineering context. Therefore it represents a semantic. Consequently the second component of a feature must be specific semantic elements to express the features functional meaning.

iii) The third component of a functional feature must be the relations between the two said components.

Chachrabarti (1993) carried out a review of functional design methodologies. He considered that reasoning about design problems and solutions in terms of their functionality, termed here as functional reasoning, is central to designing. He concluded that functional reasoning support in CAE systems should include supporting representations in terms of function and form. Supporting functional reasoning is considered to require an understanding of how function and form are created, how they can be used to represent design problems and solutions (as part of a product representation), and how one is generated from the other.

McGinnis and Ullman (1992) investigated the interplay of design objects, features, constraints and design decisions in order to identify the structures within a functional model of design. A design terminology is developed based on empirical data extracted from verbal design protocols. Constraints were identified in the protocol and classified according to source, level of abstraction, and form or function orientation. A structure was also developed which decomposes the constraints and classifies them into ten basic feature relationships. This work was the basis for the building of a formal feature/constraint representation.

2.4.3. Use of a Manufacturing model to support design for manufacture.

An early reference to Manufacturing modelling is in the IMPPACT project documentation, IMPPACT, (1991). IMPPACT defines a factory model which contains structured information on facilities such as machine tools, jigs and fixtures, tools, robots etc, and a process model which contains information on production activities and operations.

Chen and Wallace (1993) investigated the requirements of timely provision of manufacturing information to support manufacturing decisions during the early stages of design. The researchers attempted to identify the type of manufacturing information required to achieve increased manufacturability. Manufacturing information was categorised into direct and indirect information. Direct manufacturing information assists the designer in defining parts and is divided into project, material and process information. Indirect manufacturing information is to assist a designer in evaluating and modifying designs to improve manufacturability, and is divided into budget, guidance and diagnostic data. Software was developed to support the above research.

According to Corrigal et al (1992), 'a flexible representation of machine capability and process characteristics is needed in future integration systems to complement the Product model. Such a Manufacturing model would allow a range of closely coupled applications to assess common manufacturing process information as well as common product data. This introduces the possibility that functions could be performed not only in a totally integrated environment but also simultaneously'

Al–Ashaab and Young (1992), (1994) considered that in addition to product data, designers needed access to a range of information describing the characteristics and capabilities of the manufacturing process. They therefore considered that in addition to the Product model there should be the parallel concept of the Manufacturing model. It is asserted that in combination, the Product and Manufacturing models can provide the key sources of information to enable concurrency in design for function and design for manufacture. The EXPRESS information modelling language was used to capture the underlying structure of a Manufacturing model for the injection moulding process. The implemented Manufacturing model supported interacting design applications from multiple viewpoints, ie design for mouldability, mould design and moulding machine calculation. It was considered that captured manufacturing information in the Manufacturing model had to be structured in such a way that could aid the design of the product. The structure of a Manufacturing model should provide easy access to information in an unambiguous representation, and be updatable to suit changes in user requirements. According to Al–Ashaab and Young, the Manufacturing model must also capture any relationships between data contained in the model and capture the behavioural aspects of the process being considered.

In his work on virtual manufacturing, Kimura (1994) considers that although a lot of work has been carried out on Product modelling, there are still many problems with representing constraints on the product design. For this purpose it is asserted that a 'process model' is required. Kimura considered that not only functionality of products but also the performance of every necessary manufacturing process should be modelled. According to Kimura, the relationship of the process model with the Product model is as follows: Reference data or data structures are stored in Product models, and appropriate process models are invoked automatically or with human intervention. Kimura defined three different types of process modelling:

i) declarative method: derivation of results from given conditions or constraints.

ii) procedural methods: computing the result by a set of programs.

iii) computational methods: calculating the result by a set of given expressions.

Molina et al (1992) described a facility model which is able to represent product and material handling systems, and the relationships between them according to a specific flexible manufacturing system. The model is used for the design and evaluation of flexible manufacturing cells in a concurrent environment for FMS design.

Molina et al (1994)b created a Manufacturing model consisting of three information entities, namely: manufacturing resources, processes and strategies. The Manufacturing model has four levels based on a de–facto standard, (ie factory, shop, cell, station). The information model has been modelled in EXPRESS and developed using the Booch Object Oriented methodology. The model structure enables the effective representation of both manufacturing resources and processes, together with their capabilities. The taxonomies of resources and processes developed are generic enough to allow the representation of modern technologies, eg complex turning centres and related manufacturing processes.

2.4.4. Reference architectures to support integration.

In order to achieve the goals of CIM architectural guidelines are required to define an integrated methodology to support all phases of the CIM system life cycle from requirements specification, through system design, implementation, operation and maintenance, Al–Ashaab (1994). A reference architecture defines the appropriate rules and structural guidelines when integrating the constituent parts of a CIM system during its design and building process. Among others the following ESPRIT projects have undertaken research in the development of enabling technologies for CIM:

IMPPACT (Integrated Modelling of Product and Processes using Advanced Computer Technologies) ESPRIT Project No. 2165 has developed systems that allow the integration pertaining to product design, process and operation planning, including machine control data generation using product and process models, Bjorke and Mykebust (1992).

CIM–OSA (CIM–Open System Architecture) ESPRIT Project No. 688 is a CIM architecture framework for the design, development and implementation of a CIM system, BSI DD194 (1990), Kosanke (1992). CIMOSA consists of a Modelling framework and a supporting distributed environment to represent different views of an enterprise. The views are classed as the function, information, resource and organisation views in a consistent model.

The Open Distributed Processing (ODP) group of the ISO is defining a Reference Model after recognising the need for the standardisation of distributed systems. This RM–ODP considers a distributed system from different viewpoints, each of which is chosen to reflect one element of system design. These viewpoints are the enterprise information, computational and technical viewpoints, Linnington (1992). The RM–ODP is intended to provide a governing set of guidelines that will illustrate the potential interactions between different information system architectures for computer based information systems. EXPRESS is the adopted language of the information viewpoint of the RM–ODP architecture and Booch is the representation for the computational viewpoint.

Molina et al (1993) investigated the definition of a taxonomy of reference models for enterprise integration in terms of their life cycle support, system modelling and representation, modelling dimension, methodologies and system architecture. The objective was to assist in the selection of the model(s) which best sets the context and basis for the definition of a Reference Model for the Computer Aided Support of Simultaneous Engineering (RM_CASSE).

2.5. INJECTION MOULDING SOFTWARE SUPPORT APPLICATIONS.

2.5.1. Software support for injection moulding design.

Gopalakrishan and Pandiarajan (1991) describe an experimental system that defines the most appropriate manufacturing process and material for a particular component. The authors define two design stages, firstly the preliminary design stage, where the product assumes its form, shape and size as related to its functionality, and secondly the detailed design stage where the product acquires detailed geometry, material specifications and quality attributes such as surface finish and tolerance. During the preliminary design stage the key design related attributes are compared with process related attributes. Using an expert system integrated with databases, the suitability of each process is examined. Firstly those processes capable of producing the geometry of the product as a whole are determined, and then those that can produce the entire geometry of the product are examined. The alternative process options are now considered in turn regarding the materials that can be used with each of them in order to suit the functionality of the product, and that can be processed without any undesirable effects. Once the most suitable material has been identified for each process, with respect to functional requirements, the options are then ranked according to a cost algorithm.

In the detail design stage the above list is modified to accommodate new design attributes being specified at the detail design stage. The expert system compares the new design attributes against the capabilities of the material–process pairs. Once the list has been reduced to only those that can do the job, then all but one option is eliminated according to cost criteria. The primary source of information for the knowledge base, as related to the manufacturing processes and materials, was from published literature.

Cutkosky and Brown and Tenenbaum (1989) describe a program for the concurrent design of machined and injection moulded parts named 'FIRST CUT'. The design process has explicit dependence upon design features and the process plan steps to achieve them. In the authors view the essence of concurrent engineering is to use process planning information to complete the partial design. For injection moulding there is the ability to complete a 'rough scetch' of the moulding and clean it up via the application of process operations. The result is a mouldable design. All of the above implies the need for backtracking, since a designer may suggest a particular design change that cannot be permitted for processing reasons. Therefore the system is required to keep a history of previous design states and associated process plans.

Cutkosky and Tenenbaum (1990) describe how for injection moulded design using FIRST CUT, the authors have capitalised on the fact that most injection moulded parts are thin walled, and the authors have therefore provided a thin walled primitive for defining the overall component shape. The designer can then paste a number of primitives such as ribs, bosses, holes etc onto the thin walls. The designer begins by drawing cross sections of the main walls of the component in a commercial CAD system. The drawing is loaded into their program, and with designer guidance, feature–finders (for example wall–finders and corner–finders) identify features. The program then modifies the geometry to produce cross sections with preferred draft angles, thicknesses and corner radii. The designer adds small features such as bosses and ribs. Finally the part is shipped back to the CAD system for display.

The main goal of Ishii, Hornberger and Liou (1989), Ishii and Hornberger (1991), was to develop state of the art design rules for injection moulded products and establish a systematic methodology that applied these rules to part designs in their early stages. The authors approach was to model the process in which good design teams incorporate producability concerns in the early design. They use the concept of 'design compatibility analysis' (DCA). The concept views the goal of plastics design as achieving sound compatibility among i) User requirements, ii) Process constraints, and iii) The candidate design. Knowledge bases are used to evaluate a given design situation, give a compatibility rating, and provide improvement suggestions. It is considered that the above concept gives a reasonable model of the concurrent design and review process carried out by a design review team. An important aspect is to identify the different design values associated with the product for which there are more or less separate experts. Each expert breaks down the part design into building blocks or design elements, which act on information associated with design rules. It is important to identify the decomposition by each expert, as the identified elements serve as data organisation keys in the computer knowledge base. The idea is to model the 'round table' design reviews from different viewpoints, discuss the design and suggest improvements. At any one time the designer can check the proposed design by comparing it with the compatibility knowledge base. The compatibility knowledge base comprises design rules called 'C data'. DCA uses the knowledge base to evaluate the soundness of the candidates design with respect to engineering (or other) objectives of all the viewpoints. The compatibility is graded, for example good, bad, poor etc, and if necessary improvements are suggested.

Huh and Kim (1989) and Kang and Kim (1990), describe a system for the design of ribs and gates on injection moulded parts. A CAD solid modeller is used with an expert system that has heuristics to make decisions. For example standard stress calculations are carried out to decide if ribs are required or not, and if so then a formulae is used to decide how many, their size and dimensions etc. The system would ask the designer to indicate a rib and its dimensions, and then use its rules to decide if this is acceptable, for example the rib may be too wide etc. According to Huh and Kim, since the injection moulding process has direct consequences for the performance of the product, the design of supplementary features such as bosses and ribs should include knowledge regarding not only machining operations of the mould but also causal effects on the performance and mouldability. The system itself comprises three functional groups of software, a geometrical modeller, knowledge based modules and CAE programs. Overall control and the user interface is managed by an expert system. The geometrical modeller displays the primary shape and accommodates supplementary geometry as guided by the expert system. Heuristic knowledge of ribs, bosses and other supplementary features are formalised as rules in the knowledge base module, called RIBBER, which generates a recommendation of the optimum rib structure for a given primary geometry. As well as the primary geometry, also considered are the material loading conditions and structural requirements. The system also contains GATEWAY, which is an expert system for the number and location of the necessary gates. Each expert module in the system triggers the necessary modules for flow simulation, microstructure prediction, mechanical behaviour etc. Thus it is asserted that knowledge based CAD has been created by the integration of an expert system and a geometric modeller. It is considered that the modular structure of the expert system could be readily expanded to cover most attributes of injection moulded design by adding rule based modules. Since features in the form of supplementary geometry are generated through CSG, the corresponding manufacturability features can be simply extracted from the supplementary geometry specifications, for example a part with a rib is equivalent to a slot in the mould.

Hanada and Leifer (1989), developed a knowledge based system to support the design of injection moulded products. A three stage 'design with' features process is supported. Initially skeleton features are used to generate a functional description of the part. In the second stage a representation of the injection moulding viewpoint is generated using a 'design translation' module and interactively providing mouldability feedback to the designer. Finally a second 'design translation' module generates the mould representation from the representation of the injection moulded part. Mould manufacturing feedback advice is interactively provided to the designer as the mould representation is built up.

Poli, Kuo and Graves (1992) developed a design for injection moulding methodology to support the design of parts for economical manufacture. The methodology is based on group technology in order to group part design according to similar tooling and processing costs to provide the designer with the following:

i) a comprehensive listing of difficult to produce features (cost drivers) so that designers can minimise difficult–to–mould features, provided that component functionality is not affected.

ii) a consistent and systematic method for analysing part designs from the viewpoint of manufacturability.

iii) a qualitative and quantitative feel for the impact of part attributes on moulding complexity and thus cost.

iv) a convenient and meaningful basis for organising relative cost data.

A systematic, knowledge based, approach for drawing designers attention to those factors of a part which tend to increase its cost to mould is described.

Mochizuki and Yuhara (1992) developed a solid modeller based system for detecting potential undercuts in plastic injection mouldings and identifying the optimal withdrawal direction. The system identifies feasible parting line positions, ie either where the die plates come into contact or where an undercut occurs, and then evaluates each potential solution for the absence or extent of undercuts, using a methodology developed by the authors. The designer is notified of the optimum solution(s) and is allowed to choose which is to be the solution of all those available.

Manoochebir (1994) and Sebastian (1994) presented a function based intelligent system called 'Designer Apprentice'. The system enables the designer to associate the product function and process knowledge with part geometry. The system uses features as the basic building blocks of part design. Features are used as smart building blocks to capture logic and intent in terms of the form and functions of the part. Design rules attached to features are based on material properties, mouldability, strength, processing and mould making requirements. The Designer Apprentice is implemented in C++ and uses Pro/Engineer as a solid modeller and front end of the system.

2.5.2. Gating and feed system design software.

Irani, Kim and Dixon (1989)a,b describe an automated gating design system for injection moulded parts, which includes determination of the best number, type and location of gates. Five inputs are required for the system. The first input is a features representation of the object to be moulded, the second the material properties, such as mould temperature, process temperature range and degradation temperature, the third is machine data, maximum injection pressure, maximum clamping force, the fourth is the problem class, either one or two. Class one is where the gate location and numbers have been specified, and class two is where nothing is specified. Finally the fifth input is any constraints such as aesthetics (ie no gates on certain faces, regions where no weld lines are permitted (for strength) etc.

The program has an interactive redesign scheme, using initially a coarse global search for the candidate design, and then a local search for the near optimum. In the global search a number of trial gating plans are tried (based on an algorithm) with respect to three design variables, mould temperature, injection temperature and injection time. The plan with the best overall quality becomes the initial local search. During the local search gate plans are tried around the area of the initial plan to try to improve the quality. Eighteen performance parameters are used for evaluation in the local search. These are classified into geometry related and process related parameters. The geometry type depended on the gate locations, and the process type on cavity and inlet conditions.

There are five key areas for the redesign model:

i) Design variables– The program aims to use the minimum number of gates. The operator can specify the initial gate locations for the global search or the program will generate alternatives based on the geometry.

ii) Performance parameters– The criteria used to analyse the quality of the design. The analysis program is based on three nested loops. Geometry type parameters are derived from the results of analysis, whereas process type parameters are obtained directly from the results.

iii) Satisfaction curves– For each performance parameter there is an indication of how satisfactory the intended design has been (excellent, good, fair, poor, unacceptable). These curves can be based upon constraints or heuristic information.

iv) Priority ratings– All performance parameters have priority ratings, of which there are five, very high, high, moderate, low, very low. These are used to steer the gating plan to the overall goals.

v) Design quality– The quality of every iteration is judged based on a table. According to the preset table the design quality is for example excellent if all very high and high priority ratings have excellent satisfaction, all moderate priority have good or better satisfaction, all low priority have fair or better satisfaction, and all very low have poor or better.

Irani, Kim and Dixon (1990) considered that there is a need for computer programs that interpret and make decisions based on the results of analysis. The above gating design software is considered to be such a system, which is based on iterative redesign. Since design is a sequential process it is asserted that knowledge is best expressed in terms of procedures that initiate yet more procedures in a sequential manner, rather than an if–then, ie rule based system, which the authors consider does not lend itself to design. The result is a system that having defined an initial gating location on a product can then use the results of mouldflow analysis etc to iteratively achieve the best location for the gating.

Many authors have been attempting to 'automate' the design process of gating and runner systems using finite element analysis methods (FEM), for example Michaeli and Galuschka (1994) and Michaeli et al (1995) describe recent developments in the CADmould simulation package, whereby simulation of the filling and holding phases of the injection moulding process is available as well as shrinkage and warpage. The system provides this information back to the designer to help his configuration of the gating and runner system. Avoidance of weld lines is aided by simulation of the cascade injection moulding technique, whereby particular gates are opened and closed at particular times by taking appropriate mould and control measures. Using this technique it is possible to bring about a situation where there is only one flow front in the cavity at any time, thus preventing weld lines being formed.

Jong, Chan, and Wang (1994) describe a system based on C–FLOW which automates the problems of 3D runner balancing in a multi cavity mould using a finite element model of the component and feeding system to simulate the flow of material through the mould.

Ni and Guinn (1994) cited the runner system design as the largest single factor in the warpage of injection moulded products and described a system based on C–MOLD for predicting the effect of different runner system configurations on component warpage using finite element analysis techniques to predict pressure distribution during the moulding process.

Ong, Prombanpong and Lee (1994) describe CADFEED, a knowledge based and object oriented system for the design of feed systems for plastic injection moulds. CADFEED combines the use of knowledge based and object oriented tools with a draughting tool. CADFEED's design methodology is a four stage process:

i) CADFEED determines the feasible gating regions. A gate type is assigned to each of these gating locations to form potential gating configurations.

ii) A disqualification process examines preliminary gating configurations for their satisfaction of mould and material requirements. Some configurations are ruled out if they do not meet mould and material requirements.

iii) The third stage is an evaluation of the remaining gating configurations on how well they satisfy product specifications. This stage determines a value for each of the product specifications using rules in a knowledge base.

iv) A final score is obtained for each gating configuration, and the one with the highest score is recommended to the designer. Once a configuration is selected CADFEED proceeds to calculate the dimensions of the gate and runner used.

2.5.3. Mould design software.

Hui and Tan (1992),(1994) presented a methodology to select the parting and side core directions of an injection mould. It is pointed out that it is much more complicated to integrate CAD/CAM for injection moulding then for machining, because the mould cavity and core are machined, and not the component itself. One of the most pertinent issues in injection moulding design is the determination of the main parting and side core directions. The parting direction is directly related to the location of the parting lines, and hence the parting surfaces. This in turn affects the choice of side core mechanism, the location of gates, ejection pins and cooling ducts. Therefore determination of the main parting and side core directions is a prerequisite to any subsequent design process. The first step to automation is to establish a means of selecting an optimised parting direction with associated side core directions. The authors then use a boolean subtraction to create the mould cavity. In order to determine the parting direction a mesh of test points is generated over the surface of the intended product. Then a semi–infinite ray is cast from the test point in the direction of a given prospective parting direction. A blocking equation is used to give a blocking factor for each direction tested. All possible parting directions are tested and the criteria are no or minimum undercuts, and that the area in shear contact with the mould plate should be the minimum. Having chosen the main parting direction, then if there is more than one possible side core direction, the blocking check is again used to select the direction with the lowest number of side cores.

Kang, Park and Hong (1992) describe a CIM system for mould design and manufacture used in a mould manufacturing company. Using this system the basic product geometry is built up to the final product by adding secondary geometry such as ribs etc and defining the parameters of these secondary features to the system. Based on this geometry, the geometry and feature data for the mould is generated. The process parameters such as flow rate, packing pressure and melt temperature are determined as well. A solid modeller creates the mould geometry by parting the product geometry, expanding it to take account of shrinkage, and subtracting the expanded product geometry to form the cavity. Kang, Kim and Chong–won (1990) when describing the same system considered that mould polishing was a major bottleneck, and therefore mould design should aim to minimise this.

Delbressine and Hijink (1991) describe a system for checking the manufacturability of components produced by metal removal, but the methodology could equally be used for the machining of mould cavities. The approach taken is the definition of design transformations that can be manufactured, termed 'manufacturing design transformations' (MDT's). A transform consists of an application to a manufacturable object (MO) of a design and manufacture process planning transform. The manufacturing transform consists of tools, machines and set up constraints in manufacture. The MO consists of two geometric forms, the initial geometric state and the final geometric state together with a set of application rules. Thus backtracking is possible. The rules ensure that the MO can be applied (the transform) and therefore the object can be made. It is considered that for the above work the best design representation was a combination of boundary representation and constructive solid geometry (CSG), the first defining the object in terms of a topological boundary, and the second based on the concept that one can represent a solid object as a series of CSG transforms. The CSG user specifies an initial state and then gives the design transforms to get the final design. The process tree can then be used as a guide for process plan generation.

The mould design system being developed by Rho, Sheen and Lee (1990) used machining features to represent the various components of the mould, ie runners, gates, cooling passages, pockets, holes etc. The features are applied to the cavity and core plate from the viewpoint of required machining operations. Pocket and hole features are used to represent the mould cavity. The authors were developing an application to support the use of the features in mould design based on MOLDCAPP software.

Gerdes, Webb and Contantun (1994) developed an expert system called 'Mold Fabrication Process Planner' (MFPP). The software supports the designer in building up the mould design using standard parts. However each standard part is represented as a 'feature type' which has a process plan attached for fabrication operations. Therefore as the designer builds up the mould design, the process plan is also created. The MFPP has been written in C++ and runs under Microsoft windows. To describe the features of the mould, 'Material Removal Shape Element Volumes' are used.

The expert system has been developed from the formal methodology of mould making it can:

i) Produce work orders with consistent instructions independent of the mould maker.

ii) Generate work orders using only resources that are available and in a format that can be used by scheduling and procurement packages.

iii) Simplify the development of work orders for complex moulds.

iv) Provide time and cost estimates.

Cinquegrana (1990)a, (1990)b describes a knowledge based system called CAD–mould–maker. The system automatically generates mould designs (using standard parts) based on parameters, rules and constraints captured in the knowledge base. The application requires the designer to import the model of the moulded part in IGES format. The system determines mould layout based on the lowest percentage scrap for the specific number of cavities. The actual size of runners for each layout configuration is determined using injection pressure and viscosity models. The designer is given a choice of layout options and estimated (material) cost per 1000 parts of each option. Once the layout has been chosen the remainder of the mould is built up using standard parts. The cavity geometry is established by subtraction of the part geometry from the mould plate.

Lodenstien, Romps and Tan (1994) describe a mould design support system called 'mould–ease'. A designer is allowed to build up a model of the component in a CAD system. The part design is analysed and the designer is asked to make decisions on the number of cavities, material used, and the size of the mould base. Following on from these decisions the designer executes the mould design program which contains a catalogue of standard mould components, and allows the designer to go through the mould design process, selecting standard parts from the catalogue. The designer works through the design process using a series of menus.

Tseng Lia and Peng (1994) combined mould base software with a CNC programming package (to generate CNC part programs), using a common geometrical data source. Once the part geometry is created the mould design and machining data generation processes are started at the same time. The CAD system has a mould base data bank from which the designer is able to select.

2.6. Summary

From the above work it can be seen that there are many areas where research is being carried out into concurrent engineering, and many different software applications being developed.

The main points to be extracted from the above literature are considered to be as follows:

i) The accepted approach for design for manufacture software is a general system architecture within which there exists a set of independent but cooperating subsystems or functional modules.

ii) There must be a defined set of relations between different functional modules within a system. It is very important to handle dependency relations between different models effectively and changes to one model must not happen without updating other models with dependant relations.

iii) Design for manufacture considerations should be brought in to the design cycle as early as possible, but the implication of this is that the designer has too much information to remember at the beginning of the design. Design for manufacture software should therefore be used to provide and manage this information. Information should not be provided all at once but the important details at the required time. However design for manufacture software should not act as a constrainer on the designers activity, but rather in an advisory capacity, providing information and advice on possible design problems.

iv) The appropriate representation of geometry has been cited as the key issue for design for manufacture software. Features are the accepted medium to store information on geometry and to attach heuristics and constraints to that geometry. Features also provide the designer with information at his own level of cognition, ie reasoning about geometrical features rather than simply lines and points etc.

v) Most of the above authors have advocated or used 'design with' features for the reason that feature recognition is a difficult programming task. However the designer is limited by 'design with' features to relatively simple geometry, and this in turn limits the design for manufacture software that is developed. The problem of geometrical complexity is therefore a limiting factor for the development of design for manufacture software.

Chapter 3.
Background to injection moulding information support system.

3.1. INTRODUCTION.

This chapter provides the reader with the context of the research by describing the MOSES (Model Oriented Simultaneous Engineering System) project, to which the research of the author has contributed. The objectives of MOSES are described, the role of the Manufacturing strategist (which contains injection moulding design support applications) within the MOSES system, and the role of information modelling with respect to the capture of manufacturing capabilities and constraints. Also described are the formal methods used in the research and the experimental environment. The objectives of MOSES are described in section 3.2, the role of the Manufacturing strategist in section 3.3, and the role of information modelling in section 3.4. Section 3.5 describes the formal methods used in the research and section 3.6 the experimental environment.

3.2. THE MOSES RESEARCH PROGRAM.

A computer aided engineering (CAE) system to support simultaneous engineering called MOSES (Model Oriented Simultaneous Engineering System) has been researched by Loughborough and Leeds University. The research has been undertaken in order to define how Product and Manufacturing Models can be exploited in future CAE systems that will support Simultaneous Engineering. The sharing of common, consistent product and manufacturing data between a range of software applications and design teams is considered key to the effective support of simultaneous engineering. Methods for the identification of conflicts that arise in simultaneous engineering are also being explored. The objectives of the MOSES project are; i) to provide a reference model for CAE systems based on Product and Manufacturing Models, ii) to define a knowledge and software environment in which Design

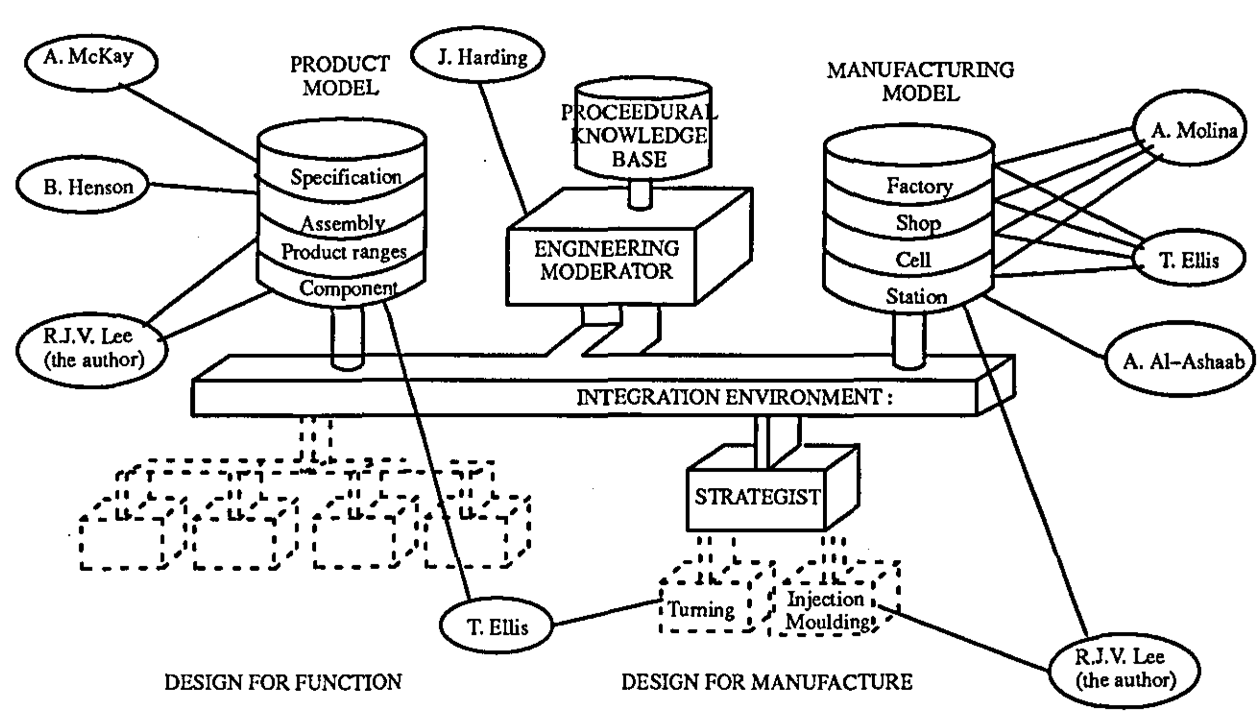


Figure 3.1. The conceptual research system of the MOSES project.

for Function and Design for Manufacture can be applied simultaneously, and iii) to provide intermediate and end of project demonstrations of the results using an industrial case study.

The conceptual research system of the MOSES project is illustrated in Figure 3.1. The figure shows the key elements of the MOSES project and the areas of contribution by members of the MOSES research group. In the context of the MOSES project the Product model captures information related to the product throughout its life cycle and the Manufacturing model captures information related to the manufacturing resources and processes to support product realisation. The MOSES framework allows the sharing of common data between a diverse range of design teams and software applications. A specialist application, called an Engineering moderator identifies conflicts within the Product model which may arise due to discordant outputs from the different application environments that populate the Product model.

The MOSES architecture is intended to support many software applications. These software applications are grouped into a range of application environments. Each application environment contains broad groups of applications that support a specific element of the product life cycle. Typical examples include a group of applications collectively called the Design for Manufacture Environment (DFME) and also a group called Manufacturing Information Generation which is concerned with post design activities such as process planning. Other non manufacturing elements of the product life cycle could be supported by a Design For Function Environment (DFFE) for which the content has not yet been defined. Both the DFFE and DFME will provide tools to populate the Product model. All engineering application environments interact with the data models via an integration environment.

3.3. THE ROLE OF THE MANUFACTURING STRATEGIST WITHIN THE MOSES SYSTEM.

One of the engineering software applications in the DFME is the Manufacturing strategist. Strategist applications are specialist expert applications which assist users of the CAE system to evaluate, modify and extend the product design using criteria which are closely allied to particular design perspectives. A manufacturing strategist assists users in design for manufacture activities, using product information from the Product model and manufacturing resources and capabilities information from the Manufacturing model. The manufacturing strategist enables the interaction of product and manufacturing information in order to asses the consequences of changes to a product for manufacturability. Feedback advice is provided to the designer.

At the higher level the manufacturing strategist is concerned with identifying the most suitable manufacturing processes and strategies for use of resources. The lower levels of the strategist are divided into applications which act as DFM experts for a particular manufacturing process, for example, machining, injection moulding, forming etc. The MOSES project is addressing at the lower level the areas of machining rotational components with prismatic elements and injection moulding. This research has been concerned with the structure of a lower level application to support design for injection moulding.

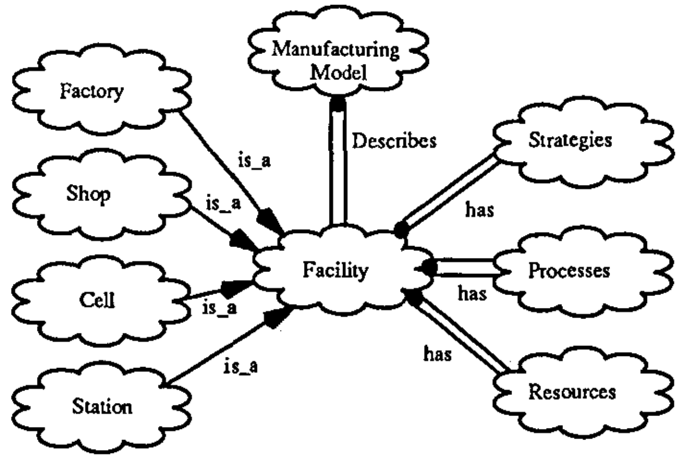


Figure 3.2. Top level classes of the Manufacturing Model.

3.4. THE MOSES MANUFACTURING MODEL.

Each design perspective, eg design for injection moulding, design for machining has its own design criteria, rules and heuristics to which the product design should conform. A Manufacturing model captures all data related to process capabilities, characteristics and manufacturing resources held by the enterprise, Ellis et al (1993)a. Having a Manufacturing model, parallel to the Product model, offers the basis for the integrity of both product and manufacturing data. Ideally a Manufacturing model is the source of manufacturing information from which any engineering software application (DFM, process planning, production planning etc) can draw information in order to support the decisions made in that application. The structure and content of the Manufacturing model in the MOSES project has been developed using the Object Oriented techniques of the Booch methodology, Booch (1991), (1993). Figure 3.2 shows the top level classes of the Manufacturing model. This shows that the Manufacturing model describes a Manufacturing Facility at four levels, factory, shop, cell and station. A manufacturing facility has a 'has_a' relationship with the 'Manufacturing Resources', 'Manufacturing processes' and 'Manufacturing Strategies' classes to capture all aspects of manufacturing information. The Manufacturing Resources class captures all the physical resources that are available in an enterprise such as machines, cutting tools, handling equipment etc. The Manufacturing Processes class captures the capabilities and characteristics of all manufacturing processes such as machining, injection moulding etc. The Manufacturing strategies class captures the aspects related to how the Manufacturing Processes and Manufacturing Resources are structured and used in the company.

A. Al–Ashaab investigated the principles of a Manufacturing model for the design of injection moulded parts to support interacting design applications from multiple design viewpoints, ie design for mouldability, mould design and moulding machine calculation, Al–Ashaab and Young (1992), (1994).

In order to support the operation of a lower level design for injection moulding support application, the authors research has examined the lower (station) level of the Manufacturing model structure, to capture the capabilities and constraints of the injection moulding process. This work extends that of Al–Ashaab by considering 3D geometry instead of the simple 2D representations used by Al–Ashaab. The use of 3D enabled the investigation of establishing a direct link between the geometry of the product and that of the mould.

3.5. FORMAL METHODS USED IN THE RESEARCH.

The following sections describe the formal methodologies used in this research, which are IDEF0, Booch and EXPRESS. The Booch object oriented methodology and the EXPRESS data modelling language are respectively the computational and information languages of the Reference Model for Open Distributed Processing (RM–ODP), Linnington (1992).

3.5.1. The IDEF0 methodology for design activity modelling.

IDEF0 stands for ICAM Definition level 0, Colquhoun et al (1993), and is based upon a structured analysis and design technique to produce a function or activity model which is a structured representation of the functions of a manufacturing system, and of the information and objects which relate to those functions. IDEF0 is a top down hierarchical method which describes a system using a series of functions arranged sequentially, as shown in Figure 3.3. The

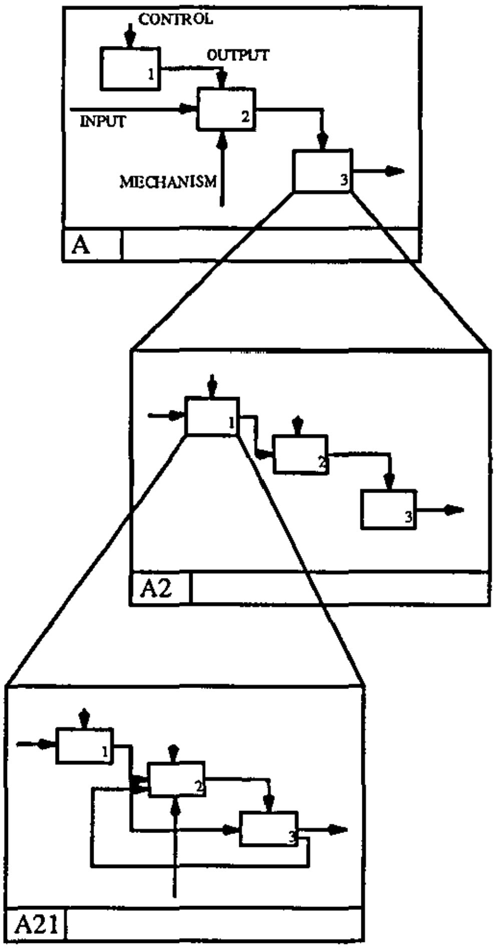


Figure 3.3. Hierarchy of IDEF 0 model.

hierarchical method allows the definition of a system in any level of detail, in order to provide an understanding of complex systems. IDEF0 is good at providing an initial clear view of the interactions of activities in a system, but it does not provide a clear definition of the information flows between activities. IDEF0 can therefore be used to provide an initial view of information flow but is inappropriate for detailed information representation.

In order to identify the functionality of an injection moulding strategist the author has performed an activity modelling of support for injection moulded product design using the IDEF0 methodology. This analysis is described in Chapter 5.

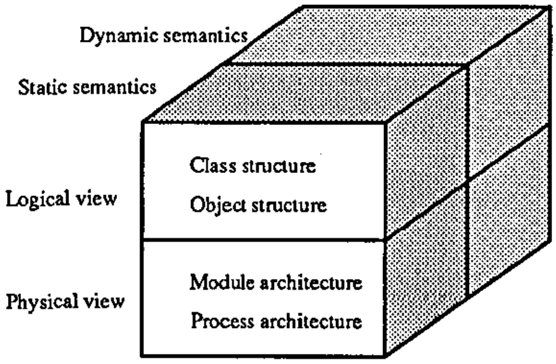
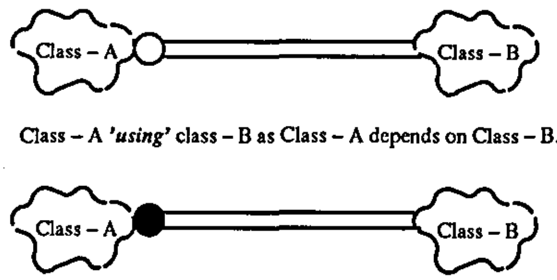


Figure 3.4. The models of the Booch object oriented design methodology.

3.5.2. The object oriented methodology of Booch.

To have a well implemented software system, it is very important to have a good understanding of the problem. This requires the designing of models that emphasize the proper and effective structuring of a software system as well as defining the relations and interactions between models. The designed models help to reason about the systems structure and provide a requirement for implementation, Booch (1991). The object oriented approach to software development is based on modelling objects from the real world (eg boss, machine, gating system) and offers several advantages to conventional approaches (algorithmic approach). Such advantages are better understanding of requirements, better handling of complex systems, smaller systems through the re–use of common mechanisms as well as leading to less complex, easy to enhance and maintain software systems, Booch (1991).

Booch (1991) defined object oriented design as 'a method of design encompassing the process of object oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design.' The results of object oriented design can be used as blueprints for completely implementing a system using object oriented programming, Booch (1991). Figure 3.4 shows the models used for object oriented design in the Booch methodology. The models are intended to capture all necessary design decisions. However object oriented design can be carried out without using all the models. To serve the purpose of the author in representing the structure of the Manufacturing and



Class – A *'using'* class – B as Class – A depends on Class – B.

Class – A *'has'* class – B ie instance of class – B *is part of* instance of class – A.

Figure 3.5. The 'using' and 'has' relations in the Booch methodology.

Product models only the static semantics of the logical view model expressed in class diagrams has been used.

A class is a set of objects that share a common structure and a common behaviour while an object is an instance of a class. A class diagram is part of the object oriented design notation to show the existence of classes and their relationships in the logical design of a system. The key relationships that are used in the Booch representation are the 'using' and 'has' relationships, as shown in Figure 3.5.

The class diagram notations of the Booch methodology have been used to represent the structure of the Manufacturing and Product models as described in Chapter 8.

3.5.3. The EXPRESS data modelling language.

The EXPRESS language has been used because no formal method exists within the Booch methodology for the representation of complex data structures within each class. Therefore EXPRESS has been used to represent the underlying data structures of the classes defined in Booch notation.

EXPRESS is a formal non–software dependant language for the representation of data structures. EXPRESS has been approved for registration as the ISO International Standard language for information modelling, EXPRESS (1992). EXPRESS is not a programming language, but it consists of language elements which allow an unambiguous object definition, with constraints on the data to be clearly and concisely stated. Classes are specified in EXPRESS as entities. The definition of an entity is in terms of its properties, which are characterised by specification of a domain and the constraints on that domain, EXPRESS (1992).

The use of the EXPRESS language in conjunction with the Booch methodology to represent the structure of the Manufacturing and Product models is described in Chapter 8.

3.6. The experimental environment.

3.6.1. C++ and Objectivity.

The Product and Manufacturing models and a design for manufacture environment to provide concurrent support for injection moulded products have been implemented using purpose written C++ code and DecObject DB, which is an object oriented database system, (a version of Objectivity). The essential information content and structure of classes of objects, and the relationships or associations between them have been captured through the attributes for object classes.

3.6.2. Unigraphics.

The geometry in the Product model is displayed using the Unigraphics V10 Modelling application. The core of the Unigraphics application is an exact Boundary representation (BRep) solid modeller that also allows geometry to be built up using constructive solid geometry (CSG).

Chapter 4.

Supporting concurrency in injection moulded product design.

4.1. Introduction.

This chapter presents the key issues in computer support for concurrent design for injection moulding and the scope of the authors contribution to this area of research is identified. The chapter then focuses on the main area of the research, discussing the issues involved in a set of design for injection moulding applications to support concurrent design for injection moulding, referred to collectively in this research as an Injection Moulding Strategist or IMS. In order to support this work some investigation has been required in the area of features support, where the feature structures are effectively captured in the information modelling environment, to provide for the information modelling support requirements of an injection moulding strategist. The key issues in computer support for concurrent design for injection moulding are identified in section 4.2. The issues involved in an Injection Moulding Strategist are identified in section 4.3, and those involved in features support for an injection moulding strategist are identified in section 4.4.

4.2. Issues in Design for Manufacture support for concurrent design.

Conventionally the design process has considered the various design viewpoints sequentially and the manufacturing consequences of functional decisions have not been considered early enough in the design cycle. The functional definition of the product would be refined through interaction with process experts, to achieve a manufacturable product definition. In the traditional design process manufacturability of the product definition is therefore limited by the form of the initial product definition, which is the input to the design for manufacture process. Historically design for manufacture (DFM) has reflected this approach, providing support for individual areas of the product life cycle, eg design for assembly Dewhurst and Boothroyd (1987). This problem is exacerbated in injection moulding because design for injection moulding (DFIM) involves not only design of the product, but also of the mould. Therefore sequential approaches to design can limit not only the manufacturability of the product, but also of the mould because the main input to the mould design process is the already designed product. Thus the traditional method of injection moulding design can require many iterations of the design process before an acceptable design is achieved. Each time there is a conflict between the viewpoints considered in the design, eg design for function, design for mouldability, mould design, design for mould manufacturability etc another iteration may be required to resolve conflict.

Research into software support tools for injection moulding design have been in two general areas; mathematical modelling tools or computer advice systems using AI techniques. Mathematical modelling tools are simulation programs that simulate the flow of plastic in the mould so that a designer can identify potential weld lines, areas of shrinkage, unfilled areas of the mould etc, eg Mouldflow, Austin (1994) and CADMOULD, Michaeli and Galhusckla (1994). This information can be used to optimise the gating positions on a product and to balance the gating and runner systems. Other programs simulate heat flow in the mould to assist the designer in choosing the configuration of the cooling system or to anticipate component warpage, eg C–MOLD, Ni and Guinn(1994). Mathematical simulation programs do not provide advice on design decisions, but are useful in providing the designer with information on the effect of different design scenarios without actually building a mould. Since simulation programs rely on simplifying assumptions, the results may be inaccurate. However as long as this is accounted for by the designer when using the program they are useful design tools. This thesis does not consider the area of mathematical simulation tools, but investigates the area of advice support for concurrent engineering using AI techniques.

Computer support for injection moulding based on AI techniques has typically addressed detail areas within the overall process. For example: design for mouldability, eg Cutcosky et al (1993), design of feeding systems, eg Cinquegrana (1990)a, (1990)b, Irani et al (1990), design of cooling systems, eg Irani et al(1990), consideration of component ejection, eg Moshizuki and Yuhara (1992), Hui and Tan (1992) , and selection of mould plates Kang, Kim Chong–won Lee (1990), Irani et al (1990), Cinquegrana (1990)a, (1990)b. The above systems capture engineering knowledge for their specific areas but do not consider interaction between areas which impinge on product and mould design. The following authors have explored interactions in product design by concurrently considering different aspects of the the life cycle, Gadh et al (1992), Ishii (1992) Hambaba et al (1992), Hanada and Leifer (1989). Although these systems provide a level of design and manufacturing information, the information that they use is captured within an application that analyses a particular area of the design process and is not accessible by other applications. For example Ishii et al (1989) and Ishii and Hornberger (1991) implemented a computational model for design review that utilised Design Compatibility Analysis (DCA). This was achieved by simultaneously evaluating a candidate design from multiple viewpoints, providing an expert for each of them. Hanada and Leifer (1989) implemented a knowledge based system that represented a plastic part and mould design knowledge, and linked it to a CAD system. Their approach aims to have different knowledge domains that could be simultaneously accessed when design decisions are made. The above systems capture design and manufacture information in a software application which represents a specific perspective on the design process, rather than that information being the kernel of the system.

The author takes the view that the kernel of a design for manufacture system can be defined as the source of product and manufacturing information which must be available for each application. To provide an appropriate design support tool based on this approach there are a number of issues that need to be resolved. The critical issues significant for the authors work are:

- How can interacting software applications, which are data driven, be structured to provide information to support design and manufacturing decision making in a concurrent manner?

- Can alternative feature representations be effectively used to support the alternative viewpoints needed in design for injection moulding?

4.3. Interactions within a set of injection moulding strategist applications.

Design for manufacture is a process oriented activity, and to bridge the (DFM) gap involves the simultaneous design of the product and the process to manufacture them, Cutcosky, Tenenbaum, Muller (1988). Therefore the important task of an intelligent Computer Aided Design (CAD) system would be to make useful and relevant manufacturing knowledge available to the designer early in the design process, Dixon (1988). If manufacturing consequences of a design are considered early in the design process, the product definition is produced as the output of the disciplines involved in design for function **and** design for manufacture, rather than as an output of functional design and an input to design for manufacture. It is the view of the author that the objective of concurrent design for manufacture software should be to provide manufacturing consequences as the designer builds up the functional geometry of the product.

In order to explore data driven concurrent design for injection moulding, the main focus of this research has been to investigate the functionality and structure of a set of injection moulding strategist applications, in order to make use of the Manufacturing model and Product model to provide the designer with design for injection moulding information in the form of feedback advice on part mouldability and mould design.

The research addresses the following issues in relation to design for manufacture work:

- To what extent can injection moulding strategist applications interact concurrently to provide alternative views of the product as it is designed?

- How should injection moulding strategist applications interact with a Product and a Manufacturing model?

The authors investigation of these issues has led to the exploration of the functionality of an injection moulding strategist as described in Chapter 5, and also into the role of the strategist in providing a translation capability between different views of product data, as described in Chapter 7.

4.4. Representing product and manufacturing data to support design for injection moulding.

The following work has been carried out in the area of features technology to support the investigation described in section 4.3. The author has investigated the appropriate sets of feature types to support the alternative viewpoints needed in design for injection moulding. The feature structures identified are effectively captured within the information modelling environment and are therefore part of the Product model or the Manufacturing model.

With respect to features in the Product model, Kjellberg and Schmekel (1992) explored the issue of parallel representations of the product to allow for the different viewpoints in the product development chain. The representations had to be updatable, able to be interrogated and have consistent relationships with the other representations. Kjellberg and Schmekels product representation is based on structures of functions, physical principles and solution principles. The structure of the functions specifies the product functionality, the structure of the physical principles specifies the physical behaviour of the product, and the solution principles specify the technical solution which has the specified behaviour and fulfils the required functionality. It was considered important by Kjellberg and Schmekel to handle relations between models effectively, and a data structure in the Product model is advocated to maintain consistency between representations.

Whilst the author agrees with Kjellberg and Schmekel, that functionality is captured in the

Product model, the interactions that take place to support concurrency are part of the strategist, rather than a data structure in the Product model. Therefore this research addresses the consistency problem within the strategist issue of how to use the interaction of Product model information on the product current state, and manufacturing process capabilities data from the Manufacturing model, in order to support concurrent design for injection moulding.

This research addresses the following issues in relation to features in the Product model:

- What alternative sets of feature types are required to represent the manufacturing views of a product?
- Can a set of feature types be defined to represent the functionality of the product?
- What links are required between feature types instantiated in the Product model to capture the data generated by the strategist?

With respect to features in the Manufacturing model, Al–Ashaab (1994) created a Manufacturing model for design of injection moulding parts which supported interacting design applications from multiple viewpoints, ie design for mouldability, mould design and moulding machine calculation. Using an interface the designer was able to build up the product using mouldability features and receive feedback on the consequences of design decisions for mouldability, for the mould and for the requirements of the moulding machine. Investigating the provision of an injection moulding strategist, capable of providing support for concurrent product and mould design has led the author to explore the appropriate sets of feature types to provide extended information structures that go beyond that of Al–Ashaab (1994)

The limitations of the work were that the designer was constrained to using mouldability features to define the shape of the product, and therefore there was no interaction between functional and manufacturing constraints. Also because the interface was focussed on demonstrating the principles of a Manufacturing model, the product was defined in terms of two dimensional feature parameters. No geometry or geometrical relationships were evaluated, and the system considered only simple 2D shapes. Since the geometry of the mould is dependant upon that of the product, no direct link could be established between the product and the mould.

This research addresses the following issues in relation to features in the Manufacturing model:

- What is an appropriate set of feature types to capture mouldability in order to support evaluation of geometry and geometrical relationships?
- What are the appropriate sets of feature types to capture the alternative ways of configuring an injection mould in order to support evaluation of geometry and geometrical relationships?

The authors investigation of the issues in this section has led to the exploration of features technology to support the interactions of injection moulding strategist applications, as described in Chapter 6, and also of the structure of a functional model termed the 'Product Range Model' representation to capture functional constraints, as described in Chapters 6 and 7.

Chapter 5.

Defining the functionality of an injection moulding strategist.

5.1. INTRODUCTION.

This chapter discusses the problems involved in defining the functionality of an injection moulding strategist. The authors investigation into the required functionality of software support tools for concurrent design for injection moulding using an IDEF0 activity model is described. This investigation has led to the definition of an implied structure of a support tool for concurrent design for injection moulding, which is presented. This structure is the basis for subsequent investigations described in the remainder of this thesis. The defined support tool structure has enabled the definition of the information model support requirements to facilitate functionality, and these are also presented and are used as the basis for subsequent investigations described in the thesis. Section 5.2 discusses the implications of concurrent design for injection moulding software support tools, and in section 5.3. the authors investigation into concurrent design support for injection moulding design is presented. Section 5.4 discusses the perceived structure of a support tool for concurrent design for injection moulding based on the above.

5.2. THE IMPLICATIONS OF CONCURRENCY FOR DESIGN SUPPORT TOOLS.

The product and the mould can be seen from different viewpoints, depending on the stage of the design process. For example a designer may look at the product with a view to achieving the functionality by satisfying the product specification, and mouldability may be considered separately. With respect to the mould design, a designer may consider the cavity or core design, and not consider feeding or cooling until a later stage. The activities involved in injection moulding design are shown in Figure 5.1. Concurrency in injection moulding design implies the concurrent interaction of the activities within product design, the concur-

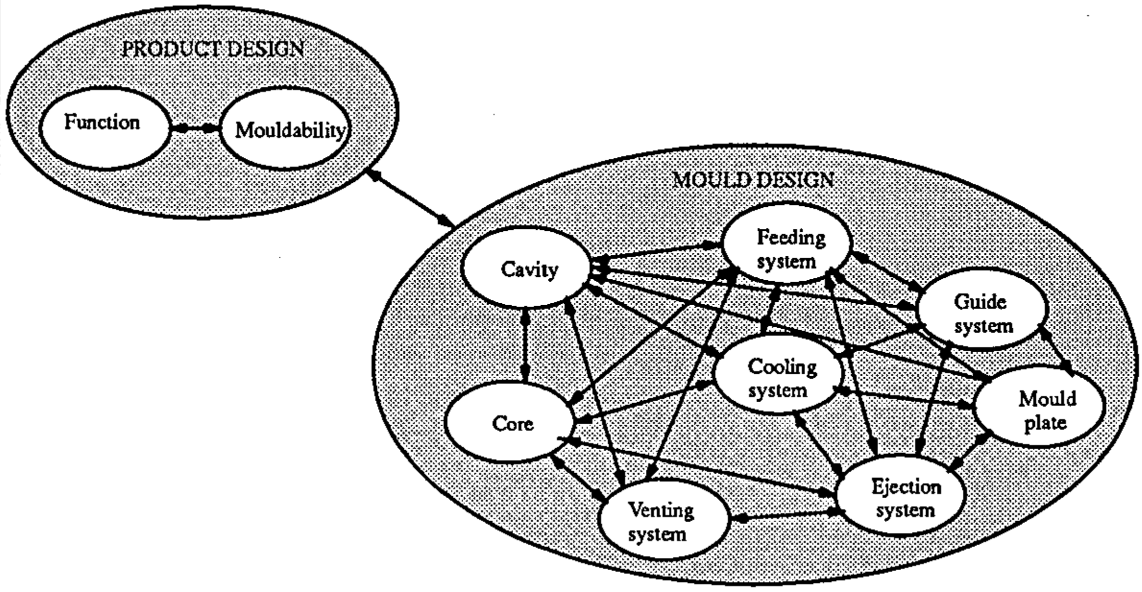


Figure 5.1. Viewpoints in injection moulding design.

rent interaction of all the activities within mould design, plus concurrency of the product and mould design process. However each activity in injection moulding has inputs that are required before the constraints of that area can be considered. Some of these inputs may be outputs from other activities. The initial phase therefore of defining the functionality and information requirements of a data driven injection moulding strategist application is to understand the activities in injection moulded design as a clearly defined set of activities, plus the information flows between them. To achieve such an understanding requires a formal method of representing the design support process. This can provide the basis for an investigation into the structure and functionality of software applications to support the concurrent design process. The author has created a formal representation of the design support process for injection moulding as described in section 5.3.

5.3. SIGNIFICANT INTERACTIONS IN DESIGN FOR INJECTION MOULDING.

5.3.1. Activity modelling to identify potential interactions.

An IDEF0 activity model of concurrent support for injection moulding design has been built by the author, and the relationships therein are analysed in the next section. The model in its entirety is shown in Appendix 1. The modeling activity has been supported by discussions with people in industry about their design methods for injection moulding, and by the use of literature about the methodology and considerations in injection moulding product and mould design. The activity model has enabled the author to break down the design support process into its constituent parts. This has provided a view of the key activities and their relationships. The activities identified from the model have provided a basis from which to begin the investigation of how concurrent design software support can be achieved.

From the IDEF0 model in Appendix 1 it has been possible to identify the possible areas for concurrent interactions as:

1. Within product design.

2. Between product and mould design.

Also no concurrency has been identified between or within the key areas (within mould design) of cavity, core, feeding system, cooling system, ejection system or mould plate design.

Using this initial contribution the functionality of an injection moulding strategist has been identified as described in the following sections.

5.3.2. Interactions between product functional design and design for mouldability.

During product development, functional product geometry must be interpreted by the injection moulding strategist (IMS) in terms of the equivalent mouldability representation. At each juncture the IMS must analyse the functional data, and provide feedback to the designer with respect to the consequences for mouldability of each modification. The designer must be able to modify the product representation in response to mouldability feedback, and any changes in the mouldability geometry must result in the equivalent change to the functional representation. Alternatively the designer must be able to ignore manufacturability advice and continue the design process.

5.3.3. Interactions between mouldability and mould design.

During product development, mouldability data must be interpreted by the IMS in terms of the equivalent mould cavity and core geometry. At each juncture the IMS must identify the equivalent mould geometry, and provide the designer with feedback with respect to the consequences for mould design of each modification to the product. The designer must be able to modify mould geometry in response to mould design feedback, and any modification of mould geometry must result in the equivalent change to the geometry of the injection moulded product definition.

5.3.4. Interactions between mould design and mould system element design.

During product development, the mould cavity and core geometry must be used by the IMS to identify the most suitable configuration for other mould system elements. At each juncture the IMS must identify the mould system element geometry, and provide the designer with feedback with respect to the consequences for mould system element design of each modifi-

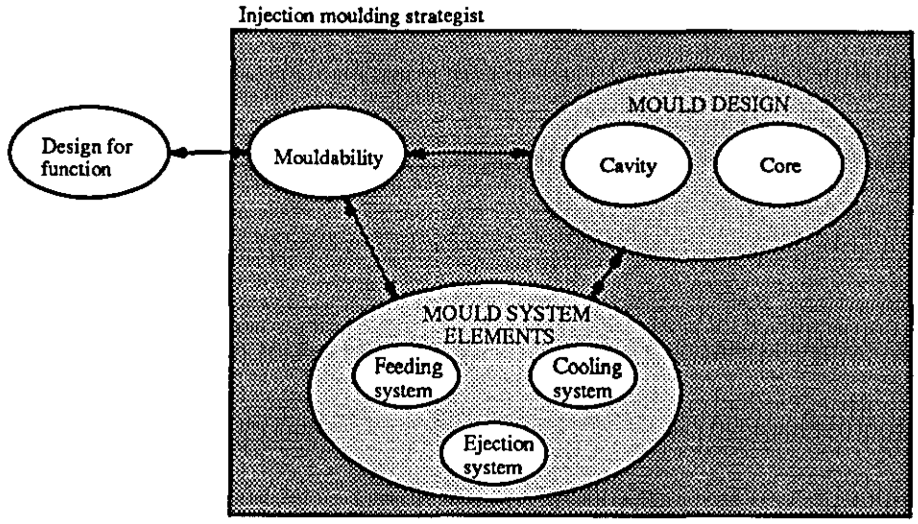


Figure 5.2. Interactions within injection moulding design.

cation of the product. The designer must be able to modify mould geometry in response to mould design feedback, and any modification of mould geometry must result in the equivalent change to the geometry of the injection moulded product definition.

5.4.THE PERCEIVED STRUCTURE OF AN INJECTION MOULDING STRATEGIST TO ENABLE INTERACTIONS.

In view of the interactions identified above the support applications within the IMS are grouped into three areas; mouldability, mould design (cavity/core) and mould system element design, as illustrated in Figure 5.2. In order to enable the interaction of functional and mouldability constraints the mouldability support applications must interact with a functional design interface outside the IMS. As identified in the modelling activity, each of the three main areas within the IMS must have interactions between them in order to support concurrent injection moulding design. From the above, within the context of the MOSES system, the implied structure of the IMS application is shown in Figure 5.3, whereby mouldability support applications are grouped together as are the mould design and mould system element design support applications. This reflects the relationship between product and mould design identified in the IDEF0 modeling activity.

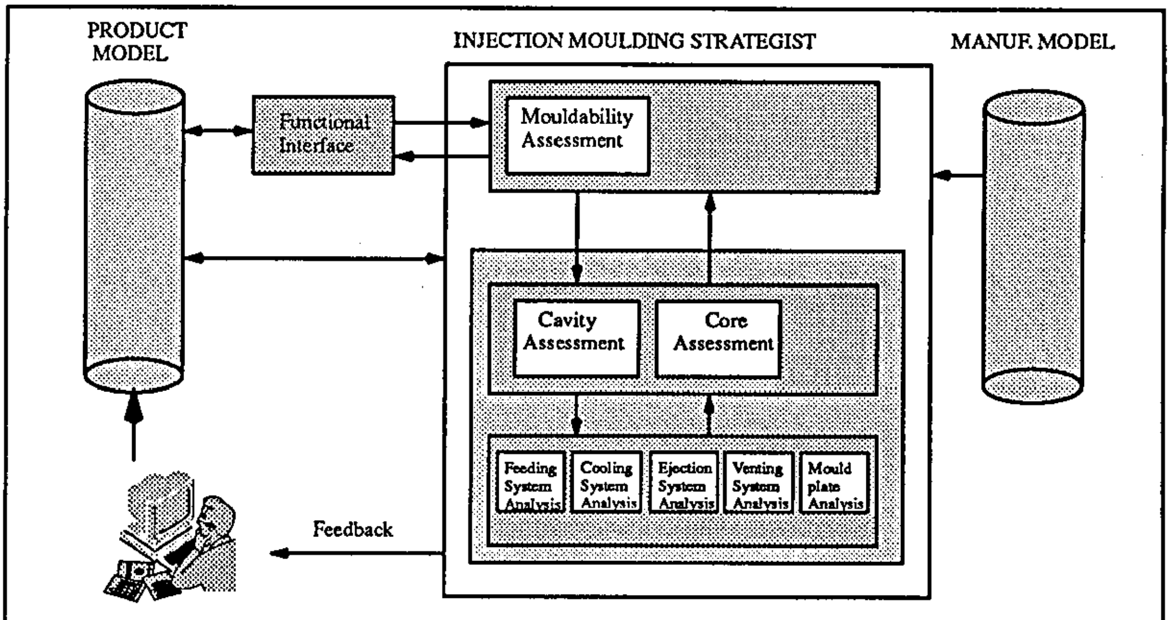


Figure 5.3. Perceived application structure.

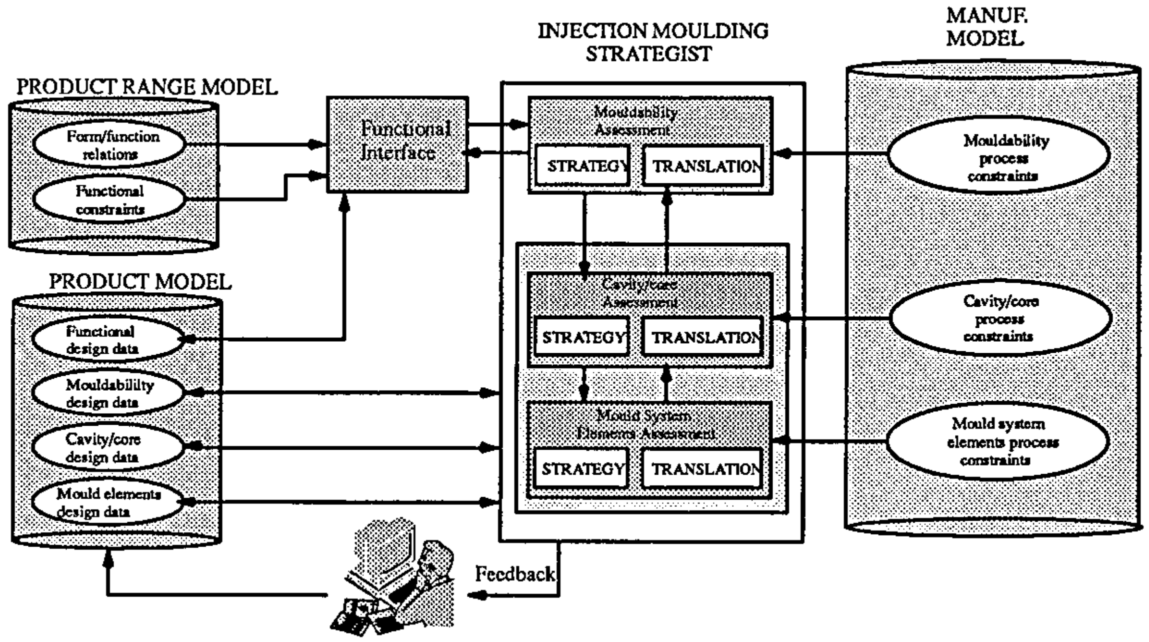


Figure 5.4. Strategist and information model interactions.

The information model support required to facilitate the strategist functionality, is as shown in Figure 5.4. A Manufacturing model is required to provide general injection moulding process capabilities data. A representation is required to support the functional interface in the association of function and form, and to capture functional constraints. In this thesis this representation is termed the 'Product Range Model'. Another representation, termed the 'Product Model', is required to represent the product and mould from all design viewpoints as they are built up by the designer. This model must be populated in such a way as to facilitate the association of the product specific data therein with the general injection moulding process capabilities data in the Manufacturing model and with the functional constraints in the Product Range Model.

From the evaluation of the strategist structure the requirements to achieve concurrency have been identified as:

i) An updatable representation in the Product model is required in terms of each viewpoint which is considered by an injection moulding strategist application. This requires a definition of the features types to represent each viewpoint.

ii) In order to provide feedback support to the designer with respect to multiple viewpoints in design a major requirement of an injection moulding strategist is to provide a translation between the viewpoints within injection moulding design.

iii) The manufacturing data representation in the Manufacturing model must be in an appropriate form to enable the association of general process constraints data with instantiations of feature types in the Product model.

iv) A representation is required of functional data in the Product Range Model. This must be in an appropriate form to enable the association of functional constraints data with instantiations of feature types in the Product model.

The successful support of concurrent design for injection moulding can be attained if each of these requirements are met. Each of the elements of design for manufacture support defined in this chapter are discussed in the following chapters.

Chapter 6.
Exploration of feature types to support interaction between strategist applications.

6.1. INTRODUCTION.

This chapter discusses the contribution made by the author to features technology in the interaction between design for function and design for injection moulding. The problems involved in providing manufacturability feedback during functional design are discussed, together with the authors analysis and the definition of appropriate sets of feature types to support interactions between strategist interactions. The remainder of this chapter is divided into two sections: Section 6.2 discusses the use of features in injection moulding design, the role that they have in the present research, and the requirement for a range of alternative feature representations. Section 6.3 discusses the authors definition of a set of feature types for each viewpoint within injection moulding design.

6.2. FEATURES IN INJECTION MOULDING DESIGN.

From the literature survey, intelligent design for manufacture CAE systems must reason about the topology and geometry of designed artifacts. Therefore a sufficient representation of geometry is a key issue, as highlighted by Cunningham and Dixon (1988). Since reasoning is in terms of features, representation should also be in terms of features. Geometric features are a very important part of product generation and according to Gadh and Prinz (1992), typically four categories of geometric feature knowledge are required: (a) feature parameters eg boss diameter, rib thickness etc, (b) feature relations such as distance and orientation between two features, (c) feature interactions, eg one feature rests on another, (d) topological entities on a feature (faces, edges, vertices). In the authors exploration, these four views seemed to be appropriate to the application of features technology to design for injection moulding. The

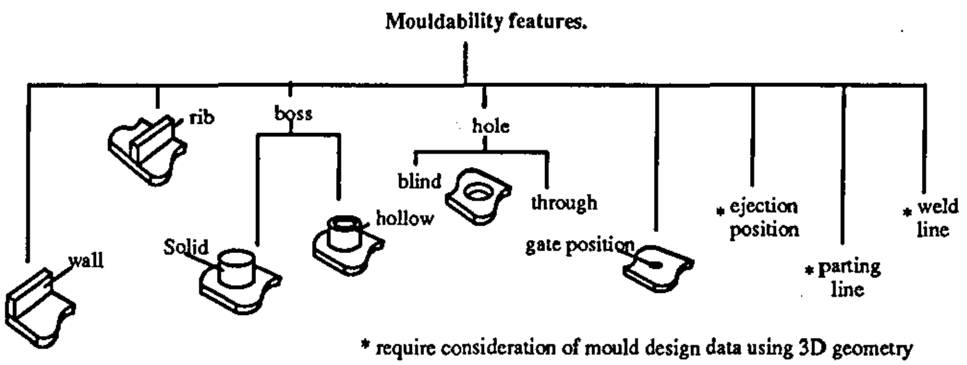


Figure 6.1. The mouldability features set of Al–Ashaab.

authors work goes further than that of Gadh and Prinz who considered only the representation of form features, whereas the authors work has considered the representation of multiple viewpoints using features.

Features are valuable in a data driven design support application because they facilitate the grouping of design data and its association with particular geometry. Thus the relevant data can be placed into an appropriate representation. In the authors research the data is stored in information models, either a Product model or a Manufacturing model. In this way general process constraints in the Manufacturing model and functional constraints in the Product Range Model can be referenced in relation to a given geometry, or combination of geometry on a particular product in the Product Model. The work represented in this thesis has built up the work of Al–Ashaab (1994) who developed a Manufacturing model for injection moulded products based on the set of mouldability features of Cunningham and Dixon (1988). The features set of Al–Ashaab is shown in Figure 6.1. This Figure shows the different geometric forms within the mouldability process. A key problem in the authors research has been to use that process related data and apply it to different products.

In the context of injection moulded product design, manufacturability information concerns not only the mouldability of the product but also the design of the mould. Therefore the problem for the author has also included the application of mould design data to the specific product. In order to achieve this it was necessary to capture a 3D representation of information in order to enable the link between the product and the mould. This goes well beyond the work of Al–Ashaab who considered only simple 2D shapes. His representation supported general feedback on the consequences of the product for mould system elements but could not support the concurrent design of a specific product and mould. The use of 3D geometry also has implications for the representation of injection moulding process capability as described in section 6.3.3.

To link the design of a specific product and a mould requires a knowledge of the equivalent cavity/core design, and the design of each of the mould system elements, so that the consequences of design decisions from the product viewpoint may be identified for the cavity/core and mould system elements and vice versa. The work in this thesis is focussing on geometry interactions between the elements in injection moulding design. Consideration of each of these elements effectively provides a distinct view of the manufacturability of the product and as such each representation requires a separate features description. This research has investigated support for the design of the following mould system elements; cavity, core, feeding system and cooling system in the mould.. The author has therefore considered five sets of manufacturing features (including the mouldability set). Whilst there are others it is considered that investigation of interactions between these five is adequate to provide a basis for the research investigation. In order to limit the scope of this work, only single impression integer moulds have been considered.

When designing the product it is necessary to consider functional requirements of the product in addition to mouldability requirements. One route to achieving this through features technology, is to constrain the designer to using mouldability features, eg Al–Ashaab (1994). However this represents a major constraint on the design process. It would seem more appropriate to allow the designer to design with features that provide the appropriate functionality for the product. This means that to offer manufacturability feedback to the designer a means is required of translating from the functional definition of the product to the mouldability definition. Thus, there is a requirement for a set of functional features in order to group functional design data, and associate this information with the particular product geometry. Therefore in addition to the manufacturing features, ie mouldability, cavity, core, feeding system and cooling system, the author has considered a set of functional features.

The representation of data in the Product model at any point in time represents the current state of the product and facilitates analysis from all required viewpoints, in order that the consequences of changes to the product may be assessed from all these viewpoints. While there are many attributes of a product that need to be considered in a design process, a basis for analysis between viewpoints is required. In this work the geometry is assumed to be the central attribute through which all viewpoints interact. Therefore each viewpoint of design must be able to be associated with the product form as it is developed, so that changes can be assessed with respect to new geometry and the combination of new and existing geometry. In this way translation between viewpoints can be achieved via a common association with the form.

Previous approaches to translation have been either i) to define the functional geometry in a conventional CAD system and then progressively replace each section of the geometry with a selection from a manufacturing feature library eg Cutcosky et al (1988), Hanada and Leifer (1989), Torbenhenau and Lianchum (1993), or ii) to bridge the gap between viewpoints in design using CAD, for example by taking a functional design geometry and changing it to a finite element representation for mouldflow analysis, or creating an offset cavity representation from the whole geometry of the product, eg Cinquegrana (1990)a, (1990)b, Tseng et al (1994), Gerdes, Webb, Chassapis (1994). The above translations provided feedback with respect to multiple viewpoints in design, but feedback was based on the whole product geometry, which had to be converted before feedback could occur from each viewpoint. Additionally there was no way of translating back to the previous representations and therefore it was not possible to re–examine an early viewpoint due to changes from a viewpoint later in the product life cycle. Thus there was no true interaction of functional and manufacturing constraints and true concurrency was not possible.

The author takes the view that the form features representation should provide the basis for linking of specific design viewpoint data. In this way the definition of a product from a particular viewpoint is a combination of the viewpoint specific data and the form. Whilst in other areas of application, for example machining, using common geometric features to relate viewpoints has had limited success, (Young and Bell (1993)), in the authors investigation of injection moulding it appears to be an appropriate approach. It is considered by the author that if viewpoint specific design data is linked to a common instance of form as other viewpoints, then backtracking is possible to earlier viewpoints in the product life cycle, and it is possible to support the proper interaction of functional and manufacturing constraints.

6.3. DEFINITION OF FEATURE TYPES TO ENABLE INTERACTION BETWEEN STRATEGIST APPLICATIONS.

6.3.1. A form features set capturing interaction constraints.

6.3.1.1. Basic form features set.

A set of forms has been derived from the rotational product ranges used in the research investigation. An illustration of these basic form features used in the research is shown in Figure 6.2. The forms are rotational wall, hollow rotational wall, spaced holes, spaced bosses, spaced ribs. Although Figure 6.2 shows a very restricted set of geometric forms, these forms are an adequate set to describe the rotational products used in the authors investigation and form the basic set for the majority of rotational products.

The author takes the view that the form features provide the common data that each design viewpoint shares. However in the investigation of form features, it has been found that ge-

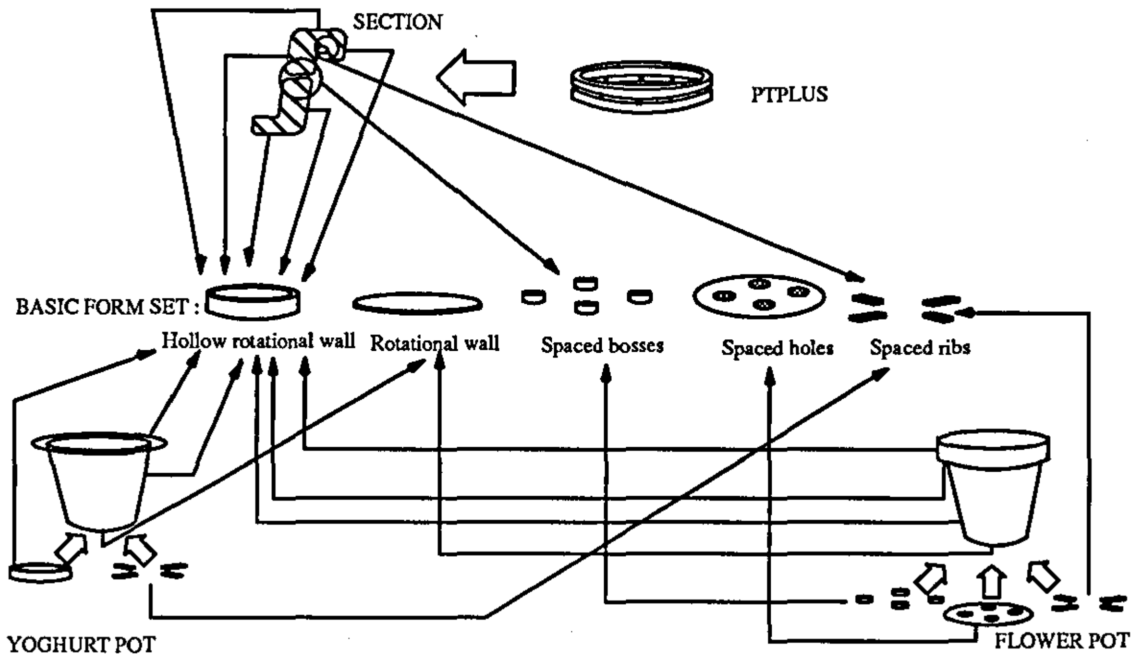


Figure 6.2. Identification of basic forms from case study product ranges.

ometry alone is not enough to support reasoning between the injection moulding design viewpoints. A critical aspect of a form feature, in addition to its geometry, is its relationship to other forms, and therefore there is a need to capture the relationships between forms.

Other authors who have considered form features and form feature relationships have looked at capturing form feature relations within specific design viewpoints. For example Mantyla et al (1994) identified the need to identify the relationships between forms, but this requirement was investigated in the specific viewpoint of functionality. Similarly, Dixon (1988) investigated the identification of relationships between forms in the specific viewpoint of mouldability. The representations of the type described by Mantyla et al (1994) and Dixon (1988) are important, and are also investigated in this thesis, shown in chapter 7. However this data is not adequate if the form features set provides a common data source for multiple design viewpoints, as it is not accessible or appropriate for other design viewpoints. Therefore extra data on form feature relationships that is viewpoint independent is required.

One way of achieving the viewpoint independent data relationships is to provide an enhanced form features set which captures the way in which one form feature can relate to another. The authors investigation into the problems involved in inter form relationships is discussed in section 6.3.1.2 and the definition of an enhanced form features set is described in section 6.3.1.3.

6.3.1.2. Inter form relationship problems.

An important attribute of any form feature is the way in which it can be connected to other form features, termed in this research the inter form relationship. The problem posed in the interaction of form features is that the connectivity between one form feature and another can be changed by their relative movement or the application of tapers to one or both features. For connectivity to be maintained it is important that the appropriate alignment is maintained between abutting faces on the form features. The ways in which breakdowns in connectivity can occur is dependant upon the spatial relationship that exists between the form features and the types of form feature involved.

There is a limited number of inter form relationships which can occur between form features in a rotational product. The author has identified six significant inter form relationships which occur frequently in rotational products, and the way in which form features can interrelate within them. These are illustrated in Figure 6.3, using a simple pot and a shallow plastic ring as examples. The six inter form relationships identified are A) a relationship where one hollow rotational wall is totally enclosed by another and aligned to the base, B) a relationship where one hollow rotational wall is partially enclosed by another and the tops are aligned, C) a relationship where a rotational wall closes off a hollow rotational wall and the outside surfaces are aligned, D) a relationship where a hollow rotational wall is placed onto the flat surface of a rotational wall, E) a relationship where one hollow rotational wall is closed off by another and the outside surfaces are aligned, F) a relationship where one hollow rotational wall is totally enclosed by another and aligned to the top. The problems of connectivity with respect to the above relationships and the definition of the resulting enhanced form features set are described in the following sub sections.

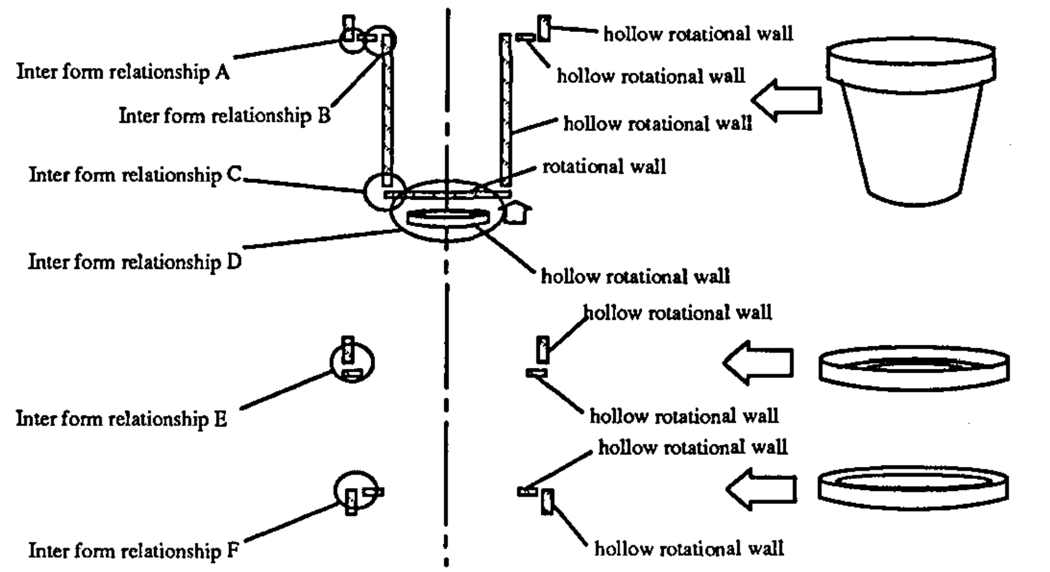


Figure 6.3. An enhanced form set is required to capture inter form relationships.

6.3.1.2.1. Inter form relationship A.

Figure 6.4 illustrates inter form relationship A. It can be seen by looking at the 'relative position constraints' that for this type of inter form relationship it is important that the base of the outer feature is flush with the base of the inner feature. If the inner form feature is not fully enclosed by the outer form feature, a thinning of section occurs at the point of contact between the two. If the inner form feature is away from the base of the outer form feature, a redundant feature is created. If the bases of the features are flush then the adding of tapers has no effect on their connectivity providing both features are tapered at the same angle, as shown in Figure 6.4 i). The following problems occur if the taper angles are different: If only the outer form feature is tapered or it has a larger taper angle than that of the inner form feature a loss of contact occurs between the form features as shown in Figure 6.4 ii). If only the inner feature is tapered or it has the larger taper angle then the features tend to merge together and the top surface of the inner is reduced as shown in Figure 6.4 iii).

If the bases of the inner and outer form features are not flush then the following problems occur if tapers are added: If the inner form feature is away from the outer form feature base it can be seen from Figure 6.4. iv) that any taper angle on the outer form feature causes a loss

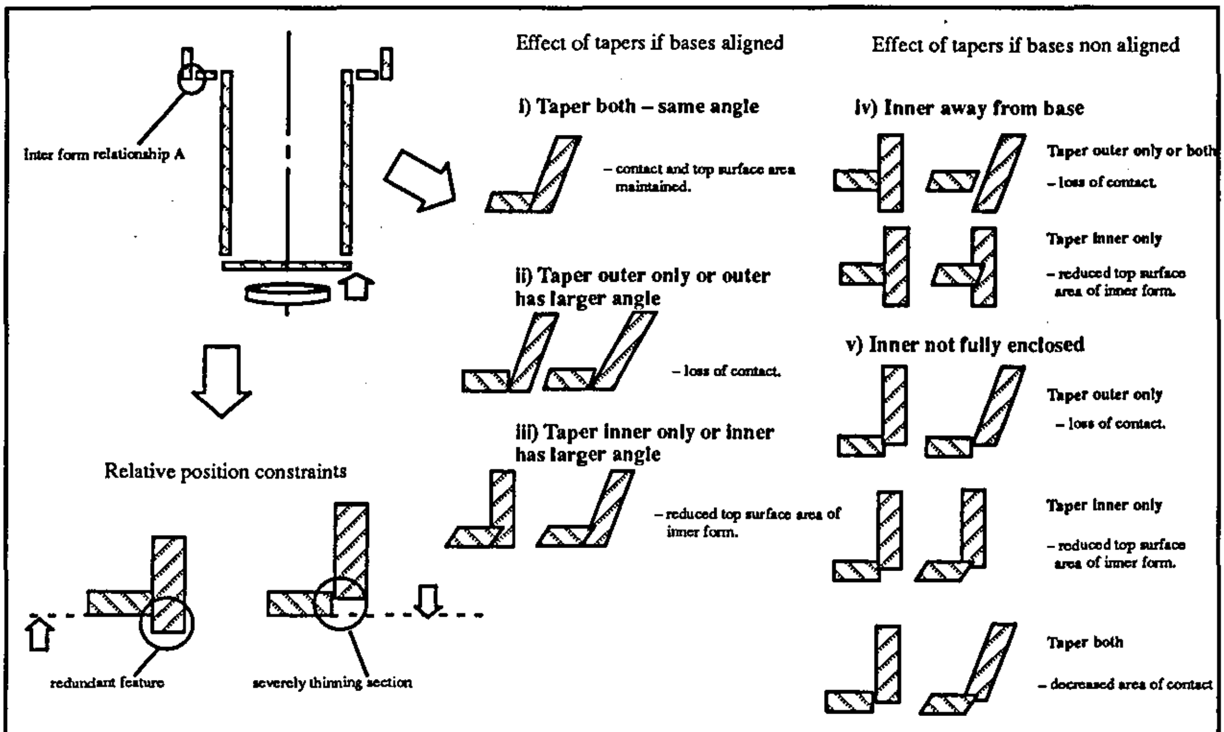


Figure 6.4. Characteristics of inter form relationship A.

of contact to occur, whatever the taper on the inner. A taper only applied to the inner form feature causes the features to merge together and the top surface of the inner is reduced. If the inner form feature is not fully enclosed by the outer it can be seen from Figure 6.4 v) that a taper only on the outer form causes a loss of contact to occur, and a taper only applied to the inner form feature causes the features to merge together and the top surface of the inner is reduced. Tapering both form features diminishes the extent of contact between the two.

6.3.1.2.2. Inter form relationship B.

Figure 6.5 illustrates inter form relationship B. It can be seen by looking at the 'relative position constraints' that for this type of inter form relationship it is important that the top of the outer feature is flush with the top of the inner feature. If only part of the outer form feature encloses the inner form feature, a thinning of section occurs at the point of contact between the two. If the outer form feature is away from the top of the inner form feature, a redundant feature is created. If the tops of the features are flush then the following problems occur with the addition of tapers: A taper applied to the inner form feature, whatever the taper on the outer, will cause the form features to merge together and the outer form feature may encroach

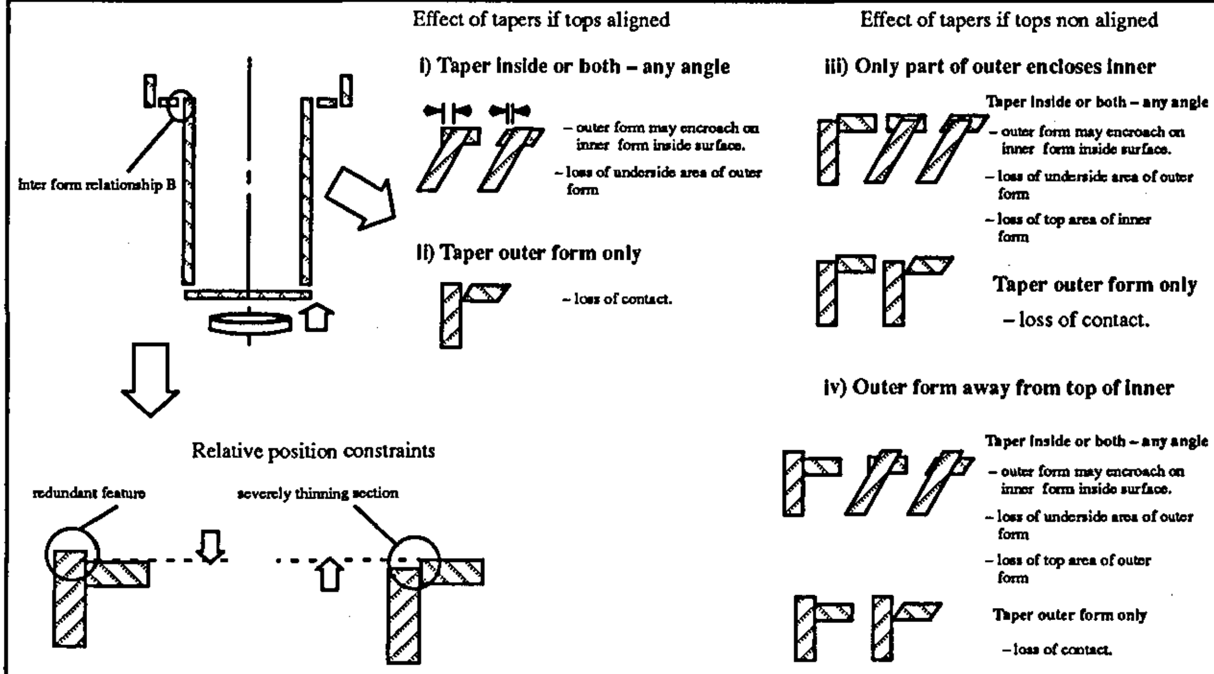


Figure 6.5. Characteristics of inter form relationship B.

on the inside surface of the inner form feature as shown in Figure 6.5 i). If a taper is added only to the outer form feature a loss of contact between the form features will occur, as in Figure 6.5 ii).

If the tops of the inner and outer form are not flush then the following problems occur if tapers are added: If only part of the outer form feature encloses the inner form feature it can be seen from Figure 6.5 iii) that a taper added to the inner form feature, whatever the taper on the outer, will cause the form features to merge together. The outer form feature may encroach on the inside surface of the inner form feature and the top surface of the inner form feature is reduced. A taper added only to the outer form feature causes a loss of contact to occur. If the top of the outer form feature is away from that of the inner form feature it can be seen from Figure 6.5 iv) that a taper added to the inner form feature, whatever the taper on the outer, will cause the form features to merge together. The outer form feature may encroach on the inside surface of the inner form feature and the top surface of the outer form feature is reduced. A taper added only to the outer form feature causes a loss of contact to occur.

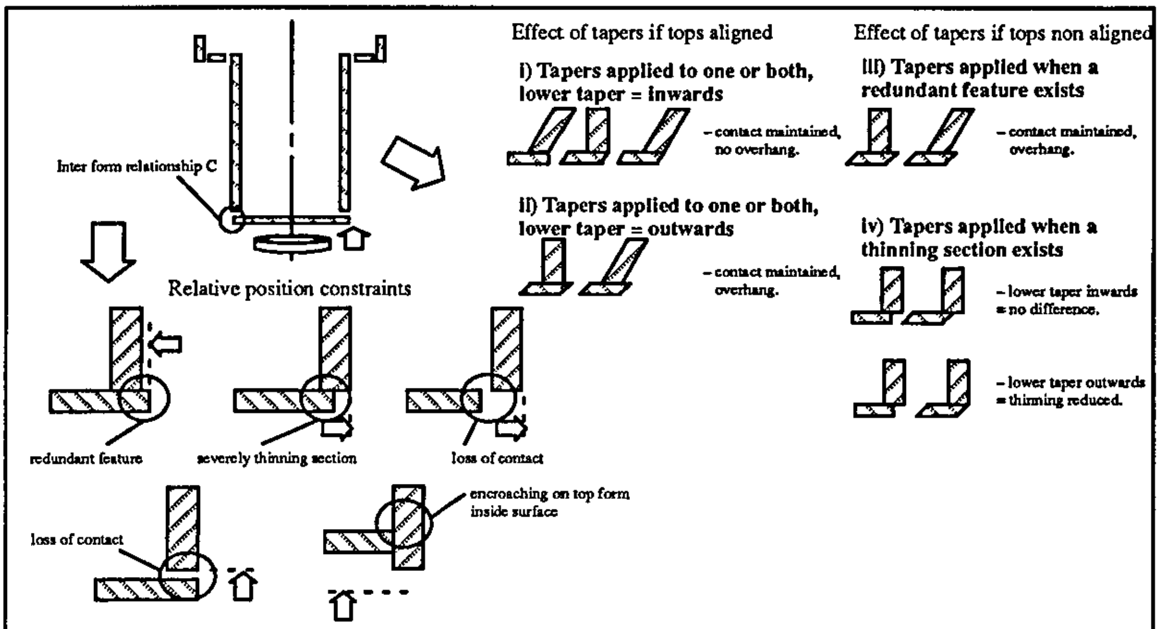


Figure 6.6. Characteristics of inter form relationship C.

6.3.1.2.3. Inter form relationship C.

Figure 6.6 illustrates inter form relationship C. It can be seen by looking at the 'relative position constraints' that for this type of inter form relationship it is important that the top of the lower feature is level with the base of the higher feature, and that the outside surfaces of the two form features are flush. If the outside surfaces are not flush, a smaller diameter on the lower form feature causes a thinning of the section at the point of contact or loss of contact between the features. A larger diameter on the lower form feature creates a redundant feature. If the top of the lower feature is not level with the base of the higher, a loss of contact can occur if the lower feature is away from the higher or the lower form feature can encroach on the inside surface of the higher.

If the outside surfaces of the higher and lower form features are flush then the addition of tapers has no effect on their connectivity, as long as the lower form feature is tapered in the opposite direction than the higher, as shown in Figure 6.6 i). However if the lower form feature is tapered outwards an overhang is created, as shown in Figure 6.6 ii).

If the outer surfaces of the higher and lower features are not flush then the following problems

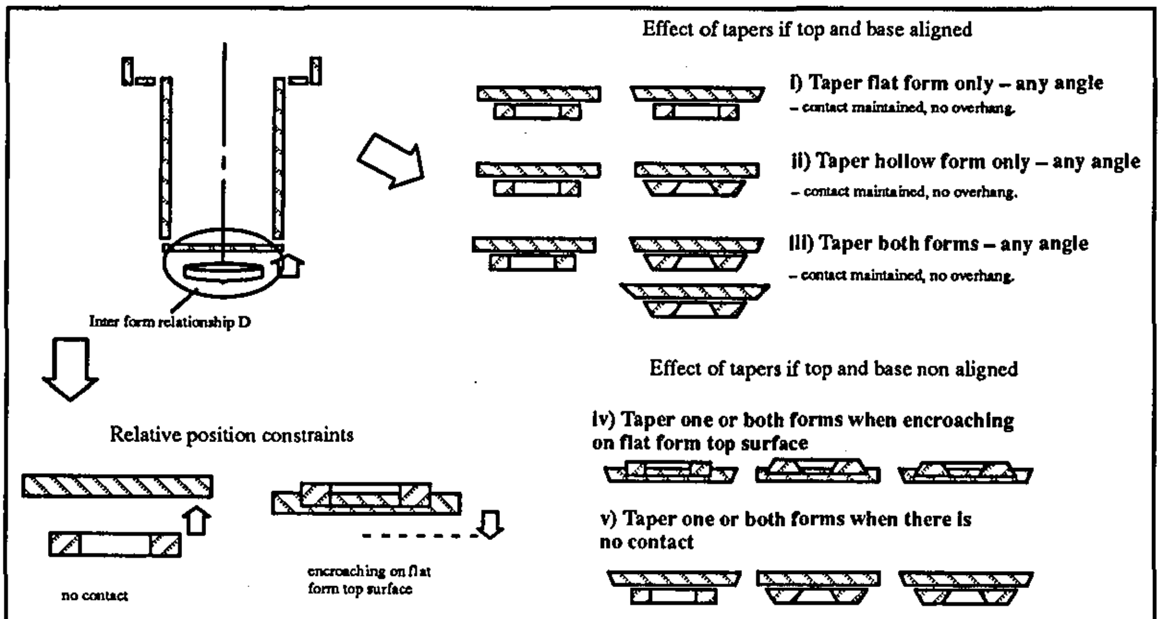


Figure 6.7. Characteristics of inter form relationship D.

occur if tapers are added: If the diameter of the lower feature is greater than the higher, and a redundant feature has been created, the addition of tapers does not change the connectivity, whether the taper on the lower form feature is applied inwards or outwards, as shown in Figure 6.6 iii). If the diameter of the lower form feature is smaller than that of the higher it can be seen from Figure 6.6 iv) that if the lower form feature is tapered inwards the connectivity is unchanged, whereas if the lower feature is tapered outwards the connectivity situation can be improved by increasing the extent of contact between the two form features.

6.3.1.2.4. Inter form relationship D.

Figure 6.7 illustrates inter form relationship D. It can be seen by looking at the 'relative position constraints' that for this type of inter form relationship it is important that the top of the hollow form feature is level with the base of the flat feature. If the top of the hollow form feature is lower than the base of the flat feature, a loss of contact occurs. If the top of the hollow form feature is too high it can encroach on the opposite surface of the flat feature onto which it was 'placed'. It can be seen from Figure 6.7 that the addition of tapers to one or both form features has no effect on the connectivity of the forms whether the relative position constraints are achieved or not.

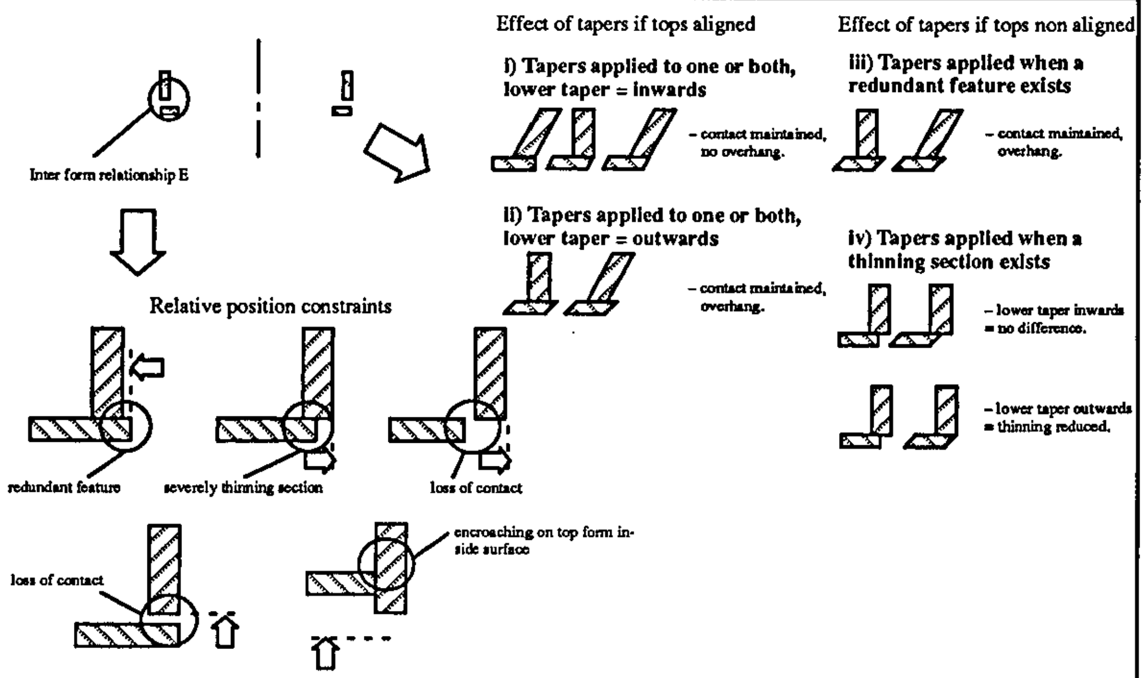


Figure 6.8. Characteristics of inter form relationship E.

6.3.1.2.5. Inter form relationship E.

Figure 6.8 illustrates inter form relationship E. It can be seen from the Figure 6.8 that the 'relative position constraints' and the effects of relative movement and taper additions are identical to those in Figure 6.6. Inter form relationship E involves two hollow rotational walls, whereas Figure 6.6. shows an inter form relationship between a hollow rotational wall and a rotational wall.

6.3.1.2.6. Inter form relationship F.

Figure 6.9 illustrates inter form relationship F. It can be seen by looking at the 'relative position constraints' that for this type of inter form relationship it is important that the top of the outer feature is flush with the top of the inner feature. If the inner form feature is not fully enclosed by the outer form feature, a thinning of section occurs at the point of contact between the two. If the inner form feature is away from the top of the outer form feature, a redundant feature is created. If the tops of the features are flush then the following problems

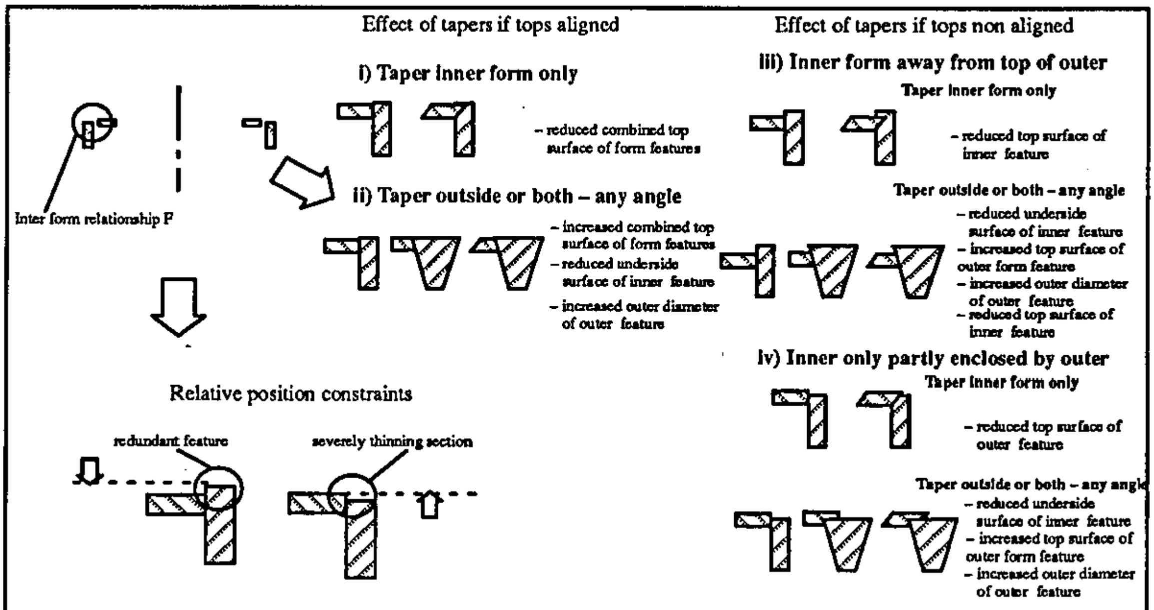


Figure 6.9. Characteristics of inter form relationship F.

occur with the addition of tapers: A taper applied to the inner form feature will cause the form features to merge together and loss of the features combined top surface as shown in Figure 6.9 i). If a taper is added to both form features or only to the outer form feature it can be seen from Figure 6.9 ii) that the underside surface of the inner form feature is reduced. The combined top surface of the form features and the outer diameter of the outer form feature are increased

If the tops of the inner and outer form are not flush then the following problems occur if tapers are added: If the inner form feature is away from the top of the outer form feature it can be seen from Figure 6.9 iii) that a taper added only to the inner form feature will cause the form features to merge together and the top surface of the inner form feature is reduced. A taper added to both form features or only to the outer form feature causes the underside surface of the inner form feature to be reduced. There is an increase in the top surface and outer diameter of the outer form feature, and a reduction in the top surface of the inner form feature. If the inner form feature is not fully enclosed by the outer form feature it can be seen from Figure 6.9 iv) that a taper added to the inner form feature will cause the form features to merge together and the top surface of the outer form feature is reduced. A taper added to both form features or only to the outer form feature causes a reduction in the underside surface

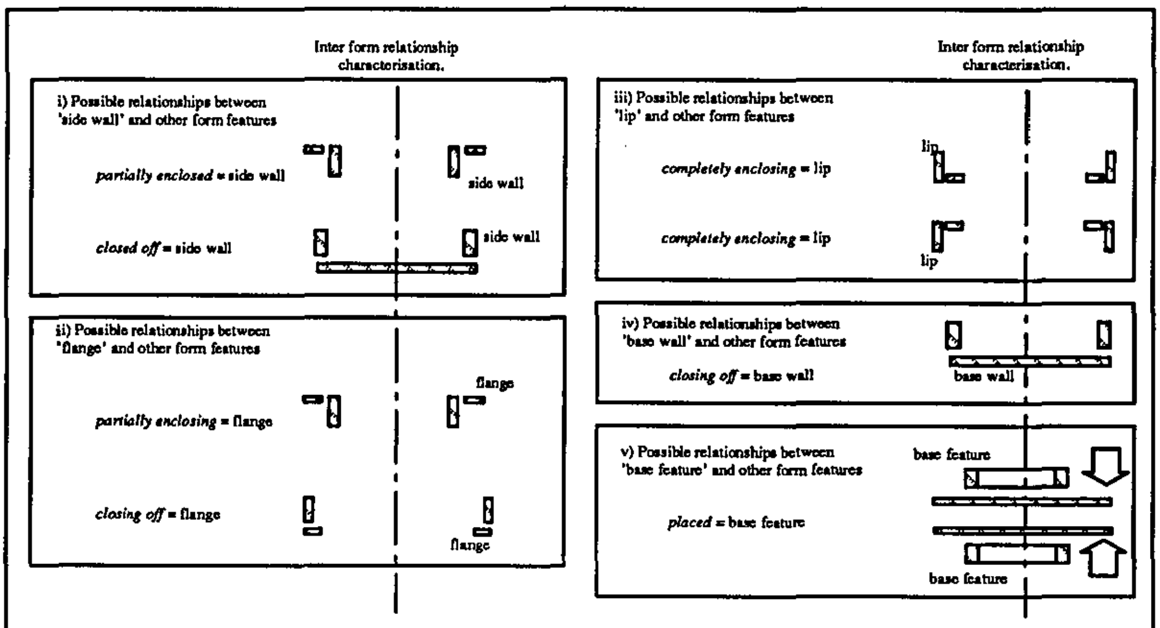


Figure 6.10. Inter form relationships associated with enhanced form features.

of the inner form feature. The top surface and outer diameter of the outer form feature is increased.

6.3.1.3. A set of enhanced form features which enable inter form relationships.

In order to identify an enhanced form features set to capture inter form relationships the author has characterised the spatial relationships between the form features in inter form relationships A) to F), as shown in Figure 6.10. For the inter form relationships described in the previous sub sections to be useful, there is a need to capture within individual form features the relevant attributes that enable the identification of these relationships. Therefore the author has defined enhanced form feature types which reflect one form features spatial relationship with another. The enhanced form feature set contains the following:– side wall, flange, lip, base wall,base feature, spaced bosses, spaced ribs and spaced holes, as shown in Figure 6.11. The nature of each enhanced form feature type which captures one form features spatial relationship with another is described below:

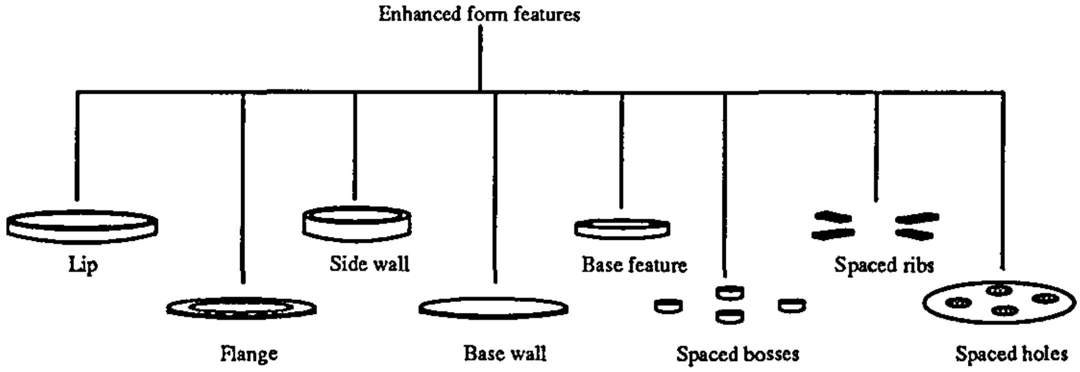


Figure 6.11. The enhanced form features set used in the research.

A ***side wall*** feature is the main body of a rotational product. A ***side wall*** can be 'partially enclosed' by another form feature and it can be 'closed off'.

A ***flange*** feature can 'partially enclose' and can 'close off' other hollow rotational form features.

A ***lip*** feature can 'completely enclose' other hollow rotational form features.

A ***base wall*** has a flat surface and can 'close off' a ***side wall.***

A ***base feature*** is a hollow rotational form feature that is placed on the flat surface of a ***base wall.***

When considering the spaced bosses, spaced ribs and spaced holes, the only addition to the basic form features is to capture one significant inter form relationship. The inter form relationship is the same as that shown in Figure 6.7 if the base feature is replaced by either spaced ribs or by spaced bosses. If the base feature is replaced by spaced holes the inter form relationship is similar to that shown in Figure 6.7, except the important alignment is between the top of the group of holes and the top of the flat form feature.

6.3.2. Functional features.

Functional design can be defined as the build up of geometry in the product in order to achieve the product functionality, eg Suh (1990), (1995), and many authors have defined a functional feature as a combination of function and form, eg Winjard et al (1992), Schute and Weber (1993) and Chakrabarti (1993). The authors agrees that a functional feature must be a combination of function and form and with the implicit assumption of the above, that to assess geometry with respect to achieving product functionality the function must be associated with form.

Previous approaches to the representation of functional design data have been either focussed on defining a general relationship between function and form, eg Kimura (1994) , Schmekel (1989), Schute and Weber (1993), or on the use of variant design, eg Dowlatshahi (1992), Cross (1989), Suh (1990), (1995). Taking the former approach offers support in the authors work, but it requires a universal set of functional features that is applicable over all products. If a functional feature is a combination of function and form the definition of such a set is unlikely as each product type has its own set of functional requirements with its own unique set of relationships between function and form, Lee and Young (1994)a. The utilisation of variant design as in the approaches of Dowlatshahi (1992), Cross (1989) and Suh (1990), (1995) offers the potential of building up a representation of the functionality of individual product ranges. Taking this approach would seem an appropriate mechanism to provide a link to an injection moulding strategist and seems appropriate to the types of product manufactured by the industrial sponsors of this work.

To define a functional features set the author has adapted the approach of Suh (1990), (1995) who divides overall product functionality into a set of functional requirements which must be satisfied by a set of design parameters. This approach has been adapted to provide a set of functional requirements of a selection of injection moulded products. These functional requirements can be associated with the form to provide a means of interaction with the mould-

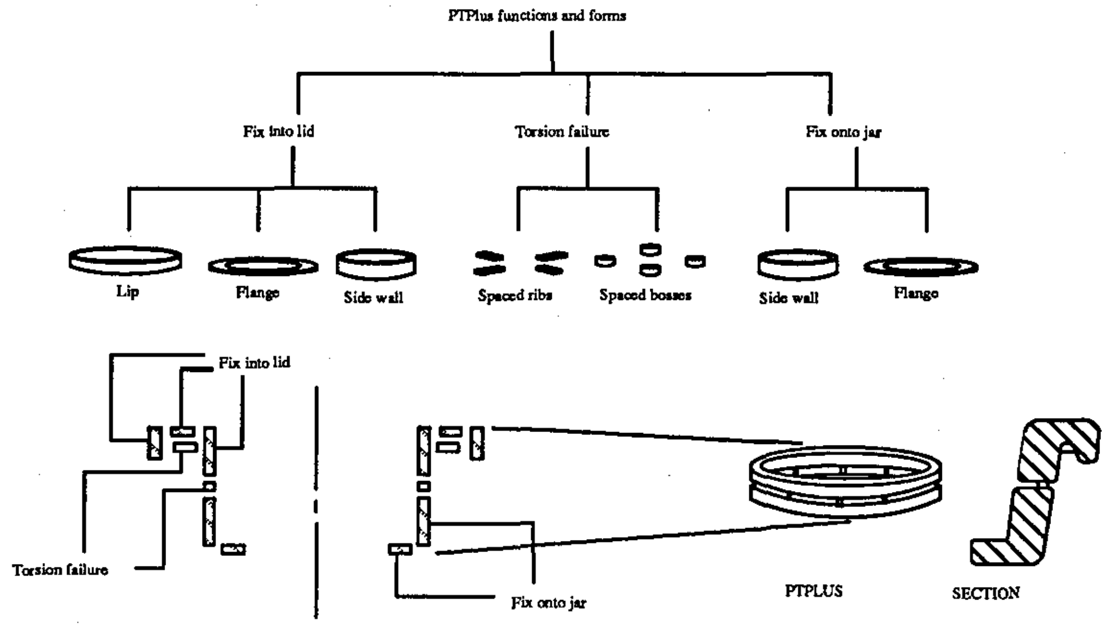


Figure 6.12. Example product range functions and forms to achieve them.

ability support application within the injection moulding strategist.

By studying examples of product ranges the author has identified the functions each product has to perform and the forms required. An example product is shown in Figure 6.12. The set of relations between the functions of a product range and the geometric forms that can be used to achieve each function is stored in the Product Range Model, which is described in Chapter 8. The authors definition of a functional feature is therefore the linking of a function of a product range and a form feature to achieve the function. This linking is driven by the designer as he builds up the product geometry to achieve a function on a particular product. This association is supported by the data in the Product Range Model. The structure of the Product Range Model to support this association is described in section 7.3.

6.3.3. Mouldability features.

For mouldability a group of features can be identified for which the rules are well known, eg Al–Ashaab (1994) (Figure 6.1), Hanada and Leifer (1989), Dixon (1988), Huh and Kim (1989). The author has identified a set of mouldability features which are: wall, rib, solid and

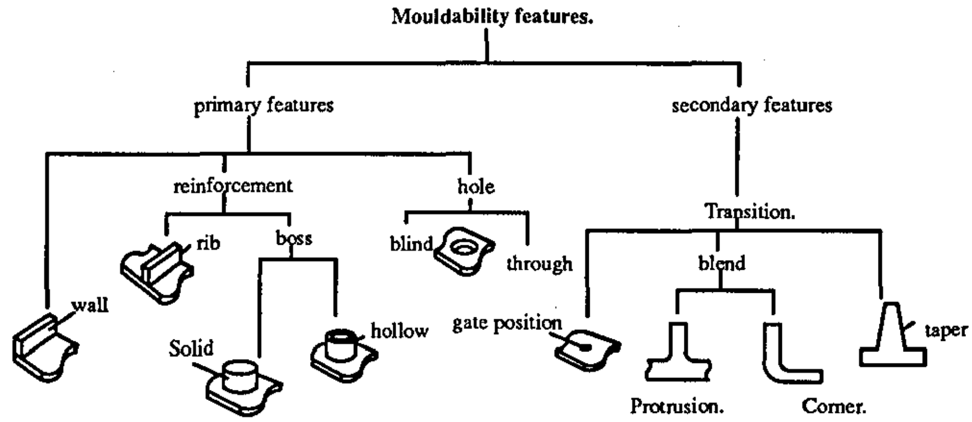


Figure 6.13. Mouldability features taxonomy.

hollow boss, hole, gate position, corner blend, protrusion blend and taper, as shown in Figure 6.13. Mouldability ribs, solid bosses and hollow bosses can be collectively termed 'reinforcement' features since they can all be used to reinforce product geometry and the design rules are also identical (see Chapter 8). Consideration of weld lines, ejection positions and parting line, which were included in the mouldability features set of Al–Ashaab (Figure 6.1), requires information on the configuration of the mould and therefore they cannot be considered at the mouldability design stage. The work of Al–Ashaab did not result in a mould design and therefore these elements were considered at the mouldability design stage using assumptions about a possible mould design.

The author has defined a taper as a separate mouldability feature as opposed to an attribute of a wall. A wall feature can be associated with varying geometry, and the way in which a taper is applied is dependant upon the nature of this geometry and its relationship with surrounding geometry. Thus a taper cannot just be applied to a mouldability feature, but is applied to one face or a set of faces.

The mouldability features set of the author has two types of blend, a protrusion blend and a corner blend. This is because when using 3D geometry a blend between two wall features (corner blend) has different parameters and mouldability process constraints than a blend between a wall feature and a rib feature or between a wall feature and a boss feature (protrusion

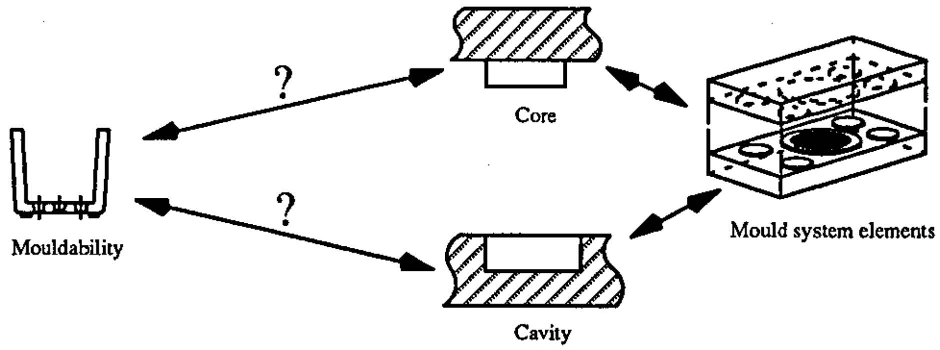


Figure 6.14. To support concurrent product design and mould **design** there must be a link between the mouldability and cavity core representations.

blend). The different mouldability process constraints for a corner blend feature and a protrusion blend feature are described in Chapter 8.

Only the mouldability features shown as 'primary' in Figure 6.13 have a direct relationship with form, ie only the wall, rib, solid and hollow boss and hole exist as a result of the existence of form. Other 'secondary' mouldability features exist as a result of the creation of a primary mouldability feature or a combination of primary mouldability features. The problem of translating between form features and the secondary mouldability features is discussed in section 7.4.

6.3.4. Cavity and core features.

As illustrated in Figure 6.14, in order for concurrency to occur between product and mould design a link must be established between the mouldability features and the cavity and core features, so that the meaning of changes to one viewpoint can be identified for the other. Additionally it must be possible to re–examine the mouldability viewpoint with respect to the consequences of cavity and core design decisions. Therefore a cavity and a core features set that provide a basis for such links to the mouldability viewpoint must be identified.

All previous work on creating a cavity representation has taken the approach of subtracting the complete product geometry from the standard mould plate. Some authors have used solid geometry eg Cinquegrana (1990)a, (1990)b, Tseng et al (1994), and some surface representations, eg Rho et al (1990), Gerdes, Webb, Chassapis (1994). None of the above work has addressed the issue of cavity design modification as the product geometry is developed, for example in order to eradicate undercuts in the mould.

Where the generation of mould cores has been addressed the whole product geometry has been the input to the process and core blocks are created using Boolean operations in a solid modeller, eg Kang et al (1990), (1992), Tseng et al (1994). Only simple shallow components have been examined eg Kang et al (1990), (1992), or components with simple geometry and hence no undercuts, eg Tseng et al (1994). Some work has examined the identification of possible parting directions on the complete component based on undercut detection, eg Mochizuki and Yuhara (1992), Hui and Tan (1992), (1994). However this work is focussed on complex algorithms for retrospective examination of the whole product geometry.

It has been identified by the author that the use of geometry alone as the input to cavity and core design breaks the link between the structure of the product geometry and that of the cavity and core, and there is no way back to re–examine mouldability in response to changes to the cavity or core. It is considered by the author that re–examination of the mouldability viewpoint with respect to cavity/core changes is only possible if a link can be established between the individual parts of the mouldability geometry and the corresponding parts of the cavity and core geometry. Therefore the cavity and core features set have been identified by the author to provide a basis for such links between mouldability geometry and that of the cavity and core. The author has identified a set of cavity 'impression' features, which are: cavity volume, cavity rim, cavity slot, cavity hole, cavity boss, cavity blend, cavity taper and cavity group volume, as shown in Figure 6.15. A set of core 'impression' features has been identified, which are: core volume, core rim, core slot, core hole, core boss, core blend, core taper and core group volume, as shown in Figure 6.16. The interaction of the cavity and core 'impression' features with the mouldability features set is discussed in chapter 7.

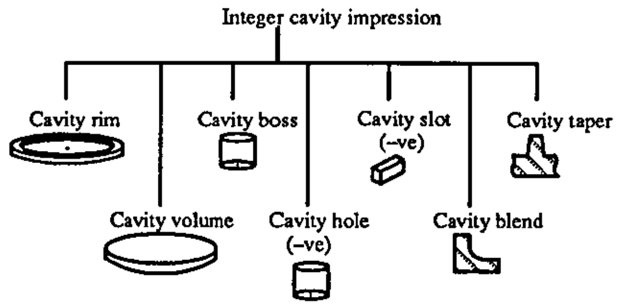


Figure 6.15. Integer cavity impression features.

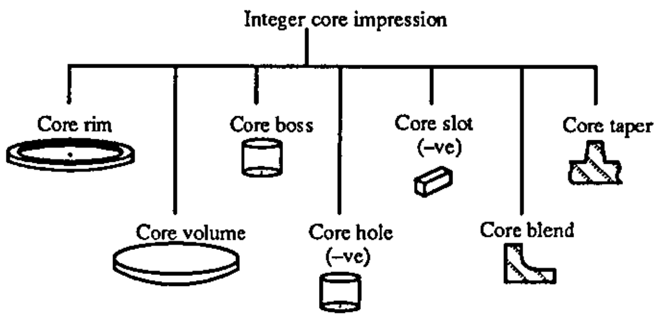


Figure 6.16. Integer core impression features.

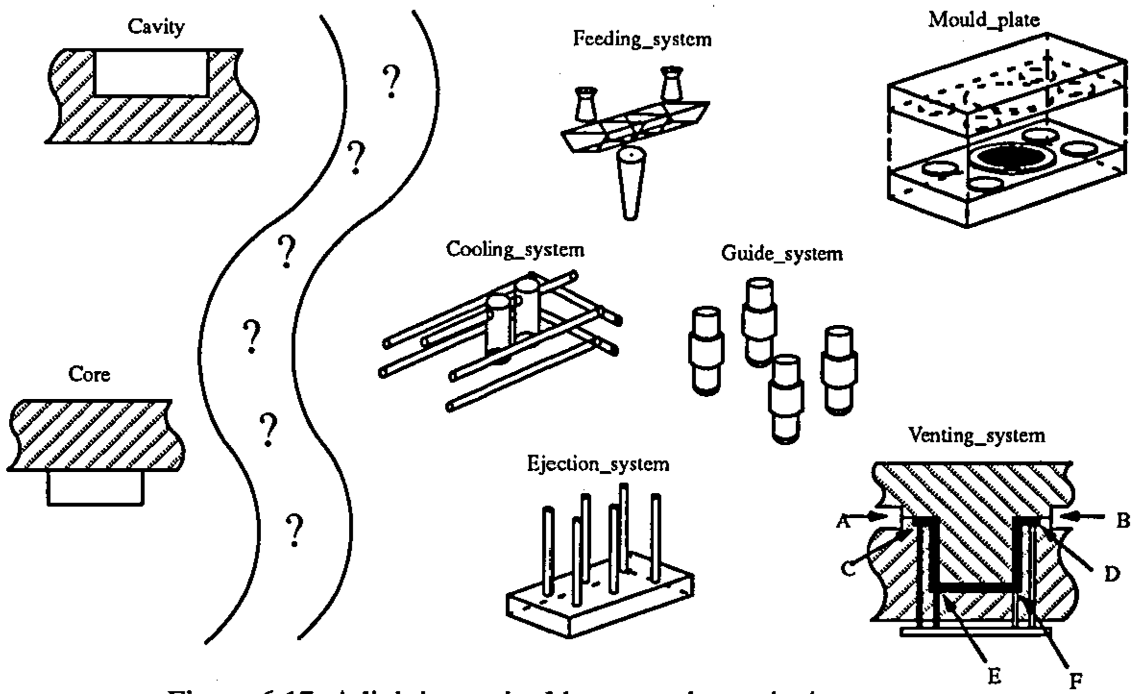


Figure 6.17. A link is required between the cavity/core and the other mould system elements

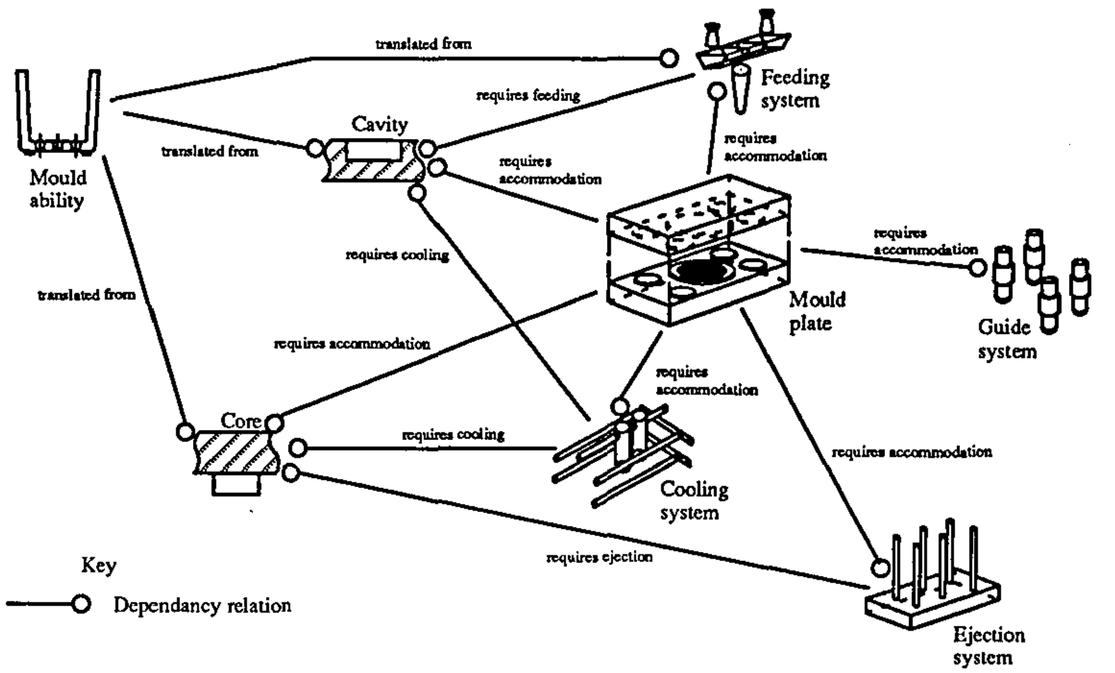


Figure 6.18. Multiple relationships between mould system elements

In order for concurrency to occur between product design and all the viewpoints in mould design, the cavity and core features sets must provide a bridge between the product geometry and the mould system elements. As illustrated in Figure 6.17, to provide a bridge requires links between the cavity/core features and the mould system elements, as well as between the cavity/core features and the mouldability features set. However as shown in Figure 6.18, a major problem is that multiple relationships exist within the mould between mould system elements. It can be seen from Figure 6.18 that many of these relationships are not focussed on the cavity/core and therefore the cavity/core features alone are not sufficient to facilitate consideration of all viewpoints in mould design and still maintain links to the product geometry and product design viewpoints. In order to enable the cavity/core to act as a bridge between the product and mould a method is required of focussing the relationships within the mould onto the cavity/core.

The author was able to accomplish the focussing of relationships within the mould onto the cavity /core by combining the features set of the cavity impression and the cavity mould plate and combining the features set of the core impression and the core mould plate, as shown in Figure 6.19. With respect to the cavity, the author has identified an expanded cavity features

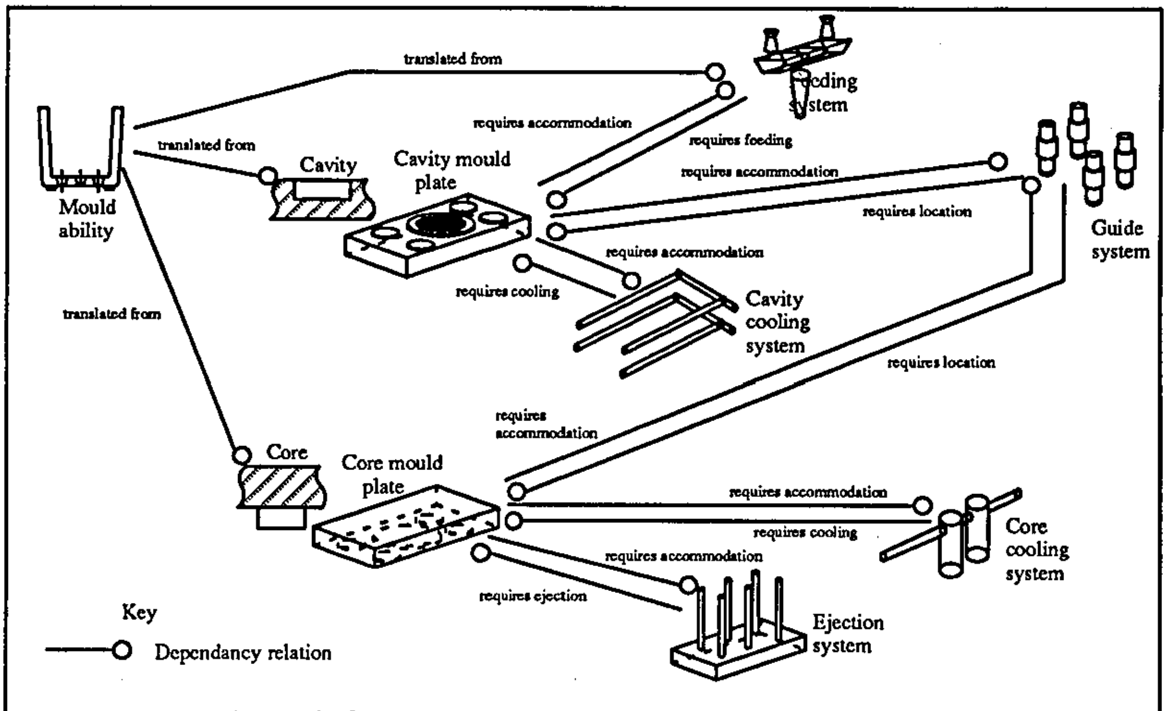


Figure 6.19. Focussed relationships to group key interactions.

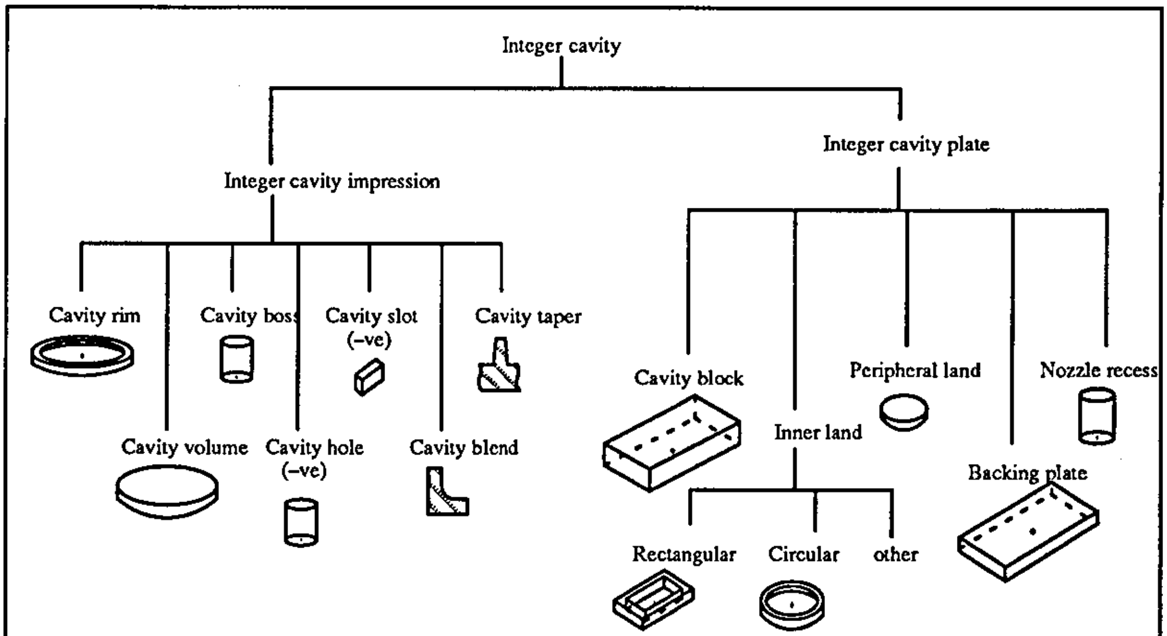


Figure 6.20. Complete integer cavity features set.

set which includes the following cavity 'plate' features: cavity block, cavity inner land, cavity peripheral land, cavity backing plate, cavity nozzle recess, as shown in Figure 6.20. The cavity plate features set includes all entities that can occur in a single cavity integer mould plate in the form considered in the present work. The interaction of the cavity 'plate' features with the mould elements and their relationship with the cavity 'impression' features is de-

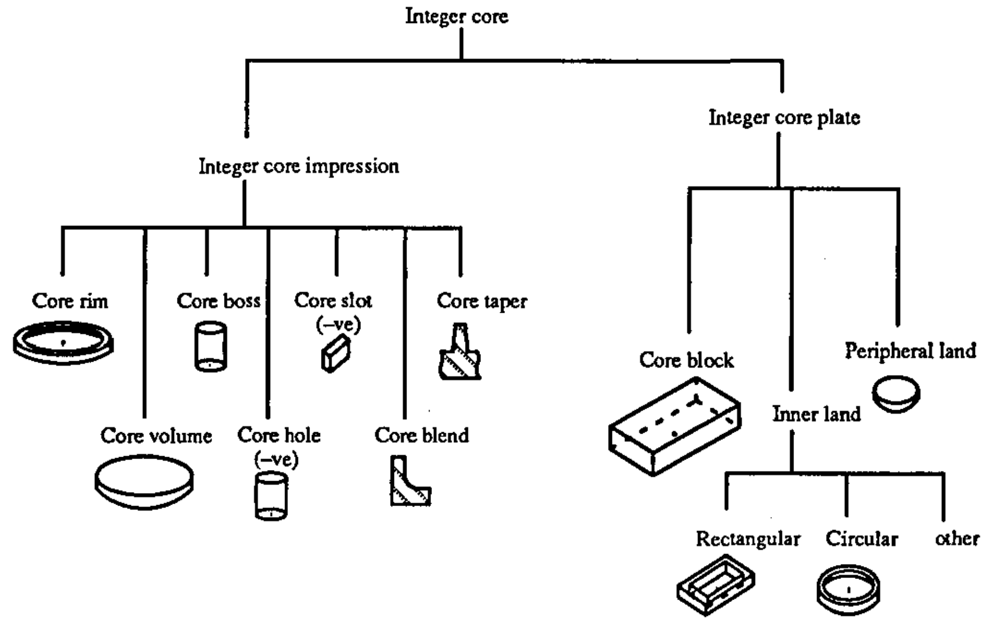


Figure 6.21. Complete integer core features set.

scribed in chapter 7.

The core has been treated similarly to the cavity and the author has identified an expanded core features set which includes the following set of core 'plate' features: core block, core inner land, core peripheral land, as shown in Figure 6.21. The core plate features set includes all entities that can occur in a single cavity integer mould core plate in the form considered in this work. The interaction of core 'plate' features with the mould elements and their relationship with the core 'impression' features is described in chapter 7.

6.3.5. Feeding system features.

The feeding system features set must capture all permutations of feeding system type and configuration available to the designer. The feeding system features set must be such that a basis is provided for appropriate links between the feeding system geometry and that of the cavity/core and the product.

All previous work has retrospectively examined the feeding system using the whole product

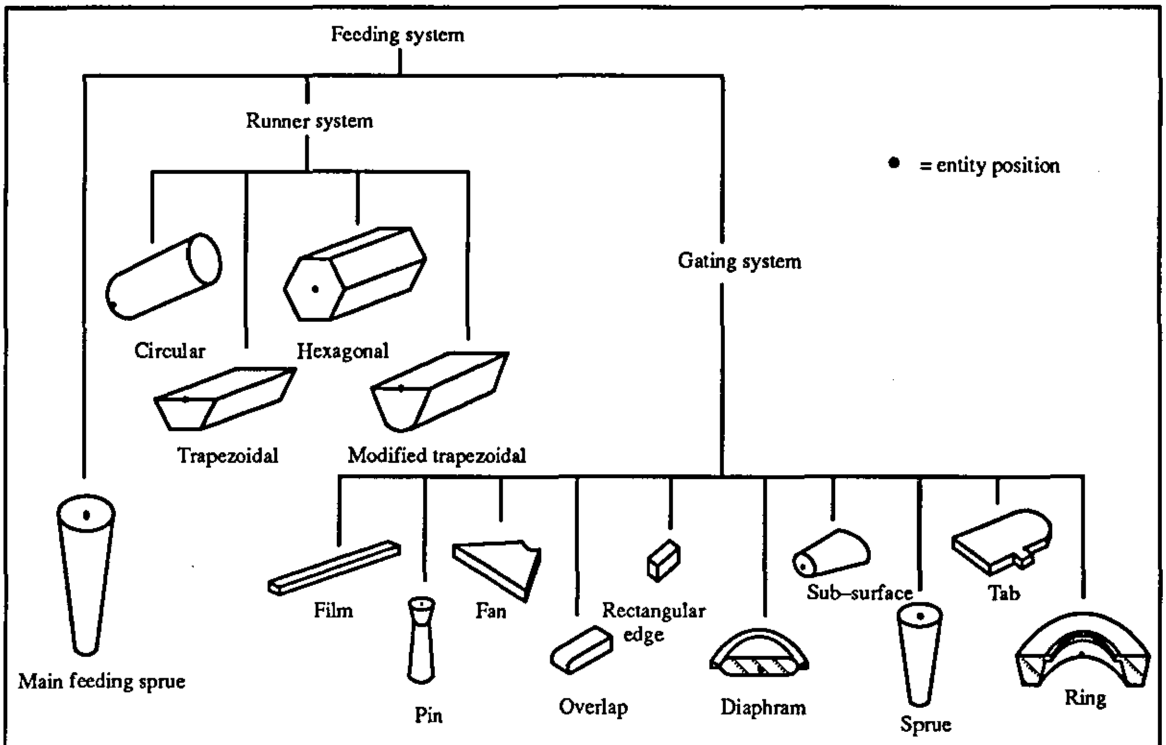


Figure 6.22. Feeding system features set.

geometry as the input. The approach has either been to examine runner and gating systems in terms of machining features, as the input to an NC machining process planner, eg Rho et al (1990), or to use simulation techniques to automatically generate the feeding system using standard parts and predetermined rules on system configuration, eg Cinquegrana (1990)a, (1990)b. None of the above work has considered the issue of feeding system design modification as the product geometry is developed.

In order to allow critical decisions to be taken in support of the design of the feeding system, and to enable interactions between the design of the product, the cavity/core and the feeding system, the necessary components of a feeding system features set are considered by the author to be the gate types, the choice of runner systems, and the main feeding sprue. Therefore the author has identified a set of feeding system features which includes i) all conventional gate types: ring gate, tab gate, sprue gate, sub–surface gate, diaphragm gate, rectangular edge gate, overlap gate, fan gate, pin gate, film gate, ii) all conventional types of runner system: trapezoidal runner, modified trapezoidal runner, circular runner, hexagonal runner, and iii) the main feeding sprue, as shown in Figure 6.22. The features set in Figure 6.22 is the basis for the linking of the feeding system geometry to that of the product and the cavity/core.

6.3.6. Cooling system features.

The cooling system features set must capture the possible types and configuration of the cooling system, available to the designer. The cooling system features set must be such that a basis is provided for the appropriate links between the cooling system and the cavity/core and feeding system.

Similar to feeding system design, all previous work has retrospectively examined the cooling system using the whole product geometry plus the completed feeding system design as the input. The approach of previous work has either been to examine cooling systems in terms of machining features, as the input to an NC machining process planner, eg Rho et al (1990), or to automatically generate the cooling system using standard parts and predetermined rules on system configuration, eg Cinquegrana (1990)a, (1990)b. No work has addressed the issue of cooling system design modification as the product geometry is developed.

The cooling system feature set must represent all the options of cooling system configuration and geometry for the designer, and allow him/her to decide what configuration and geometry is most appropriate, given the current state of the cavity/core and feeding system. In order to allow critical decisions to be taken in support of cooling system design, the author has identified a set of cooling system features, which are: standard flow way, baffle flow way and baffle blade, as shown in Figure 6.23. The features in Figure 6.23 are the basic elements of any cooling system configuration in terms of the geometry inside the mould. The cooling features set is the basis for the linking of cooling system geometry to that of the product, the cavity/core and the feeding system, thereby extending design links into the mould system elements, as described in section 7.8.

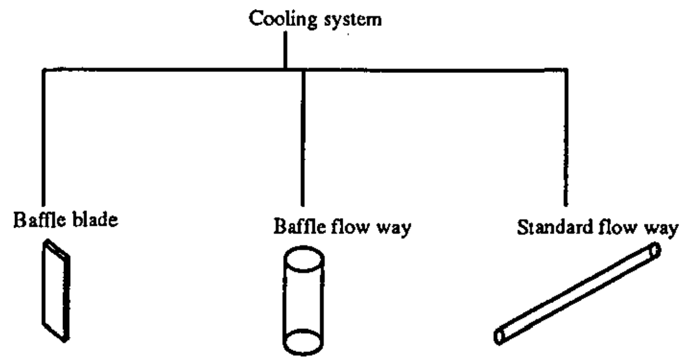


Figure 6.23. Cooling system features set.

Chapter 7.

Feature translations and application strategies to enable support for concurrent design for injection moulding.

7.1. INTRODUCTION.

This chapter builds on the work in Chapter 5 by investigating the requirements of the key modules within the design for injection moulding strategist applications. Each application has been defined as having a strategy module and a translation module to enable the interactions of applications. In this chapter it is shown possible to define translation routines which support the interactions between strategist applications. The work described in this Chapter investigates the detailed functionality of the modules which provide these translation processes and the design support strategies. Section 7.2 discusses the need for the general module functionality, Section 7.3. discusses the translation process and the design support strategy of the functional interface which provides the initial interpretation of functional design data into a form suitable for design for manufacture. Section 7.4 discusses the translation process and the design support strategy of the mouldability assessment modules. Sections 7.5 and 7.6 discuss the translation process and the design support strategy of the cavity/core assessment modules with respect to the cavity and the core respectively. Sections 7.7 and 7.8 discuss the translation process and the design support strategy of the mould system element assessment modules with respect to the feeding system and the cooling system respectively.

7.2. Overview of module functionality.

7.2.1. Design support strategy.

Research in concurrent injection moulding design support systems typically constrains the designer to using mouldability features to define the shape of the product, eg Dixon (1988), Al–Ashaab (1994), Hanada and Leifer (1989). Mould design has been considered retrospectively, and the issue of examination of mould design constraints as the product geometry evolves has not been solved, eg Cinquegrana (1990)a, (1990)b, Gerdes, Webb, Chassapis (1994), Rho et al (1990). In order to support concurrency, the author takes the view that design support systems should aim to: i) To provide the designer with manufacturing consequences of **functional** product geometry as the design evolves. ii) To provide the designer with all options and consequences for mould design as the product geometry evolves.

To achieve these aims an overall strategy is required to control the interactions of support applications so that the design process is as concurrent as possible. The limitations imposed on such a strategy have been identified in Chapter 5, where an IDEF0 activity model showed that all areas of injection moulding design cannot be addressed at once. Thus the elements of a strategy can only be executed when the appropriate information can be made available. For example the work in Chapter 5 has shown that whilst product geometry can be available to support analysis of a product design from multiple viewpoints as the design evolves, a prerequisite of mould design is identifying the parting line. Thus concurrent product and mould design is only possible if at any juncture of product development a parting line can be identified to enable reasoning about mould design, and also the consequences of mould design changes for product design can be examined.

The investigation in Chapter 5 led to a general application structure (Figure 7.1) which reflects the overall design support strategy. The information limitations on the interaction of the strategist modules has led to an overall strategy of the following form: To support concurrent *product* design every change from the functional viewpoint (functional interface activity) must trigger the mouldability support application to provide mouldability feedback in response to design changes. Re–analysis (backtracking) must be available at any juncture in the product design process to re–analyse functionality in response to mouldability or mould changes. Functional re–analysis must also trigger mouldability re–analysis. The designer must also be able to perform a mould design process at any juncture in the product design

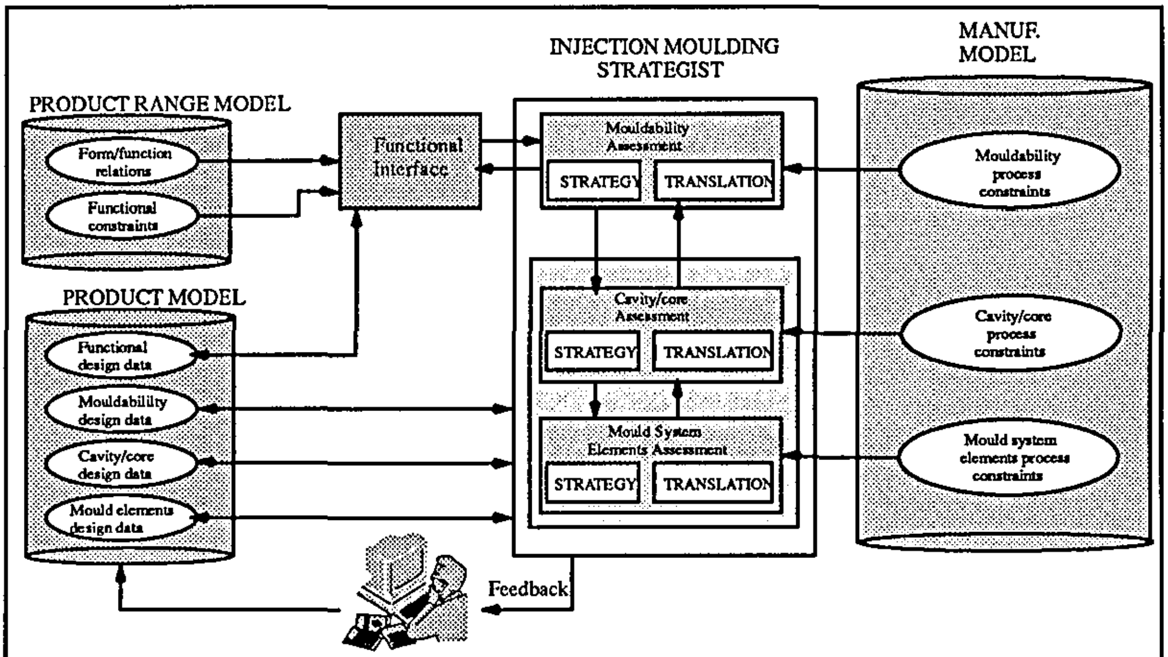


Figure 7.1. Strategist and information model interactions.

process, and it must be possible to repeat the mould design cycle as many times as required at each juncture, so that the designer can examine all options for the configuration of a mould as the product evolves. A simple flow diagram of this strategy is provided in Figure 7.2.

7.2.2. Translation mechanisms.

Each application strategy is dependant upon the availability of information which it needs as input. To obtain this information a translation process is required. The requirement is for translation mechanisms to provide feature instantiations which can be associated with data

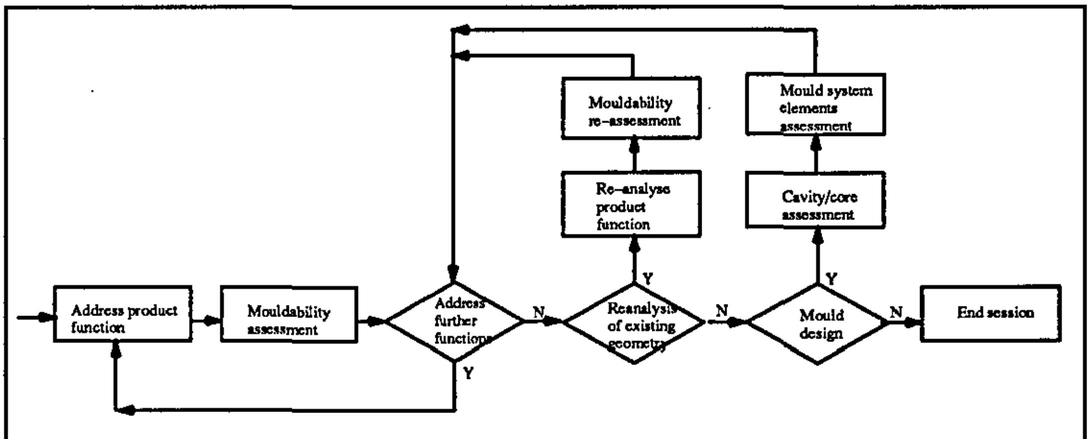


Figure 7.2. Overall design support strategy

in the information models, providing data input to an application strategy. The application strategies are driven by the relevant feature instantiations within the Product model and therefore the requirement of a translation mechanism is principally to provide the appropriate feature instantiations for each type of strategy.

Consideration of feedback advice potentially can change feature instantiations, and to enable concurrency these changes must be reflected in each of the corresponding feature types in the Product model. This is necessary in order that the consequences of design decisions from one viewpoint can be considered from another. The author takes the view that this feedback problem can be overcome by the creation of a link between the new feature instantiation and that instantiation from which it was translated.

Translation processes must support each of the application strategies and are therefore required to provide i) form feature instantiations linked to corresponding functional specification data, ii) mouldability instantiations linked to corresponding form instantiations, iii) cavity instantiations linked to corresponding mouldability instantiations, iv) core instantiations linked to corresponding mouldability instantiations, v) feeding system instantiations linked to corresponding cavity instantiations, vi) cooling system instantiations linked to corresponding cavity and core instantiations. How each of these translation processes can be achieved is discussed in the following sections.

7.3. The Functional interface.

Design by feature systems commonly have the drawback of over constraining the designer by enforcing the use of manufacturing features to build up product geometry. The designer should be able to construct designs from a functional point of view and therefore the author takes the view that the product design should be created using a range of functional features. The work of others has considered functional features by defining form from a functional viewpoint, eg Shah and Rogers (1988), Winjard et al (1992), but no work has considered the interaction of such a features set with separate manufacturing features, whose form is derived from the manufacturing viewpoint. The focus of the author has been to achieve an instantiation of functional data linked to a separate form instantiation so that geometry can be the central attribute through which application strategies interact.

7.3.1. Problems in the instantiation of linked function and form.

The instantiation of linked function and form is the front end of the interaction process between viewpoints in injection moulding design because this activity is the first to capture the form. Investigation has revealed the following problems:

i) A source of data is required which allows the designer to select a form feature to achieve a function.

ii) A method is required of evaluating the functionality of a form feature instantiation.

iii) Specification data must be available against which to evaluate functionality.

iv) Functionality may relate to the relationship with existing geometry as well as the attributes of a new form. A mechanism is therefore required to ensure context setting for new form feature instantiations.

7.3.2. Use of product ranges to support the instantiation of linked function and form.

The authors approach to the solution of these problems is based on the use of product ranges. The first problem was to support the designer with a knowledge of form features which can be used to achieve a particular function. This can be done by defining a product range which contains the necessary functions of the product type and the possible forms which can be used. An example of this can be found in Figure 6.12.

The problem of evaluating functionality can be overcome by defining a set of constraints which apply to each function within the product range. The detailed structure of this data to support the evaluation of an independent form set is discussed in Chapter 8.

Thirdly as the designer builds up the geometry of a product, specification data is required against which the functional constraints in the product range can be evaluated. The approach of Shah and Rogers (1988) was to provide a specification in the product representation, defining 'technological features' which contained information relating to performance and operation, including performance parameters and design constraints. Winjard et al (1992) also put forward a Product model structure of coupled 'technological elements' and 'form feature entities'. The approach here uses the specification data for each product function in a similar way to Shah and Rogers (1988) use of their 'technological features', as follows: The product range must contain data to support the designer in providing a specification for a product function. When a designer selects a product range function to be achieved on a particular product, the specification data provided by the designer is instantiated in the Product model. The combination of specification data and the linked form feature selected by the designer make up the functional representation of the product. An example of these relationships leading to a build up of a functional representation in a Product model is shown in Figure 7.3. Product range data is captured in the Product Range Model as discussed in Chapter 8. An example showing the relationships between the Product Range Model, functional interface and the Product model is illustrated in Figure 7.4.

A fourth problem is that many functional constraints require a knowledge of adjacent existing geometry for evaluation of functionality. Therefore the issue of identification of adjacent geometry must be resolved. This requirement has been recognised by many authors when investigating the representation of functional constraints, eg McGinnis and Ullman (1992), Linberg (1992), Sivard, Linberg and Agerman (1993). One option was to use the Product

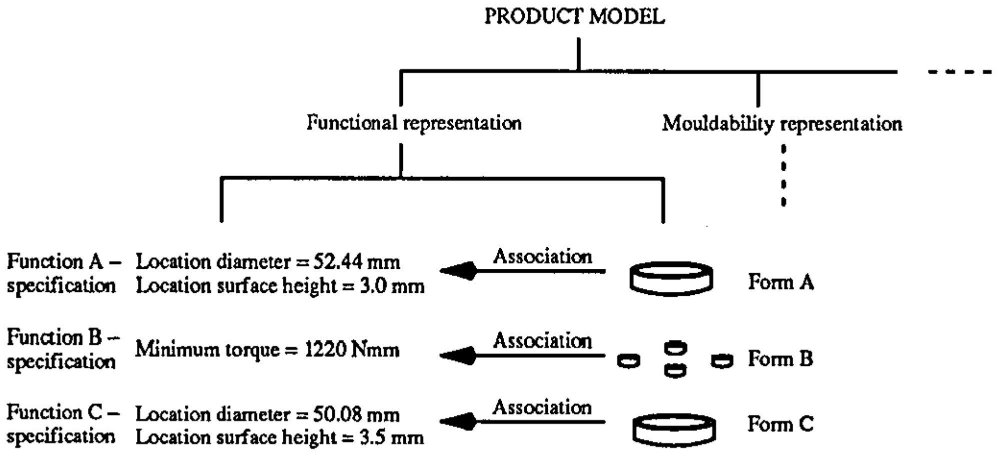


Figure 7.3. Product model example of relationships built up in the functional representation.

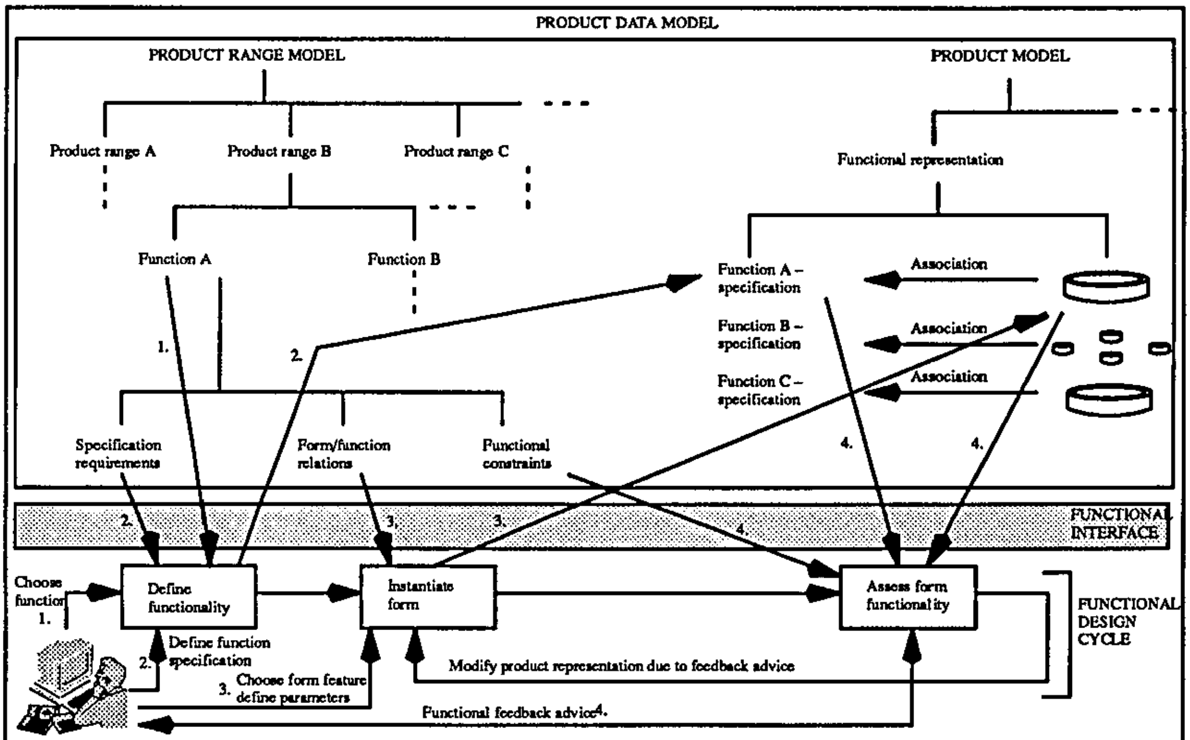


Figure 7.4. Use of Product Range Model data to support the association of function and form.

model to identify the adjacent form using spatial relationships, as in the approaches of Linberg (1992) and Sivard et al (1993). However this approach does not resolve the problem that a new instance of a form feature may be adjacent to several existing form feature instances, and the adjacent form features that affect functionality have to be identified. The approach taken in this thesis to solve this problem has been to look for the type of adjacent form feature that is identified as being significant to functionality in the Product Range Model. By looking

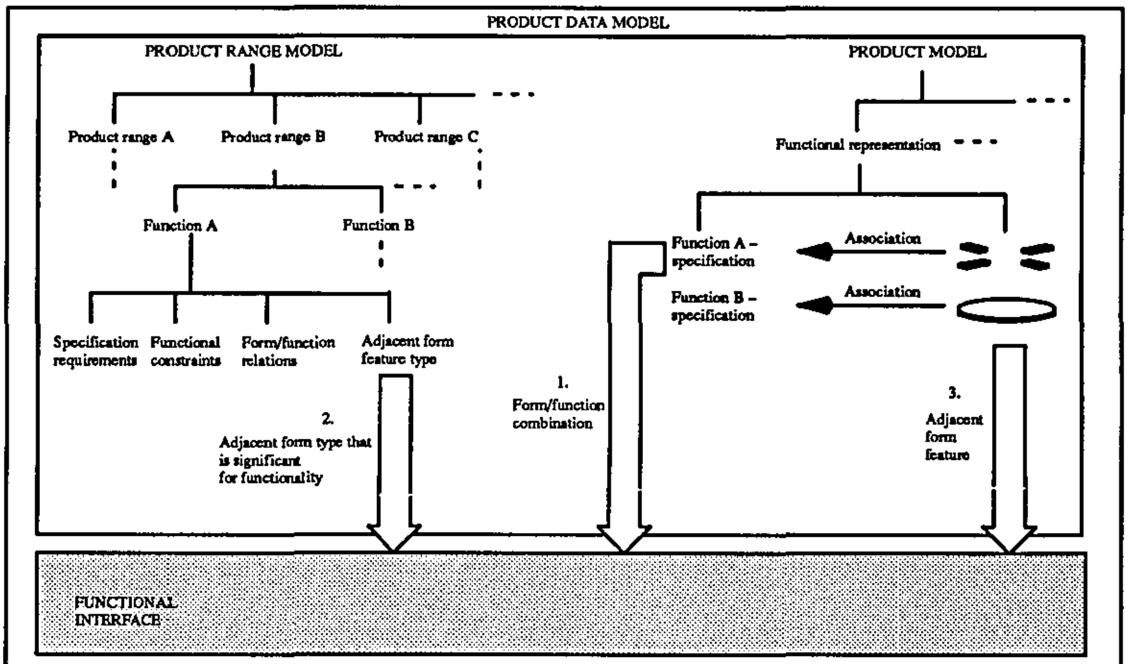


Figure 7.5. Use of Product Range Model data for identifying adjacent form features.

at example product ranges such as in Figure 6.12 it has been identified that for each form feature that helps make up the shape of a product, the type of adjacent geometry is dependant upon the function that it is achieving plus the type of the form feature itself, ie the method that has been chosen to achieve the functionality. The Product Range Model therefore includes this data within each product range function. If several pieces of geometry are adjacent to a new form feature instantiation in the Product model, only that type of form feature specified in the Product Range Model as affecting functionality is identified. This method is shown in Figure 7.5.

7.3.3. Functional design support strategy.

When using product ranges the designer is required to follow a particular design cycle: The designer is supported in choosing a product function to address from the product range and in defining a specification for the function. Support is provided in choosing a form type to achieve the function and functional feedback advice enables the designer to consider the dimensions of the form in relation to achieving the functionality. This cycle may be repeated until the build up of the product is completed. The functional design support strategy which supports the design cycle is illustrated in Figure 7.4.

Utilising product range data and the functional design support strategy a functional interface can be defined which enables the demonstration of the concurrent interaction of functional and manufacturing constraints as described in the remainder of this text. The structure of the functional constraints data within the Product Range Model is discussed in detail in chapter 8. Chapter 9 describes the design of an experimental software system that incorporates the ideas that have been described and discussed above.

7.4. Mouldability assessment.

The author has defined a mouldability features set in Chapter 6 which is modified from that of Al–Ashaab, as shown in Figure 6.13. Previous work typically constrained the designer to using mouldability features to define the shape of the product, the forms used by the designer being defined from the viewpoint of mouldability. In the approach taken by the author a translation process is required between the independent form features set and the mouldability features set, and a strategy is required to support the consideration of the mouldability of a product as the geometry evolves. Translations from form to primary and secondary features have different characteristic and are therefore treated separately in the following sections.

7.4.1. Problems of instantiation of primary mouldability features.

The problems discussed in this section are how to translate between the set of form features and the primary mouldability features set. When you look at each form feature in turn and examine the relations with the mouldability features the potential mappings between form features and mouldability features can be readily identified, and are illustrated in Figure 7.6. The authors investigation has shown the following problems in identifying the particular route to the instantiation of a primary mouldability feature:

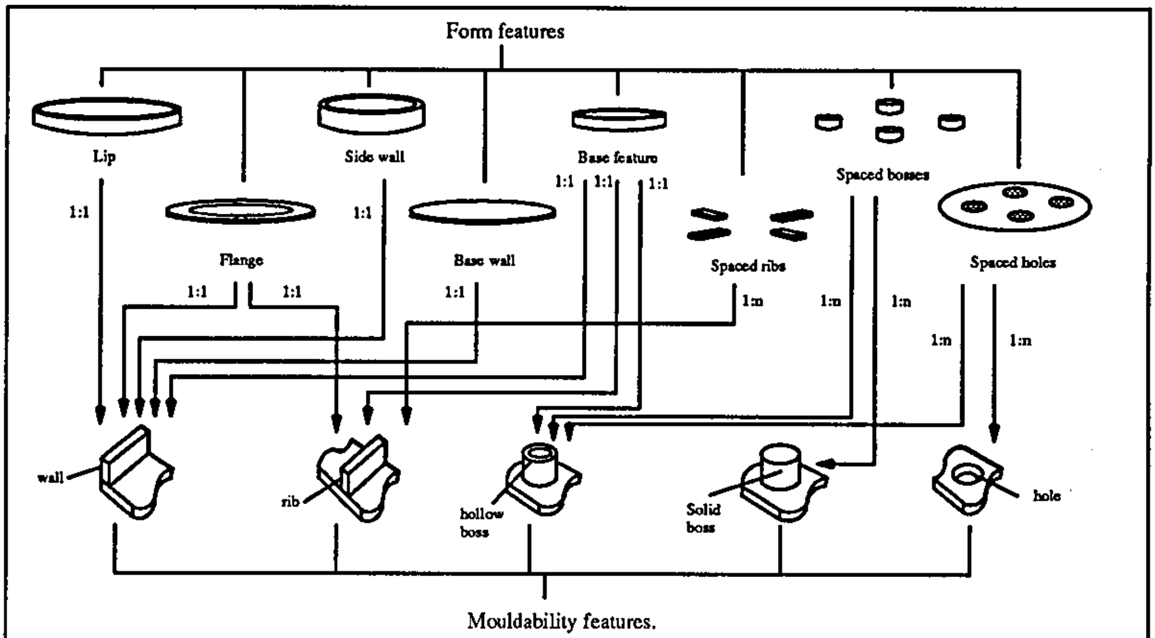


Figure 7.6. Relationships between the enhanced form features set and mouldability features set.

i) For a base feature, a flange, spaced bosses or spaced holes, the mouldability meaning of the form feature is dependant upon the context of surrounding geometry. Therefore in order to interpret the meaning of these form features with respect to mouldability, a methodology for identification of adjacent form feature instantiations is required.

ii) As illustrated in Figure 7.7 for the example of a mouldability wall feature, features of the same mouldability type may have different geometry. Since geometry is captured in the form features, the problem to be resolved is to translate parameters of the form feature to the equivalent mouldability feature parameters.

iii) As shown in Figure 7.6, some of the form features representation have a one–to–many relationship with the mouldability representation. These are spaced ribs, spaced bosses and spaced holes. Each form feature translates to as many mouldability features as there are members of the group. The instantiation of the mouldability viewpoint by translation requires consideration of positional and orientation problems of each mouldability feature, and therefore each mouldability instantiation must be examined as a separate entity.

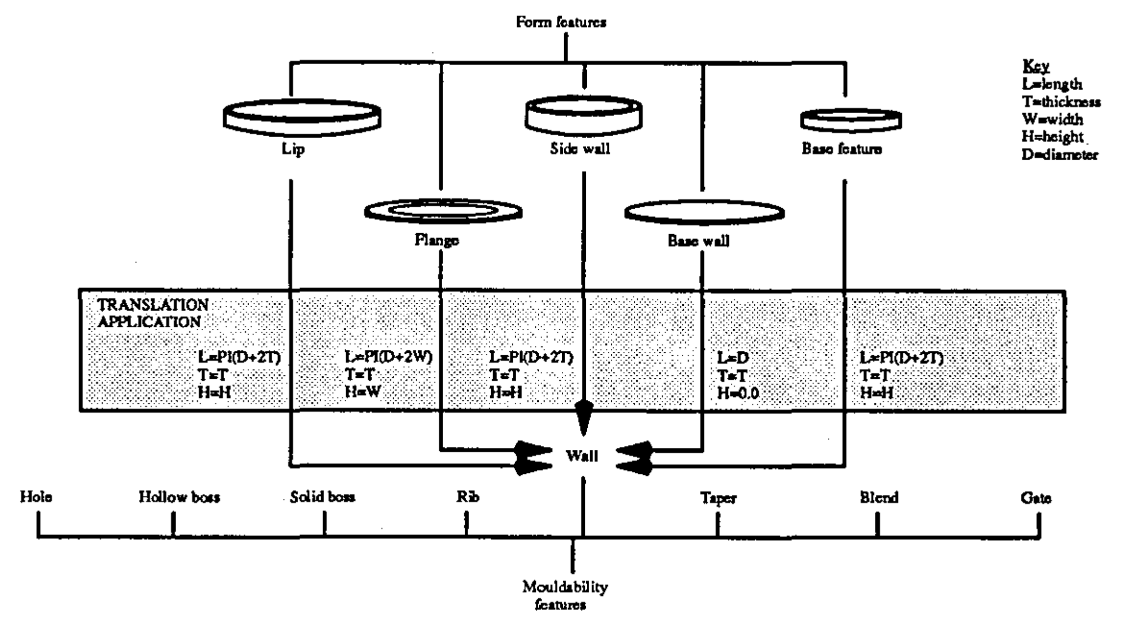


Figure 7.7. Alternative parameter translations from form features to mouldability wall feature.

iv) In order to instantiate a mouldability feature and provide feedback on its parameters with respect to mouldability constraints, a knowledge is required of adjacent **mouldability** geometry. For example the maximum size of a protrusion feature (rib, solid or hollow boss) is determined by the thickness of the adjacent mouldability wall, the optimum thickness of the wall is determined by the relative thickness of adjoining walls etc. Therefore a method is required of identifying adjacent mouldability feature instantiations.

7.4.2. A structured methodology for primary mouldability feature instantiation.

From the previous section, the requirements to be met in translation from the form representation to the primary mouldability representation are i) identifying adjacent geometry in terms of form and mouldability in the Product model, ii) identifying the mouldability meaning of form features, taking account of adjacent form features where appropriate, iii) managing one–to–many relationships between form features and the mouldability viewpoint, iv) correctly interpreting the parameters of form features when translating to mouldability. Formulated solutions for each in turn are outlined below:

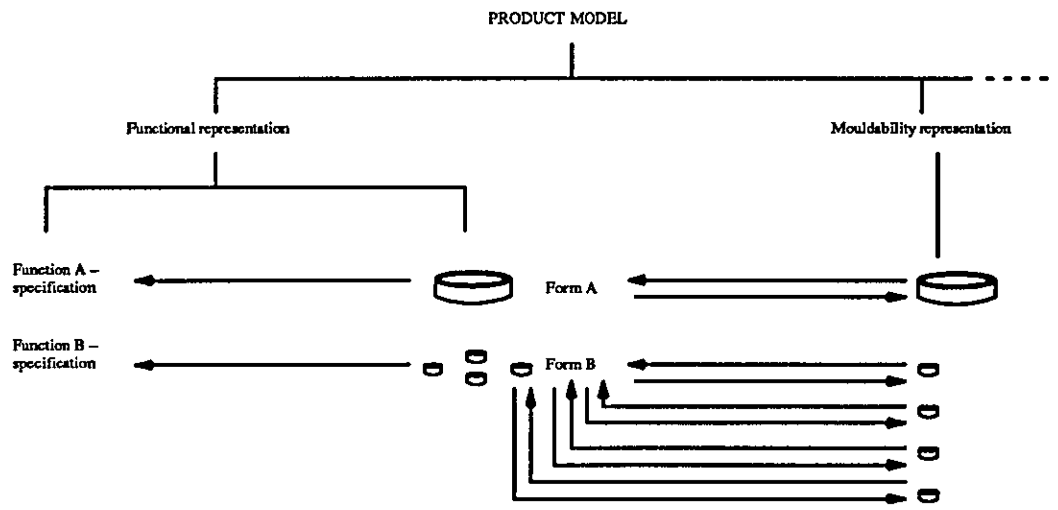


Figure 7.8. Example of relationships built up in the Product model between the functional and mouldability representation.

i) To achieve translation when a knowledge of surrounding geometry is required, the new form feature instantiation must be evaluated in the context of surrounding instantiations of form features. The author has already defined a method in section 7.3.2. which identifies adjacent instances of form feature (Figure 7.5). This method can also be used to identify adjacent mouldability instantiations as long as a two way link between the form and mouldability feature can be instantiated, (Figure 7.8): The backward link (mouldability–to–form) is driven by the translation process as a primary mouldability feature is instantiated. Taking the form feature from which the translation has been made, the adjacent form can be identified as described in section 7.3.2. The forward link (form–to–mouldability) can then be used to identify the equivalent mouldability feature of the adjacent form feature. This method is shown in Figure 7.9. By instantiating the backward link this is available when returning to re–evaluate mouldability. The above structured methods solve the problem of identifying adjacent geometry in terms of form and mouldability.

ii) Those form features with more than one mouldability equivalent are base feature, flange, spaced bosses and spaced holes as illustrated in Figure 7.6. Other form features can be directly translated to mouldability as only one mouldability equivalent exists, ie base wall, side wall, and lip always translate to mouldability 'wall' features and spaced ribs always translate to mouldability 'rib' features. Given that the adjacent geometry can be identified and spatial

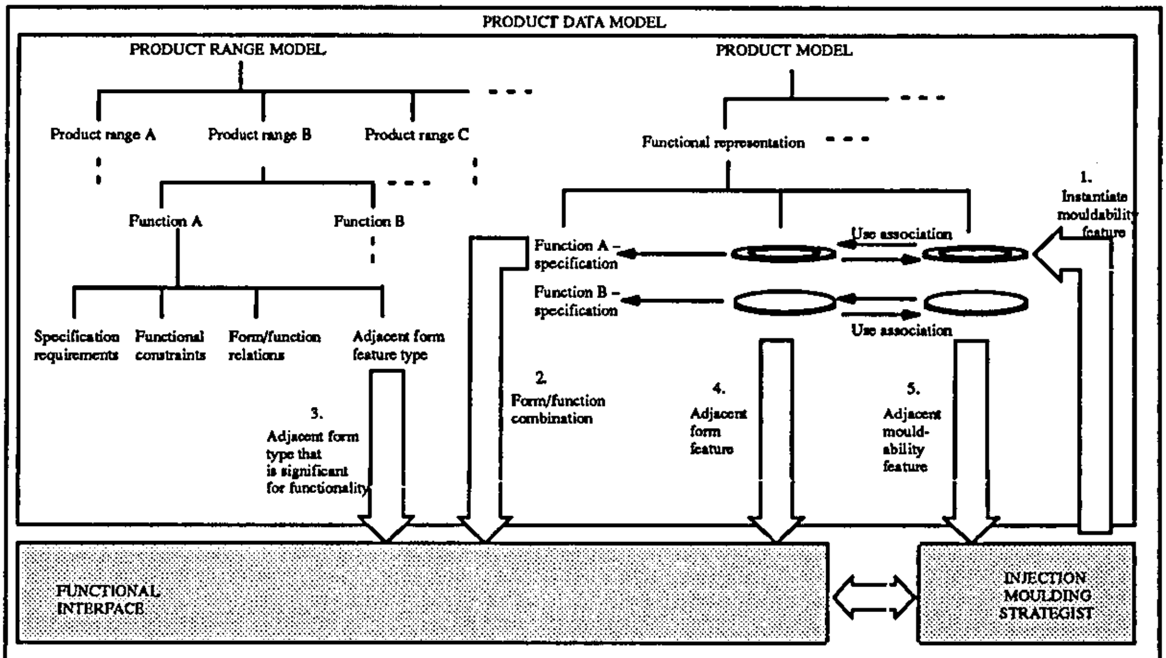


Figure 7.9. Use of Product Range Model data for identifying adjacent mouldability features.

relationships between existing geometry and new evaluated, translation of base feature, flange, spaced ribs and spaced bosses is as follows:

It can be seen from Figure 6.13. that the difference between a mouldability 'wall' feature and a mouldability 'rib' feature is that the wall exists at the extremity of geometry, ie extending geometry, whereas the same form away from the extremity, ie reinforcing the geometry, can be considered to be a 'rib' feature. Figure 7.10 shows the possible relationships between a flange and adjoining geometry in the context of a rotational product. If the flange is away from the edge of geometry, as in Figure 7.10 b), c), f), g), the translation is to a mouldability 'rib' feature. If the flange is on the edge of geometry, as in Figure 7.10 a), d), e), h), the translation is to a mouldability 'wall'.

Figure 7.11 shows the possible relationships between a base feature and adjoining geometry in the context of a rotational product. If the base feature is positioned on the edge of geometry, as in Figure 7.11 b), c), the translation is to a mouldability 'wall' feature. If away from the edge of geometry, the base feature can be translated either to a mouldability 'hollow boss' feature or to a 'rib' feature, depending on the relationship with the geometry of the base wall. In the context of a rotational product if the diameter of the base feature is smaller than that

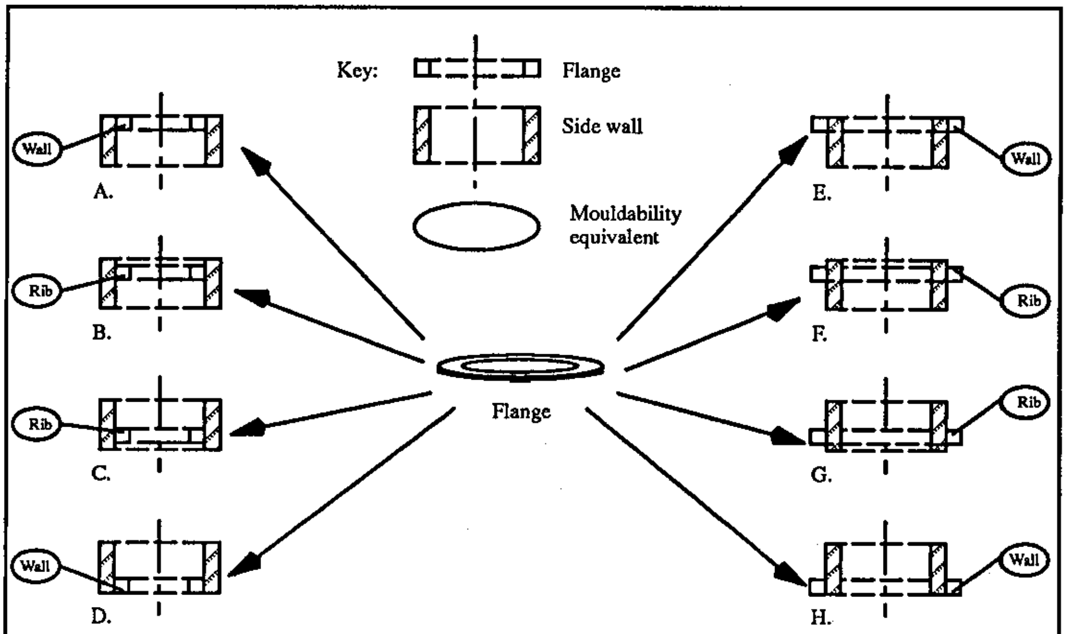


Figure 7.10. Examples of translation of flange to mouldability.

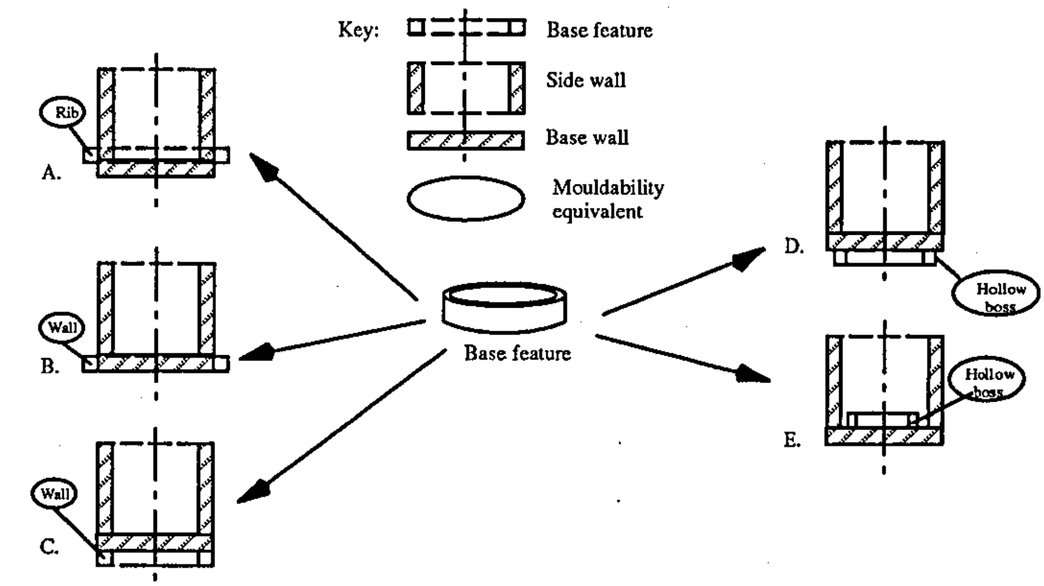


Figure 7.11. Examples of translation of base feature to mouldability.

of the base wall, as in Figure 7.11 d), e), the translation is to a mouldability 'hollow boss' feature, if the diameter is larger, as in Figure 7.11 a), the translation is to a mouldability 'rib' feature. The above constraints, if captured within a method, solve the problem of translation from form to mouldability where a form feature has more than one mouldability meaning.

The above translation of the base feature and flange emphasises the limitations of the 'design by mouldability features' approaches of Al–Ashaab (1994), Dixon (1988), Huh and Kim

(1989) and Hanada and Leifer (1989).

iii) Three form feature types (Figure 7.6) have a one–to–many relationship with the mouldability representation, namely spaced holes, spaced bosses and spaced ribs. For these form features each member of the group translates to a separate mouldability feature. Therefore there is a requirement to calculate the position of the individual mouldability features and also the orientation for the individual mouldability rib features.

As shown in Figure 7.12, the position of a spaced holes, spaced bosses or spaced ribs form feature is at the centre of a pitch circle diameter around which the group members are positioned. Group members are equally spaced around the pitch circle diameter with the first form being at 0.0 on the y axis and at the radius of the pitch circle on the x axis (assuming the orientation is such that the central axis is z). Therefore the position of each mouldability feature is calculated around the pitch circle diameter, with the centre of the mouldability feature on the circle. For a mouldability hole or solid boss this means the central axis is on the circle, and for a mouldability rib this means the centre of the base is on the circle. For each mouldability rib feature the orientation angle in radians relative to the x axis is calculated so that the length of a mouldability rib feature runs parallel to a line from the centre of the old form feature to the position of the centre of the base of the mouldability rib on the pitch circle diameter. The translation of spaced ribs, spaced bosses and spaced holes to multiple individual mouldability features is shown in Figure 7.12. Each individual mouldability feature has to be dealt with separately with respect to mouldability feedback and interaction with the designer. Potential problems with the above approach are:

For the spaced holes form feature each hole in the group translates to a separate mouldability hole, unless that hole is positioned on an existing mouldability solid boss feature, in which case the translation is to a mouldability hollow boss. The hollow boss is made up of the new hole and original solid boss. If the translation results in the instancing of hollow bosses, then the place in the Product model of the solid bosses that existed prior to the translation must

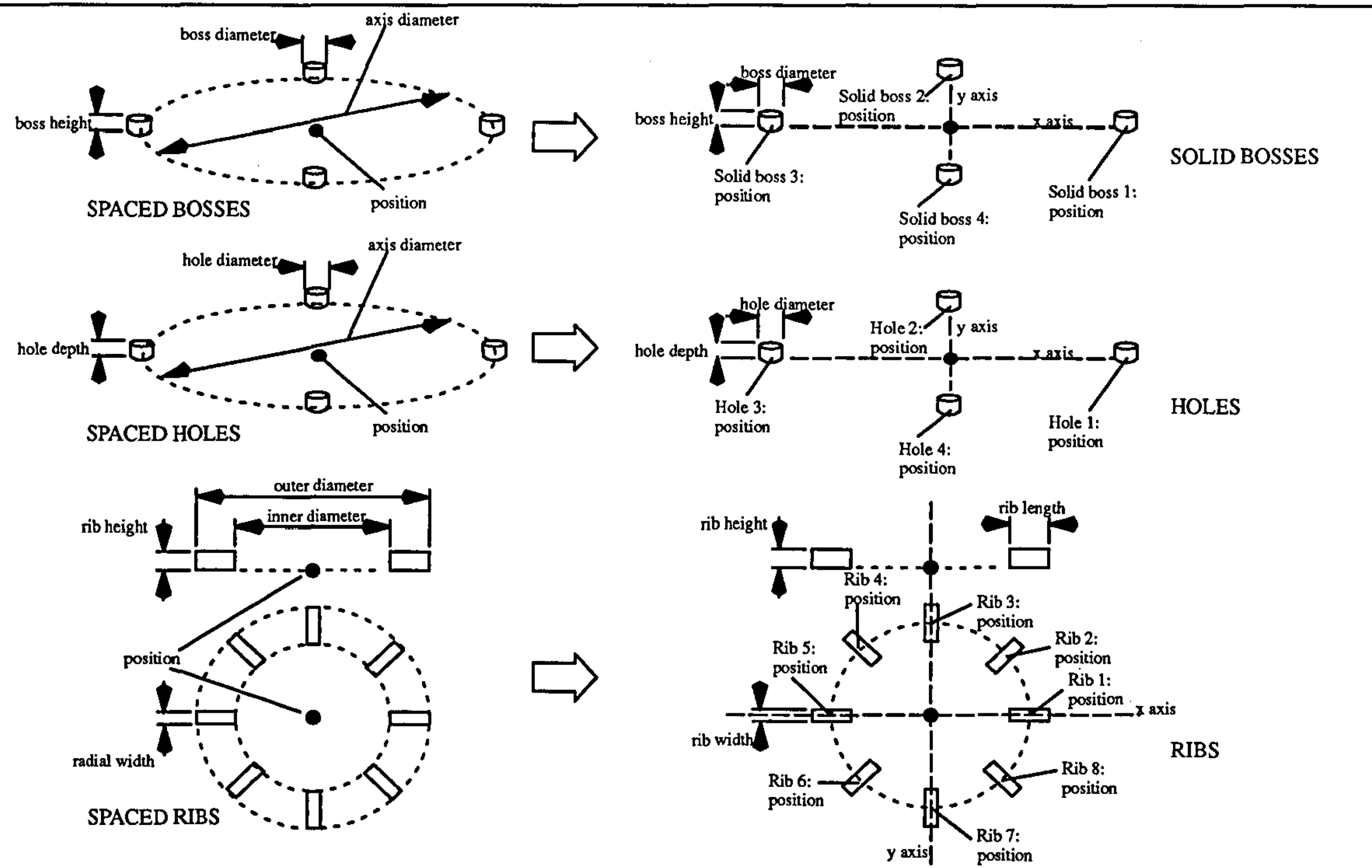


Figure 7.12. Translation of one–to–many form features to mouldability.

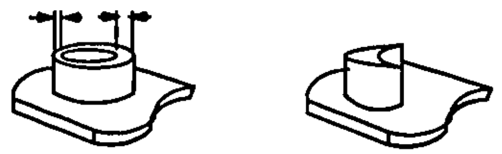


Figure 7.13. Examples of ambiguities when combining non-coaxial hole and solid boss into a hollow boss.

be taken by the new hollow bosses. The original solid bosses must be removed from the Product model.

The method for identifying adjacent geometry described in 7.2.1. and extended above to identify adjacent mouldability features, may not identify an existing spaced bosses instantiation in the Product model. For example, if the functional objective for the spaced holes form was 'drainage' on a flower pot, the method would identify a base wall as the adjacent form feature, since the holes' drainage functionality relates to their relationship with the base wall. Therefore if spaced bosses are not identified as adjacent geometry, checking for existing spaced bosses instantiations in the Product model is required.

A potential difficulty in combining instances of hole and solid boss to create a single hollow boss is that they may not be coaxial, resulting (Figure 7.13) in a varying section thickness from one side of the hollow boss to the other or the hole not being fully within the hollow boss. Neither of these geometry configurations would be permissible in an injection mould, Pye (1989). Therefore the two instances must either be made coaxial and combined or separated and considered as separate entities. The author considers that the optimum solution is to recognise a hollow boss only if the new instance of hole is coaxial with the existing solid boss geometry. Otherwise the designer must decide whether to reconfigure the new form feature so that the mouldability instances are separated or combined. Feedback is required to inform the designer of the problem and assist in the reconfiguration.

Other authors, eg Al–Ashaab (1994), Hanada and Leifer (1989), Dixon (1988), Cunningham and Dixon (1988), Cutcosky et al (1989), (1991), appear not to have considered the interaction of bosses and holes, assuming that because the instance has been chosen from a features library, instances of hollow boss arise only as a conscious decision of the designer.

Similar to the translation of spaced holes, each member of the group in a spaced bosses feature translates to a single mouldability solid boss, unless placed on an existing mouldability hole, where the new instance of a solid boss and existing hole are combined into a single instance of a hollow boss. The procedures for spaced bosses translation are close to those of the spaced holes: a) If the translation of spaced bosses results in the instancing of hollow bosses, then the place in the Product model of the holes that existed prior to the translation must be taken by the new hollow bosses. The original holes must be removed from the Product model, b) If spaced holes are not identified as adjacent geometry, checking for existing spaced holes instantiations in the Product model is required, c) A hollow boss is only recognised if the new instance of solid boss is coaxial with the existing hole geometry. Otherwise the designer must decide whether to reconfigure the new form feature so that the mouldability instances are separated or combined. The above structured methods solve the problem of managing one–to–many relationships between form features and the mouldability viewpoint.

iv) Figure 7.7 shows an example of a situation where the interpretation of form feature parameters when translating to a particular mouldability type is not always the same. The interpretation of parameters is dependant upon the type of form feature from which the mouldability instance is derived. Another situation where variations exist in the way form feature parameters are translated to mouldability is where a form feature has more than one mouldability equivalent, ie base feature, flange, spaced holes and spaced bosses. To enable the correct translation of form feature parameters to mouldability a set of rules is required for interpretation of form feature parameters. A set of rules have been defined for all permutations of translation between the form features set and that of mouldability. This set of rules is shown in Figure 7.14.

The structured methods described above solve the problems of translation between the form features set and the primary mouldability features. Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.4.3. Problems of instantiation of secondary mouldability features.

Tapers, blends and gates are secondary mouldability features. No direct translation is possible from the form features set as with the primary mouldability feature types because secondary mouldability features are instantiated as a result of considering mouldability. Secondary mouldability features do not have a form of their own but affect the form of primary mouldability features. A method for the identification of when such features are needed and how to instantiate them is needed. The problems of doing this for each secondary mouldability feature type are as follows:

The mouldability *taper* feature should be applied to all instances of primary mouldability features to facilitate removal of the product from the mould. Therefore the requirement for a taper feature instantiation must be identified each time a mouldability primary feature is instantiated in the Product model. Different constraints on tapers exist when applied to different types of geometry, depending on the primary mouldability type to which it is applied. Also, the way that a taper feature is applied to a primary mouldability feature is dependant on the type and context of the geometry to which it is applied, as shown in Figures 6.4 to 6.9. It can be seen from those figures that a taper application has a significant effect on the geometry and on the connectivity of abutting geometry. Therefore the three problems with re

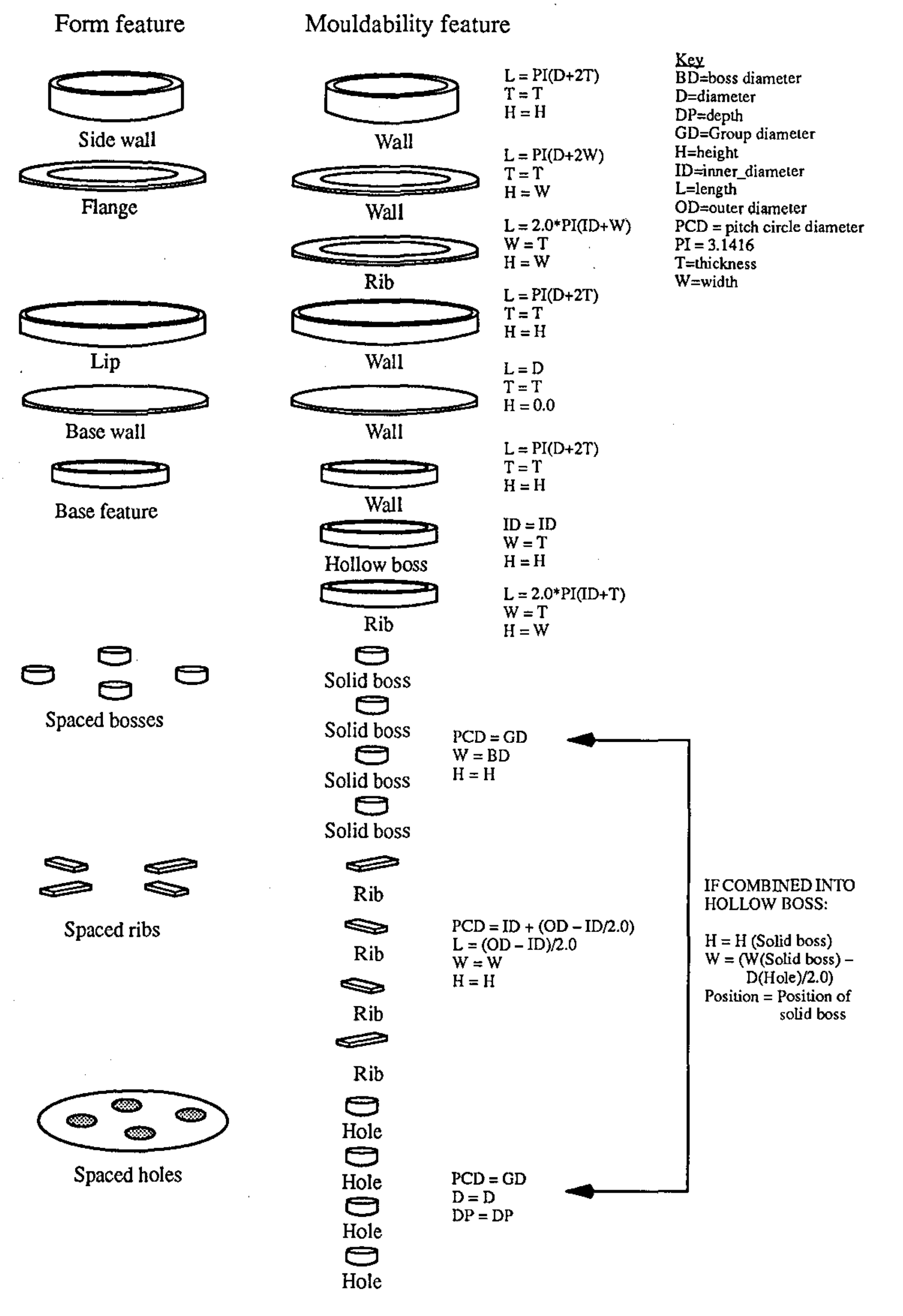
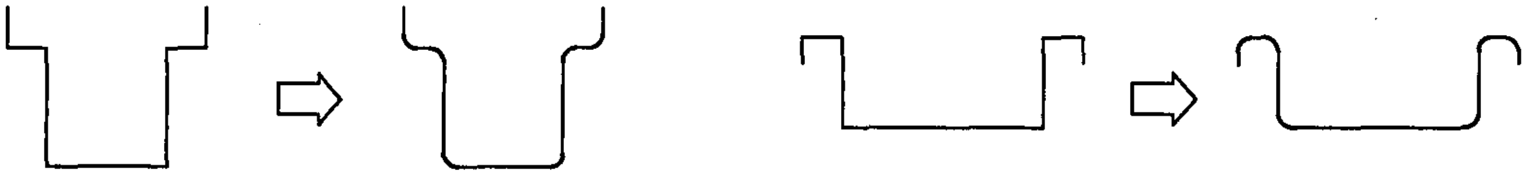


Figure 7.14. Interpretation of form feature parameters during translation to mouldability

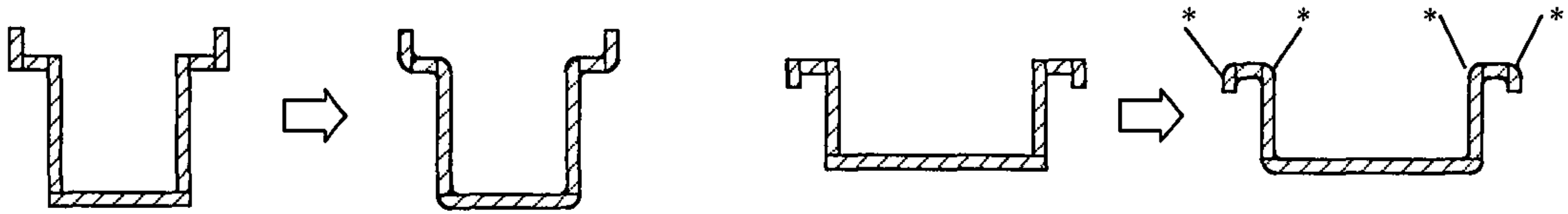
spect to taper feature instantiations are i) identifying the need for a taper feature instantiation, ii) identifying the constraints on the taper, iii) identifying a method for applying taper feature instantiations is required which allows for the effect of a taper on geometry and on the connectivity of abbutting geometry.

The mouldability *blend* feature must be applied between two wall instantiations or between an instantiation of a wall feature and that of a mouldability reinforcement feature. Therefore the requirement for a blend feature instantiation must be identified each time a reinforcement feature is instantiated in the Product model adjacent to an instance of a wall feature or a wall feature is instantiated adjacent to an instance of a reinforcement feature or another instance of a wall feature. Different types of blend are applied to different combinations of primary mouldability feature types, ie abbuting wall feature instantiations require a corner blend, whereas an abutting wall and reinforcement feature instantiation require a protrusion blend. Previous work has ignored a separate protrusion blend type, eg Dixon (1988), Huh and Kim (1989).

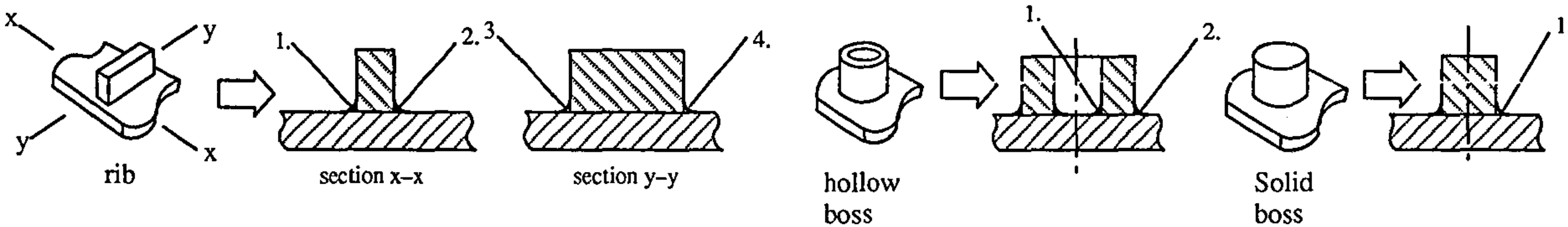
Corner blends must be applied differently depending on the context of the geometry, as shown in Figure 7.15. Figure 7.15a) shows the application of corner blends on 2D geometry, as in the work of Al–Ashaab (1994), and Figure 7.15b) shows the application of corner blends to the same corners using 3D geometry. Using 2D geometry, (as in Figure 7.15a) it can be seen that the relationship between two walls is always the same. Using 3D geometry, as in Figure 7.15b) it can be seen that the relationships between abutting walls can change, this having an effect on how the blend is applied, and individual radii within a corner blend can be applied to different edges and/or faces on different wall feature instantiations. Previous work has allowed a designer to select a blend feature from a library of pre–determined feature primitives, eg Dixon (1988) Hanada and Leifer (1989). However when translation is required the part of each wall feature instantiation to which the blend must be applied must be identi–



A. Corner blends in 2D geometry are all the same.



B. Corner blends in 3D geometry vary in their application.



C. Protrusion blends in 3D geometry require a varying number of radii applications depending on the reinforcement type.

* These radii are not applied if on the parting line – prevent removal of product from mould or require concave core.

Figure 7.15. Problems in applying blends in 3D geometry.

fied.

As shown in the figure, a corner blend outside radius on the parting line is not applied on a 3D product as this prevents ejection of the product from the mould unless the parting line is moved to the edge of the blend and a concave core is used. Figure 7.15c) shows that blends applied between reinforcement feature instantiations and wall feature instantiations vary in their nature according to the type of reinforcement feature. The problems arising are i) to identify the need for a blend feature instantiation, ii) to identify the required type of blend, iii) A structured method for the application of blends to 3D geometry is required which takes account of the combination and context of 3D geometry to which the blend is applied.

A *gate* feature is not required on every primary mouldability feature instantiation. However at least one gate feature must be instantiated somewhere on an injection moulded product to allow plastic material to enter the mould cavity. For adequate feeding a gate must be placed onto a large section, therefore gate features should only be applied to wall feature types. In the context of mouldability the gate position is evaluated with respect to the feeding requirements of the product. This involves calculation of the largest distance that material must flow (in the mould) from the gate position to the extremities of the product geometry. Previous work on mouldability support has evaluated the gate position either by assuming a central gate position and calculating the distance material must flow through the mould by adding up the wall lengths of 2D geometry, Al-Ashaab (1994), or has retrospectively evaluated the gate position on simple geometry (eg slabs with protrusions attached) using FEM type software to identify potential weld lines, eg Irani, Kim, Dixon (1989)b,(1990). Modern FEM analysis software can be used to evaluate gate positions on complex geometry, eg Gerdes et al (1994), Michaeli et al (1995). The problem here is that such software requires data on the dimensions of the mould elements such as the gate, runner system and sprue. This data is unavailable until the mould design stage and it is considered by the author that it is important that some information about gating should be defined during the product design phase. This thesis therefore takes the following approach: The designer chooses the gate position and this could be anywhere on the component. The longest distance material has to flow in any direction from the gate must be identified and compared with the maximum feeding distance in the Manufacturing model.

The gate position is evaluated in relation to the type of gating system, eg pin gate, sprue gate etc and this evaluation is considered during the mould design stage, as discussed in section 7.7. However the appropriate gate type(s) is/are dependant on the configuration of the product geometry, eg tubular–thin walled, shallow prismatic–thick walled etc. Furthermore, the type of gate has an influence on the evaluation of the distance material is required to travel from the gate to the product extremity, (section 7.4.4). Therefore the gate type needs to be considered at the mouldability stage, ie product design stage rather than mould design.

Problems arising are i) identifying the need for a gate feature instantiation, ii) identifying the configuration of the product geometry hence the appropriate gate type(s), iii) To evaluate the feeding distance from a gate feature instantiation to the product extremity when the sequence of geometry creation is unknown and the flow of material from the gate is multi directional. The following section describes the formulated solutions to the problems of the instantiation of secondary mouldability features outlined in section 7.4.3.

7.4.4. A structured method of secondary mouldability feature instantiation.

A method has been defined which identifies when a *taper feature* needs to be instantiated as follows: The requirement for a taper is identified via the mouldability constraints associated with a primary mouldability feature. For example, a wall feature constraint in the Manufacturing model identifies the need for a taper to be applied to a wall feature instantiation. The designer is then advised on the need for a taper as part of the mouldability assessment process of the primary feature. If the designer chooses to instantiate a taper the constraints associated with the taper feature within the Manufacturing model are evaluated. These evaluated constraints are compared with the designer specified values and appropriate advice given to the

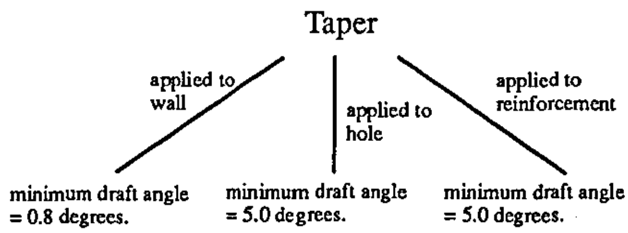


Figure 7.16. Taper draft angle constraint depends on primary feature.

designer as necessary.

In the previous work of Dixon (1988) and Al–Ashaab (1994) the constraints that apply to a taper are known because the taper is an attribute of a primary mouldability feature. However using a separate taper instantiation as herein, the different angle constraint that applies to the taper is dependant upon the type of primary mouldability feature to which the taper is applied. At the time of primary mouldability feature instantiation, the identity and type of the primary mouldability feature is known, and this data is thus used to support the instantiation of the taper feature (Figure 7.16).

The final problem of taper feature instantiation is the definition of a structured method of taper application, taking account of the effect of the taper on geometry and on the connectivity of abbutting geometry. All previous work into mouldability, eg Dixon (1988), Hanada and Leifer (1989), has used predefined mouldability feature types, selected by a designer from a menu. Tapers have been applied by associating attributes of wall instantiations with areas of a solid model of the whole product geometry, with no requirement to consider the effect of a taper on inter form relationships. When considering multiple viewpoints in product design, as currently, a taper feature instantiation changes the geometry of the form feature from which the primary mouldability feature has been instantiated. The enhanced form features set defined by the author supports a structured method of taper applications, taking account of the effect of taper feature instantiations on inter form relationships. The successful identification of the inter form relationship enables the identification of the manner in which the taper must be applied to the geometry of the form feature. Alternative taper applications and their effect on product geometry is illustrated in Figure 7.17.

How the geometry is modified by a taper application can be defined from three factors; the type of form feature instantiation, the context of the form feature with respect to other form features in the Product model and the parting line. The first two of these factors can be readily identified as already discussed in sections 7.3.2 and 7.4.2. The relative position of the parting line must be established. However, in the authors work, considering rotational products, it can be assumed that the parting direction is parallel to the central axis of the wall feature instantiations. The identification of a parting line position can be achieved as discussed in section 7.5. The above structured methods solve the problem of taper feature instantiation.

The above approach is illustrated in Figure 7.18, which illustrates the process of primary and secondary mouldability instantiations. The approach can also be applied to the blend or gate features. However there are some structural differences in the way mouldability is assessed for the blend and gate features, as discussed in the remainder of this section.

A mouldability *blend* feature should be applied between all reinforcement feature instantiations and adjacent walls, and between all abutting wall feature instantiations. With the exception of the first feature to be instantiated in a Product model, where there is no blend requirement, the identification of a blend requirement can be accomplished in the same way as for a taper, and follows the same instantiation procedure as illustrated in Figure 7.18.

The work of Dixon (1988) and Al–Ashaab (1994) which also provided feedback on the requirement for a separate mouldability blend feature instantiation using the mouldability constraints of other mouldability features only considered corner blends, and relied upon the designer to respond to feedback advice by selecting a blend from a predetermined feature menu. In this work the instantiation of mouldability features is driven by the functional design viewpoint (via translation), and Dixon and Al–Ashaab's approach is not appropriate as there is no form equivalent of a mouldability blend feature.

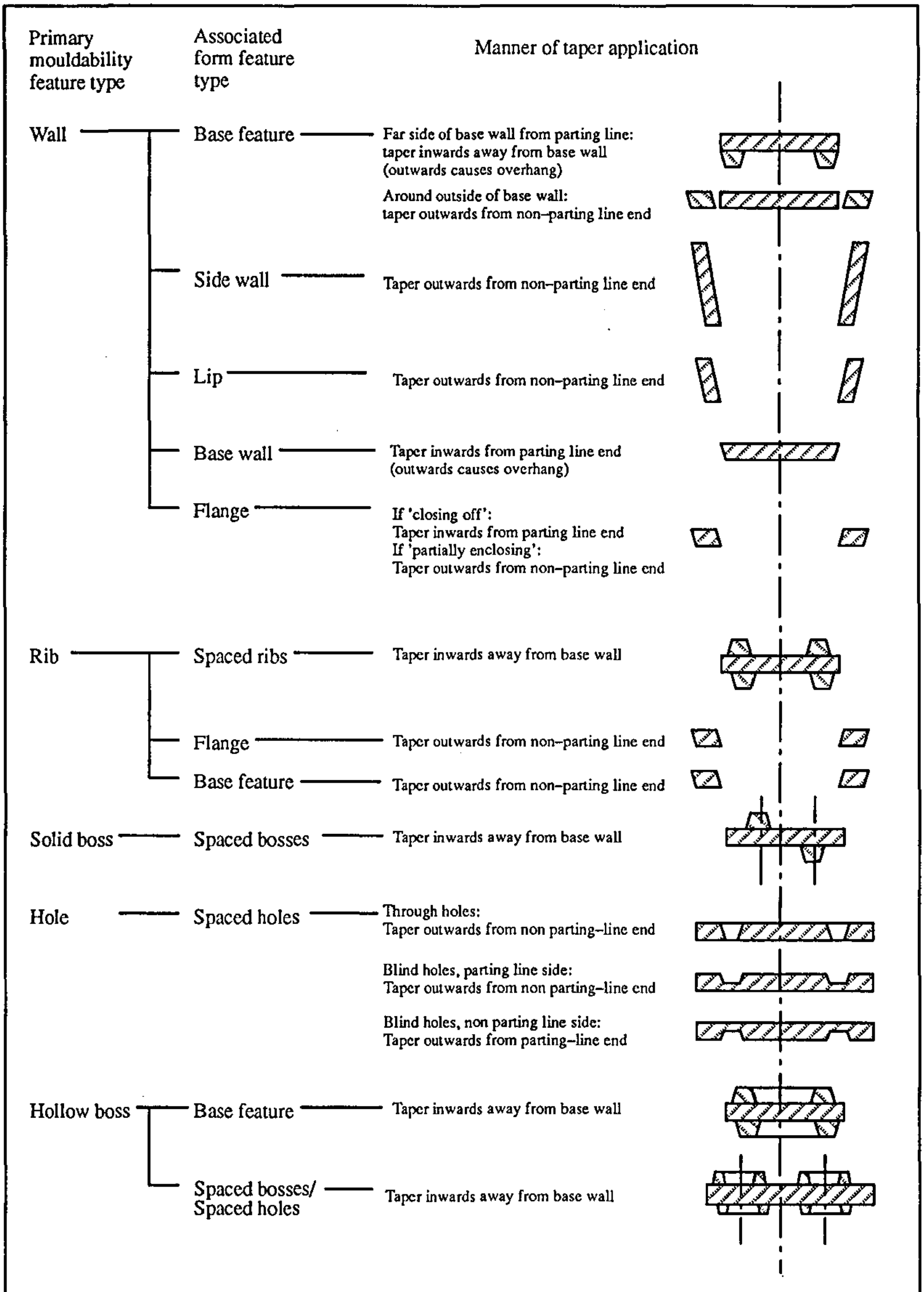


Figure 7.17. Application of tapers to 3D geometry.

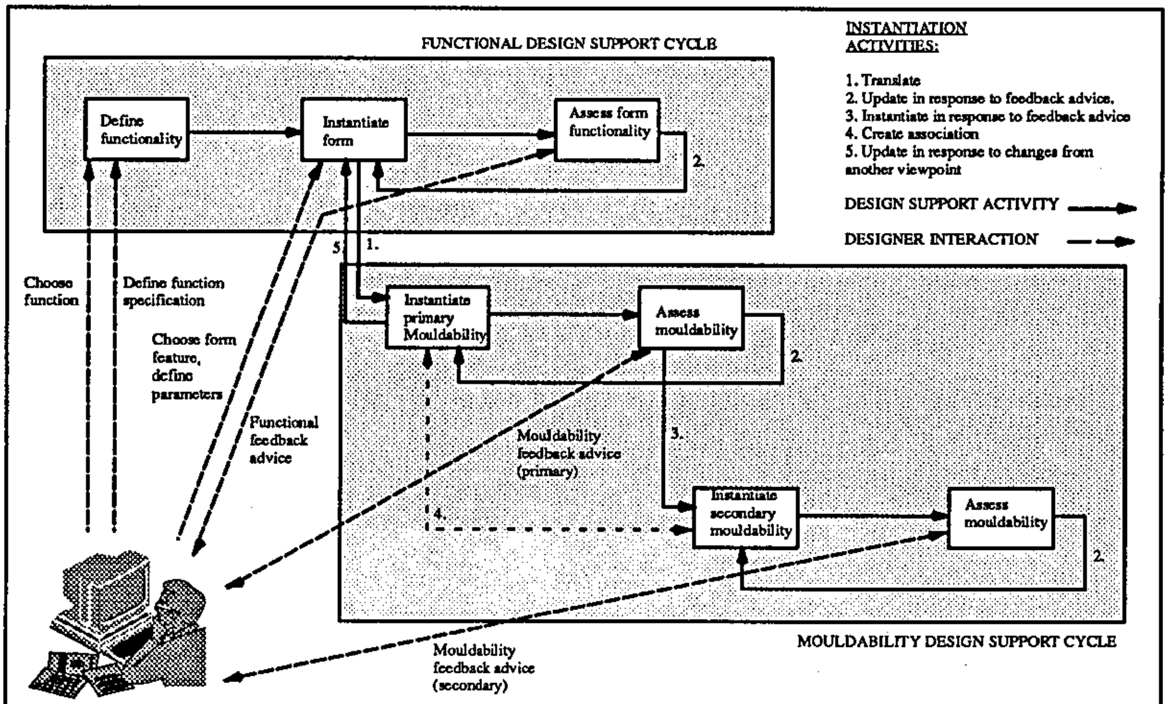


Figure 7.18. The process of instantiation of primary and secondary mouldability features.

It is also necessary to identify which type of blend should be applied. A protrusion blend is applied between a reinforcement feature and a wall feature, and a corner blend is applied between abutting wall features. Therefore a wall feature constraint in the Manufacturing model identifies the requirement for a corner blend if adjacent existing geometry in the Product model has been identified as a wall feature instantiation, or identifies the requirement for a/ some protrusion blends if adjacent existing geometry in the Product model has been identified as a/some reinforcement feature instantiation(s). A reinforcement feature constraint in the Manufacturing model identifies the requirement for a protrusion blend between a reinforcement feature instantiation and existing geometry in the Product model.

Finally there is the problem of the definition of a structured method of blend application, taking account of the combination and context of the geometry to which the blend is applied. The same approach can be taken as described for the taper feature. The actual consideration of blend features and their effect on geometry is illustrated in Figure 7.19.

In all previous work, eg Ong et al (1995), Al–Ashaab (1994), Irani, Kim, Dixon (1989), the application of a *gate* feature has been considered retrospectively to the build up of product

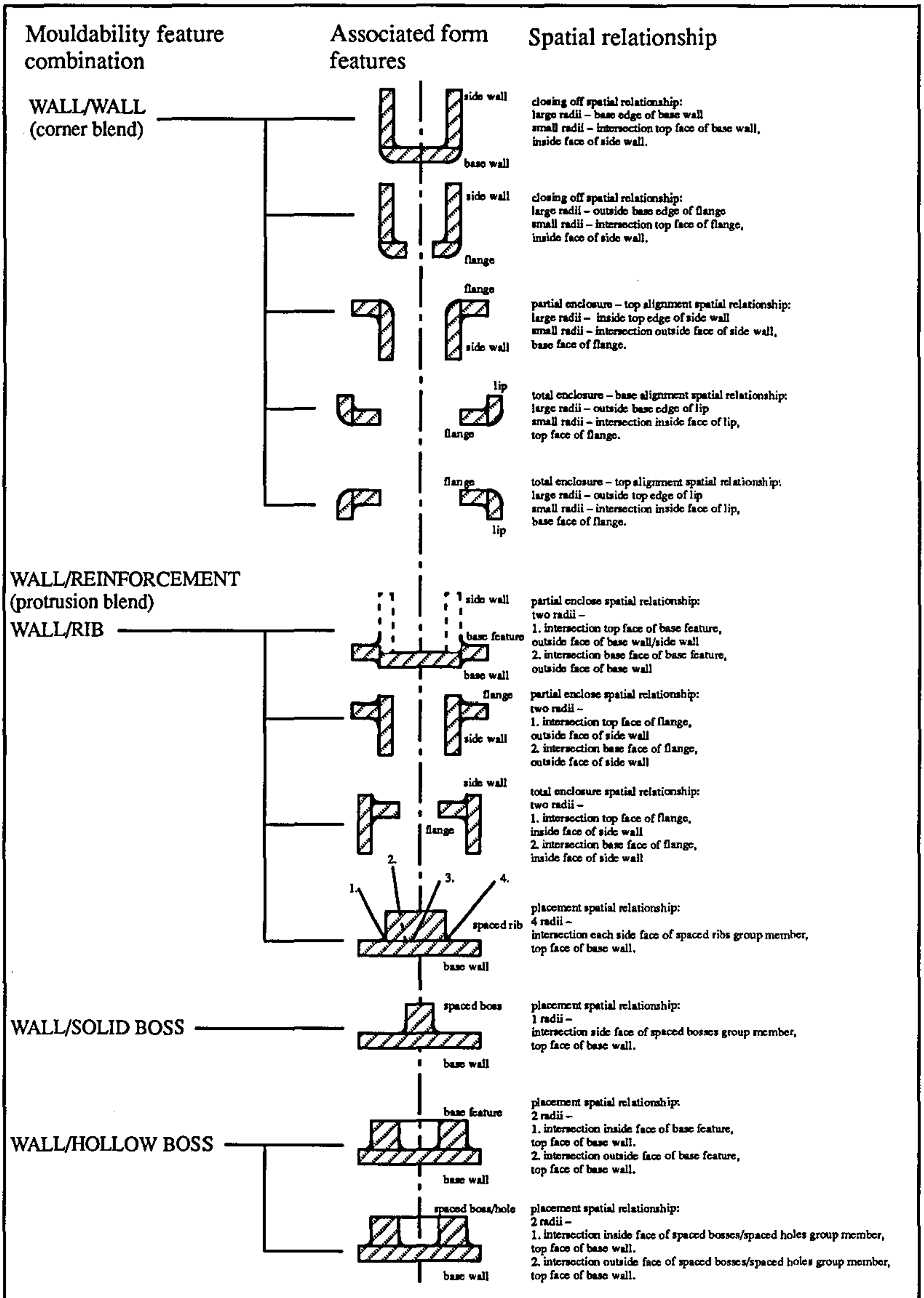


Figure 7.19. Application of blends to 3D geometry.

geometry, which however, precludes the consideration of different gating options as the product geometry is built up. It is also not possible to consider mould design without first considering the gating of the product, and so retrospective consideration of gating, precludes the possibility of concurrent product and mould design. Therefore the designer must be able to consider gating during the build up of the product geometry.

At least one gate feature is essential on an injection moulded product and gate features are only applied to wall feature types. Therefore using the defined method of managing secondary mouldability feature instantiations (Figure 7.18), a wall feature constraint in the Manufacturing model identifies the requirement for applying at least one gate feature to a wall feature instantiation in the Product model. The designer is advised of the need, and given the opportunity of creating a gate on a particular primary feature instantiation as part of the mouldability assessment process of the primary feature.

It is also necessary to identify the overall configuration of product geometry, and hence the type of gate, in order to evaluate the feeding distance and provide a basis for examination of gating system process constraints during the mould design stage. Two options exist for identifying the overall product geometry; i) to analyse the form feature instances in the Prod uct model using an algorithm in a solid modeller, and ii) to anticipate the configuration of the product geometry using part family analysis, storing this data in the Product Range Model. If gating is to be considered during the build up of product geometry, then using the former option, the configuration of geometry could change as the designer builds up the product in the Product model. In such a case the wrong type of gate may be chosen by the designer for the eventual configuration of product geometry, with possible erroneous feeding distance evaluation which is influenced by the type of gate. Furthermore design effort may be wasted during the mould design stage evaluating the process constraints for the wrong gate type and consequent mould configuration.

The latter method, that of storing data on product configuration in the Product Range Model, ensures that the appropriate gate type(s) is/are known to the designer at the outset of the design process, enabling him/her to make the correct decision on gate type. In the authors work this information is made available to the designer when a gate feature is instantiated in the Product model, via the gate feature constraints in the Manufacturing Model. Feedback advice on gate type is in relation to the configuration data in the Product Range Model and wall thickness attributes data from the Product model, eg tubular – thin walled.

The final problem of gate feature instantiation was to evaluate the feeding distance from a gate feature instantiation to the product extremity to identify potential filling problems. The interpretation of flow length is related to the position of the gate relative to the wall geometry and the relationship of a wall feature instantiation with surrounding geometry. Two factors influence this. The first is that features of the same mouldability type may not have the same geometry. The second is that the length attribute of a feature does not always correspond to the flow length of the plastic material. The feeding distance through a particular wall geometry is dependant on where the material enters the geometry and where it exits or reaches the product extremity. Below the author presents a structured method of measuring the flow length through wall feature instantiations based on their association with instantiations of the enhanced form features set in the Product model:

To identify the relative position of a gate and wall feature instantiation in the Product model, the chosen gate type of the designer can be classified as either an under gate or side gate. Examples of side gates are rectangular edge gates, ring gates, film gates, fan gates, overlap gates, sub–surface gates and tab gates, Pye (1989). Examples of under gates are diaphragm gates, pin gates and sprue gates, Pye (1989). These gate types are discussed further in section 7.7. Side gates are so called because they are placed on or near the parting line of a mould, whereas under gates are placed away from the parting line. If the type of form feature from which a wall feature instantiation was translated can be identified, the feeding length through the wall geometry can be ascertained for the associated form feature, depending on whether the product is side gated or under gated, using feeding lengths identified in Figure 7.20. The

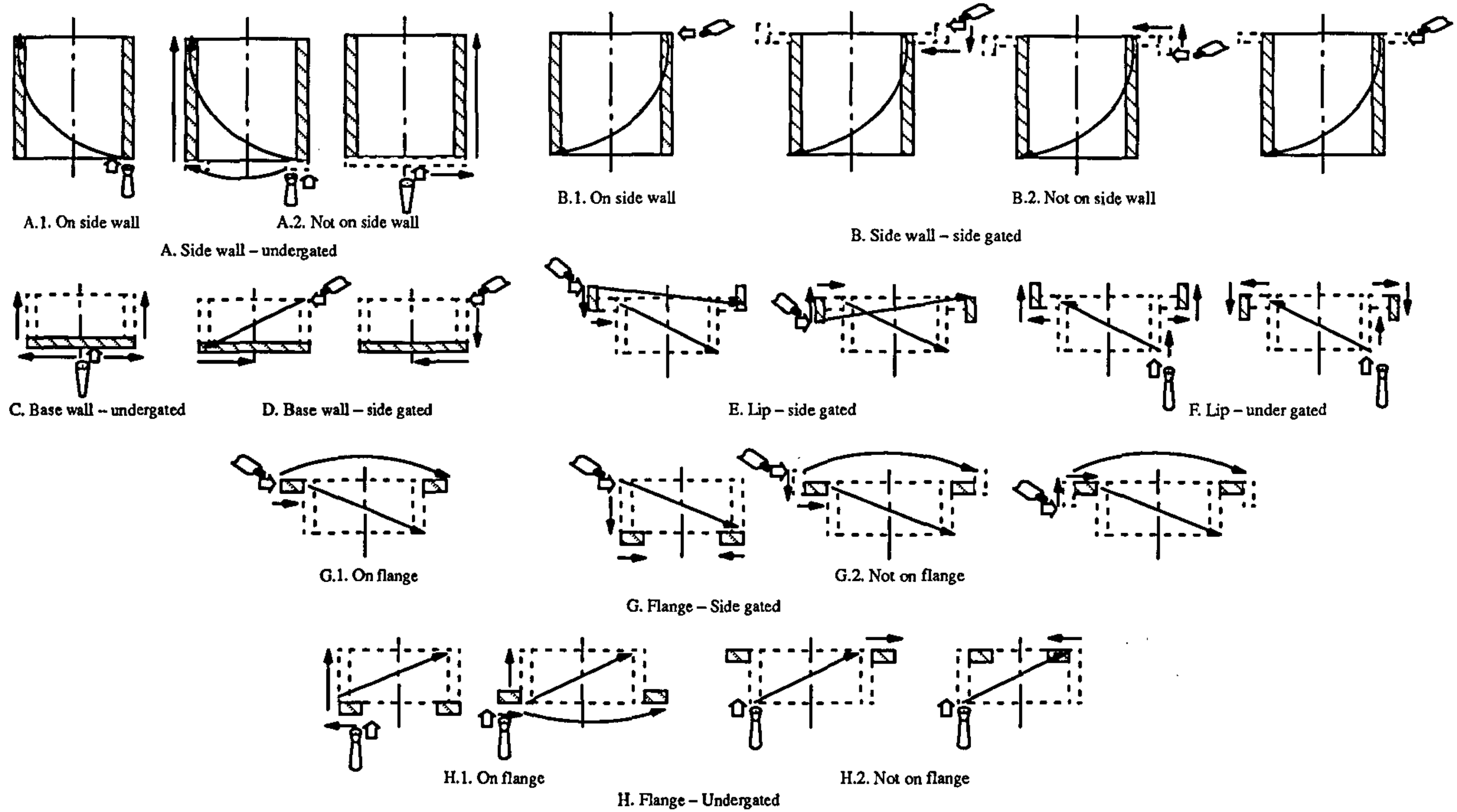


Figure 7.20. Flow length of material through 3D form features in an injection mould.

Type of associated form feature	Calculation of feeding length using mouldability feature attributes	
	undergated	side gated
Side wall	gate on side wall feeding distance = SQRT(SQ(height) + SQ(length/2.0)) gate not on side wall feeding distance = height	gate on side wall feeding distance = SQRT(SQ(height) + SQ(length/2.0)) gate not on side wall feeding distance = SQRT(SQ(height) + SQ(length/2.0))
Base wall	feeding distance = length – (shortest distance from gate to wall edge)	feeding distance = length/2.0
Lip	feeding distance = height	route 1 feeding distance = height route 2 feeding distance = SQRT(SQ(height) + SQ(length/2.0))
Flange	gate on flange route 1 feeding distance = height route 2 feeding distance = length/2.0 gate not on flange feeding distance = height	gate on flange route 1 feeding distance = height route 2 feeding distance = length/2.0 gate not on flange feeding distance = height
Base feature	diameter > adjacent base wall feeding distance = thickness diameter < adjacent base wall feeding distance = height	diameter > adjacent base wall feeding distance = thickness diameter < adjacent base wall feeding distance = height

Figure 7.21. Flow length calculation through 3D geometry

rules for calculation of the feeding length of plastic material through form feature instantiations in the Product model have been defined in Figure 7.21. It can be seen from the figure that some form feature types have two possible routes of material flow that reach the extremity of geometry. In Figure 7.21 the route 1 represents the feeding distance of plastic material through a wall geometry and out the other side. Route 2 is used in the feeding distance calculation only if the particular instantiation is the last feature into which material must flow, ie it is at the extremity of geometry from the gate. This can be ascertained using the adjacent feature identification technique discussed in section 7.4.2.

Once the feeding length through every feature instantiation is known they can be added together. The total feeding distance is evaluated and feedback advice is provided to the designer via the gate feature 'maximum feeding distance' constraint in the Manufacturing model. The designer is informed if additional gates are required on a product and/or the gate instantiation must be re–positioned. If more than one route to the extremity of geometry exists, the largest

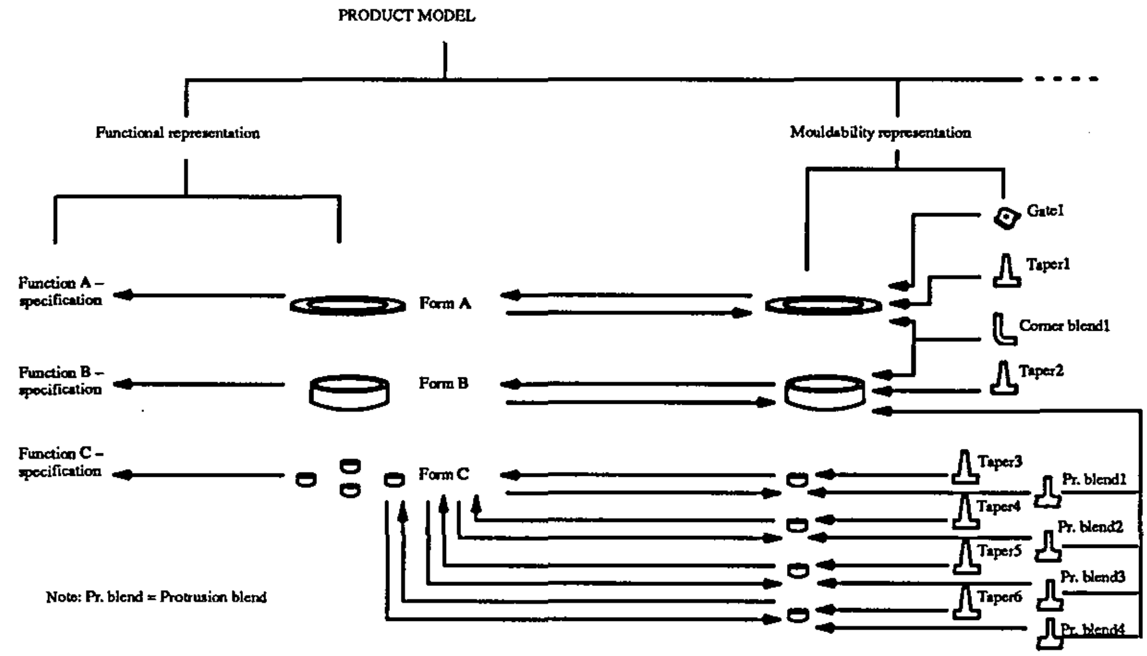


Figure 7.22. Example of relationships built up in the Product model including secondary mouldability representation

feeding distance must be the input to the gate feature 'maximum feeding distance' constraint in the Manufacturing model. The above provides a solution to the problem of evaluating feeding distance using rotational product geometry when the sequence of build up of geometry is unknown and the above structured methods provide a solution to the problem of gate feature instantiation.

This section has described the authors formulated solutions to the problems of instantiation of secondary mouldability features. The instantiation of secondary mouldability features in the Product model, with no direct relationship with form feature instantiations, requires the linking in the Product model of each secondary mouldability feature with the associated primary mouldability feature(s), so that the effect of taper, blend and gate instantiations can be evaluated with respect to other viewpoints in injection moulding design, eg the design of the cavity and core. Therefore the relationships built up in the Product model including secondary mouldability data are as shown in the example in Figure 7.22.

7.4.5. Mouldability design support strategy.

The mouldability design support strategy is shown in Figure 7.18. Any form modification results in the modification of the mouldability viewpoint and the provision of mouldability feedback advice. The designer may change geometry due to mouldability feedback and initiate the instantiation of secondary mouldability features. Any modification from the mouldability viewpoint results in corresponding modifications to the form.

Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.5. Cavity/core assessment – Cavity.

Previous work into cavity design has considered cavity design retrospective of design of the product, eg Cinquegrana (1990)a (1990)b, Gerdes, Webb and Chassapis (1994), Rho et al (1990), Huh and Kim (1989). However for concurrency a designer must be able to consider the consequences of product geometry for cavity design as the product evolves and modify the design of a product in response to cavity process constraints. To overcome this problem cavity impression features must be instantiated as the product is designed. This work has resolved the problem by defining both a translation process between individual mouldability features and cavity design features, and a strategy to support the designer in consideration of the cavity design as the product evolves. The problems involved and the formulated solutions are now outlined.

7.5.1. The instantiation of cavity impression features.

This section discusses the problems of instantiating the features in the cavity impression from the existing product definition data available defined in terms of functional, mouldability and form features. A procedure which supports the build up of the cavity (and core) geometry

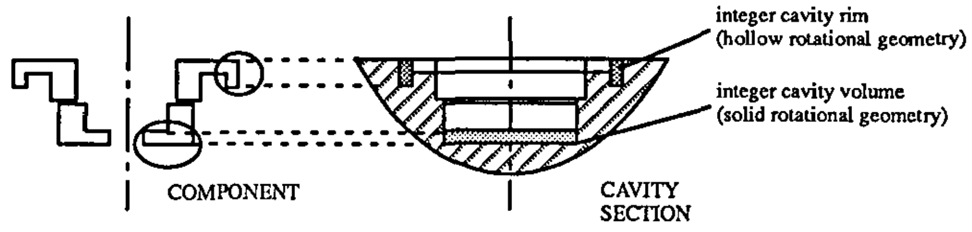


Figure 7.23. Translation problems: Identifying type of cavity space

by translation from the mouldability viewpoint has been identified in Appendix 1 (Drawing B1). This involves the identification of the parting line, the definition of the main cavity geometry, the definition of the main core geometry, the definition of any local inserts and finally the generation of any additional parting lines. These problems are discussed in this section with the exceptions of the parting line which is a cavity plate feature and is therefore discussed in section 7.5.2, and the main core geometry which is discussed in a separate section.

The initial problem for the instantiation of the cavity impression features defined in section 6.3.4 was to identify the main cavity geometry from mouldability wall features. Mouldability wall features can translate either to an *integer cavity volume* or to an *integer cavity rim* as illustrated in Figure 7.23. One must therefore identify whether a particular wall instantiation translates to a volume or to a rim. In most cases the translation of a wall feature will be to an integer cavity volume, which has solid rotational geometry. However non detection of a rim, which has a hollow rotational geometry, will result in the wrong shape of cavity, hence a structured method of rim identification is required.

The following method identifies whether a translation is to an integer cavity volume or an integer cavity rim which is shown in Figures 7.24 and 7.25. Wall feature instantiations in the Product model are identified (Figure 7.24) in order of the proximity of their base to the parting line (whose instantiation is discussed in section 7.5.2). Working from the furthest wall feature instantiation towards the parting line, the geometry of each wall instantiation is compared with that of the next furthest in order to check for the occurrence of rims. There are two ways in which a rim occurs in the cavity; A rim occurs (Figure 7.25 a)) when a wall fea-

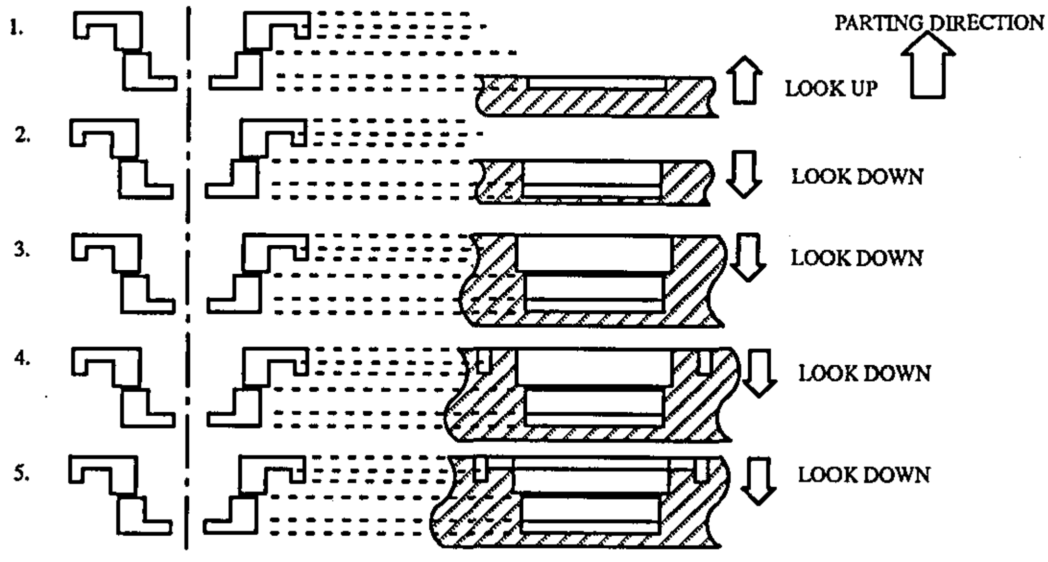


Figure 7.24. Sequential build up of cavity to identify type of geometry

ture instantiation with hollow rotational geometry is placed on the non parting line side of a wall instantiation with non–hollow rotational geometry, and also (Figure 7.25 d)) when an overhang is created in a direction parallel to the plane of the parting line.

To achieve translation, checking for the conditions in Figure 7.25 a) and 7.25 d) is carried out whilst working from the furthest wall instantiation towards the parting line. This requires that the geometry of the furthest wall feature instantiation be evaluated against the next furthest to look for the conditions in Figure 7.25 a), and that the geometry of each of the other wall feature instantiations be evaluated against the adjacent wall whose base is further from the parting line to look for the conditions in Figure 7.25 d). If the furthest wall instantiation has hollow rotational geometry and the next furthest has non–hollow geometry, the furthest wall translates to an integer cavity rim (Figure 7.25 a)). If the geometry of the next furthest is also hollow then the furthest wall translates to an integer cavity volume (Figure 7.25 b)). With respect to the remaining wall feature instantiations, for a rim to exist a wall must partially enclose an adjacent wall whose base is further away or the same distance from the parting line as in Figure 7.25 c) and d). If no contact exists between the encloser and the enclosed, as in Figure 7.25 d) the encloser wall instantiation translates to an integer cavity rim. If there is contact between the encloser and the enclosed the encloser wall instantiation translates to an integer cavity volume, as in Figure 7.25 c). The above provides a solution to the problem

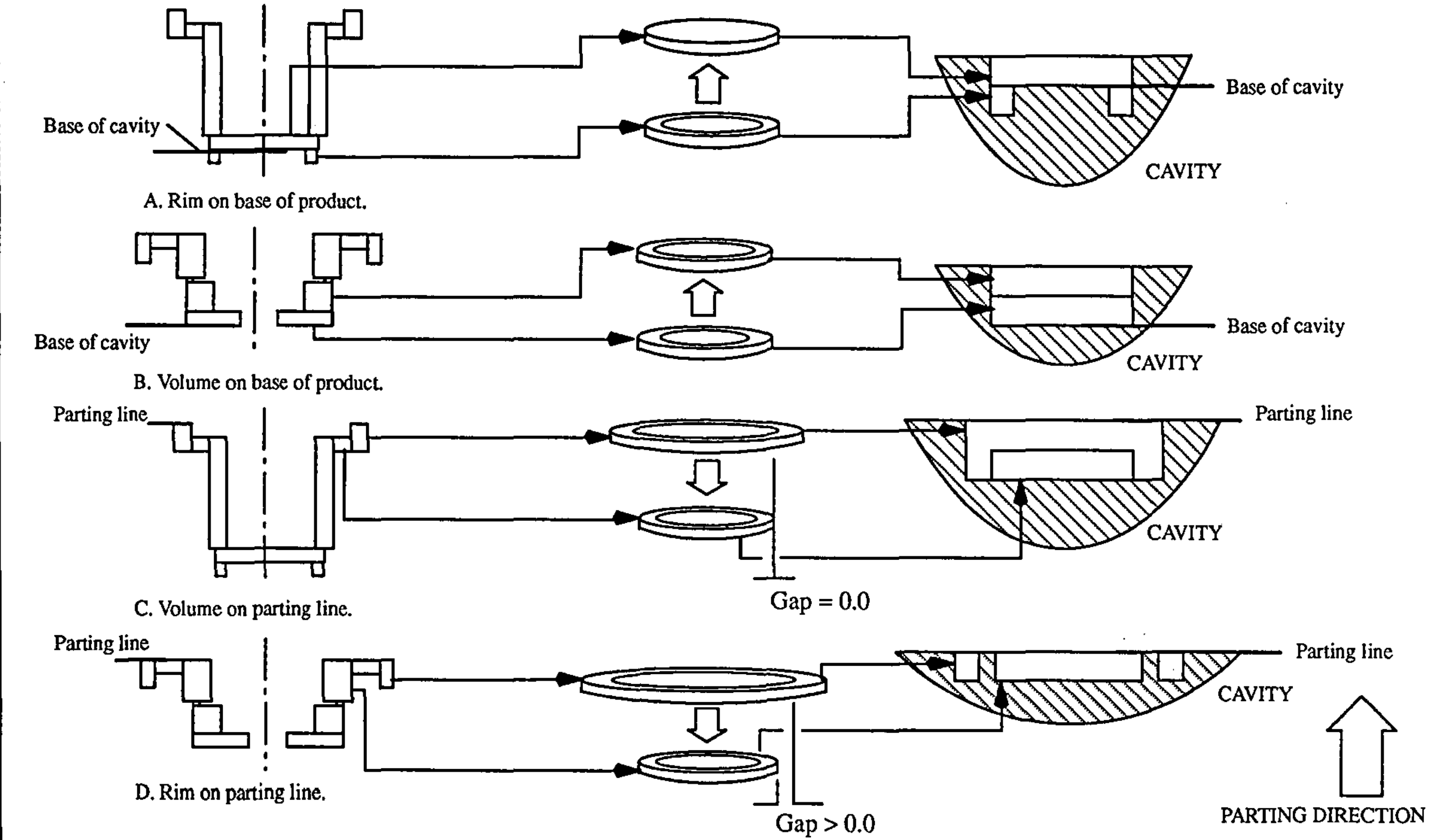


Figure 7.25. Differentiating between rims and volumes during cavity creation

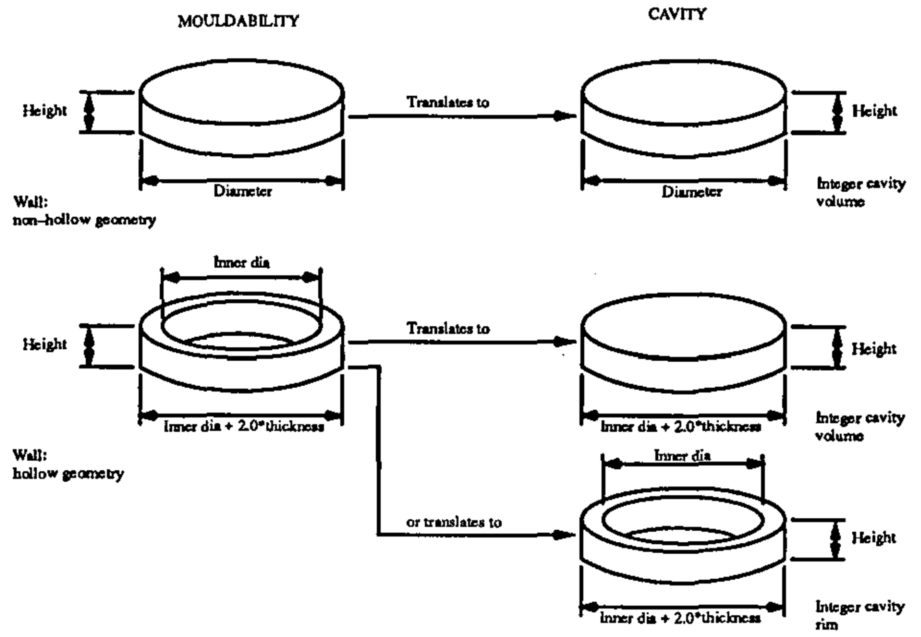


Figure 7.26. Integer cavity rims and integer cavity volumes take mouldability wall parameters unchanged

of identifying equivalent cavity feature types of mouldability wall feature instantiations in the Product model. As shown in Figure 7.26, both the integer cavity volume and integer cavity rim take their position, height and outer diameter, from the equivalent wall geometry. The integer cavity rim also takes the inner diameter of the equivalent wall geometry.

While the comparison of these cavity feature types provides a basis for the identification of most potential moulding problems, this cannot cope with a situation where a gap may occur in the main cavity(/core) geometry due to reinforcement features in a product bridging the gap between two walls, as shown in Figure 7.27. Therefore a method of considering the process constraints of the cavity(/core) in relation to gaps in the cavity(/core) geometry is required. While the work of (for example) Hui and Tan (1994), and Mochizuki and Yuhara (1992) has been in the area of automatic undercut detection in moulded components using complex algorithms in a solid modeller, it has been performed on a completed mould geometry with no potential links back to product design and has not addressed the specific problems created by such gaps. Concurrent feedback on gaps has been addressed in this thesis by the inclusion of the *integer cavity group volume* in the cavity features set which must be instantiated when a gap has been identified.

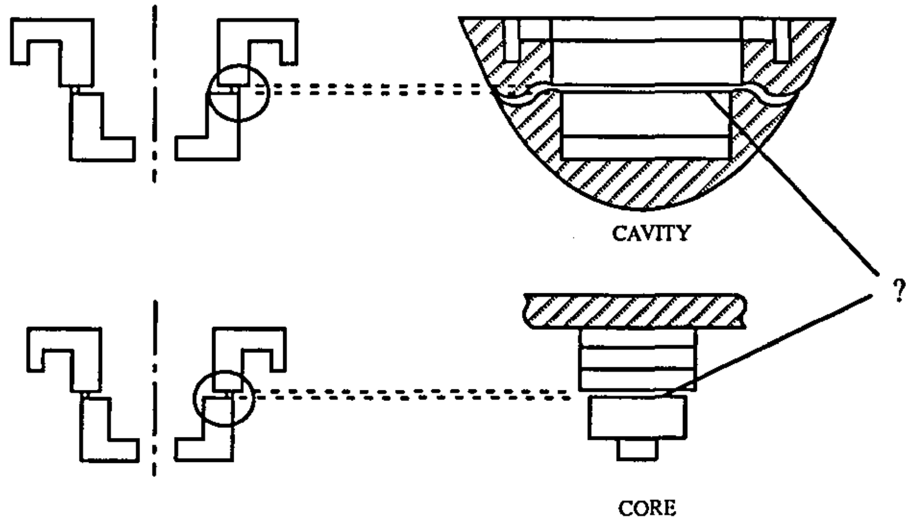


Figure 7.27. Translation problems: Dealing with gaps in cavity/core walls.

A gap between adjacent integer cavity volume instantiations can be identified simply by the relative position and dimensions of the geometry. Once a gap is identified, it must be established that the gap is bridged by reinforcement instantiations. To bridge a gap a reinforcement instantiation must be within the gap vertically and within the horizontal boundaries of the cavity impression at the level of the furthest of the two integer cavity volumes from the parting line.

Having established that an integer cavity group volume must be instantiated, the dimensions of the group volume must be obtained from the group of reinforcement instantiations in the gap. In the context of this work, such a group of reinforcement instantiations can only be mouldability solid bosses or mouldability ribs. The author has formulated below a structured method of obtaining integer cavity group volume dimensions from a group of reinforcement instantiations so that the process constraints of a gap in the cavity(/core) geometry can be considered.

Group volume dimensions are taken from the form feature instantiation that is associated with the reinforcement instantiations in a gap, as shown in Figure 7.28. In the context of a gap in the cavity the associated form feature instantiation can only be spaced ribs if the mouldability instantiations are of the rib type, or spaced bosses if the mouldability instantia-

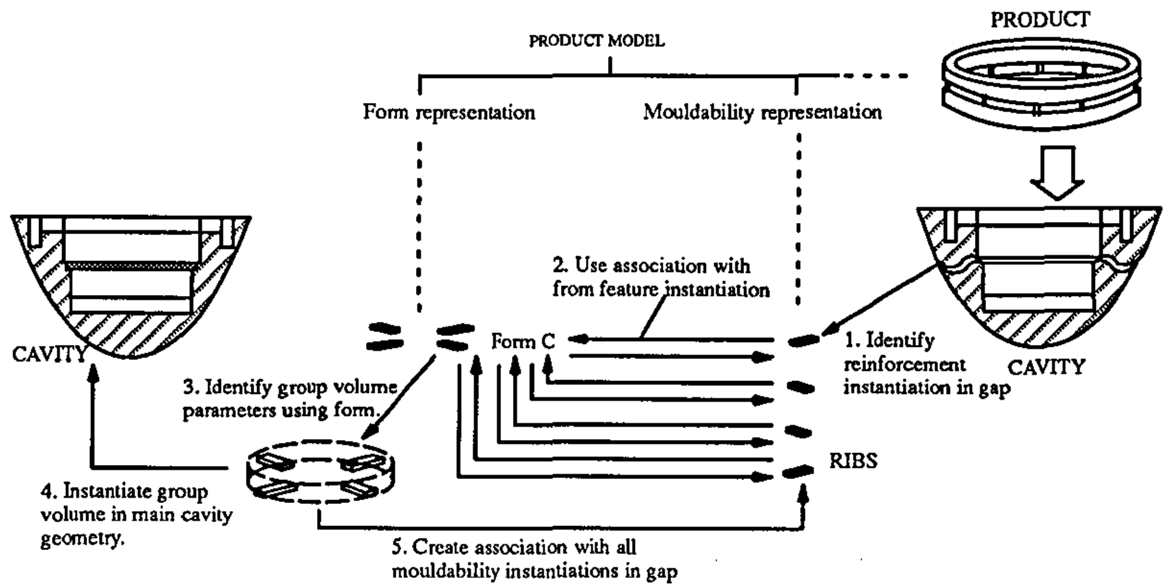
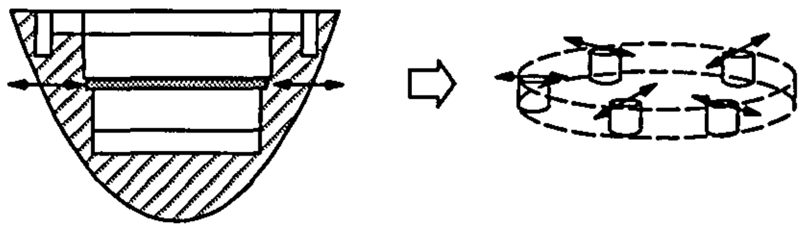


Figure 7.28. Instantiation of integer cavity group volume from ribs and solid bosses in a gap.

tions are of the solid boss type. Selecting one mouldability instantiation, the association with a form feature instantiation is used to obtain the identity of the other mouldability instancations in the gap, via their associations with the same form feature instantiation, the parameters of the integer cavity group volume being obtained from the form feature instantiation. If the form feature instantiation is spaced ribs the diameter of the integer cavity group volume is the outer diameter of the ribs grouping, the height is the height of the rib group members and the position is the position of the base of the central axis of the spaced ribs instantiation (see Figure 7.12). If the form feature instantiation is spaced bosses the diameter of the integer cavity group volume is the axis diameter of the spaced bosses plus the diameter of a boss group member, the height is the height of the boss group members and the position is the position of the base of the central axis of the spaced bosses instantiation (see also Figure 7.12). The above method enables integer cavity group volume instantiation.

The integer cavity group volume enables consideration of process constraints of gaps in the main cavity geometry as follows. Adjustments of integer cavity group volume dimensions in response to feedback advice, eg for undercuts in the cavity, are used to adjust the position of the associated mouldability feature instantiations (via the form) so that the mouldability instantiations' relationship with the dimensions of the integer cavity group volume is main-



A. Adjustment of group volume dimensions due to feedback advice   B. Corresponding adjustment of mouldability instantiation positions.

Figure 7.29. Constant relationship between dimensions of integer cavity group volume and mouldability instantiations in a gap.

tained, as shown in Figure 7.29. The constraints associated with the integer cavity group volume in the Manufacturing model are discussed in Chapter 8. The above structured solution solves the problem of considering process constraints of gaps in the main cavity geometry bridged by reinforcement instantiations. The consideration of gaps in the main core geometry is discussed in section 7.6. The above structured methods for wall translation and dealing with gaps in the cavity solve the problems of instantiation of the main geometry in the cavity impression.

Having instantiated the main cavity and core geometry, the next problem is the instantiation of any local inserts. Previous work has carried out geometric analysis of a completed cavity and has not identified local inserts separately, eg Cinquegrana (1990)a (1990)b, Gerdes, Webb, Chassapis (1994). The authors work identifies local inserts as individual features, namely *integer cavity boss, integer cavity hole* and *integer cavity slot* features, plus an *integer cavity rim* if it is translated from a hollow boss instantiation rather than that of a wall. Local inserts are translated from mouldability reinforcements and holes. Reinforcement features that are associated with an integer cavity group volume should not be translated to local inserts because their process constraints in the cavity have been considered during instantiation of the group volume. As shown in Figure 7.30, it is necessary next to identify whether a reinforcement or hole instantiation translates to a local insert in the cavity or in the core. Instantiation of local inserts in the core is discussed in section 7.6. If a reinforcement or hole feature translates to a local insert in the cavity, then a third problem arises, to identify the type of cavity feature to which it translates, ie a cavity rim, boss, hole, slot or a combination.

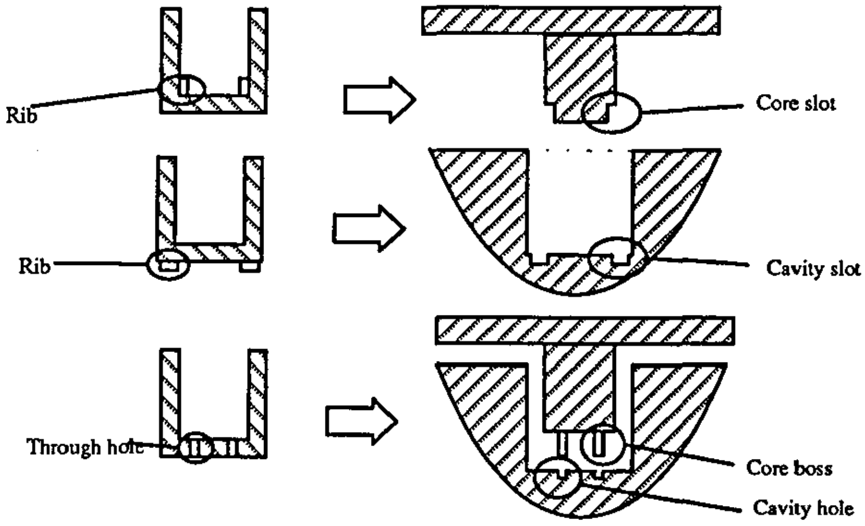


Figure 7.30. Translation problems: identifying equivalent entities created in cavity or core by reinforcements and holes

As shown in Figure 7.28, reinforcement instantiations inside a group volume (and therefore excluded from translation to a local insert) can be identified by the association in the Product model between an integer cavity group volume and those reinforcements from which it was instantiated. The rules for translation of reinforcement and hole instantiations have been identified by the author as in Figure 7.31. Figure 7.31 a) shows that solid boss instantiations translate to integer cavity holes if their geometry is below the base of the core, or to integer **core** holes if their geometry is above the base of the core. Similarly in Figure 7.31 b) hollow boss instantiations translate to integer cavity rims if their geometry is below the base of the core, or to integer **core** rims if their geometry is above the base of the core. From Figure 7.31 c), if the geometry of a rib instantiation is below the core base the translation is to an integer cavity slot. If the rib instantiation is above the base of the core, then if it is inside the diameter of the cavity impression the translation is to an integer **core** slot, otherwise it is to an integer cavity slot. The rules for the translation of hole instantiations are shown in Figure 7.31 d). If a hole instantiation is blind and above the base of the cavity the translation is to an integer **core** boss. If a blind hole instantiation is on or below the base of the cavity the translation is to an integer cavity boss. However the translation of a through hole is to an integer **core** boss **and** an integer cavity hole. The integer cavity hole provides location for a core pin in the mould.

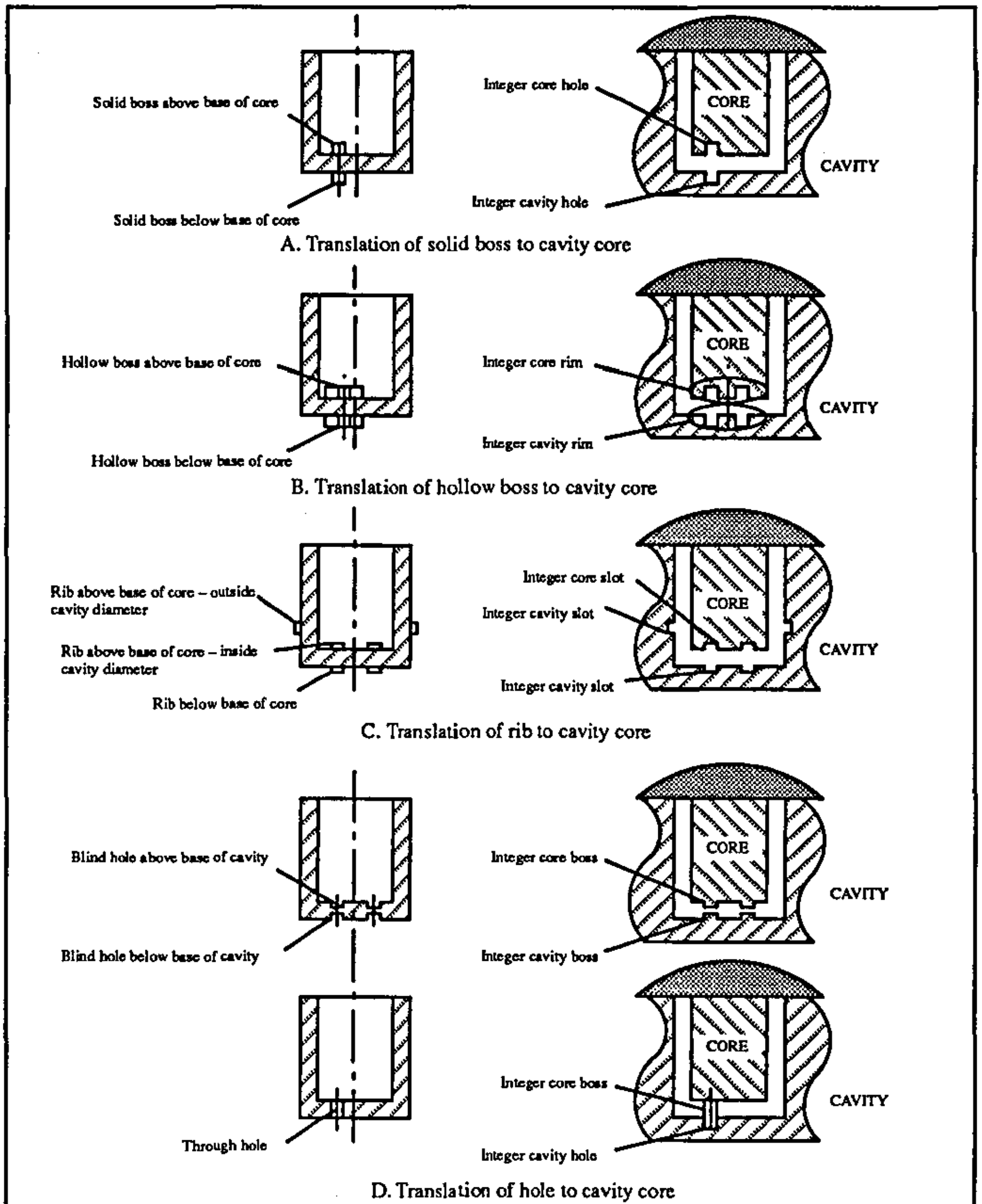


Figure 7.31. Translation of reinforcements and holes to the cavity/core.

As shown in Figure 7.32, local inserts are the equivalent of reinforcement and hole instantiation geometry in the cavity. Therefore for the translation of ribs, solid bosses, hollow bosses and blind holes, the dimensions for the corresponding integer cavity slots, holes, rims and bosses are taken unchanged. However for the translation of a through hole the length of an integer core boss is increased downwards by 5 mm to give some location in the cavity, and the depth of the corresponding integer cavity hole is the distance from the base of the cavity to the base of the integer core boss. The above structured solution solves the problems of instantiation of local inserts in the cavity. The instantiation of local inserts in the core is dis-

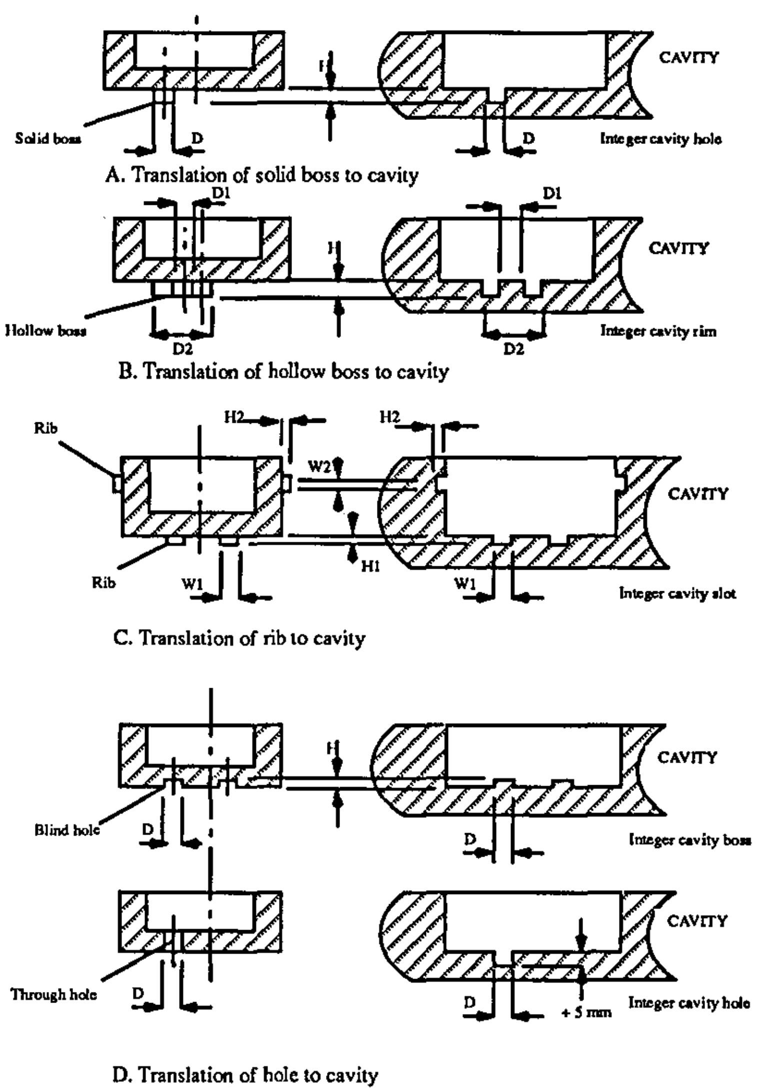


Figure 7.32. Reinforcements and holes parameters are unchanged in cavity feature instantiations.

cussed in section 7.6.

A further problem that relates to both the main cavity geometry and the local inserts is the translation of secondary mouldability feature instantiations to the cavity viewpoint. As indicated in Figure 7.33, it is necessary to first identify whether there is a need for an *integer cavity blend* in the cavity. As a blend is between two primary mouldability instantiations, it is essential to establish to which other cavity geometry the new integer cavity blend instantiation should be attached. If the mouldability blend is of the corner blend type, one needs to know whether the larger radius or the smaller radius should be applied to the cavity, ie which

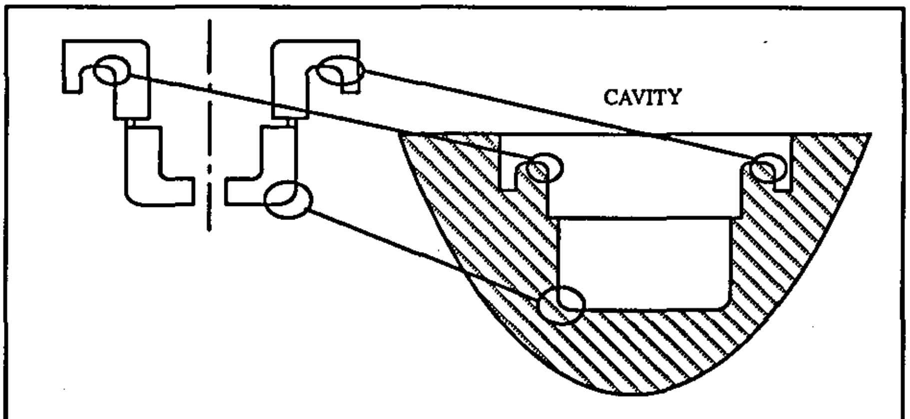


Figure 7.33. Translation problems: identifying blends between cavity entities

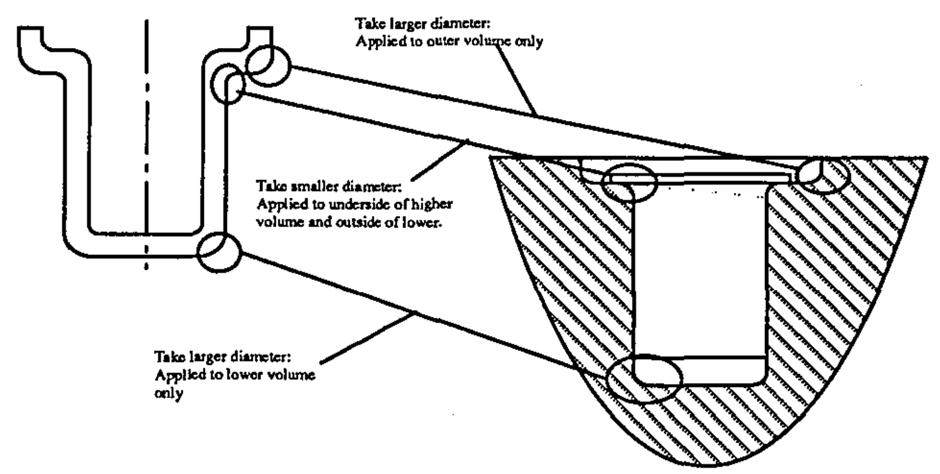


Figure 7.34. Translation problems: Applying blends correctly to cavity geometry

radius should be on the cavity and which on the core. An additional problem (Figure 7.34) is how to apply an integer cavity blend to the cavity geometry to correctly replicate the outside product geometry. It can be seen in the figure that the manner of application of an integer cavity blend to cavity geometry is not the same as the application of a mouldability blend to product geometry.

Mouldability taper features translate to *integer cavity tapers*. The instantiation problems of an integer cavity taper are similar to those of an integer cavity blend: Firstly a requirement for an integer cavity taper in the cavity must be identified. It is then essential to know how to apply an integer cavity taper to the cavity geometry in order to correctly replicate the outside product geometry as illustrated in Figure 7.35.

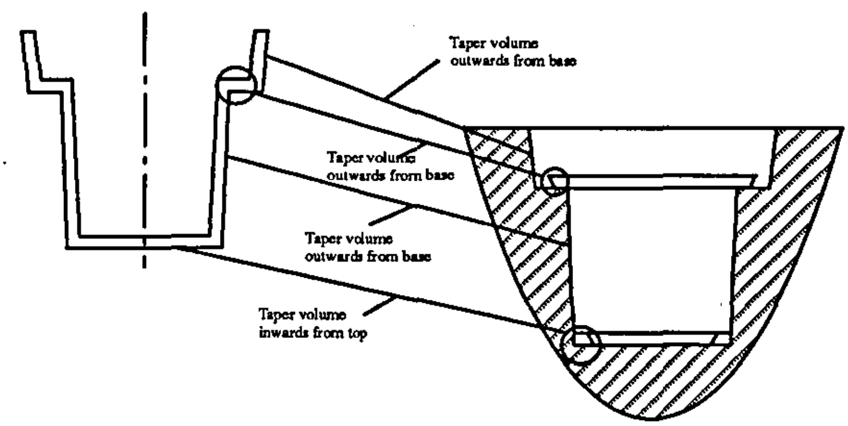


Figure 7.35. Translation problems: Applying tapers correctly to cavity geometry

To identify the requirement for an integer cavity taper and/or an integer cavity blend in the cavity the existence of a taper and/or a blend on a translated primary mouldability feature instantiation must be detected. This can be achieved using the Product model structure in Figures 7.22 and 7.36. The associations in Figure 7.36 identify the equivalent primary mouldability instantiation of a cavity feature instantiation, and therefore using the associations between primary and secondary mouldability feature instantiations shown in Figure 7.22, the existence of tapers and blends on the equivalent primary mouldability instantiation can be identified. If a blend exists on a primary mouldability instantiation, there is a problem of identifying the adjacent geometry in the cavity to which an integer cavity blend instantiation should be attached. This can also be identified using the associations in the Product model shown in Figures 7.22 and 7.36. Having identified the equivalent primary mouldability instantiation and the existing blend, the associations in Figure 7.22 allow identification of the other primary mouldability instantiation to which the blend instantiation is attached. The cavity equivalent of the former is then identified using the associations in Figure 7.36.

With respect to the translation of blends a further problem exists if a blend is of the corner type, namely to identify whether the smaller or larger radius is to be used in the cavity or the core, ie which is to be used in the integer cavity blend instantiation. A method of radius identification for integer cavity blend instantiations is shown in Figure 7.37. Using the associations in Figure 7.36 and 7.22 the equivalent wall feature instantiation for each integer cavity volume instantiation is known. Therefore which radius from a corner blend to be used

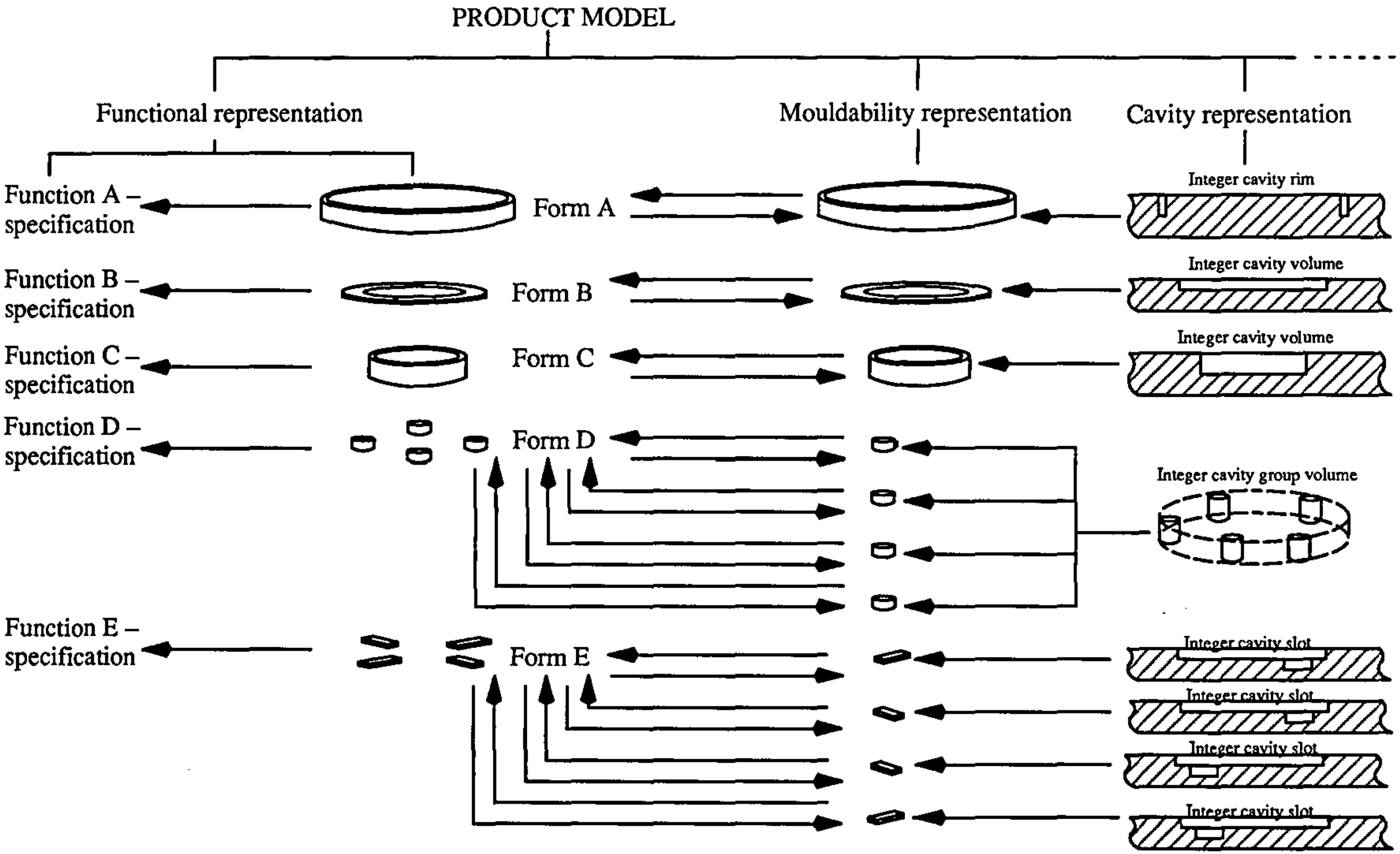


Figure 7.36. Example of relations built up in Product model including cavity representation

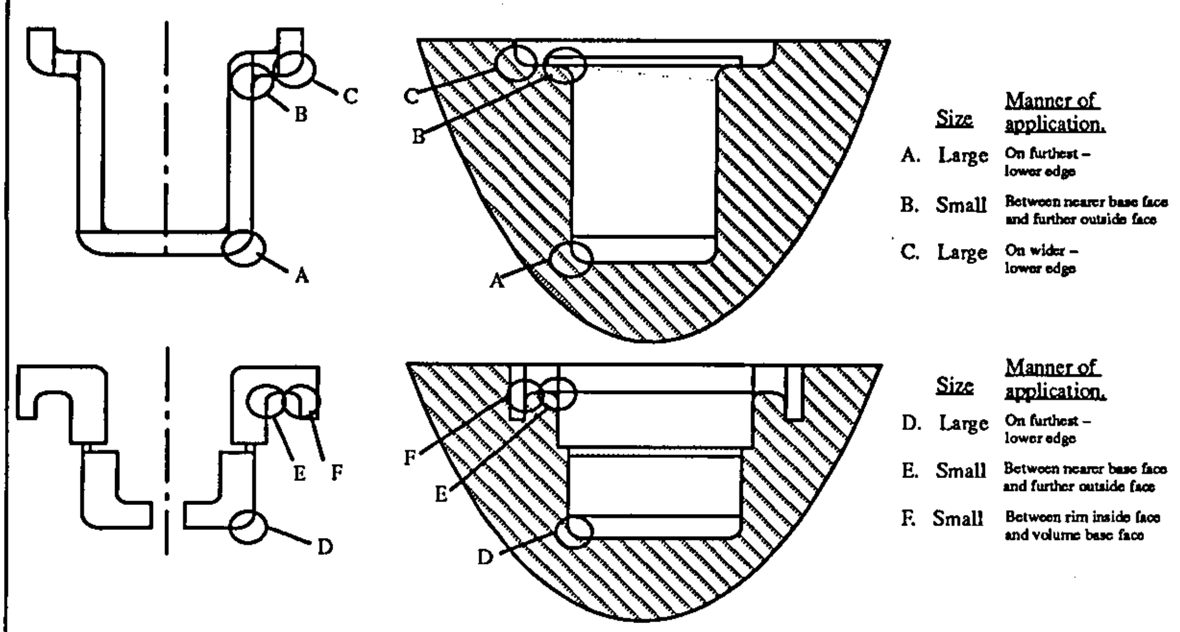


Figure 7.37. Identifying blend radii and manner of application between cavity entities

in an integer cavity blend instantiation is identified by comparing the dimensions and positions of the equivalent blended wall feature instantiations in the Product model, working from the furthest wall instantiation from the parting line to the nearest. Note that a wall feature instantiation that translates to an integer cavity rim is considered as the nearest to the parting line, even though an enclosed wall instantiation will be nearer. Thus the dimensions of those wall instantiations that translate to an integer cavity volume are compared as far as the parting line, and the last of these wall instantiations is compared with the one that translates to an integer cavity rim. The comparison in the product is as follows: If the further of two wall instantiations is 'closing off' (Figure 7.37 a) and d)) the larger of the radii in the corner blend is instantiated in the cavity. If two wall feature instantiations are the same distance from the parting line (Figure 7.37 c)) the larger of the radii in the corner blend is instantiated in the cavity. In any other comparison the smaller of the radii in the corner blend is instantiated in the cavity.

The formulated solution to the application of integer cavity blends and integer cavity tapers in a manner that replicates the outside geometry of the product is shown in Figures 7.37 and

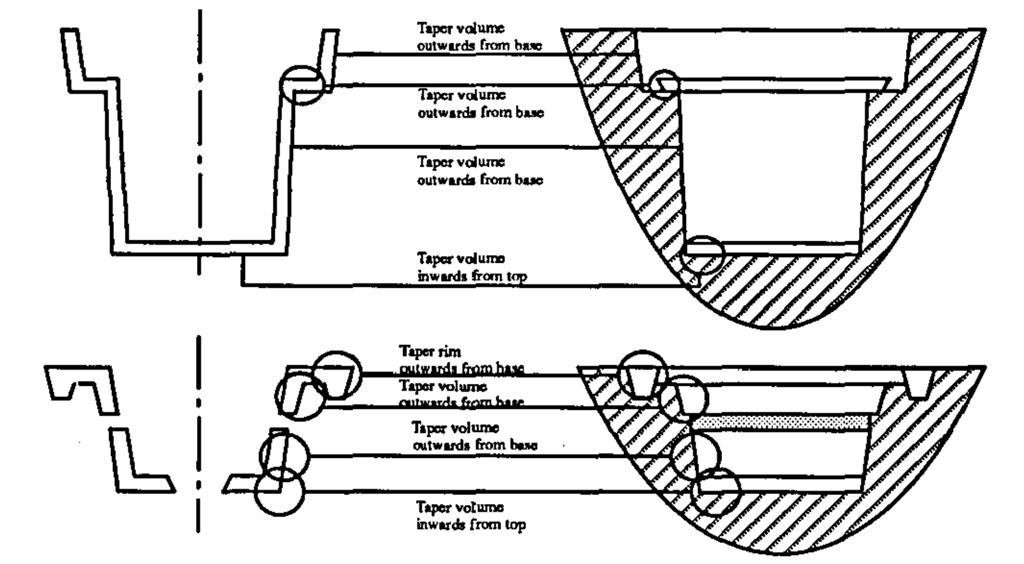


Figure 7.38. Applying tapers correctly to cavity geometry using the rules that apply to the corresponding mouldability geometry.

7.38 respectively: Looking at wall to wall relationships in the product; in 'closing off' scenarios 7.37 a) and d) an integer cavity blend is applied to the lower outside edge of the furthest integer cavity volume instantiation from the parting line. In the same distance scenario of 7.37 c) an integer cavity blend is applied to the lower outside edge of the wider integer cavity volume instantiation. In a partial enclosure scenario (Figure 7.37 b) and e)) an integer cavity blend is applied between the base face of the wider integer cavity volume instantiation and the outside face of the other. In a full enclosure scenario (Figure 7.37 f)) an integer cavity blend is applied between the inside face of the integer cavity rim instantiation and the base face of the integer cavity volume. As shown in Figure 7.38, the same rules apply for the application of integer cavity tapers in the cavity as for mouldability tapers on primary mouldability geometry. The above provides a structured solution to the problems of the instantiation of integer cavity blends and integer cavity tapers in the cavity.

The structured methods described above provide a solution to the problems of translation from the mouldability features set to an alternative set of cavity impression features which maintains the links between the structure of geometry of the two viewpoints. Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.5.2. The instantiation of cavity plate features.

A set of geometric relationships can be defined between the cavity impression and the plate such that as the impression is instantiated, so feedback can be provided on the consequences of product geometry for the cavity plate design as the product geometry evolves. Therefore the design of the cavity plate can be modified as the product is developed by the designer. For cavity plate feature instantiation, these geometric relationships must be evaluated, most of which require an additional input of the geometry of mould system elements such as the feeding or cooling system, and/or the geometry of other cavity plate feature instantiations. These prerequisite data requirements and the geometric relationships can be captured in relation to each cavity plate feature type. This captured data can be used to support a strategy for instantiation of the cavity plate features. This places a requirement on the cavity design strategy as described in section 7.5.3. The problems of the above approach for instantiation of each cavity plate feature is considered in the remainder of this section, and the formulated solutions are described.

7.5.2.1. Cavity parting line feature instantiation.

The instantiation of the *integer cavity parting line* is a prerequisite to instantiation of the cavity impression features, because undercuts in the mould must be considered with respect to the position of the parting line. Furthermore the parting line is required to identify the types of some features in the cavity impression, as described in section 7.5.1. Hui and Tan (1992) (1994) using sweep operations, and Mochizuki and Yuhara (1992) using direction vectors of half spaces, used complex algorithms in a solid modeller to identify the optimum withdrawal direction(s) of a component. All retrospectively analysed the developed component geometry. In investigating the concurrent design of the product and mould, a method of identifying the parting line at each juncture of product development is required, so that the consequences of the product geometry for mould design can be considered and vice versa.

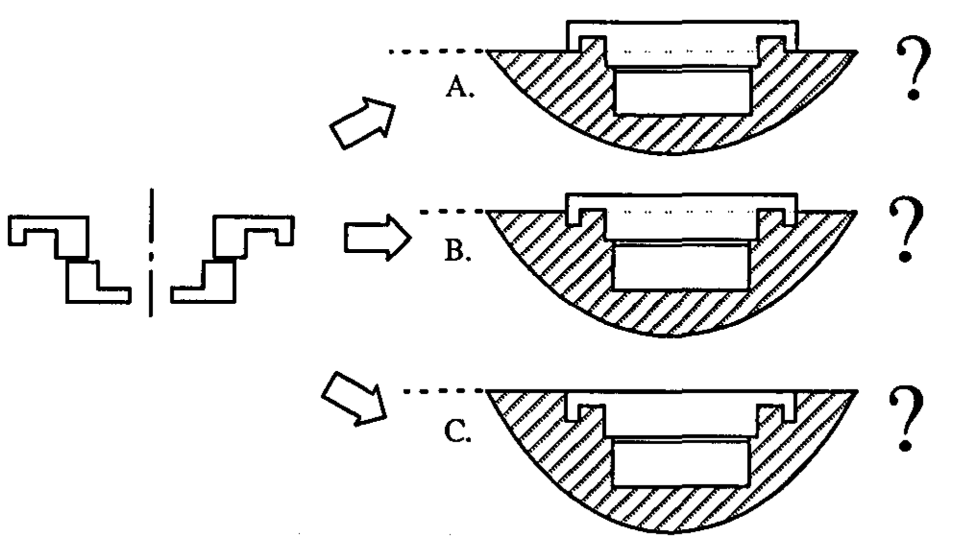


Figure 7.39. Problem of identifying a parting line position.

For rotational products the draw direction can be assumed to be along the central axis of a component, the parting line being assumed to be flat and in the horizontal plane. The parting line can therefore be identified as being on the widest part of the product when viewed in the draw direction, and where on that widest section to place the parting line (see Figure 7.39) must be identified. Another problem is to identify a requirement to create a split mould and the resultant requirement for a vertical parting line instantiation.

As shown in Figure 7.39, there are three possible solutions for instantiation of a horizontal parting line, the simplest of which is 7.39c. The methods in Figure 7.39a and 7.39b require a hollow core, and in Figure 7.39b careful alignment of the cavity and core is required to avoid a visible mark on the product or a mis–shapen product. Using the method in Figure 7.39c the parting surface is perfectly flat and the need for a hollow core and the problem of careful alignment of edges is avoided. Furthermore the parting line is not visible on the product unless flash develops.

Identifying the requirement for a vertical parting line is possible by comparison of individual integer cavity volumes, group volumes and rims during instantiation of the cavity impression features. Each individual integer cavity volume, group volume or rim can be evaluated with respect to overhangs in the cavity. If an overhang is detected feedback advice can be provided to the designer via process constraints data associated with the feature. This offers the opportunity to make changes as an alternative to instantiation of a vertical integer cavity parting line. To demonstrate a vertical parting line instantiation the simplifying assumption has been made that a split line is in the plane of the y and z axis. Considering only rotational products the widest part of a product has been assumed to be along the central axis of the main product geometry.

Having instantiated the integer cavity parting line(s) and the cavity impression features, the remaining features in the cavity plate need to be instantiated. The examples in Figure 7.40 show that the cavity plate features have a geometric relationship with the parting line and with the cavity impression. However it can be seen from the figure that the nature of this relationship is also dependant upon the type of feeding system. The example in Figure 7.40 shows how different cavity plate features may be instantiated depending on the type of feeding system, in this case a two plate mould (Figure 7.40b) or a three plate mould (Figure 7.40a). Therefore the feeding system geometry must be instantiated in the Product model before the remaining cavity plate features can be considered. The instantiation of the feeding system features is discussed in section 7.7.

As described in the following sections, it has been identified that data on cavity lands is a prerequisite to cavity block instantiation, and cavity block data is a prerequisite to instantiation of a backing plate or a nozzle recess (if required). The order of instantiations must therefore be i) lands, ii) block, iii) backing plate or nozzle recess (mutually exclusive).

7.5.2.2. Cavity land features instantiation.

Having instantiated the feeding system features, the next cavity plate feature to be instantiated should be the integer cavity inner land. The inner land is a small area around the edge of the impression where the parting surfaces are bedded down to provide an effective seal. The remainder of the parting surface in the vicinity is relieved to a minimum depth of 2.4

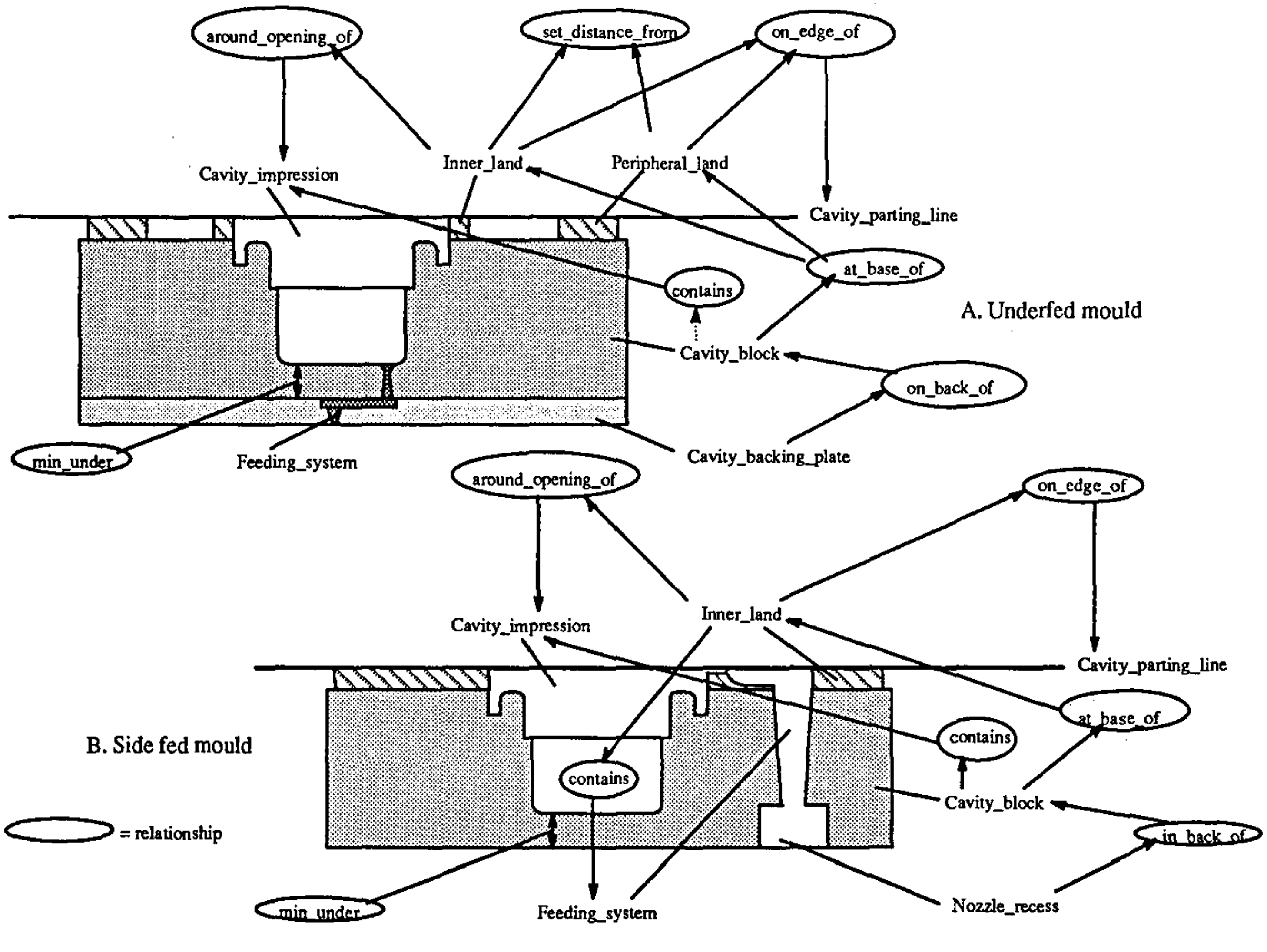


Figure 7.40. Geometric relationships of impression with the mould plate.

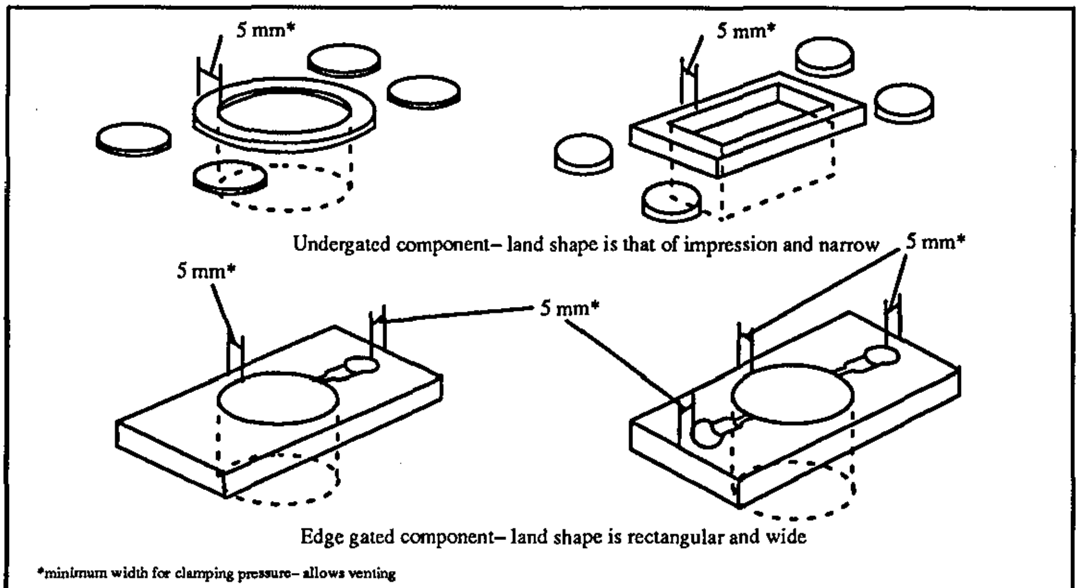


Figure 7.41. Land size and shape is related to gate type and numbers.

mm. As shown by the examples in Figure 7.41, the general problems of inner land instantiation are, i) an inner land must enclose the cavity impression, and to accomplish this enclosure, the minimum size and the shape of the land is dependant upon the type and geometry of the feeding system as well as the width of the cavity impression, ii) the size of an inner land must take account of venting considerations, which are again dependant upon the type and geometry of the feeding system. In practice the shape of an inner land can be any shape, depending on the shape of the opening and the nature of the feeding system. In order to consider any shape of inner land a complex analysis capability is required, which is not available in this work. To demonstrate that cavity impression and feeding system data can be used at an early stage as an input to decision making with respect to the size and shape of the lands, and the extent to which this can be in parallel with the design of the product, the following method has been used.

Considering only rotational products the simplifying assumption has been made that an inner land can be of two types, an *integer cavity circular land* or an *integer cavity rectangular land.* It can be seen from Figure 7.41 that a rectangular land instantiation is required in order to achieve enclosure of a cavity impression with a side gate feeding system. If a cavity impression is under gated a circular land instantiation is required. The gate type in the Product model thus defines the required shape. If the gate is a side gate, ie fan gate, rectangular edge gate, film gate, sub–surface gate, tab gate, overlap gate or ring gate, an integer cavity rectangular land is instantiated. If the gate is an under gate, ie sprue gate, pin gate or diaphragm gate, an integer cavity circular land is instantiated.

The land position can be assumed to be on the centre of the cavity impression, at the parting line minus the depth of the land. The depth of a land is assumed always to be the minimum value of 2.4mm, Pye (1989). Therefore to achieve enclosure the size of the inner land instantiation can be calculated based on the venting requirements, as shown in Figure 7.41. The diameter of an integer cavity circular land instantiation is thus the width of the cavity impression at the parting line plus 10 mm, ie 5 mm all round to allow venting of the cavity impression. A side gated feeding system can be assumed to always be along the x axis and therefore the x dimension of an integer cavity rectangular land instantiation has been calculated as the width of the cavity at the parting line plus the length of the feeding system plus 10 mm, allowing venting of the feeding system as shown in Figure 7.41. The y dimension is the width of the cavity impression at the parting line plus 10 mm. The above methods solve the problems of instantiating an inner land.

If a narrow inner land is instantiated, ie an integer cavity circular land in the context of the authors work, the surface area at the parting line is likely to be insufficient to withstand the injection forces and deformation and flash formation may occur, Pye (1989). Therefore *integer cavity peripheral lands* are required to increase the surface area. As shown in Figure 7.42, the problems of peripheral land instantiation are, i) the total surface area of peripheral lands should make up the difference between the surface area of a narrow land and the area of an equivalent wide land. This requires identifying the surface area of each land, and ii) in the literature, and in practice there is a wide variety of shapes, spacing and numbers of peripheral lands, and no specific guide-lines exist, other than to provide balanced clamping forces. This creates a problem of the identification of these factors for instantiation of peripheral lands.

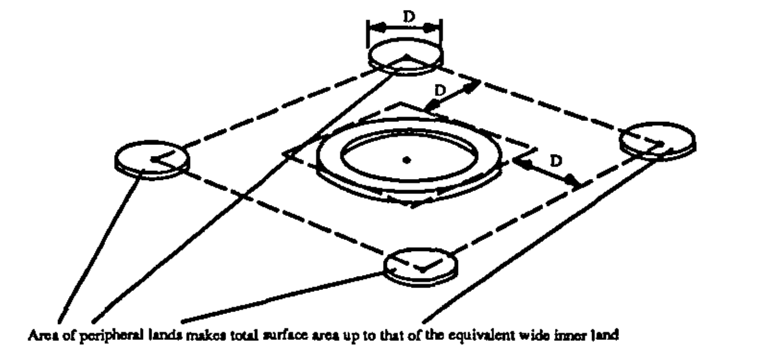


Figure 7.42. Peripheral lands are required to make up the surface area of narrow inner lands

As shown in Figure 7.42, in the absence of specific guidelines the number of peripheral lands has been assumed to be four, equispaced around the cavity impression, and the shape has been assumed to be circular. The surface area of an integer cavity peripheral land instantiation is therefore one quarter of the difference between the actual surface area of a narrow inner land instantiation and the surface area of one that is wide. In order to enable bedding down of the parting surface over the entire area, the maximum width of an inner land is 25 mm, Pye (1989). Therefore this dimension has been used to identify the surface area a wide inner land instantiation would have. Having done this the depth is the same as that of the inner land (2.4 mm), leaving the problem of spacing. As no guide-lines have been found in practice or in the literature for the spacing of peripheral lands, the author has assumed a spacing relating to the diameter of an integer cavity circular land instantiation and the diameter of the peripheral land instantiations, as shown in Figure 7.42. A minimum spacing constraint may exist due to machining access. However this is outside the scope of the present work.

7.5.2.3. Cavity block feature instantiation.

The examples in Figure 7.40 show that the general problems of *integer cavity block* instantiation are, i) to provide a depth under the cavity impression that avoids distortion of the mould due to injection forces, ii) to provide sufficient depth in the block to accommodate the mould system elements. The mould system elements that have a potential effect on the depth of an integer cavity block instantiation are the feeding system and the cooling system, and iii) to provide sufficient width and length in the block to accommodate the cavity impression, **lands** and the mould system elements. Those mould system elements with a potential effect on the width and length of an integer cavity block instantiation are the cooling system, and the guide system. Using the following method it can be demonstrated that cavity impression, feeding system and cavity lands data can be used at an early stage in product design as an input to decision making with respect to the dimensions of a cavity block, and the extent that these can be considered in parallel to product design can be shown.

With respect to the depth of a cavity block, in the absence of a complex analysis capability the minimum depth of steel beneath a cavity impression can be calculated using Releaux's formula for a circular flat plate secured all around the edge with a uniformly distributed load over the unsupported area, where the unsupported area is the projected area of the cavity impression, Pye (1989). Considering the depth requirement for the feeding system, if the cavity impression is under gated the depth of an integer cavity block instantiation under the impression must be sufficient to accommodate the sprue (sprue or diaphragm gated) or the gate (pin gated). If the cavity impression is side gated then the total depth of an integer cavity block instantiation must be sufficient to accommodate the main feeding sprue. Thus if the base of a sprue or pin gate is below that of an integer cavity block instantiation, the depth must be increased so that the base of the sprue or gate is on the base of the integer cavity block.

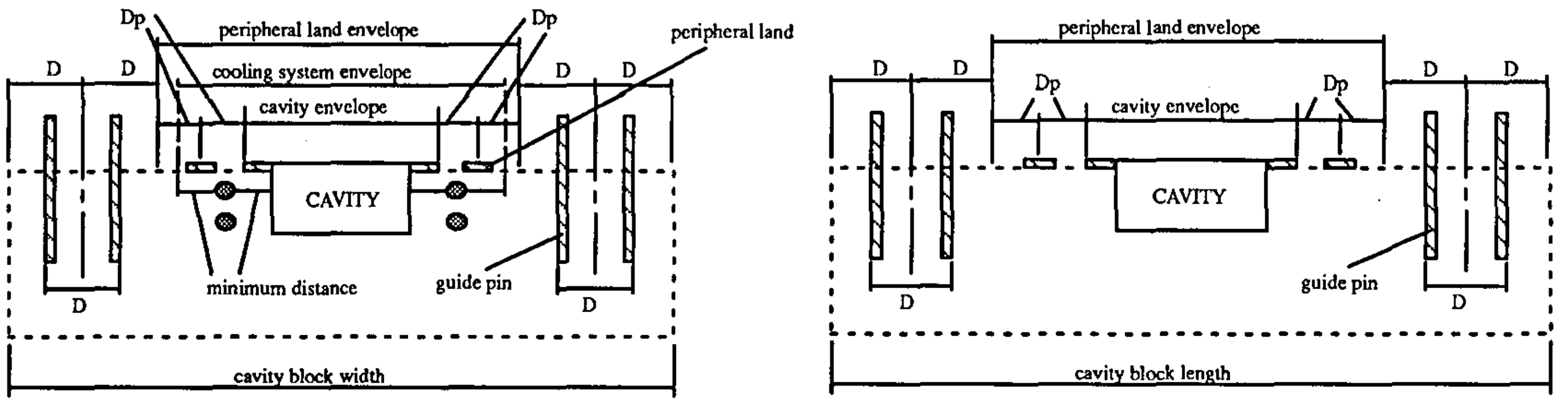
Rules can be specified with respect to the cooling system. For example a gap of 16 mm is required around a cooling tube to avoid the effects of local cooling and a similar gap is required to allow for machining, Pye (1989). This would then effect the depth of an integer cavity block. Thus for example in a shallow mould a single cooling tube would require a minimum integer cavity block depth of the cooling tube diameter plus 32 mm. Therefore during integer cavity block instantiation the depth of the cooling system plus 32 mm is calculated, and if this is greater than the total depth of the integer cavity block instantiation, the depth is increased.

The calculation of the width and length requirements of an integer cavity block instantiation is shown in Figure 7.43. To enable the accommodation of the cavity impression and the lands, their width and length can be easily identified as an envelope around the outside of a rectangular land instantiation or around the outside of peripheral land instantiations. The requirement of 16 mm clearance around a cooling tube is used to create a cooling system envelope for an integer cavity block instantiation. The guide system is not instantiated in the authors work but Pye (1989) related the required dimensions of the guide pins to the size of the mould by tabulating standard pin sizes against mould dimensions for a large number of working moulds. This data is used in this work to calculate a guide system envelope around the outside of the cooling system or feeding system envelope as shown in Figure 7.43. For- guide pin size calculation, the size of the mould is taken as the envelope around the outside of a rectangular land instantiation or around the outside of peripheral land instantiations.

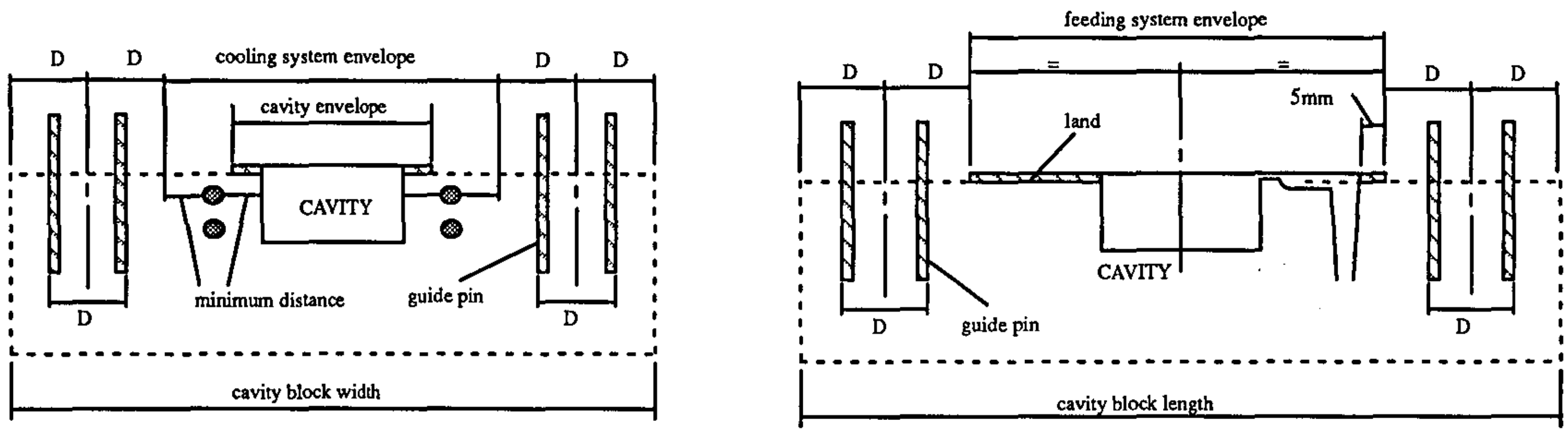
7.5.2.4. Backing plate feature instantiation.

From Figure 7.44a) it can be seen that the geometry of some feeding systems dictates that a three plate mould is required to enable ejection. Thus an *integer cavity backing plate* instantiation is required to accommodate the runner and sprue. The general problems of integer cavity backing plate instantiation are, i) the requirement for an integer cavity backing plate instantiation must be identified, ii) the depth of a backing plate must be related to the minimum strength requirements of a mould and accommodation requirements of the runner and sprue in a feeding system, iii) the width and length of a backing plate must be related to those of the integer cavity block. Using the following method it can be demonstrated that cavity impression, feeding system and cavity block data can be used at an early stage in product design as an input to decision making with respect to the dimensions of a cavity backing plate, and the extent that these can be considered in parallel to product design can be shown.

An integer cavity backing plate instantiation is required when a pin gate is used. Therefore after integer cavity mould block instantiation, if a pin gate exists in the Product model an integer cavity backing plate is instantiated. In the absence of a complex analysis capability



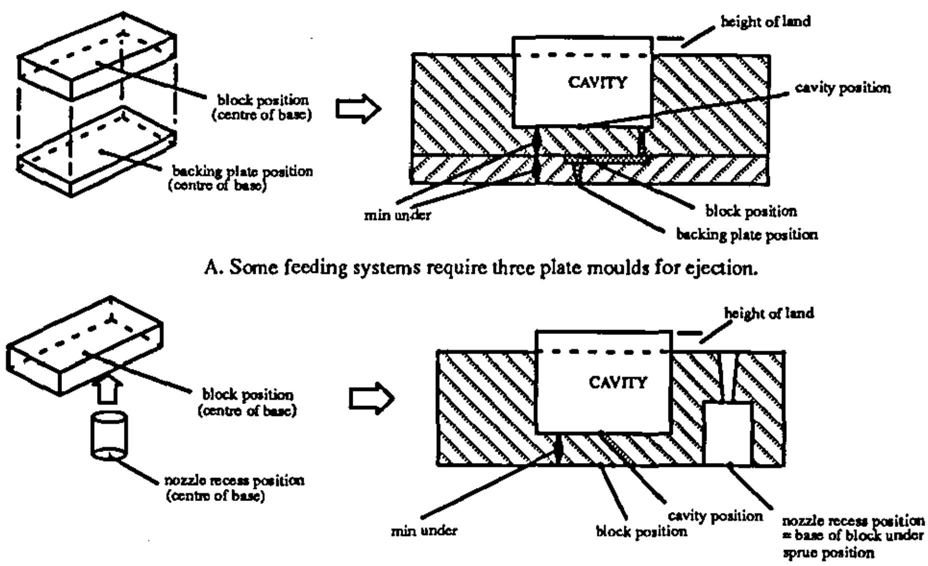
A. Block size calculation with narrow land



B. Block size calculation with wide land

= cooling tube
Dp = diameter of peripheral land

Figure 7.43. Examples of identification of integer cavity block dimensions.



A. Some feeding systems require three plate moulds for ejection.

B. Some feeding systems require a recess for connection with the machine nozzle.

Figure 7.44. Extra cavity instantiations due to feeding system requirements.

the minimum depth of an integer cavity backing plate to withstand the injection forces without distortion is assumed to be dependant upon the same minimum depth of steel calculation used for the integer cavity block. However if the depth requirement of a feeding system inside an integer cavity backing plate is greater than that for the minimum steel depth to avoid distortion, the depth of an integer cavity backing plate is increased. It can be seen from Figure 7.44 that the depth requirement of a feeding system inside an integer cavity backing plate is equal to the width of the runner plus the length of the sprue. With respect to the width and length of a backing plate, these are related to those of an integer cavity block. To avoid uneven clamping forces on the block or the plate which may cause distortion and/or flash to occur, it is assumed that the width and length of an integer cavity backing plate instantiation must match those of the integer cavity block. In any event the plates of an integer mould are usually made from a single block of steel, Pye (1989), and the backing plate has a similar requirement to accommodate the guide system.

7.5.2.5. Nozzle recess feature instantiation.

Because of a limit on the maximum diameter of the elements of a feeding system, Pye (1989), (Figure 7.44 b) a deep cavity impression may imply that a tapered main sprue cannot reach the base of a deep integer cavity mould block with sufficient diameter to correctly interface with an injection machine nozzle. In such a case an *integer cavity nozzle recess* instantiation is required. The general problems of integer cavity nozzle recess instantiation are, i) that an instantiation requirement needs to be identified, ii) from Figure 7.44 b), the depth of a nozzle recess must be related to the the dimensions of the integer cavity block and of the feeding system that must be connected to the machine nozzle, and iii) The diameter of a nozzle recess must be related to the diameter of a machine nozzle it has to accommodate. Using the following method it can be demonstrated that feeding system and cavity block data can be used at an early stage as an input to decision making with respect to the size and shape of a nozzle recess, and the extent to which this can be in parallel with the design of the product.

In the authors work, if an integer cavity block is instantiated with a depth that is greater than the maximum length of a main feeding sprue instantiation an integer cavity nozzle recess is automatically instantiated. The maximum length of a main feeding sprue is the length required to taper down from the maximum dimension (see section 7.7) to the diameter required to interface with an injection machine nozzle. The depth of a nozzle recess is the distance from the base of a sprue instantiation of maximum length to the base of an integer cavity block instantiation. In order to relate the diameter of a nozzle recess to that of a machine nozzle, an all round clearance of at least 7 mm is required to avoid heat transfer from a hot machine nozzle to the mould, Pye (1989). Therefore the diameter of an integer cavity nozzle recess instantiation is equal to the diameter of an injection nozzle plus 14 mm.

The structured methods described above provide a solution to the problems of instantiation of the cavity plate features.

7.5.3. Cavity design support strategy.

The above investigation into the interaction of the cavity and mouldability viewpoints has identified a set of precedence relationships which are reflected in the cavity design support strategy (Figure 7.45a). The strategy is i) identify the parting line, ii) instantiate main cavity geometry, ii) instantiate cavity group volumes, iii) instantiate local inserts, iv) instantiate cavity plate. With respect to cavity plate instantiation, the precedence relationships identified for evaluation of geometric relationships described in section 7.5.2 mean that the strategy has been defined as i) instantiate parting line, ii) instantiate cavity lands, iii) instantiate cavity block, iv) instantiate backing plate or nozzle recess if required.

The above solutions provide a link between the geometry of the product and that of the mould. Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.6. Cavity/core assessment – Core.

There has been no reported examination of process constraints during the build up of mould cores. Where the generation of mould cores has been addressed the whole product geometry has been the input to the process and core blocks are created using Boolean operations in a solid modeller, eg Kang et al (1990), (1992), Tseng et al (1994). Only simple shallow components have been examined eg Kang et al (1990), (1992), or components with simple geometry and hence no undercuts, eg Tseng et al (1994). None of the above work has addressed the issue of core design modification as the product geometry is developed. This work aims to identify the functionality of the support applications to enable this type of concurrent design. The requirement is for a translation process between individual feature types of the mouldability and core design viewpoints, and a strategy to support the designer in consideration of the core design as the product evolves. The problems involved and the formulated solutions are now outlined.

7.6.1. The instantiation of core impression features.

This section discusses the problems of instantiation of the features in the core impression

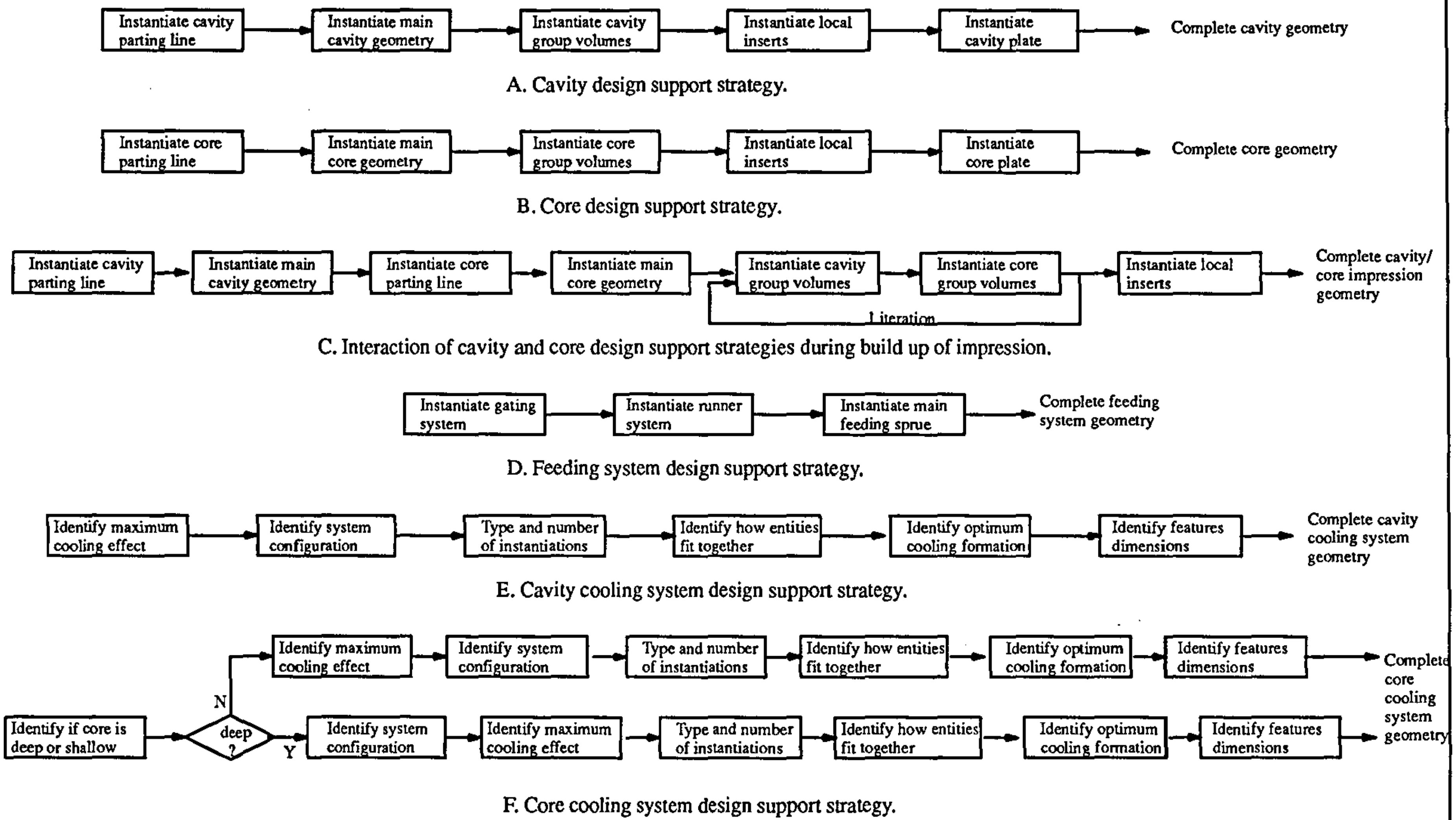


Figure 7.45. Design support strategies in the mould.

from existing product data available defined in terms of functional, mouldability and form features. A procedure which supports the build up of the (cavity and) core geometry by translation from the mouldability viewpoint has been been identified in Appendix 1 (Drawing B1). As outlined in the previous section, this involves identification of a parting line, the definition of the main cavity geometry, the definition of the main core geometry, the definition of any local inserts and finally the generation of any additional parting lines. The problems that relate to the core are discussed in this section, with the exception of the parting line which is a core plate feature and is therefore discussed in section 7.6.2.

Similar to the cavity, the initial problem for the instantiation of the core impression features defined in section 6.3.4. was to identify the main core geometry from mouldability wall features. Mouldability wall features all translate to an *integer core volume*, but one must correctly interpret the parameters of a wall instantiation for an integer core volume instantiation. A method of translating from the mouldability representation to the main core geometry follows.

As for the build up of the main cavity geometry, the wall feature instantiations in the Product model are identified in order of proximity of their base to the parting line. Working from the furthest wall feature instantiation towards the parting line, the geometry of each wall instantiation is compared with that of the next furthest in order to identify the equivalent dimensions of an integer core volume. The geometry of an integer core volume is not always the same as the inside of the equivalent wall feature instantiation, requiring i) Identifying the base of the core, ii) instantiating the corner geometry in the core, and iii) dealing with wall feature instantiations whose equivalent is an integer cavity rim in the cavity.

Identifying the base of the core: From Figure 7.46. it can be seen that if the furthest wall feature instantiation from the parting line has non–hollow rotational geometry (Figure 7.46 a)), then the top of the wall is recognised as the base of the core, and an integer core volume is instantiated of equal thickness above the top surface of the wall instantiation. The diameter

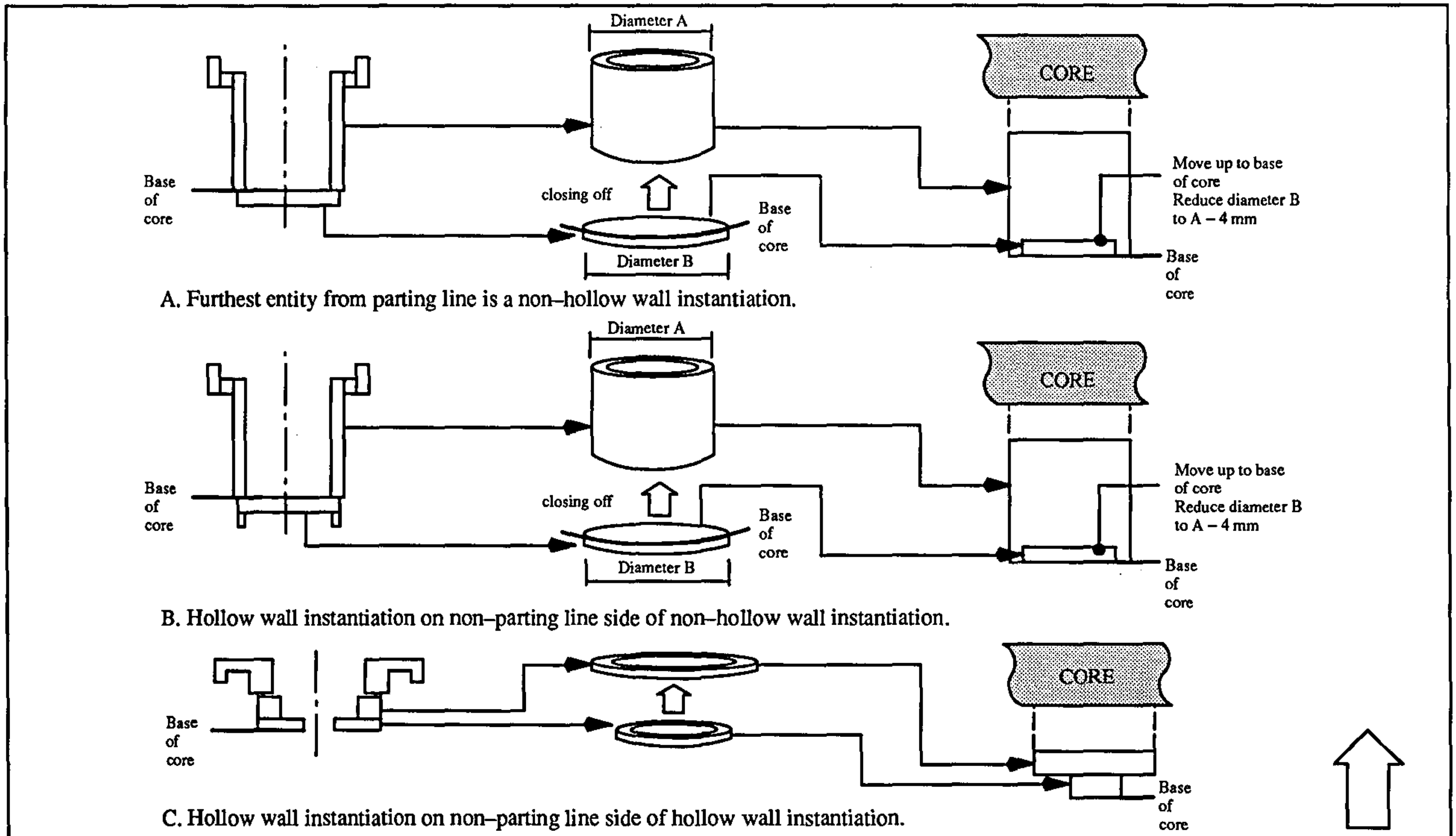


Figure 7.46. Identifying the base of the core

of the integer core volume cannot be that of the inside geometry of a non–hollow wall instantiation, and so to create the equivalent geometry of the inside of the product the diameter is related to that of the inside of the next furthest wall feature instantiation. The diameter is 4 mm less than the inside diameter of the next furthest wall instantiation to allow for the instantiation of integer core blends, which is discussed later in this section. If the furthest wall instantiation has hollow geometry as in Figure 7.46 b) and c) then the base of the core may or may not be related to this wall feature instantiation. It can be seen in Figure 7.46 b) that if the next furthest wall instantiation has non–hollow geometry there is no core equivalent for the furthest wall geometry, and the base of the core is instantiated in the same way as in Figure 7.46 a). If however, the next furthest wall instantiation is also hollow as in Figure 7.46 c), then the base of the core is identified as the base of the furthest wall instantiation. The equivalent geometry of the furthest wall instantiation is an integer core instantiation with the same height and position and with the inside diameter of the wall.

Figure 7.47 shows the formulated method of instantiation of corner geometry in the core. It can be seen from the figure that the manner in which the geometry of a corner is instantiated in the core is dependant on the spatial relationships between two wall feature instantiations. As shown in Figure 7.47 a) in a partial enclosure relationship as between walls 2 and 3, the enclosed wall instantiation translates to an integer core volume which has the same geometry as the inside of the wall instantiation, ie identical position, height and the inside diameter of the wall. However to correctly reproduce a corner in the core, the encloser wall instantiation translates to an integer core volume of the same diameter as the inside of the **enclosed** wall instantiation, otherwise a step is created in the core. The integer core volume has the same position and height as the encloser wall instantiation. Figure 7.47 b) shows that in a full enclosure relationship as between walls 3 and 4, the encloser wall translates to an integer core volume which has the diameter of the inside of the wall geometry. However the height of the integer core volume is reduced to the difference between the encloser and the enclosed wall instantiation and the position is relocated to the top of the enclosed wall instantiation.

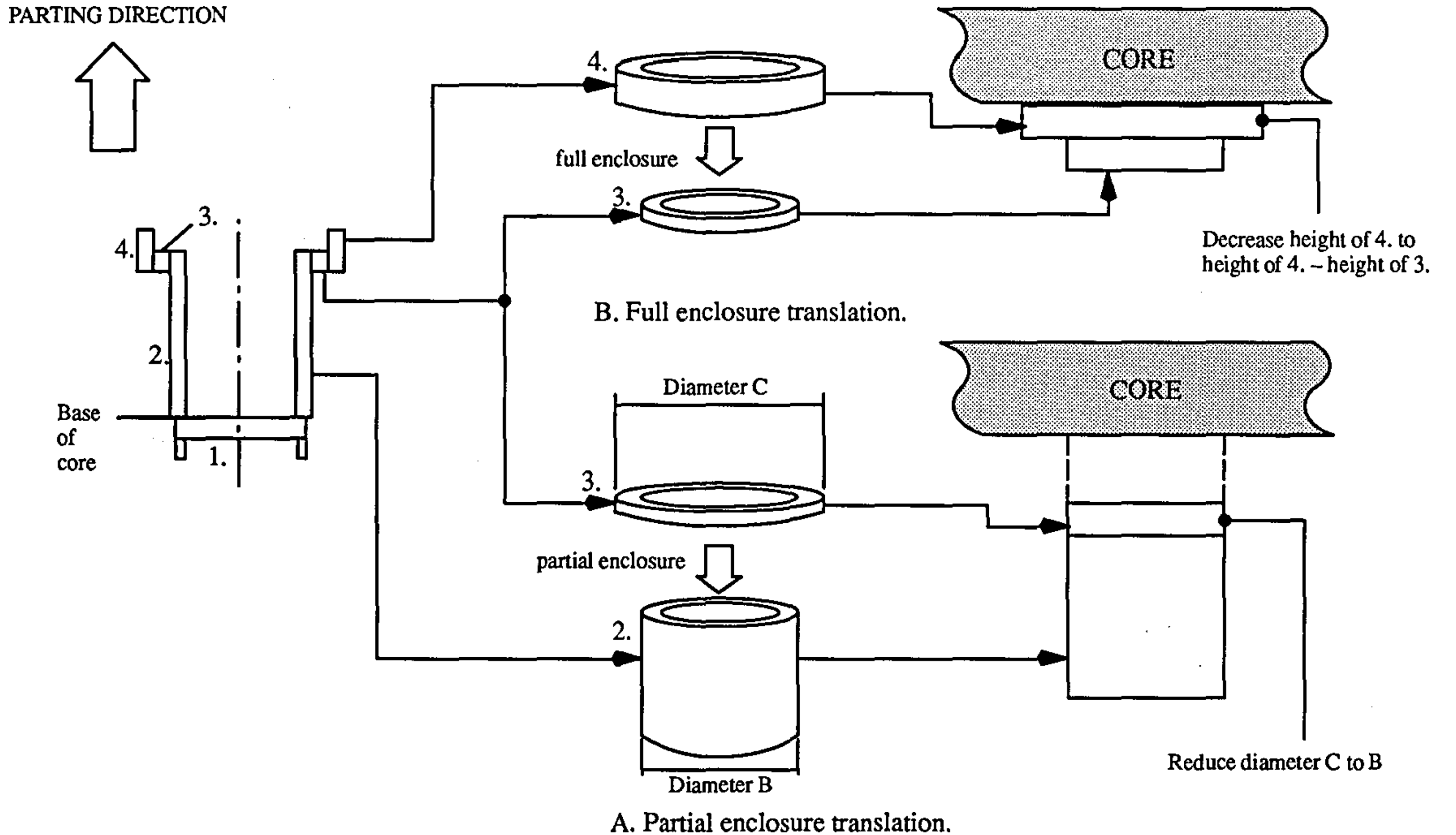


Figure 7.47. Instantiation of corner geometry in the core

The formulated method of dealing with wall feature instantiations whose equivalent is an integer cavity rim in the cavity is shown in Figure 7.48. It can be seen from the figure that the instantiation of integer core volumes is based upon the relationships between the 'rim' wall and the adjacent wall instantiations. In Figure 7.48 a) the further wall instantiation geometry has a partial enclosure relationship with the 'rim' wall instantiation, and therefore the instantiation of integer core volumes is the same as in Figure 7.47 a), ie the enclosed wall instantiation translates to an integer core volume which has the same geometry as the inside of the wall instantiation, ie identical position, height and the inside diameter of the wall. The 'rim' wall instantiation translates to an integer core volume of the same diameter as the inside of the **enclosed** wall instantiation, otherwise a step is created in the core. The integer core volume has the same position and height as the 'rim' wall instantiation. Figure 7.48 b) shows a full enclosure relationship, but unlike in Figure 7.47 b) it is the wall instantiation **nearer** the parting line that is enclosed. In this scenario the nearer wall instantiation translates to an integer core volume whose diameter is the same as that of the integer core volume equivalent of the 'rim' wall, otherwise a step is created in the core. The height and position are identical to that of the nearer wall instantiation.

A problem for the core as well as for the cavity is that a gap may occur in the main (cavity/)core geometry due to reinforcement features in a product bridging the gap between two walls. A method of considering the process constraints of the core in relations to gaps in the main geometry is required, a problem hitherto not considered the closest work being that of Hui and Tan (1994) and Mochizuki and Yuhara (1992) in the area of automatic undercut detection using complex algorithms in a solid modeller. As outlined their work has been performed on completed mould geometry with no potential links back to product design. Here the problem has been addressed by including in the core features set an *integer core group volume* which is instantiated from a group of reinforcement instantiations bridging a gap between two wall instantiations. As in the cavity, a requirement for an integer core group volume must be identified, ie that a gap exists in the main core geometry which contains reinforcement instantiations. It is also necessary to correctly identify the dimensions of an integer

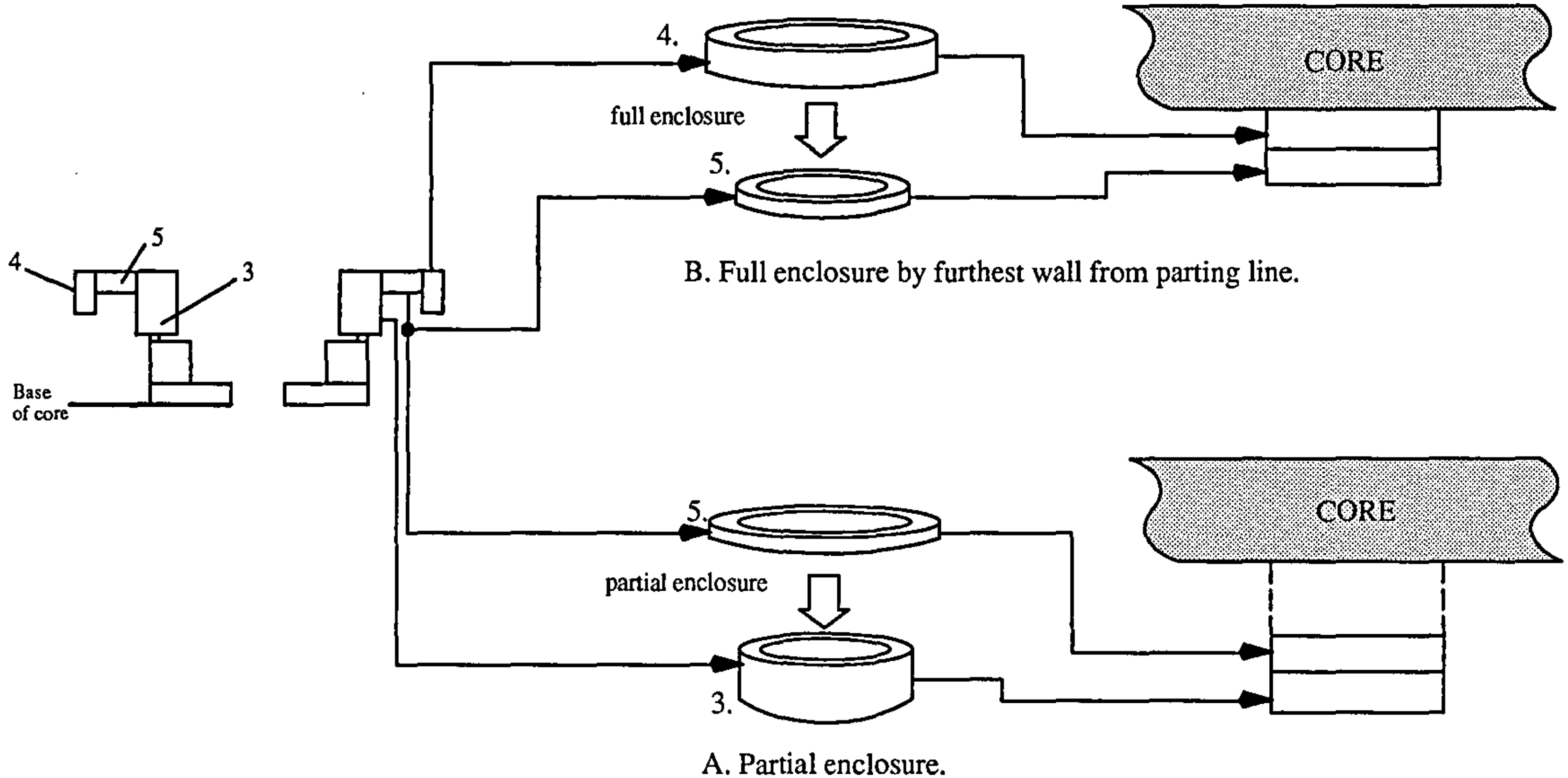


Figure 7.48. Dealing with wall instantiations whose **cavity** equivalent is an integer cavity rim.

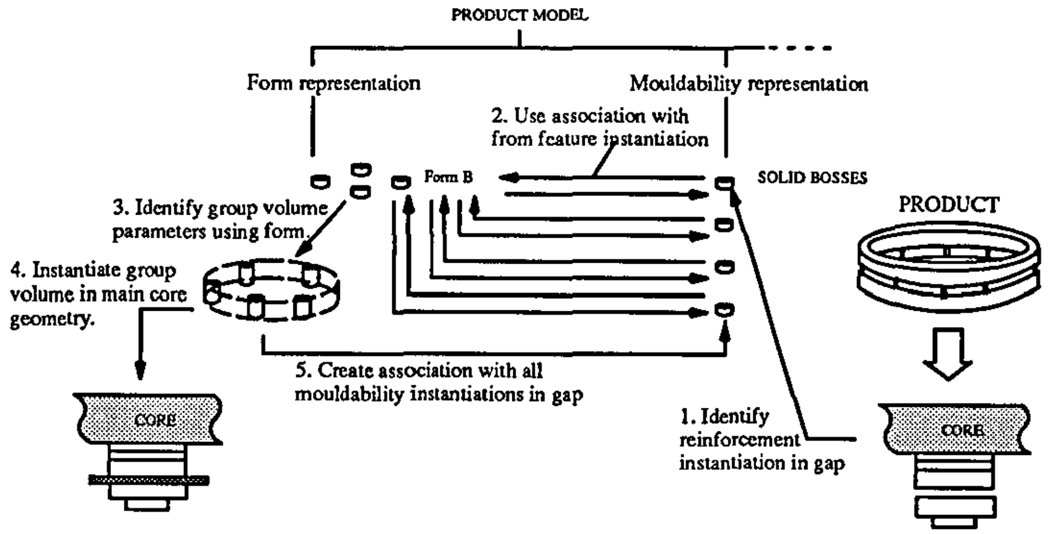


Figure 7.49. Instantiation of integer core group volume from ribs and solid bosses in a gap.

core group volume from a group of reinforcement instantiations bridging a gap in the core.

As in the cavity, a gap can be identified simply by the relative position and dimensions of geometry. Once a gap is identified, reinforcement instantiations in a gap are identified in the same way as in the build up of the cavity. Having established that an integer core group volume must be instantiated, the dimensions of the group volume must be obtained from the group of reinforcement instantiations in the gap. Similar to the build up of cavity geometry, dimensions are taken from the form feature instantiation that is associated with the reinforcement instantiations in a gap, as shown in Figure 7.49. Selecting one mouldability instantiation, the association with a form feature instantiation is used to obtain the identity of the other mouldability instancations in the gap, via their associations with the same form feature instantiation. As shown in the figure, the parameters of the integer core group volume are obtained from the form feature instantiation in the same way as for the instantiation of the cavity group volume.

The integer core group volume enables consideration of process constraints of gaps in the main core geometry in a similar manner as the integer cavity group volume does for gaps in the cavity: Adjustments of integer core group volume dimensions in response to feedback advice, eg for overhangs in the core, are used to adjust the position of the associated mouldability instantiations (via the form) so that the mouldability instantiations' relationship with the dimensions of the integer core group volume is maintained, as shown in Figure 7.29. The above structured methods for wall translation and dealing with gaps in the core solve the problems of instantiation of the main geometry in the core impression.

Having instantiated the main core geometry, we must next instantiate the local inserts in the core. Local inserts have previously been considered only as a part of the overall core geometry in a solid model, eg Kang et al (1990), (1992), Tseng et al (1994) not individually. As in the cavity, the authors work identifies local inserts as individual features, namely *integer core boss*, *integer core hole*, *integer core slot* and *integer core rim* feature types. Local inserts in the core are translated from reinforcements and holes, and similar to the cavity, reinforcement features that are associated with an integer core group volume should not be translated to local inserts because their process constraints in the core have already been considered during instantiation of the group volume. A method of identifying whether a reinforcement feature translates to a local insert in the cavity or in the core has been described in section 7.5.1, is shown in Figure 7.31. If a reinforcement or hole translates to a local insert in the core, then there is a requirement to identify the type of core feature to which it translates, ie a core boss, hole, slot or rim. A solution to this problem has also been described in section 7.5.1, and the permutations of equivalent core entities to mouldability reinforcements and holes are shown in Figure 7.31. A final problem is the interpretation of the parameters of the reinforcement or hole instantiations from which the local inserts are translated.

As shown in Figure 7.49, reinforcement instantiations inside a group volume can be identified in a similar way as during the build up of cavity geometry, by the association in the Product model between an integer core group volume and those reinforcements from which it was instantiated. As illustrated in Figure 7.50, considering the remaining problem of identifying the parameters of local inserts, for the translation of ribs, solid bosses, hollow bosses and blind holes the dimensions are taken unchanged for the corresponding integer core slots,

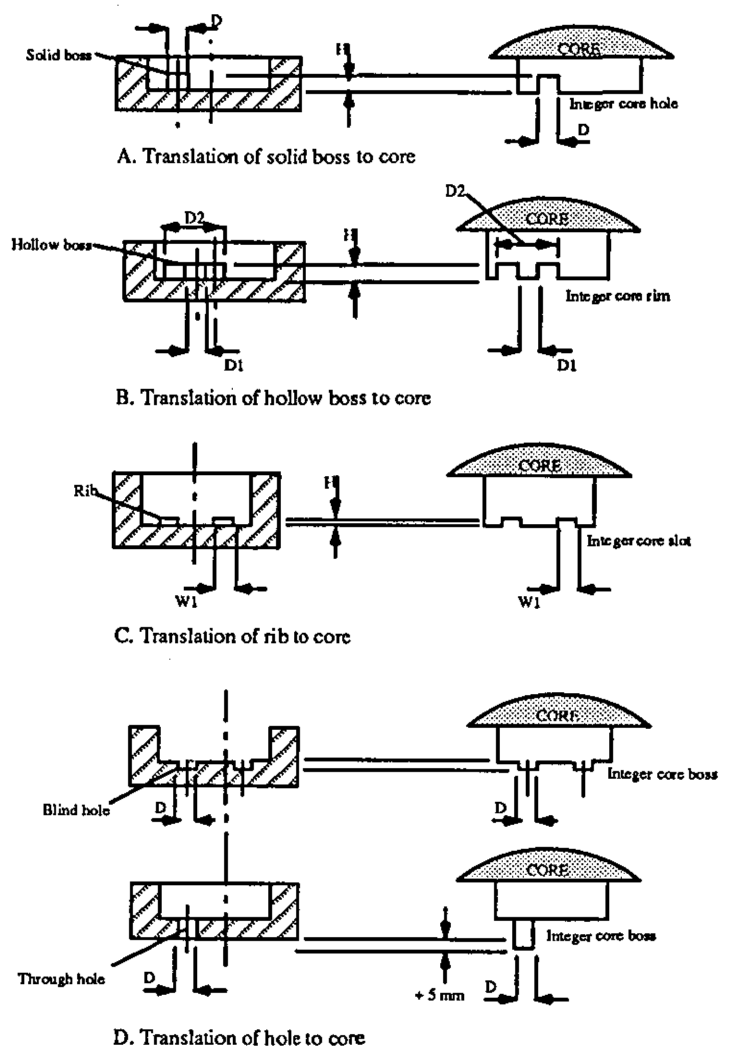


Figure 7.50. Reinforcements and holes parameters are unchanged in core feature instantiations.

holes, rims and bosses. However for the translation of a through hole the length of an integer core boss is increased by 5 mm to give some location in the cavity. The above structured solution solves the problems of instantiation of local inserts in the core.

A further problem that relates to both the main core geometry and the local inserts is the translation of secondary mouldability feature instantiations to the core viewpoint. The requirement for an *integer core blend* in the core must first be identified. As a blend is between two primary mouldability instantiations, one must establish to which other core geometry the new integer core blend instantiation should be attached. If the mouldability blend is of the corner blend type, the problem is to identify whether the larger radius or the smaller radius should be applied to the core. An additional problem (Figure 7.51) is how to apply an integer

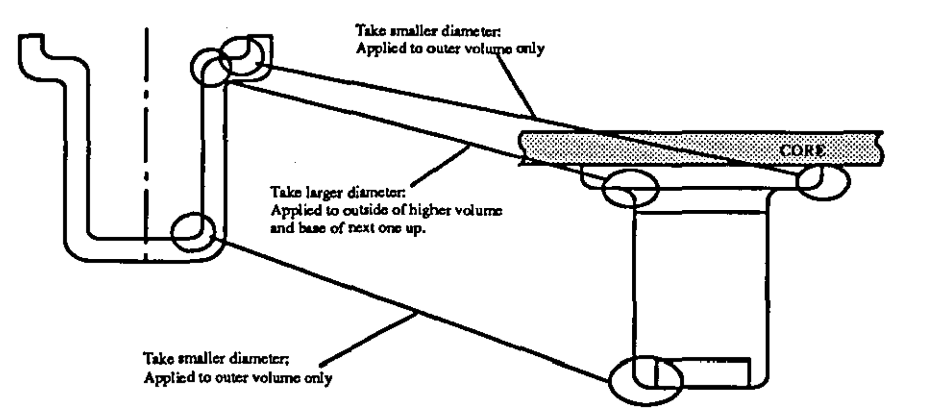


Figure 7.51. Translation problems: Applying blends correctly to core geometry

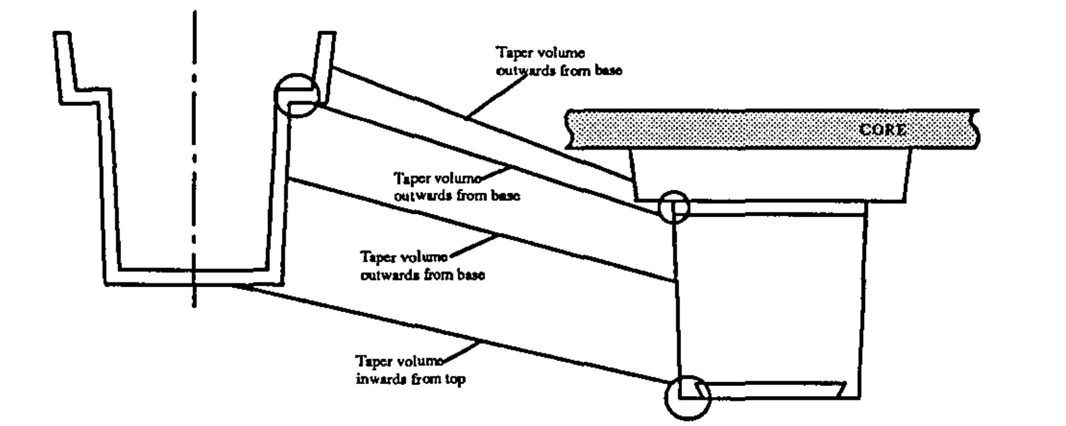


Figure 7.52. Translation problems: Applying tapers correctly to core geometry.

core blend to the core geometry to correctly replicate the inside product geometry. It can be seen that the manner of application of an integer core blend to core geometry is not the same as the application of a mouldability blend to product geometry.

Mouldability taper features translate to *integer core tapers*. The instantiation problems of an integer core taper are similar to those of an integer core blend: A requirement for an integer core taper in the core must be identified. The second problem is how to apply an integer core taper to the core geometry to correctly replicate the inside product geometry, as illustrated in Figure 7.52.

To identify the requirement for an integer core taper and/or blend in the core the existence of a taper and/or a blend on a translated primary mouldability feature instantiation must be detected. This can be achieved using the Product model structure in Figures 7.22 and 7.53.

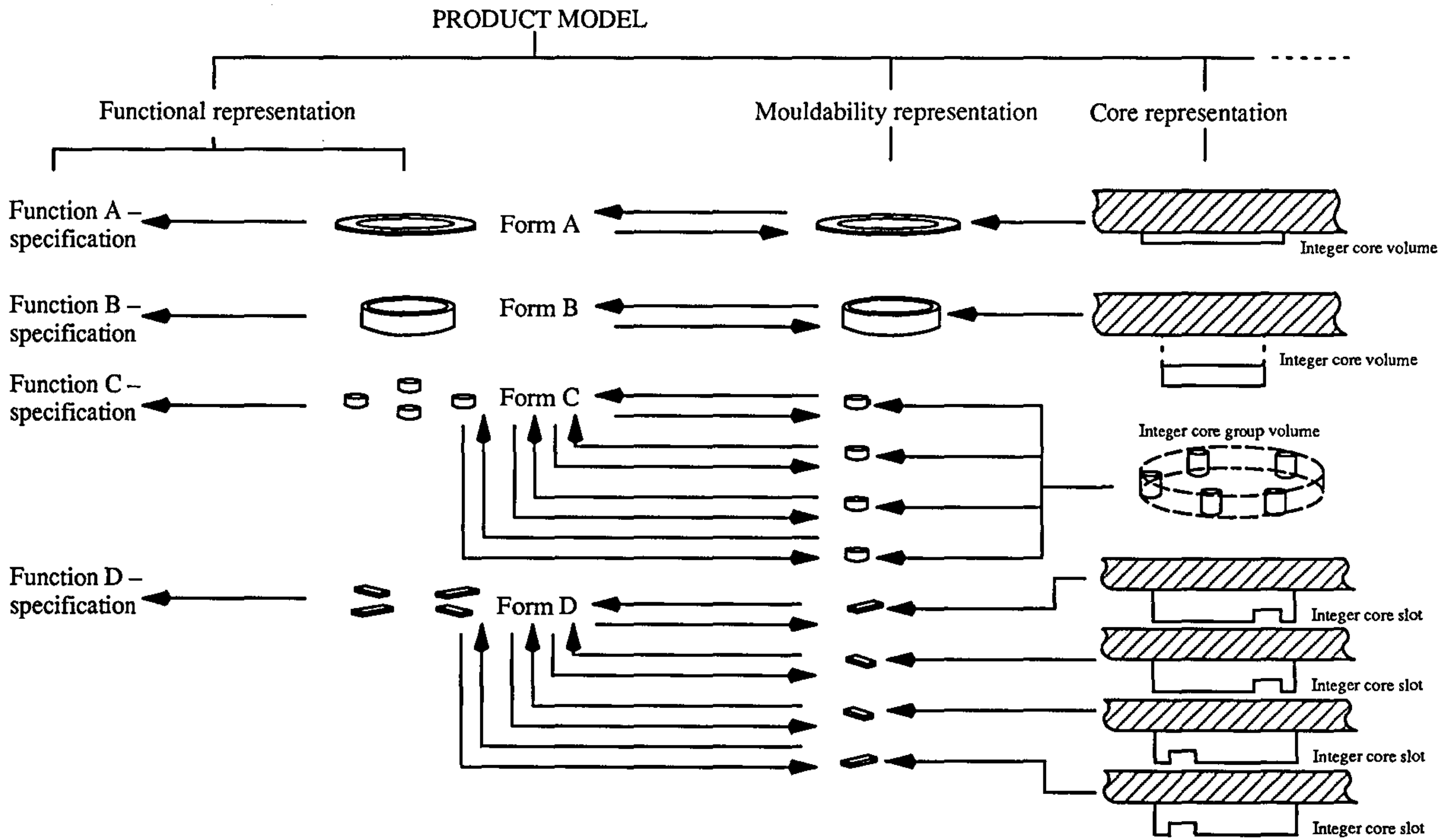


Figure 7.53. Example of relations built up in Product model including core representation

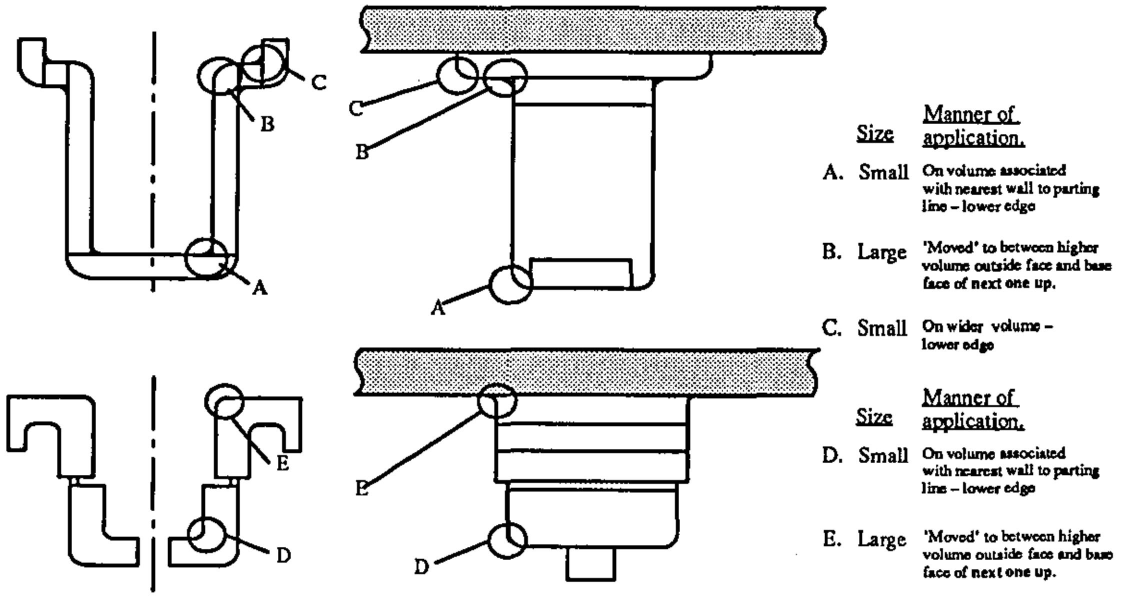


Figure 7.54. Identifying blend radii and manner of application between core entities

The associations in Figure 7.53 identify the equivalent primary mouldability instantiation of a core feature instantiation, and therefore using the associations between primary and secondary mouldability feature instantiations shown in Figure 7.22, the existence of tapers and blends on the equivalent primary mouldability instantiation can be identified.

If a blend exists on a primary mouldability instantiation, the adjacent geometry in the core to which an integer core blend instantiation should be attached needs identification, also done by using the associations in the Product model shown in Figures 7.22 and 7.53. Having identified the equivalent primary mouldability instantiation and the existing blend, the associations in Figure 7.22 allow identification of the other primary mouldability instantiation to which the blend instantiation is attached. The core equivalent of the former is then identified using the associations in Figure 7.53.

What if a blend is of the corner type? The problem is now to identify whether the smaller or larger radius is to be used in the cavity or the core, ie which is to be used in the integer core blend instantiation. Figure 7.54 exemplifies a method of radius identification for integer core blend instantiations. Using the associations in Figure 7.53 and 7.22 the equivalent wall feature instantiation for each integer core volume instantiation is known. Therefore, the radius from a corner blend to be used in an integer core blend instantiation is identified by comparing the dimensions and positions of the equivalent blended wall feature instantiations in the Product model, working from the furthest wall instantiation from the parting line to the nearest. Note that a wall feature instantiation that translates to an integer **cavity** rim (no walls translate to rims in the core) is considered as the nearest to the parting line, even though an enclosed wall instantiation will be nearer. Thus the dimensions of those wall instantiations that translate to an integer **cavity** volume are compared as far as the parting line, and the last of these wall instantiations is compared with the one that translates to an integer **cavity** rim. The comparison in the product is as follows: If the further of two wall instantiations is 'closing off', as in Figure 7.54 a) and d), the smaller of the radii in the corner blend is instantiated in the core. If two wall feature instantiations are the same distance from the parting line, as in Figure 7.54 c), the smaller of the radii in the corner blend is instantiated in the core. In any other comparison the larger of the radii in the corner blend is instantiated in the core.

The final problem of the translation of secondary mouldability features to the core design viewpoint is the application of integer core tapers and integer core blends in a manner that replicates the outside geometry of the product. Figure 7.54 offers a solution for the application of integer core blends: Looking at wall to wall relationships in the product; in 'closing off' scenarios 7.54 a) and d) an integer core blend is applied to the lower outside edge of the integer core volume instantiation that is associated with the nearer wall instantiation to the parting line. In the same distance scenario of 7.54 c) an integer core blend is applied to the lower outside edge of the wider integer core volume instantiation. In the partial enclosure scenarios in Figure 7.54 b) and e) an integer core blend is not applied to the lower integer core volume at all, but the integer core blend is **'moved'** to be between the higher of the two integer core volumes and the **next one up**. In the case of the scenario in Figure 7.54 e) this means that the integer core blend is between an integer core volume and the integer core inner land. Note that no integer core blend is applied between two integer core volumes if one has an equivalent wall instantiation that translates to an integer **cavity** rim.

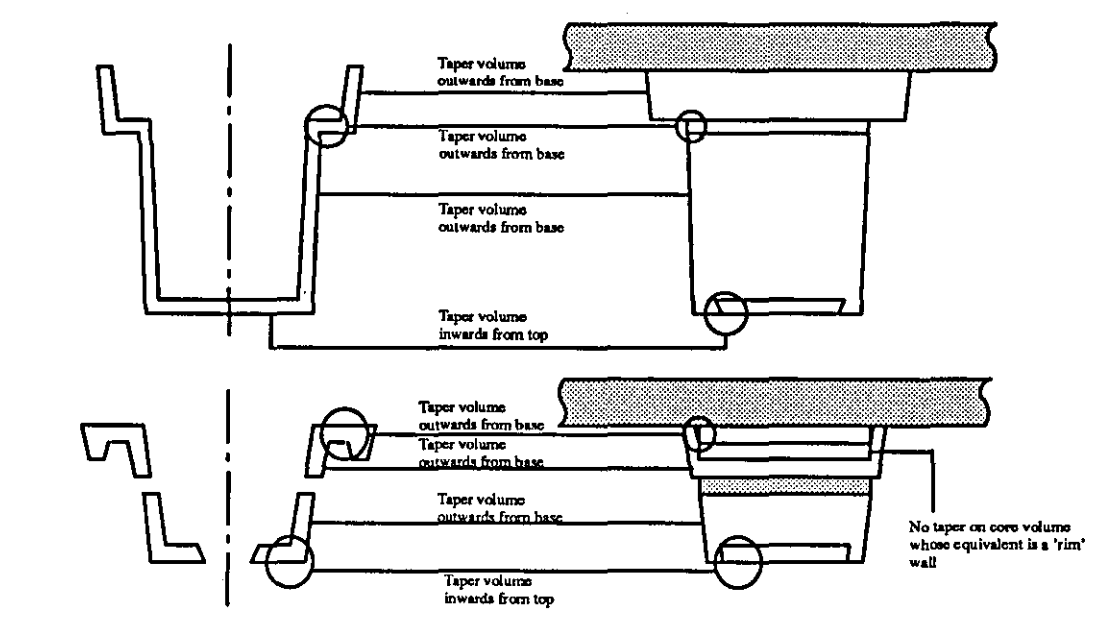


Figure 7.55. Applying tapers correctly to core geometry using rules that apply to corresponding mouldability geometry.

From Figure 7.55, the author has shown that the same rules apply for the application of integer core tapers in the core as for mouldability tapers on the product. However no taper is applied to an integer core volume whose equivalent wall feature instantiation translates to an integer **cavity** rim. The above provides a structured solution to the problems of the instantiation of integer core blends and integer core tapers.

The above structured methods provide a solution to the problems of translation from the mouldability features set to an alternative set of core impression features which maintains the links between the structure of the geometry of the two viewpoints. Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.6.2. The instantiation of core plate features.

A set of geometric relationships can be defined between the core impression and the plate, and between the cavity and the core plate, such that as the impression is instantiated, so feedback can be provided on the consequences of product geometry for the core plate design as the product geometry evolves. Therefore the design of the core plate can be modified as the product is developed by the designer. For core plate feature instantiation, these geometric relationships must be evaluated, some of which require an additional input of the geometry of mould system elements such as the feeding or cooling system, and/or the geometry of other core plate feature instantiations. These prerequisite data requirements and the geometric relationships can be captured in relation to each core plate feature type. This captured data can be used to support a strategy for instantiation of the core plate features. This places a requirement on the core design strategy as described in section 7.6.3. The problems of the above approach for instantiation of each core plate feature is considered in the remainder of this section, and the formulated solutions are described.

7.6.2.1. Core parting line feature instantiation.

The first problem that must be addressed in the translation between mouldability and the core is the instantiation of the *integer core parting line*. A knowledge of the parting line position is a prerequisite to the instantiation of the core impression because process constraints such as overlaps in the core must be considered with respect to the position of the parting line. Furthermore, the parting line position is required to identify the geometry of some feature types in the core during translation. The problem of integer core parting line instantiation is to identify the position. The position of the parting line on the product is identified during analysis of the cavity design viewpoint, and therefore the integer core parting line position can simply be taken from that of a horizontal integer cavity parting line instantiation.

7.6.2.2. Core land features instantiation.

Having instantiated the integer core parting line and the core impression features, consider next the instantiation of the remaining features in the core plate. As in the cavity, integer core inner lands must be instantiated as a prerequisite to instantiation of the core block, since core lands geometry is an input into the core block instantiation process. To match the cavity, the

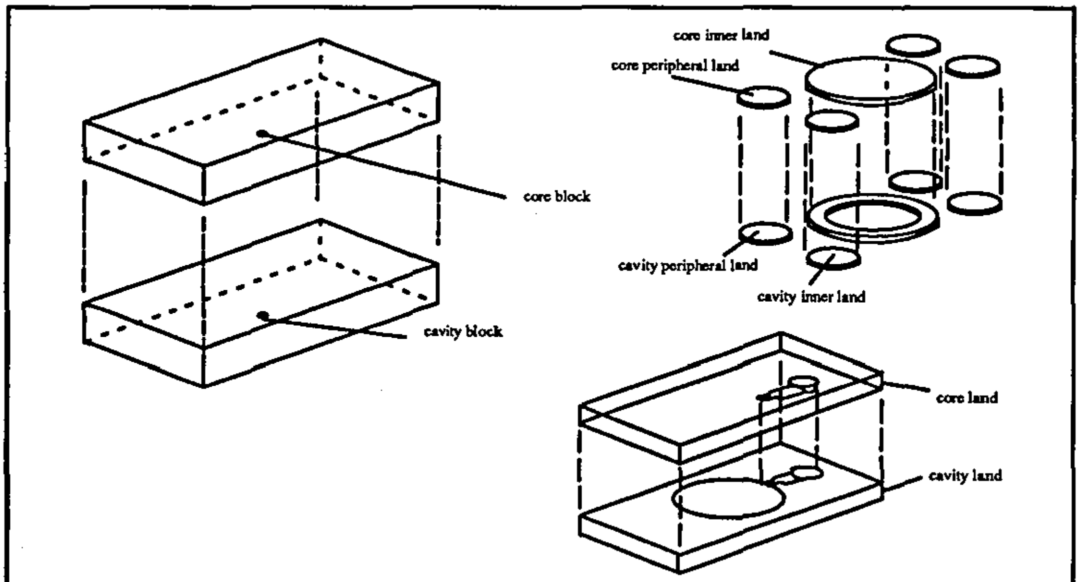


Figure 7.56. Relationship between cavity and core elements other than the impression.

assumption here has been made that a core inner land can be of two types, an *integer core circular land* or an *integer core rectangular land.* The general problems of an integer core inner land instantiation are similar to those for an integer cavity inner land ; i) the inner land must enclose the core impression, and to accomplish this enclosure, the minimum size and shape of the land is dependant upon the type and geometry of the feeding system as well as the width of the core impression, ii) the size and shape of an integer core inner land instantiation must also take account of venting considerations, which are related to the type and geometry of the feeding system. To demonstrate that core impression, parting line, and cavity plate data can be used at at early stage in product design as an input to decision making with respect to the size and shape of the lands, and the extent to which this can be in parallel with the design of the product, the following method has been used.

It can be seen from Figure 7.56 that the mating and sealing requirements between lands in the cavity and core means that lands instantiated in the core must match those instantiated in the cavity. Therefore if an integer cavity circular land is instantiated in the cavity, an integer core circular land is instantiated in the core and if an integer cavity rectangular land is instantiated in the cavity, an integer core rectangular land is instantiated in the core. It can also be seen that the geometrical relationship that exists between lands in the cavity and core identifies most dimensions of the integer core inner land. The enclosure requirements of the integer core inner land with respect to the impression and feeding system geometry have already been considered during instantiation of the integer **cavity** land. Only the depth remains to be identified, and as with the cavity lands, this is assumed to be the minimum requirement identified by Pye (1989) of 2.4 mm.

If a narrow integer core inner land is instantiated (a circular land in the current context), the surface area of the inner land is likely to be insufficient to withstand the injection forces and deformation and flash formation may occur, Pye (1989). Therefore *integer core peripheral lands* are required to increase the surface area. As for the integer cavity peripheral lands the problem of integer core peripheral land instantiation is to identify the numbers, shape and spacing of the peripheral land instantiations and the surface area of each.

It can be seen from Figure 7.56 that the shape and numbers of integer core peripheral land instantiations has already been identified during instantiation of the integer **cavity** peripheral lands. It can also be seen that the geometrical relationship that exists between lands in the cavity and core identifies the surface area and spacing of the integer core peripheral lands.

7.6.2.3. Core block feature instantiation.

Having instantiated the integer core lands the last of the core plate features to instantiate is an *integer core block*. The general problems of integer core block instantiation are i) to provide a depth of block behind the impression to avoid distortion of the block due to injection forces, ii) the depth of an integer core block instantiation must take account of the requirement to accommodate mould system elements such as the cooling system and possibly some of the feeding system, the width and length must be sufficient to accommodate the width of the core impression and lands, the cooling system, the guide system, and ejection system. Using the following method it can be demonstrated that core impression and lands data, cavity plate data, and feeding and cooling system data can be used at an early stage in product design as an input to decision making with respect to the dimensions of a core block, and the extent to which these can be considered in parallel can be shown.

As described in section 7.5.2.3, tubes in a cooling system require a clearance of 16 mm in order to avoid localised cooling effects. Therefore the depth of an integer core mould block instantiation is the depth of the cooling system in the core block (some cooling tubes may be in the centre of the core impression), plus 32 mm. Thus for example for a single layer of cooling tubes across the core block, the block depth dimension is the diameter of the tubes plus 32 mm. With respect to the feeding system, as shown in Figure 7.40, if the impression is side gated half of the runner system is in the cavity plate and half in the core. Furthermore, to aid ejection of the feeding system a sprue puller is located in the core plate (see section 7.7). A 16 mm clearance between the cooling system and the feeding system must be maintained in order to avoid premature solidification of the sprue and runner and the possible consequences of inadequate filling of the impression. Any adjustments in the position of the cooling tubes to accommodate this require a corresponding increase in the depth of the core block.

It can be seen from Figure 7.56 that the dimensions of the integer cavity block can be used to identify the width and length of an integer core block instantiation. As shown in Figure 7.43 the dimensions of an integer cavity block are determined by the need to accommodate the guide system, which has the largest envelope of all the mould system elements. The guide system in the integer core block must match that of the cavity block to ensure mating of the two halves and therefore the two blocks can be assumed to have the same size.

The structured methods described above provide a solution to the problems of instantiation of the core plate features.

7.6.3. Core design support strategy.

The above investigation into the interaction of the core and mouldability viewpoints has identified a set of precedence relationships which are reflected in the core design support strategy (Figure 7.45b). The core design strategy is i) instantiate main core geometry, ii) instantiate core group volumes, iii) instantiate local inserts, iv) instantiate core plate. It has been identified that the strategies of the cavity and core applications for instantiating the impression are interrelated and a set sequence of interactions can be identified (Figure 7.45c). With respect to core plate instantiation, the precedence relationships identified for evaluation of geometrical relationships described in section 7.6.2 mean that the strategy has been defined as i) instantiate core parting line, ii) instantiate core lands, iii) instantiate core block.

The above solutions provide a link between the geometry of the product and that of the mould. Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.7. Mould system elements assessment – Feeding system.

For true concurrency a designer must be aware of the choices that exist in terms of the configuration and geometry of the feeding system as the geometry of the product evolves. However all previous work has considered the design of the feeding system retrospectively to the design of the product. Much of the previous work has focussed on the automation of feeding system design and has considered some kind of optimisation algorithm, eg Cinquegrana (1990)a (1990)b, Rho et al (1990). The work of Ong et al (1995) considered the type of gating system in relation to the product geometry in a solid model, taking account of possible parting lines. However no gating system or runner system geometry was produced. Al–Ashaab (1994) captured the elements of a feeding system in a Manufacturing model and provided feedback on the appropriate type and parameters of a gating and runner system in a mould. However no product or mould geometry was considered and a question and answer routine was required to identify the overall shape of a product, eg tubular. All of the above work has considered the feeding system after completion of the product design.

In this work a strategy is required to support the designer in consideration of the feeding system design during the build up of product geometry. To provide the designer with choices with respect to the configuration and geometry of a feeding system at each juncture in the development of a product a translation process is required between the cavity viewpoint and individual feature types of the feeding system. The problems involved and the formulated solutions are now described.

7.7.1. Problems of instantiation of feeding system features.

This section discusses the problems of instantiating the features in the feeding system, ie ring gate, tab gate, sprue gate, sub-surface gate, diaphragm gate, rectangular edge gate, overlap gate, fan gate, pin gate, film gate, trapezoidal runner, modified trapezoidal runner, circular runner, hexagonal runner and main feeding sprue. A prerequisite must be the instantiation of a gating system, since all other entities in the feeding system are joined to the cavity via the gate. Therefore the geometric relationship with the cavity/core of other entities instantiated in a feeding system, and thus their dimensions, are dependant upon the position and dimensions of the gating system as well as those of the cavity/core.

The first problem in instantiating one of the gating systems included above is to identify where it should be positioned. It has been recognised that each gating system has its own geometric relationship with the cavity impression in order that each type shall have the correct feeding position. For example a rectangular edge gate must be on the parting line as well as on the edge of the cavity, whereas a sprue gate must be on the base of the cavity, ideally in the centre. The required type of gating system instantiation has to be identified and then allocated the correct position in relation to the cavity impression. Previous work has considered the gate position solely with respect to the geometry of a product, eg Ong et al (1995), Irani et al (1989). The final problem is to identify the dimensions of a gating system instantiation.

Next it is necessary to identify which feeding system feature types must be instantiated to build up the remainder of a feeding system. This is dependant upon the type of gating system. For example, a rectangular edge gate requires a circular runner and a main feeding sprue, a diaphragm gate requires only a main feeding sprue and a sprue gate requires no other elements in the feeding system. If additional feature types are needed they must then be instantiated.

The available runner systems are *trapezoidal runner, modified trapezoidal runner, circular runner* and *hexagonal runner* feature types. If a runner is needed the next steps are to identify the geometrical relationship between a runner instantiation and a gating system and decide upon appropriate dimensions. In general a runner can be said to be 'on the end of' a gating system, but the meaning of these words changes depending upon the type of gating system and the type of runner feature.

Following this stage a *main feeding sprue* may have to be instanciated. The geometrical relationship between a main feeding sprue instantiation and a runner, or a gating system if a diaphragm gate has been used needs to be determined. Again a main feeding sprue can be said to be 'at the end of' a runner system. However the meaning of these words varies depending upon the type of runner or gating system instantiation. Finally the main sprue dimensions need to be established.

7.7.2. A structured methodology for feeding system feature instantiation.

Solutions to the problems of feeding system instantiation are now described.

<u>Instantiation of a gating system</u>. It is first necessary to identify the requirement for a gating system instantiation. To do this the existence of a mouldability gate instantiation on the primary mouldability geometry must be detected. This can be achieved using the Product model structure in Figure 7.22. The associations between the primary and secondary mouldability instantiations in the Product model mean that by checking all wall instantiations on a particu-

lar product the existence of a gate feature instantiation on any wall can be identified.

The type of gating system to instantiate and its position in relation to the the cavity. Previous work has considered the gate position solely with respect to the geometry of a product, eg Ong et al (1995), Irani et al (1989). Subsequently Ong et al (1995) identified the possible gating system types based on the gate position on the product and possible parting lines. The work in this thesis has identified that the type of gating system influences the configuration of the cavity and core plate (sections 7.5. and 7.6) so that the choice of gating system must be related to the product geometry in order to then provide the most appropriate configuration of cavity and core for the given product geometry. The position of the gating system in relation to the cavity/core can then be considered by the designer in relation to the specific process constraints of a given type of gating system during the feeding system design stage. The possibly significant consequences of the position of a gating system in relation to the cavity/core geometry has to date been ignored. To take an example, a rectangular edge gate away from the parting line prevents ejection of the feeding system and renders a mould non–manufacturable.

Because the appropriate type of gating system must be considered with respect to the geometry of the product this can be considered during instantiation of a mouldability gate feature, as described in section 7.4.4. Thus the required type of gating system can be identified from the mouldability gate feature instantiation identified in the Product model.

The appropriate position of a gating system in relation to the the cavity is dependant upon the gate system type and can be identified as follows. Consider Figure 7.57. This figure shows where the position of each gating system feature type is measured and where the position should be in relation to the cavity/core geometry. A gating system inherits the position of the associated mouldability feature instantiation, and this position can be evaluated with respect to process constraints data in the Manufacturing model associated with each gating system type.

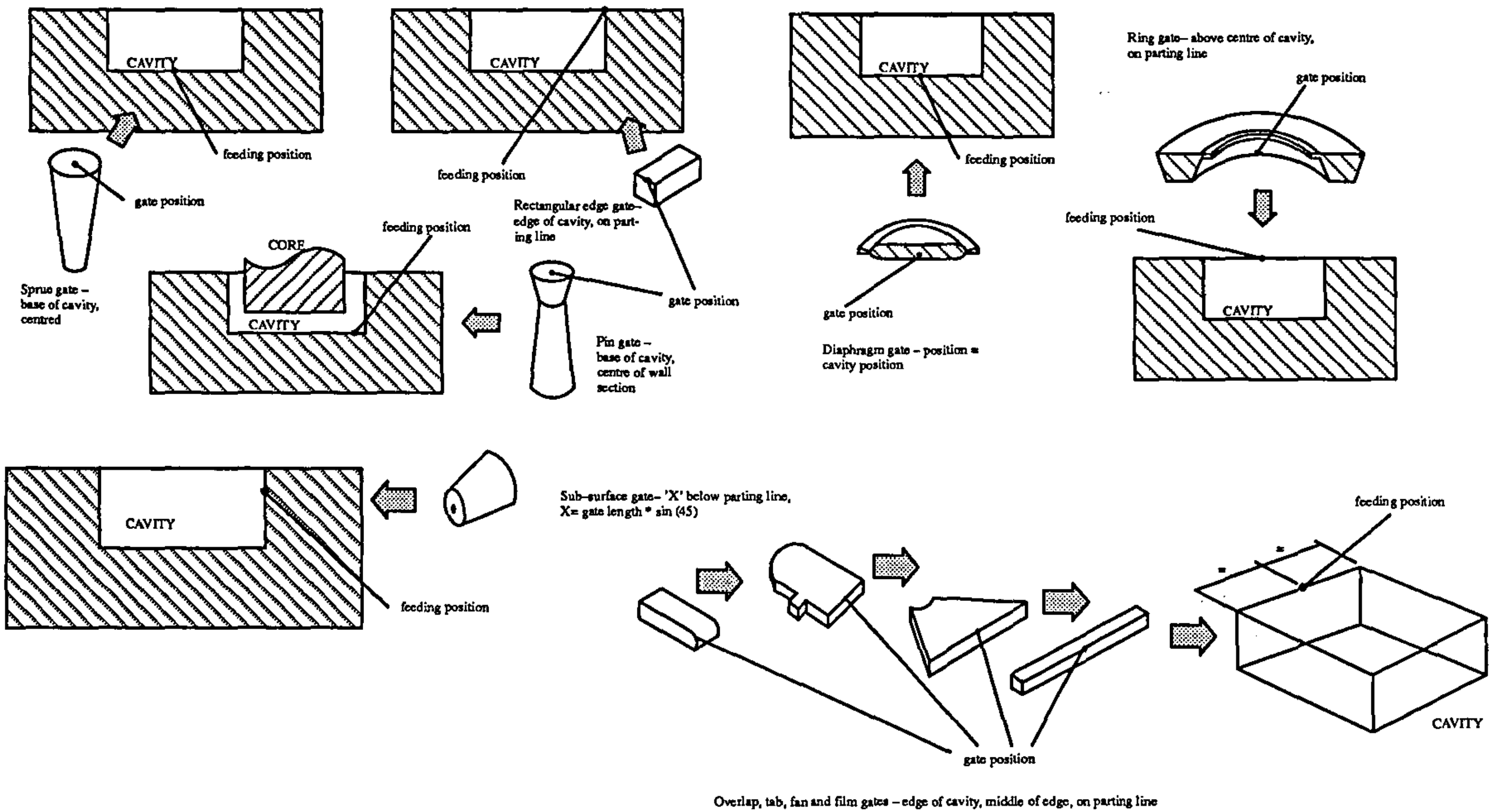
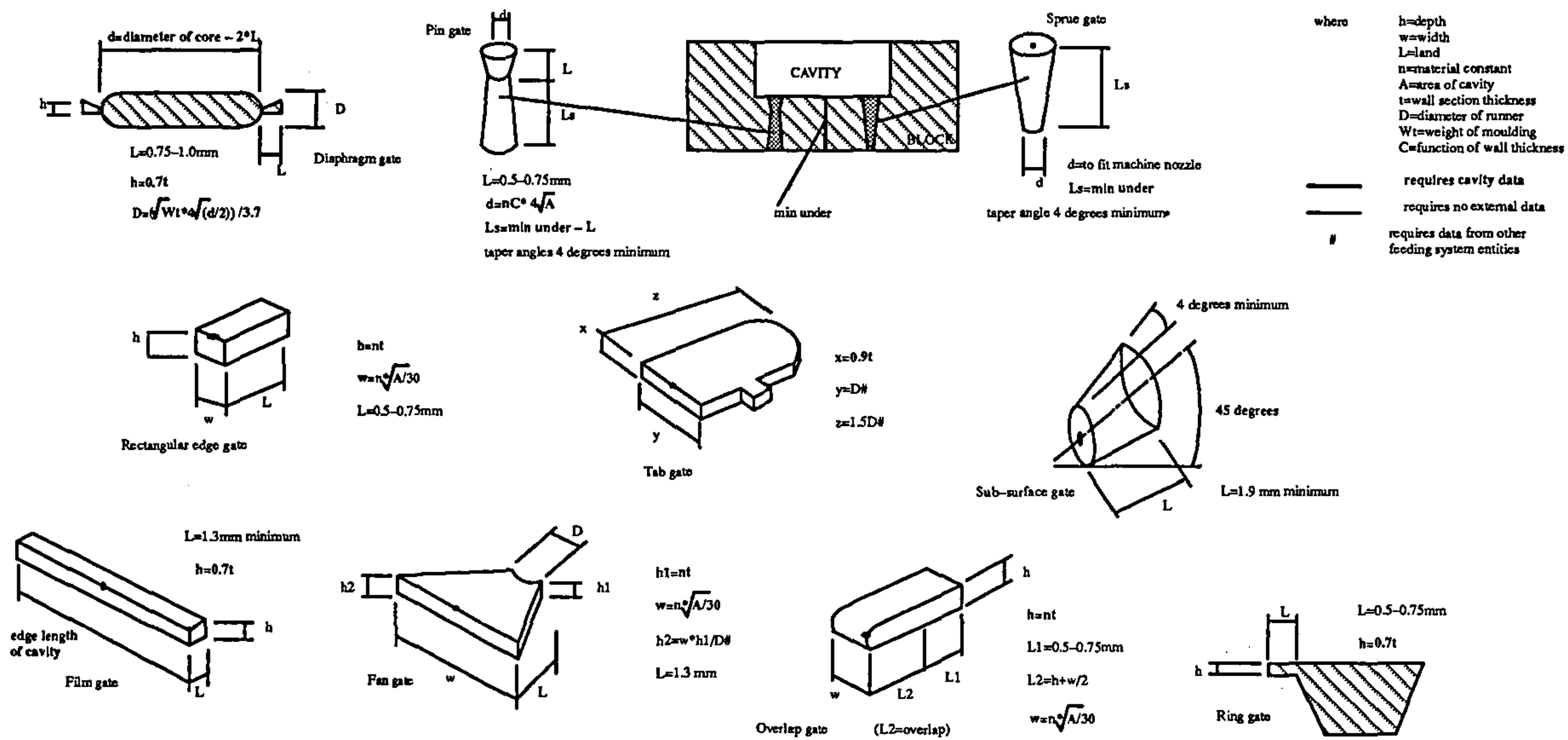


Figure 7.57. Appropriate gating position is dependant upon gate type

Finally the dimensions need to be determined. The rules (see Figure 7.58) for identifying the dimensions of most gating systems in relation to the geometry of the cavity/core can be identified from the literature, eg Pye (1989), Dym (1987). These rules can be captured in relation to each gating system feature type. Al–Ashaab (1994) and Ong et al (1995) also captured process constraints on dimensions in relation to gating system types but no mould geometry was available in their work and therefore the dimensions were considered with respect to product geometry, and no gating system geometry was created.

Here, where the geometry of the cavity/core is available, it can be seen that to identify the dimensions of a tab gate and a fan gate during instantiation requires the diameter of the runner system. The dimensions of a runner system instantiation are either related to the cavity geometry or can be calculated without external data (see below). These calculations must be made during the instantiation of the gating system in order to identify all dimensions of a tab gate or a fan gate. Therefore the process constraints in the Manufacturing model relating to fan gate and tab gate dimensions must contain data on runner design. It can also be seen from Figure 7.58 that the lengths of a pin gate and a sprue gate instantiation are related to the depth of the cavity block under the impression. The identification of the depth of an integer cavity block has been described in section 7.5. However as described in section 7.5.2. the instantiation of the feeding system is a prerequisite to the instantiation of all cavity plate features except the integer cavity parting line. Therefore at the time of instantiation of the gating system the integer cavity block has not been instantiated and the process constraints in the Manufacturing model that relate to pin gate and sprue gate dimensions must contain data on integer cavity block design. The above structured methods provide a solution to the problem of instantiation of a gating system.

Having instantiated a gating system, the next problem is to identify the feature types which make up the remainder of a feeding system. This can be related to the type of gating system, and can be identified from practice and the literature, eg Pye (1989), Dym (1987). The author has related the type of gating system instantiation to the other feature instantiations required



source – Pye (1989)

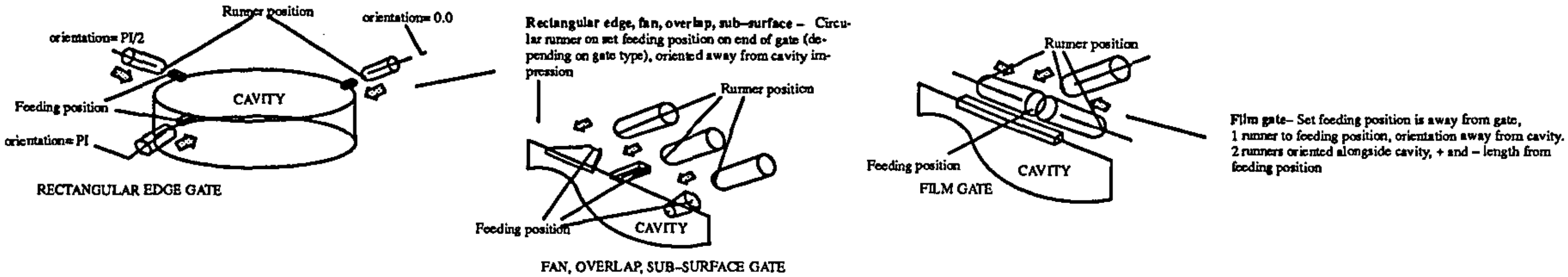
Figure 7.58. Gate parameters can be set by empirical formulae, but most formulae require data from other entities.

Gate type	Runners		Main feeding sprue
Ring gate	Trapezoidal/ Modified trapezoidal	1	1
Tab gate	Circular/Hexagonal	1	1
Sprue gate	————	0	0
Sub-surface gate	Circular/Hexagonal	1	1
Diaphragm gate	————	0	1
Rectangular edge gate	Circular/Hexagonal	1	1
Overlap gate	Circular/Hexagonal	1	1
Fan gate	Circular/Hexagonal	1	1
Pin gate	Trapezoidal/ Modified trapezoidal	1	1
Film gate	Circular/Hexagonal	3	1

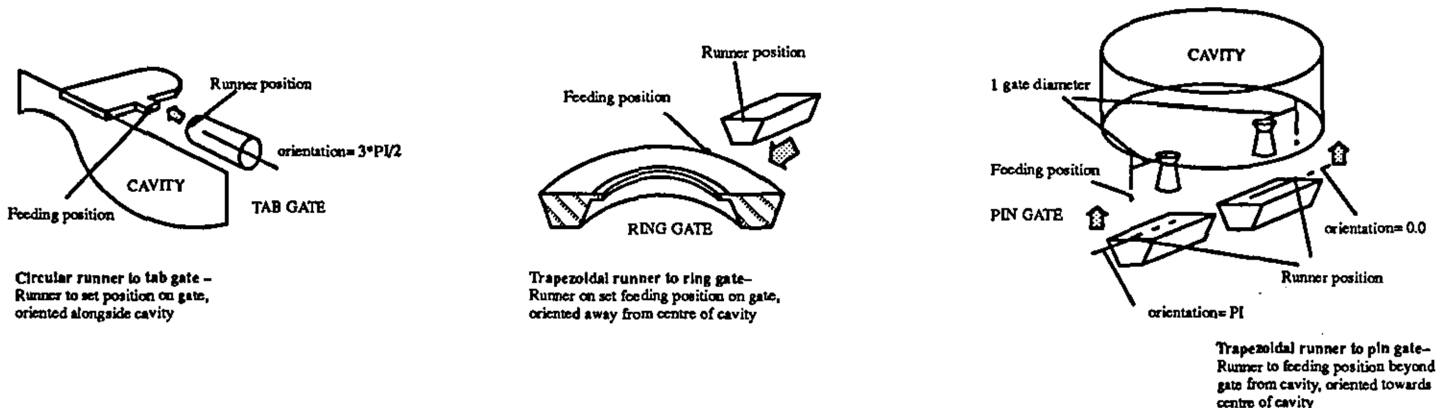
Table 7.1. Types and numbers of runner and main feeding sprue instantiations associated with different gating system types.

to complete the feeding system, as shown in Table 7.1. This data has been captured in relation to each gating system feature type. The instantiation of the remainder of the feeding system as identified in Table 7.1 is discussed in the remainder of this chapter.

Assuming a runner is required, this is now considered. The author has identified the geometric relationships between runner instantiations and the various types of gating system in Figure 7.59. The figure shows where the position of each runner instantiation should be measured and where the runner position should be in relation to the geometry of the gating system and the cavity. Figure 7.60 shows that the rules for identification of the dimensions of a runner in relation to the geometry of the gating system and the cavity/core can be identified from the literature, eg Pye (1989), Dym (1987). These geometric relationships and process constraints on dimensions have been captured in relation to each runner system type. Ong et al (1995) used the same approach to provide feedback on the parameters of a runner system, but with no mould geometry a question and answer routine was required to obtain from the designer information on desired runner/cavity volume ratio and the number of secondary runners etc. No runner geometry was created and only diameter feedback information could be provided. In this work, where the geometry of the cavity/core is available, it can be seen from Figure 7.60 that in all cases the dimensions of a runner are dependant upon the

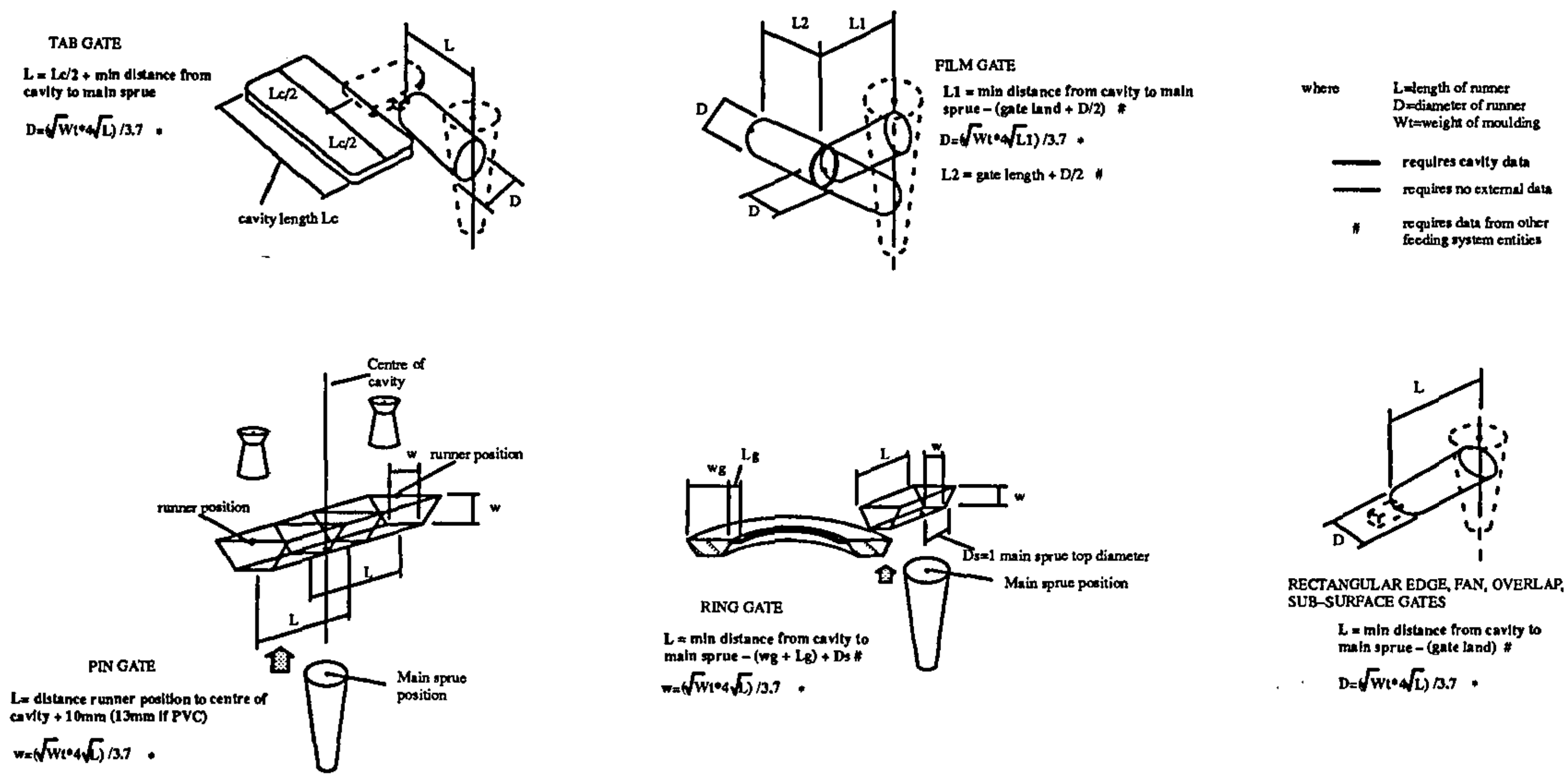


A. Circular runner onto rectangular edge, fan, overlap, sub–surface and film gate.



B. Circular runner onto tab gate, trapezoidal runner onto pin and ring gate.

Figure 7.59. Runner onto gate – 'on end of' has a different meaning for each gating system type.



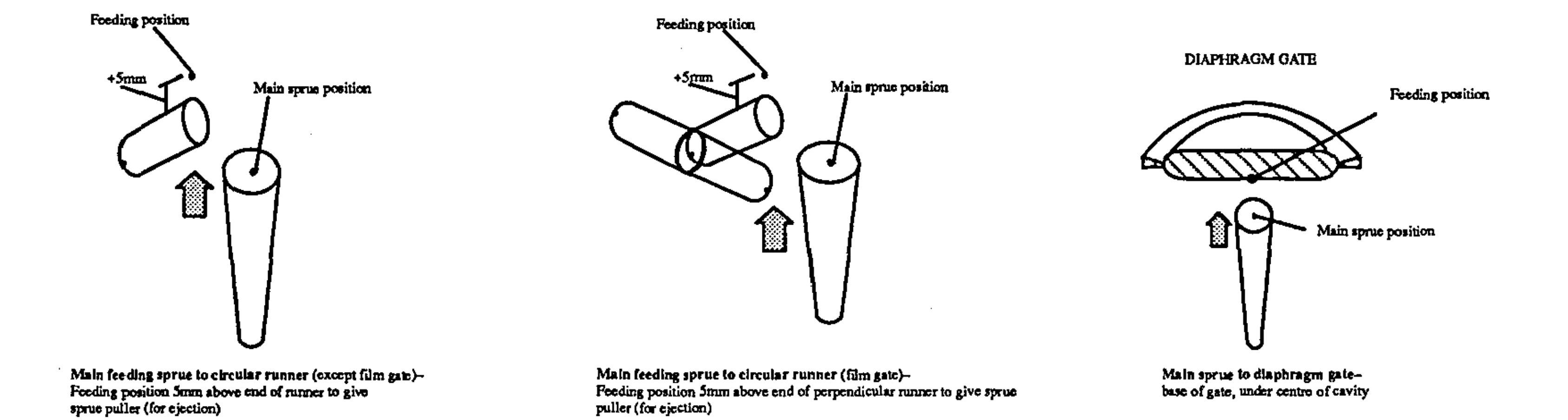
* source – Pye (1989)

Figure 7.60. Runner parameters have to be derived from gate, sprue and cavity data

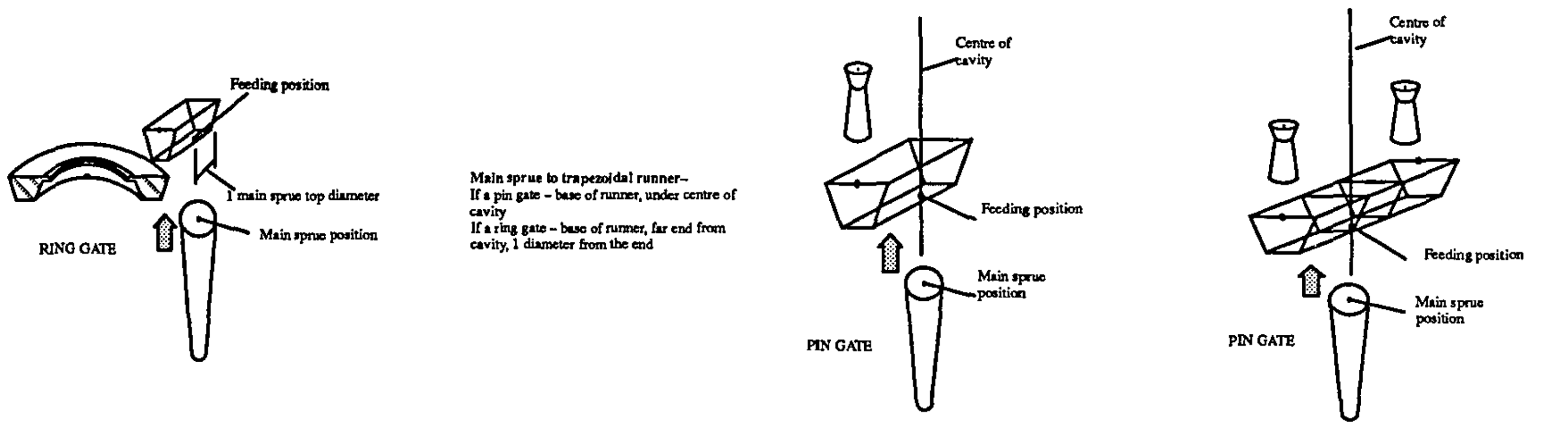
position of the main feeding sprue relative to the cavity. Therefore the process constraints in the Manufacturing model that relate to runner dimensions must contain data on main feeding sprue design. The above structured methods provide a solution to the problem of instantiation of a runner system.

Finally the instantiation of a main feeding sprue. We need first to identify the geometric relationship between a main feeding sprue instantiation and a runner system or diaphragm gate instantiation. In Figure 7.61 the author identifies the geometric relationship between a main feeding sprue and a runner system or a diaphragm gate. From the figure the correct geometric relationship is attained if the main feeding sprue position is the same as the feeding position. It can be seen that the geometric relationships between a runner instantiation and a main feeding sprue is dependant upon the type of gating system. These geometric relationships and their relevance to particular types of gating system have been captured within the data associated with a main feeding sprue in the Manufacturing model

With regard to the dimensions, as shown in Figure 7.62 these can be related to an integer cavity block, and these relations can be captured in association with a main feeding sprue feature type: If a mould cavity is side fed as in Figure 7.62 a) and b), the length of a main feeding sprue is directly related to the depth of an integer cavity block unless i) the top diameter of a tapered main feeding sprue with a base diameter to match an injection machine nozzle has a lesser diameter than the width of a runner instantiation, or ii) a tapered sprue of maximum width 10 mm at the top (Pye (1989)) does not have sufficient length to reach the base of the cavity block with a base diameter large enough to interface with an injection machine nozzle. Scenario i) requires an extension of the integer cavity block to allow an increase in the top diameter of the main feeding sprue to be equal to that of the runner. Scenario ii) requires the instantiation of an integer cavity nozzle recess as described in section 7.5. It can be seen that if a mould cavity is under fed using a pin gate as in Figure 7.62 c), and the top diameter of a tapered main feeding sprue with a base diameter to match an injection machine nozzle has a lesser diameter than the width of a runner instantiation, then an extension of the integer cav–



A. Main feeding sprue onto circular runner and diaphragm gate.



B. Main feeding sprue onto trapezoidal runner.

Figure 7.61. Main feeding sprue onto runner – 'at end of' has a different meaning for each gate type

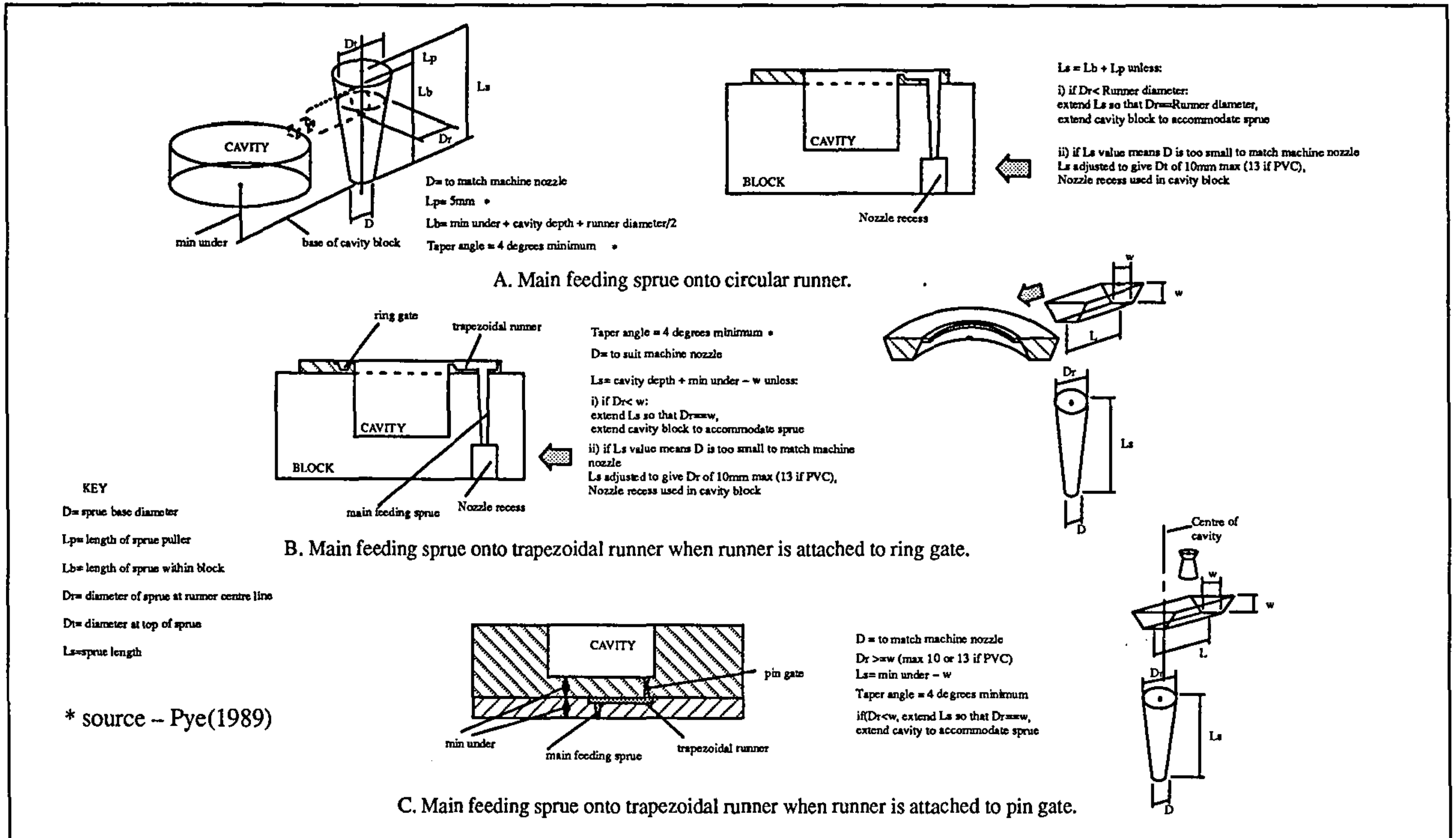


Figure 7.62. Main feeding sprue onto runner – parameters derived from gate, cavity and runner data.

ity backing plate is required to allow an increase in the top diameter of the main feeding sprue to be equal with that of the runner. The above structured methods provide a solution to the problem of instantiation of a main feeding sprue.

To enable concurrency, changes to a feeding system in response to consideration of process constraints must be reflected in the other viewpoints in injection moulding design. The only area earlier in the design cycle where changes to the feeding system affect another design viewpoint is mouldability, which is affected with respect to a gate instantiation that corresponds to a gating system. If the position of a gating system is changed due to analysis of process constraints, the position of a corresponding mouldability gate instantiation must also be changed, so that a later analysis of the feeding system does not show the same design problems as existed in a current analysis.

The above structured methods provide a solution to the instantiation of the feeding system features identified by the author, enabling a linking between the feeding system design viewpoint and those of product and the cavity/core design.

7.7.3. Feeding system design support strategy.

The above investigation has identified the information precedence relationships in feeding system feature instantiation, and the feeding system design support strategy is as expected for a mould design, ie i) gating system instantiation, ii) runner system instantiation, iii) main feeding sprue instantiation (Figure 7.45d). After execution of stage i) of the strategy, the first stage of the cooling strategy in the cavity (Figure 7.45e) is a prerequisite to executing stages ii) and iii) which require a knowledge of the minimum depth of the cavity plate.

Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above.

7.8. Mould system elements assessment – Cooling system.

For true concurrency a designer must be aware of the choices that exist in terms of the configuration and geometry of the cooling system as the geometry of the product evolves. However, as with the feeding system, this has always been examined retrospectively in the past using the whole product geometry plus the completed feeding system design as the input. The approach until now has either been to examine cooling systems in terms of machining features, as the input to an NC machining process planner, eg Rho et al (1990), or to automatically generate the cooling system using standard parts and predetermined rules on system configuration, eg Cinquegrana (1990)a, (1990)b.

In this work a strategy is required to provide the designer with choices in respect of the configuration and geometry of a cooling system at each juncture in the development of a product. In order to provide the designer with this choice a translation process is required between the viewpoints of the cavity/core and individual feature types of the cooling system. The problems involved and the formulated solutions are now described.

7.8.1. Problems of instantiation of cooling system features.

This section discusses the problems of instantiating the features in the cooling system, ie standard flow way, baffle flow way and baffle blade. The order in which the following problems arise is dependant upon the type of cooling system, and whether it is in the cavity or the core, as discussed in section 7.8.3.

One consideration is which configuration of cooling system to use, ie a 'U' tube, 'Z' tube, or more complex design. The optimum configuration of a cooling system is dependant upon the geometry of the cavity/core and feeding system that must be cooled and whether the cooling system is in the cavity or the core. The temperature of the mould is important as it governs a portion of the overall moulding cycle, Pye (1989). The melt flows more freely in a hot mould but a greater cooling period is required before the solidified moulding can be ejected. Alternatively, while a melt solidifies quickly in a cold mould it may not reach the extremities of the impression. During the impression filling stage the hottest material in the mould will be in the vicinity of the entry point, ie the gate, and the coolest will be at the point furthest from the entry. The temperature of the coolant fluid however, increases as it passes through the mould. To minimise the cycle time the configuration of a cooling system must provide as far as possible an even cooling effect over the surface of the moulding, Pye (1989), Dym (1987). To achieve this it is necessary to locate the incoming coolant fluid adjacent to the 'hot' moulding surfaces and to locate the channels containing heated coolant fluid adjacent to the 'cool' moulding surfaces. With respect to cooling system design, it is not always possible to adopt the idealistic approach and common sense is required if unnecessarily expensive moulds are to be avoided, Pye (1989).

Having identified the best possible configuration, it is necessary to identify which types and quantity of cooling feature instantiations are required to build the chosen cooling system configuration, ie what types of flow ways and how many, does the cooling system require baffle blades to change the flow of coolant through the system etc. For example a 'U' tube configuration of cooling system would require three standard flow ways. Also to be identified is the way in which the instantiations fit together to create the chosen configuration.

Another problem is to maximise the potential cooling effect of the system in order to provide the minimum cycle time. This problem is related to the capacity of a cooling system to carry coolant around the mould and the surface area that is made available for heat transfer between the coolant and the mould. As shown in Figure 7.63, a further problem of cooling system instantiation is the avoidance of local variations in the temperature across the impression with the consequent moulding problems. Thus the optimum cooling system formation provides the maximum cooling effect without causing local variations in the temperature of the impression.

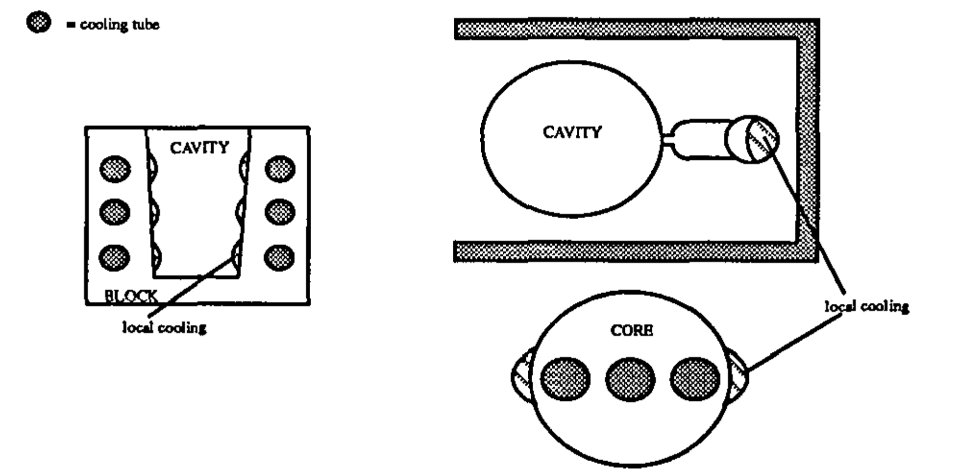


Figure 7.63. Optimum cooling formation must give maximum cooling without local variations

The final problem of the instantiation of a cooling system is to identify the dimensions of the cooling feature instantiations making up the cooling system geometry. The dimensions of a cooling feature instantiation are dependant upon i) the final formation of a complete cooling system, ii) which part of the overall cooling system is provided by the instantiation eg is it the bottom of the 'U' in a 'U' tube or on the side etc, and iii) the dimensions of the cavity or core block instantiation within which the cooling system is contained.

7.8.2. A structured methodology for cooling system feature instantiation.

The problems of instantiation of the cooling system features described above can be dealt with as follows. The configuration of a cooling system must as far as possible minimise the cycle time by providing an even cooling effect over the surface of the moulding without necessitating an unduly expensive mould, Pye (1989). The configuration problem is different for a cooling system in the cavity and for a cooling system in the core, and so are dealt with separately.

With respect to a cavity, the options for cooling system configuration are limited by the geometrical requirement to envelope the cavity impression. Additionally, as indicated by Pye (1989), where possible highly complex cooling systems which increase the cost of the mould should be avoided. Therefore two configurations of cooling system in the cavity are considered; the 'U' tube configuration and the 'paired' tube configuration. The author has determined that the configuration of the cavity cooling system can be related to the type of gating system and whether the impression is single or multi gated, as shown in Figure 7.64. A cavity impression is either undergated, using a pin gate a diaphragm gate or a sprue gate, or else it is side gated, using a ring gate, a fan gate, a film gate, a sub-surface gate, an overlap gate, a rectangular edge gate or a tab gate. Looking at the side gates, as shown in Figure 7.64 a) a single side gate produces a hotter moulding at the gated end than at the other. Therefore in order to produce as even a cooling effect as possible a 'U' tube configuration should ideally be used to provide extra cooling at the gated end. It can be seen from the figure that if more than one layer of cooling tubes is used the coolant can be passed in opposite directions in the layers to alleviate the effects of the coolant heating up whilst travelling through the mould. From Figure 7.64 b), with more than one side gate a 'U' tube configuration would actually promote an uneven cooling effect. Therefore a system of 'paired' tubes is preferable.

With respect to an undergated cavity impression, if a single gate is used then if an undergate is in the centre of the base as in Figure 7.64 c) a 'paired' tube configuration is preferable as a 'U' tube configuration promotes uneven cooling. If an undergate is acentral as in Figure 7.64 d) then a 'U' tube configuration is preferable to provide additional cooling at the gated end. If multiple undergating is used as in Figure 7.64 e) a system of 'paired' tubes is preferable. As shown in Figure 7.64 f) all types of gating system can be related to a preferred configuration of a cavity cooling system. Capturing the above relations for cavity cooling systems solves the problem of identifying the configuration of a cooling system in the cavity.

The configuration of a cooling system in a core depends amongst other things upon whether the core is a 'deep' core or a 'shallow' core. A core is considered to be deep if the depth of the cavity impression is greater than 25 mm, Pye (1989), in which case cooling tubes are required inside the core impression as well as the core block. As shown in Figure 7.65 a), for the cooling of a deep integer core three types of cooling system are available, ie i) a stepped circuit system, ii) an angled hole system, and iii) a baffle tube system. It can be seen from

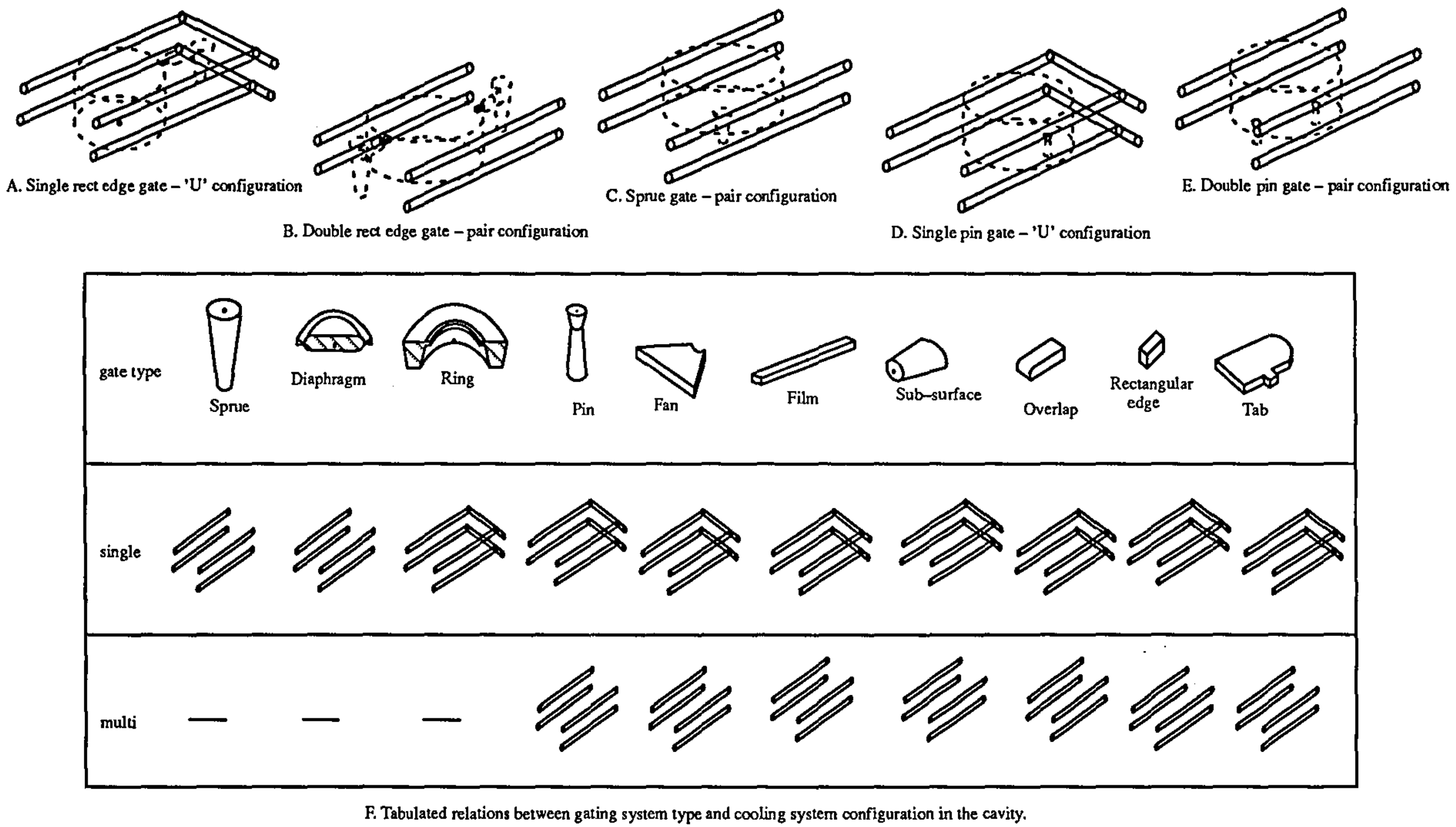


Figure 7.64. Preferred cavity cooling configuration is related to the gating system.

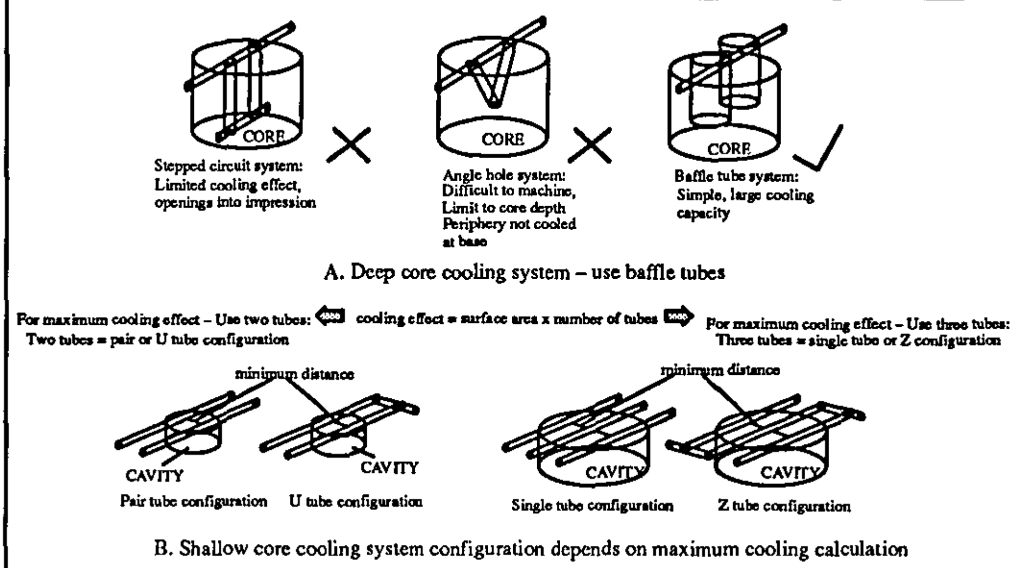


Figure 7.65. Possible cooling configuration in the cavity and core.

the figure that a stepped circuit system requires holes drilled through the side of the core, which are subsequently plugged and refinished. The angled hole system may cause difficulties in making sure the angled holes meet up in a very deep core, and it can be seen that this configuration places a limit on the depth of core that can be reasonably cooled. Because of the geometry of the angled hole system, the lower end of the core is only cooled in the centre. The baffle tube system uses baffle blades to divert the coolant so that it flows into and out of the core in the vertical flow ways. It can be seen that the baffled hole system has a greater capacity to carry coolant around the mould than the others and it has none of the problems of configurations i) and ii). Therefore this is the only deep cooling system considered in the authors work.

As shown in Figure 7.65 b), the options for the configuration of a cooling system in a shallow core depends on the number of tubes across the top of the core. In the authors work this is determined by the maximum cooling capacity evaluation, which is discussed later in this section. The number of tubes across the top of a core impression in a core block can be any number from a single tube upwards. As shown in the figure, with two tubes a cooling system configuration can be a 'U' tube or a 'paired' tube. Any number of paired or U tubes can be used where the number of tubes across the core is a multiple of two, and the direction of flow can be alternated to provide more even cooling. With three tubes a cooling system configur-

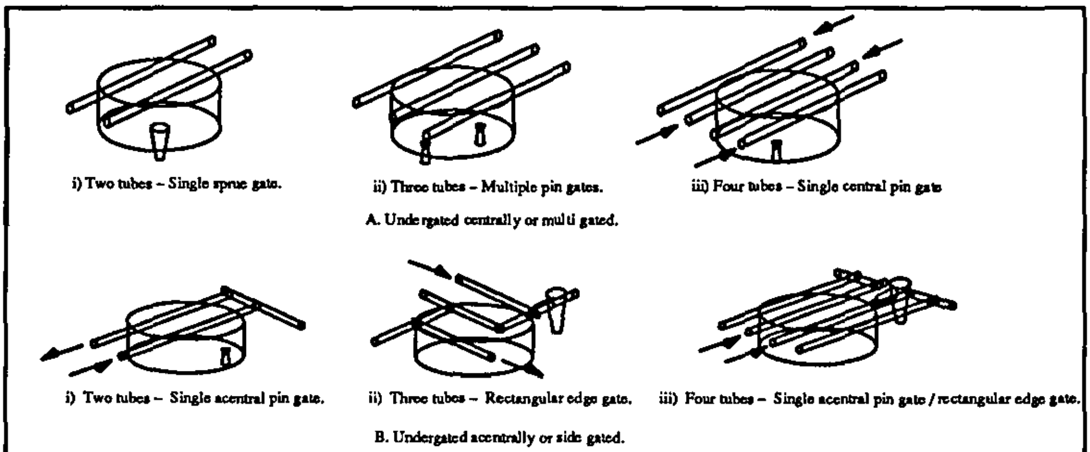


Figure 7.66. Relating cooling system configuration to gating in a shallow core.

ation can be 'Z' tube or a 'single' tube. Any number of Z tubes can be used where the number of tubes across the core is a multiple of three, and the direction of flow can be alternated to provide more even cooling.

In order to identify the preferred configuration of cooling system across the top of a shallow core, the appropriate configuration can be related to the gating system type and the number of gates, as shown in Figure 7.66. It can be seen from the figure that if a cavity is undergated centrally or with multiple gates as in Figure 7.66 a), then for two tubes across the core the cooling system configuration should be 'paired' tubes, for three tubes across the core the configuration should be 'single' tubes, and for four tubes across the core the configuration should be a single tube system with the coolant flow as shown. This is the same for a cavity that has multiple side gates as in Figure 7.64 b). In Figure 7.66 a) a 'U' tube in scenario i), or a 'Z' tube in scenario ii) would promote uneven cooling in the mould. As shown in Figure 7.66 b) if a cavity is undergated acentrally or is side gated then for two tubes across the core the cooling system configuration should be a 'U' tube, for three tubes across the core the configuration should be a 'Z' tube, and for four tubes across the core the configuration should be a balanced 'U' tube system. These configurations give extra cooling to the 'hot' part of the mould. Capturing the above relations for core cooling systems solves the problem of identifying the configuration of a cooling system in the core.

Having identified the configuration of a cooling system in the cavity and the core, the types and numbers of cooling feature instantiations required to make the geometry of the system has to be identified and how they fit together. The types and numbers of cooling feature instantiations for each configuration, and how they fit together have been identified in Table 7.2, and have been captured in relation to cavity and core cooling systems.

A further problem of cooling feature instantiation is to identify the maximum cooling effect. The author has identified a method of evaluating the maximum cooling effect of a cooling system as shown in Figure 7.67. The maximum cooling effect is related to the capacity of a cooling system to carry coolant around the mould and the surface area available to allow heat transfer from the mould to the coolant. The surface area of a cooling tube per unit of length is related to the diameter. However there is a limit on the range of tube diameters that can be used in the mould. The smallest tube diameter that should be used is 7 mm as a smaller tube has insufficient surface area for good heat transfer, and the largest diameter tube that should be used is 10 mm, otherwise too much coolant will be needed to be pumped around the mould, Pye (1989). Therefore the maximum cooling effect evaluation aims to maximise cooling within the range of diameters above. The exception is the baffle flow way in a deep core. Owing to the two way flow and the requirement to contain a baffle blade, the diameter of these tubes can range from 12 mm to 16 mm, Pye (1989). As shown in Figure 7.67, to identify the maximum cooling effect the maximum number of cooling tubes of each diameter that can be fitted a) alongside the cavity, b) across the top of the core, and c) inside the core is identified. This number is multiplied by the surface area per unit length. The diameter of cooling tube which provides the highest total surface area is used in the mould cooling system.

In order to avoid local variations in cooling effect a clearance of 16 mm is recommended around a cooling tube, Pye (1989). As shown in Figure 7.67, this distance is taken as the minimum distance between a) layers of tubes in the cavity, b) tubes across the top of the core, and c) tubes inside the core. Figure 7.68 shows how the optimum cooling formation is identified in the cavity and core. This involves ensuring the minimum distance of 16 mm between a

Cavity cooling systems:	Standard flow way	Baffle flow way	Baffle blade
Paired tubes	2	——	——
U tube	3	——	——

Core cooling systems:	Standard flow way	Baffle flow way	Baffle blade
Baffle tube	1	As many as fit in core	As many as baffle flow ways
Paired tubes	2	——	——
U tube	3	——	——
3 single tubes	3	——	——
Z tube	5	——	——
4 single tubes	4	——	——
Balanced U tube	6	——	——

A. Number and type of cooling system features in each configuration of cooling system.

Tube orientation:	Standard flow way number					
	1	2	3	4	5	6
Cavity paired tubes	0.0	0.0	——	——	——	——
Cavity U tube	0.0	0.0	PI/2	——	——	——
Core Baffle tube	0.0	——	——	——	——	——
Core paired tubes	0.0	0.0	——	——	——	——
Core U tube	0.0	0.0	PI/2	——	——	——
Core 3 single tubes	0.0	0.0	0.0	——	——	——
Core Z tube	PI/2	3PI/2	3PI/2	0.0	PI	——
Core 4 single tubes	0.0	0.0	0.0	0.0	——	——
Core balanced U tube	0.0	0.0	0.0	0.0	PI/2	3PI/2

B. Orientation of standard flow way instantiations in each configuration of cooling system.

Table 7.2. Components of each cooling system configuration and how they fit together.

cooling system and the closest parts of the impression and the feeding system. Capturing the above relations for cavity and core cooling systems solves the problems of identifying the cooling system that gives the maximum cooling effect and of avoiding the problems of local

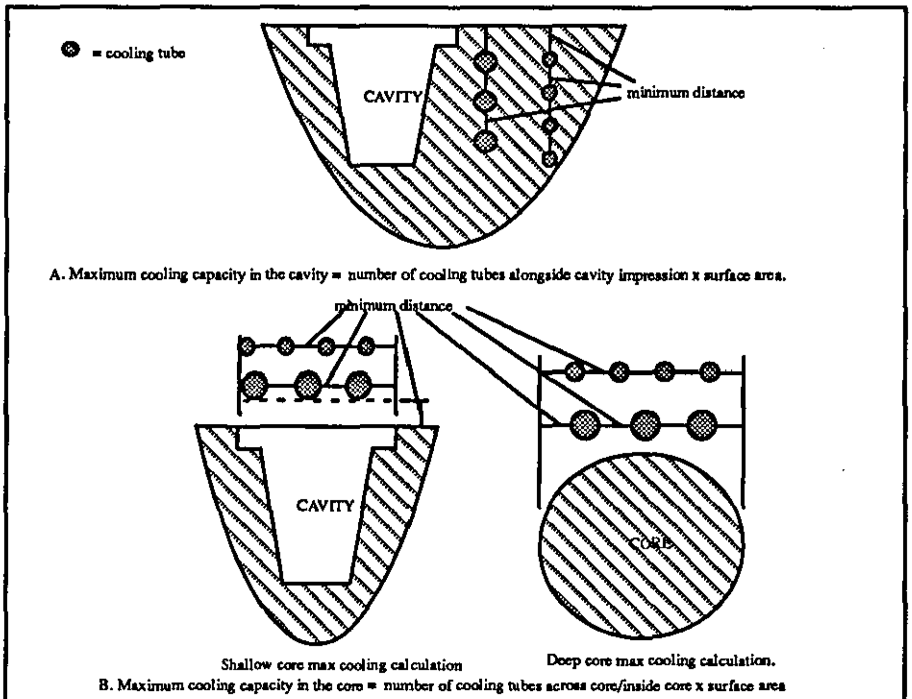


Figure 7.67. Maximum cooling capacity in the cavity and core.

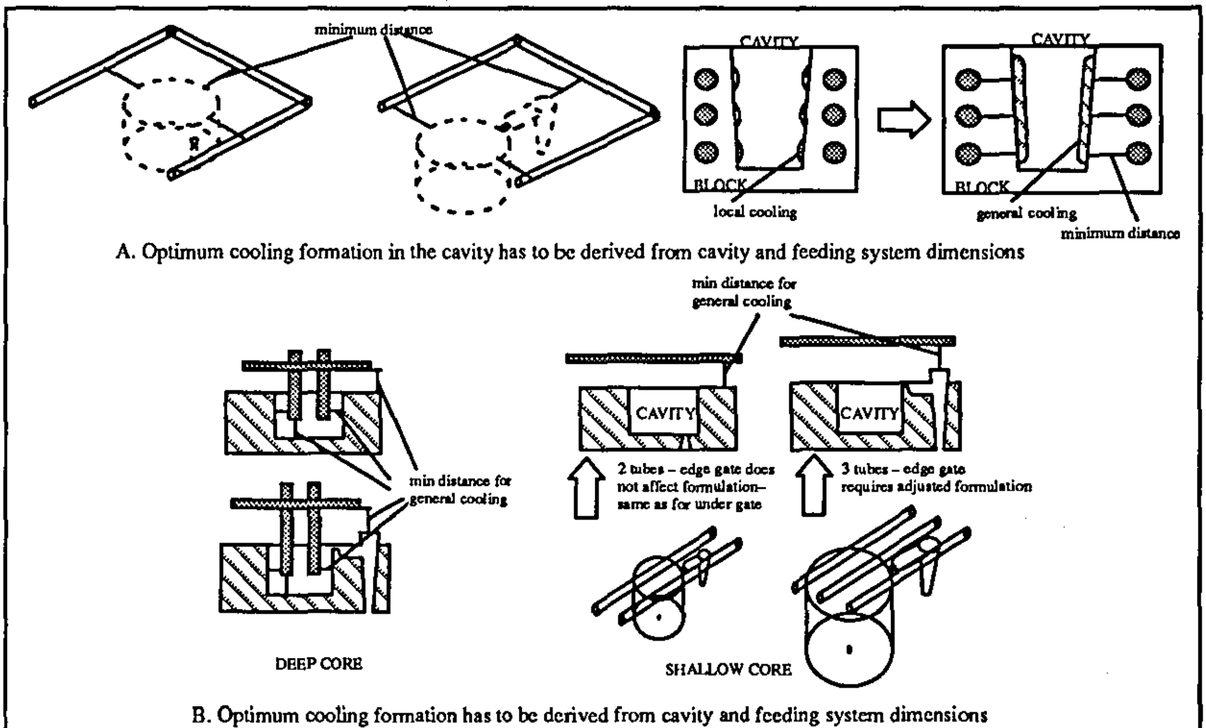


Figure 7.68. Optimum cooling formation in the cavity and core.

cooling effects.

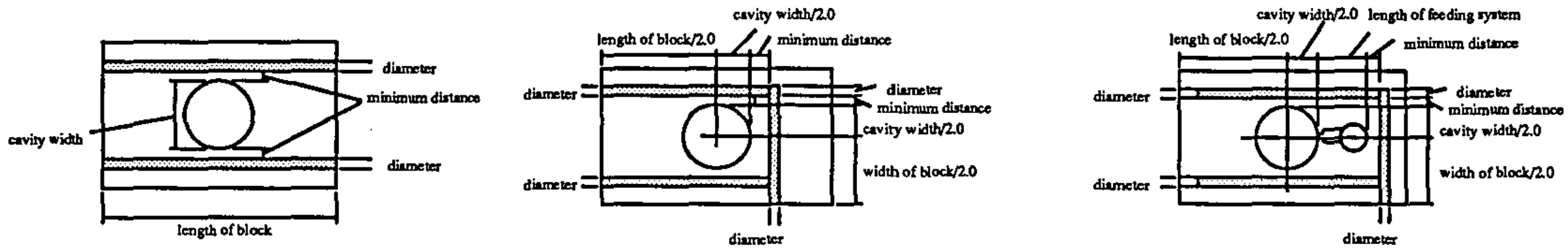
The final problem of instantiation of the cooling system is to identify the dimensions. The dimensions of the cooling features that make up a cooling system can be related to the dimensions of a cavity or core block and the size of the impression and feeding system. These relationships have been identified by the author and are shown in Figure 7.69, and have been captured in relation to individual cooling system feature types.

The above structured methods solve the problems of instantiation of a cooling system in the cavity and core.

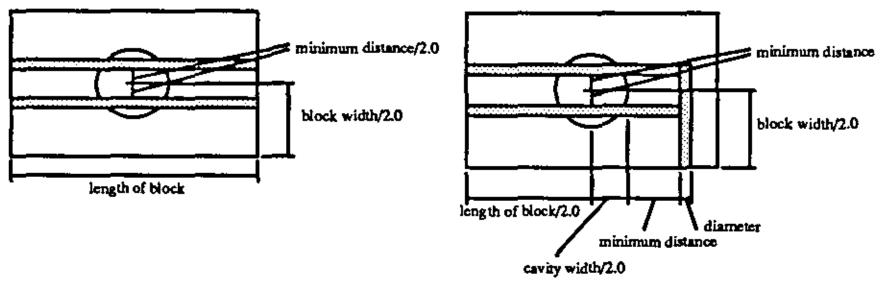
7.8.3. Cooling system design support strategies.

The work described above has enabled the identification of the precedence relationships to support cooling system design. The strategy is not always the same for the cavity and the core. In the cavity the first stage of the strategy has to be 'identify maximum cooling effect' as the data from this activity is a prerequisite to the latter stages of the feeding system design strategy (section 7.7.3). The remaining stages of the cavity cooling system design strategy are executed after feeding system design. The strategy in the cavity is i) identify maximum cooling effect, ii) identify cooling system configuration, iii) identify type and number of feature instantiations, iv) identify how instantiations fit together, v) identify optimum cooling formation, vi) identify dimensions (Figure 7.45e). The last stage in the cavity cooling system design strategy has the final stage of the cavity design strategy (Figure 7.45a) in the middle because some dimensions of the cavity plate are dependant upon those of the cooling system and vice versa. In the core the stages i) and ii) interchange, depending on whether the core is a deep or shallow (Figure 7.45f). This is because in a shallow core the first stage identifies the number of tubes across the core, which is a prerequisite to identifying the options for configuration of the cooling system.

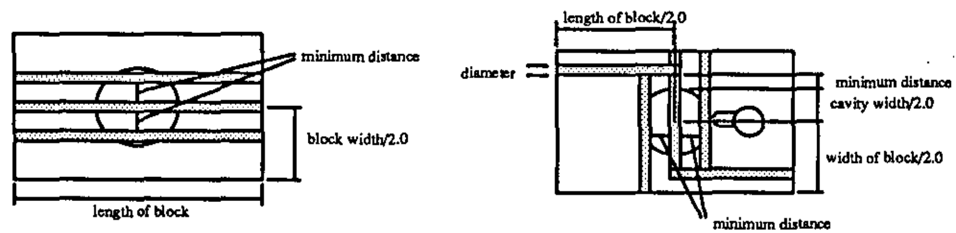
Chapter 9 describes the design of an injection moulding strategist application that incorporates the ideas that have been described and discussed above. As a result of the above work the IDEF0 activity model of the functionality of a design for injection moulding support application has been significantly modified (Appendix 2).



i). Dimensions of cavity cooling systems related to the cavity impression, cavity block and feeding system.

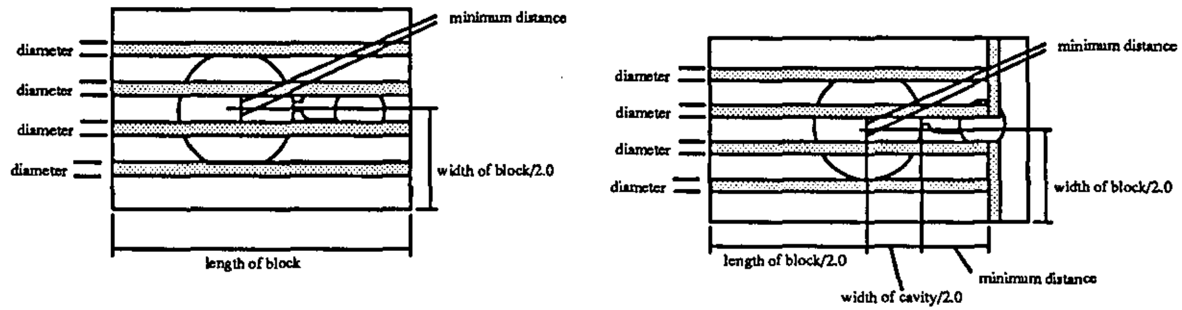


ii). Dimensions of shallow core cooling systems related to the cavity impression and core block – two tube systems.

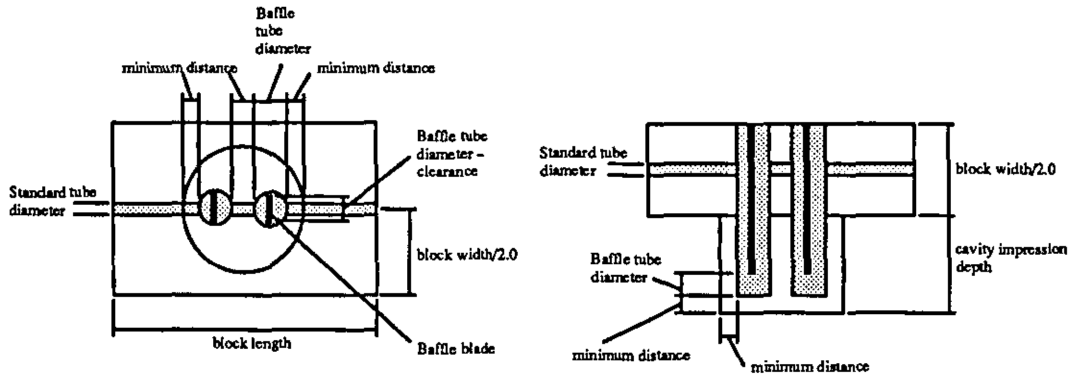


iii). Dimensions of shallow core cooling systems related to the cavity impression and core block – three tube systems.

Figure 7.69a. Relationship of cooling system dimensions with the mould block, impression and feeding system.



iv). Dimensions of shallow core cooling systems related to the cavity impression and core block - four tube systems.



iv). Dimensions of deep core cooling systems related to the core impression and core block.

Figure 7.69b. Relationship of cooling system dimensions with the mould block, impression and feeding system.

Chapter 8.

Experimental information modelling environment.

8.1. INTRODUCTION.

This research has investigated the range of feature types and feature interactions required to support concurrent design for injection moulding. The types of feature and their associated constraints are captured in information models. The information model environment has been identified as a Product model and a Manufacturing model, and a need has been identified for a Product Range Model. This Chapter describes the design of an information model environment based on the need to support the strategist applications and interactions between them. Section 8.2 discusses the structure of the Product Range Model, Section 8.3 the Manufacturing model structure, and section 8.4 the Product model structure.

8.2. PRODUCT RANGE MODEL.

8.2.1. General Product Range Model Structure.

In order to provide a representation of the functional viewpoint, this work has focussed on the use of variant design. The utilisation of variant design as in the approaches of Dowlatshahi (1992), Cross (1989) and Suh (1990),(1995) offers the potential of building up a representation of the functionality of individual product ranges. The approach of the authors work has been to define a structured representation of product functions to support variant design. This representation is referred to as the 'Product Range Model' (PRM) and this section describes the authors investigation into the structure of a Product Range Model to capture the relationship between the functions of product ranges and form, and thereby support the association of function and form in a Product model.

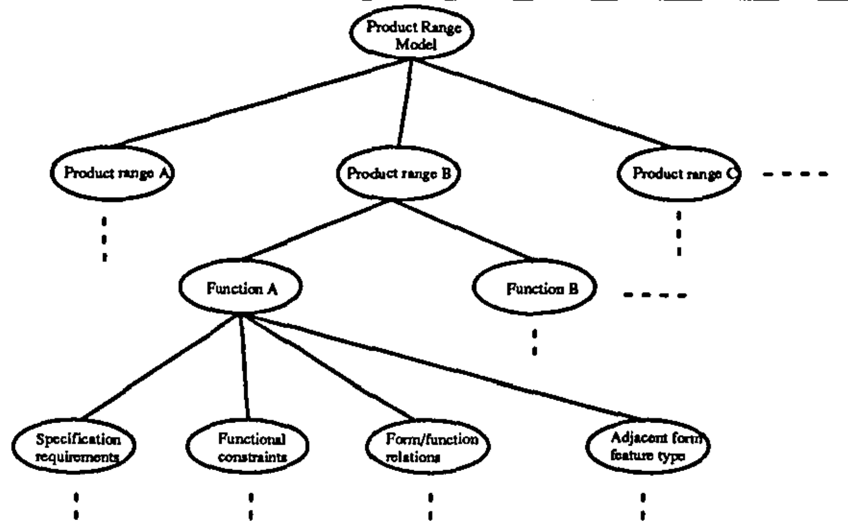


Figure 8.1. Conceptual data structure of the Product Range Model.

Figure 8.1 shows the conceptual structure of the Product Range Model that was identified by the author in Chapters 6 and 7. The figure shows a product range as one of many captured in the Product Range Model. Each product range representation captures information describing the functions of a product range, relating to specification (data) requirements, possible forms that can be used to achieve a function in the context of the product range, functional constraints and the type of adjacent geometry that affects functionality for a particular function. As described in section 7.3, this information structure enables functional constraints data for a product range to be evaluated with respect to the specific product. The Booch methodology has been used to create a formal representation of the Product Range Model data structure (Figure 8.2). In addition to the data described above, Figure 8.2 also-shows initial product definition data, which is discussed in section 8.2.2. Figure 8.2 illustrates how the Booch methodology can be used to capture the hierarchical structure of the elements in the Product Range Model. The EXPRESS language is used to capture the data structures within the elements in the Product Range Model.

8.2.2. Product range data.

From Figure 8.2, the main areas to be considered are form/function relations data, functional requirements and initial product definition data. From section 6.3.2, the association of function and form is driven by the designer as he/she builds up the geometry of a product

Title: Class Diagram: Product_range_model
Printed on: Friday, March 1, 1996
Printed by: enrjvl

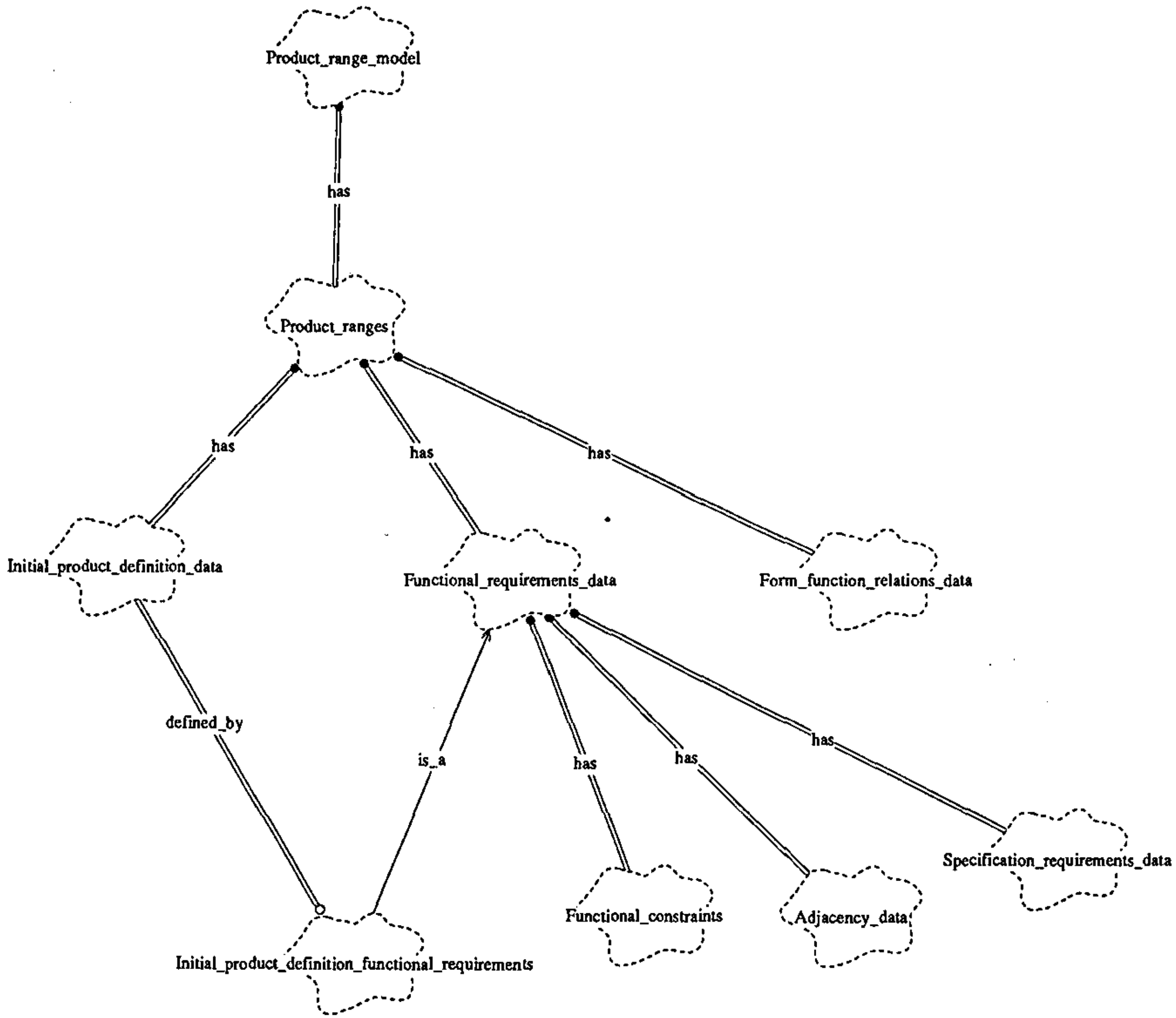


Figure 8.2. The Booch representation of the general Product Range Model structure.

in the Product model. *Form/function relations data* supports the designer in choosing a form feature to achieve a particular function of a product range. From Chapter 6, the author has adapted the approach of Suh (1990),(1995) who divides overall product functionality into a set of functional requirements which must be satisfied by a set of design parameters. This approach has been adapted to provide a set of functional requirements of a selection of product ranges. These are yoghurt pots, flower pots and a tamper evident ring for food packaging called PTPlus. The example products are all rotational products. Additionally the yoghurt pot and the PTPlus were made by the sponsor company. The flower pot example allowed the analysis of a minimum variation on the yoghurt pot range that is similar in overall shape. Thus a comparison could be made with respect to the differences in features that make up the similar products. Also with respect to the mouldability viewpoint, the flower pot case study allowed the analysis of features such as holes.

The author has identified the functions each product range has to perform and the forms required, as shown by the example in Figure 6.12. Figure 8.3 shows the product functions identified by the author for all the example products. It can be seen that to achieve most product functions requires more than one form feature. This is true for i) all PTPlus product functions, ie fix into lid, torsion failure, fix onto jar, ii) two yoghurt pot functions, ie enclose volume, destack, and iii) all flower pot functions, ie enclose volume, destack, drainage.

Within the approach of this thesis the problem is how to associate the product functions with individual form features in order to create functional features. The author has identified the relationships between product functions and individual form features by decomposing product functions requiring more than one form feature into sub–functions which can be achieved using individual form features, as shown in Figures 8.4, 8.5 and 8.6 for PTPlus, yoghurt pots and flower pots respectively.

Figure 8.4 shows that for PTPlus the sub–functions of the fix into lid function are locate in lid, hold in lid and cover lid edge. The sub–functions of the torsion failure function are break in torsion and prevent rotation and the sub functions of the fix onto jar function are

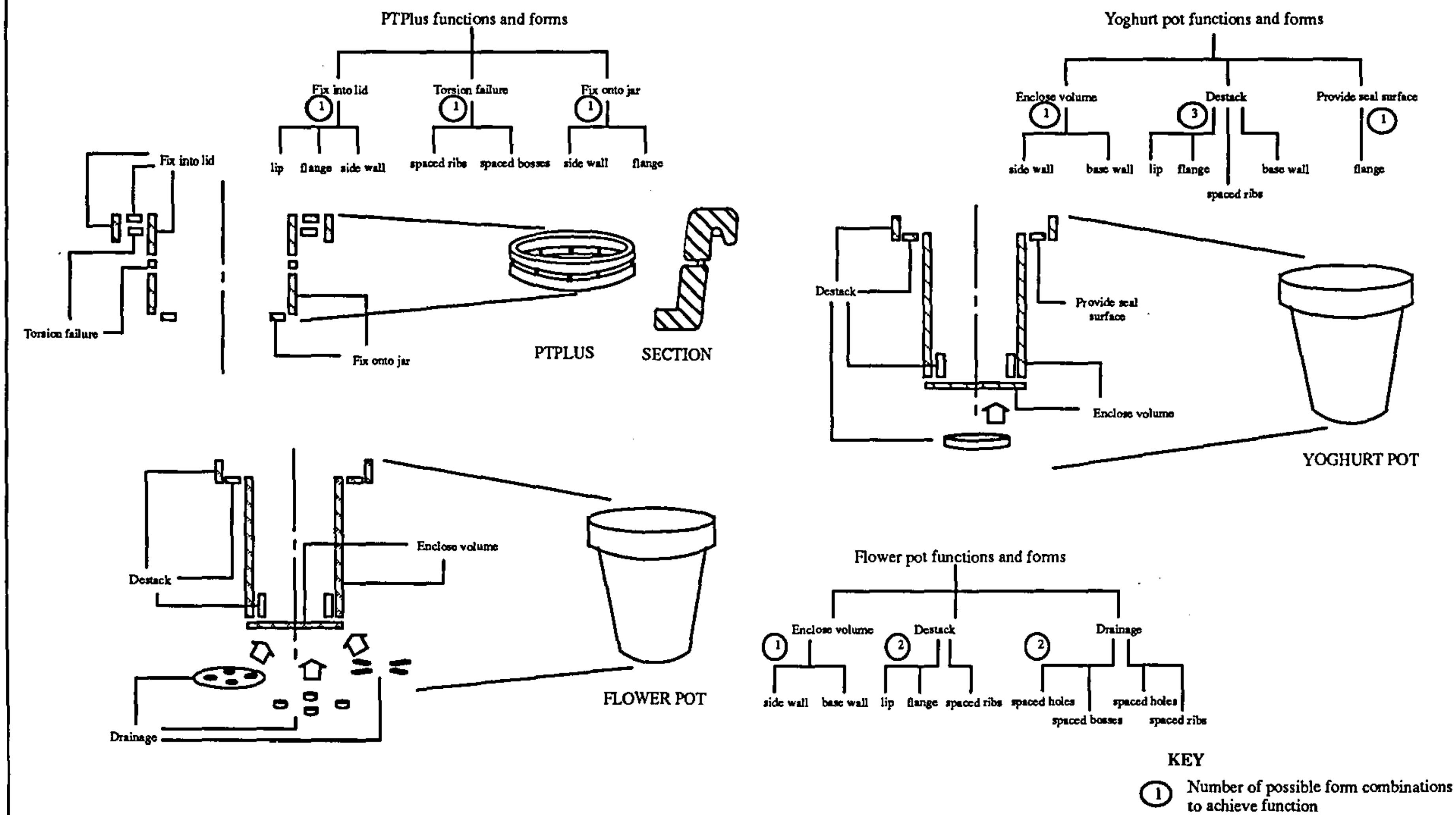


Figure 8.3. Product range functions and forms to achieve them.

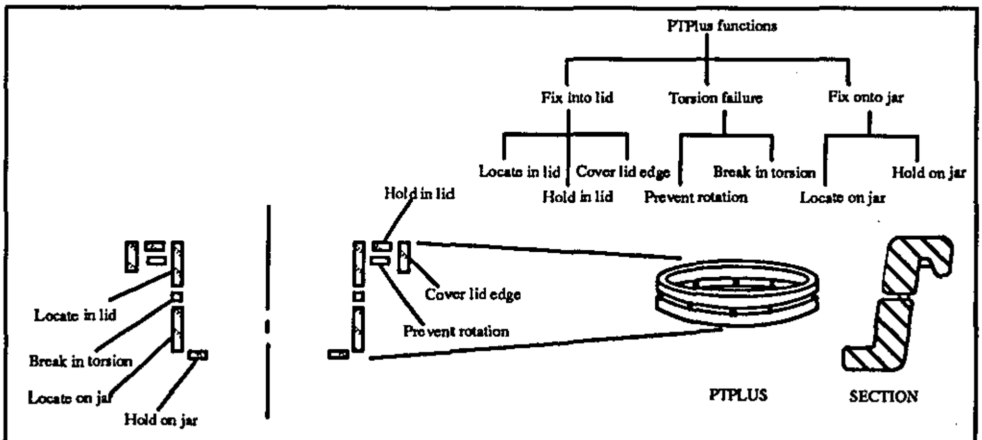


Figure 8.4. Decomposition of PTPlus functions.

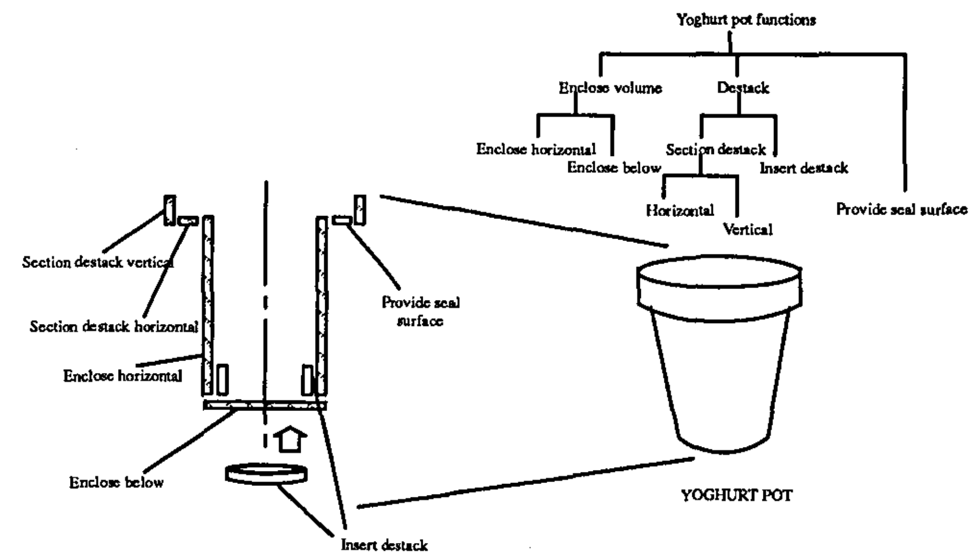


Figure 8.5. Decomposition of yoghurt pot functions.

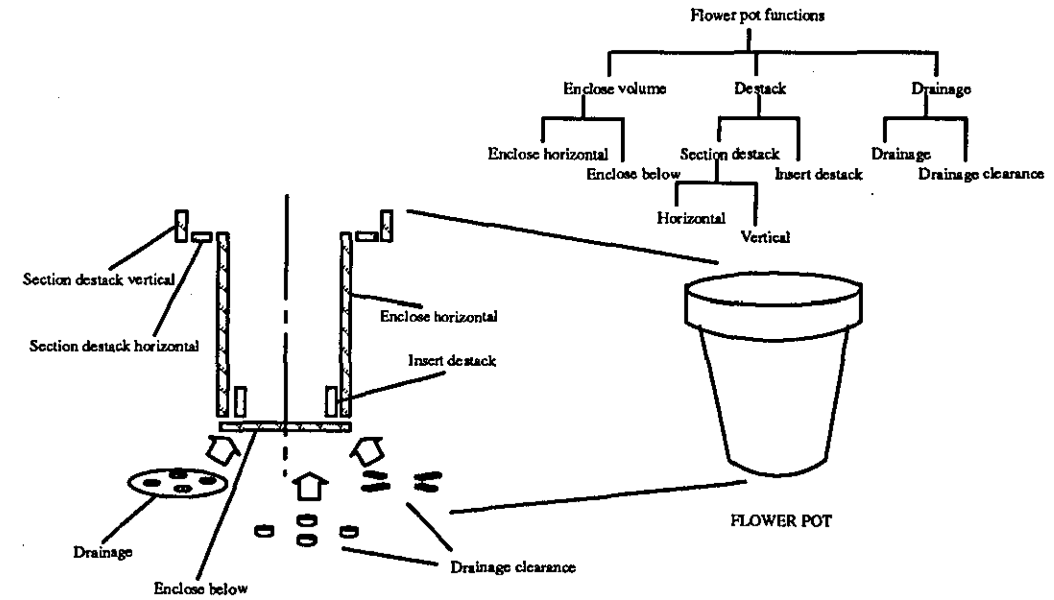


Figure 8.6. Decomposition of flower pot functions.

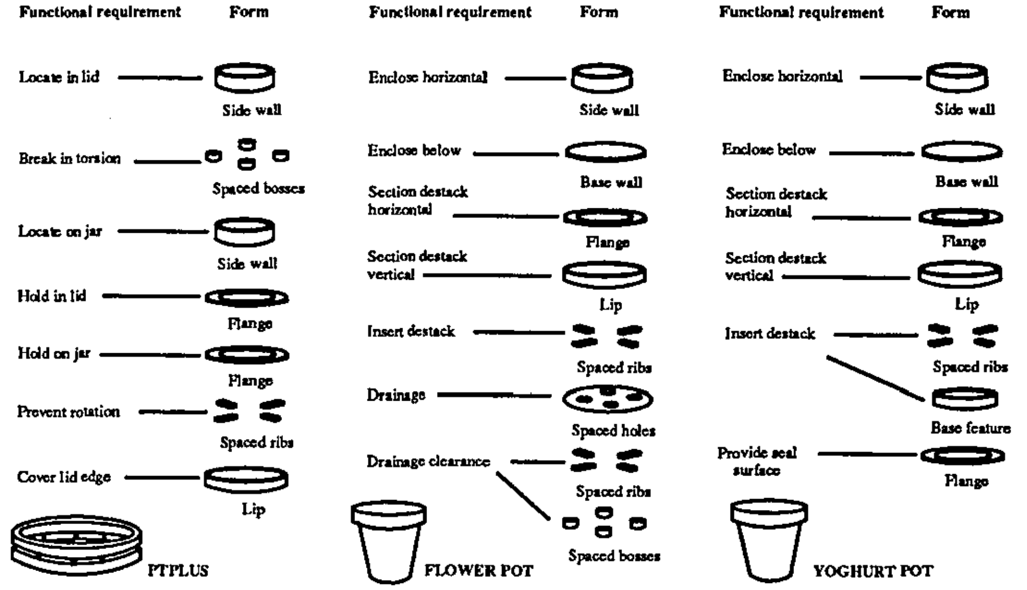


Figure 8.7. Relationship between product range functional requirements and form.

locate on jar and hold on jar. Figure 8.5 shows that for the yoghurt pot the enclose volume function has the sub–functions of enclose horizontal and enclose below and the destack function has the sub–functions of section destack horizontal, section destack vertical and insert destack. For the flower pot product range, as shown in Figure 8.6 the enclose volume function has the sub–functions of enclose horizontal and enclose below, the destack function has the sub–functions of section destack horizontal, section destack vertical and insert destack, and the drainage function has the sub–functions of drainage and drainage clearance. The above decomposition of product functions requiring multiple form features into lower level functions which can be achieved using only one form feature has enabled the identification of a set of relations between the functions of a product range and the geometric forms that can be used to achieve each function (Figure 8.7). This data is captured in the form/function relations data and is used to provide support for the designer as he/she chooses a particular function of a product range to address. The authors definition of a functional feature is therefore the association of a function or sub–function of a product range and a form feature to achieve the function or sub–function. The above structured methodology solves the problem of supporting the association of function and form data in the Product model. A formal representation of form/function relations data is presented in section 8.2.3 using the Booch methodology, and product examples are described in section 8.2.4.

In order to support the evaluation of functionality of a form feature the Product Range Model contains *functional requirements data.* From section 7.3. there are three requirements of functionality evaluation that affect the structure of functional requirements data; i) Functional constraints data must be available in the Product Range Model to be associated with a functional feature in the Product model, ii) data is required in the Product Range Model to support the instantiation of specification data in the Product model against which to evaluate functional constraints, and iii) in order to evaluate functionality in relation to the geometrical relationship of a form feature with surrounding geometry, because several forms may be adjacent to a functional feature instantiation, data is required on the type of adjacent geometry that is significant to functionality.

From Figure 8.2, the above requirements i) to iii) are captured as elements of the functional requirements data: The functional constraints capture the dimensional requirements for a form feature instantiation in the Product model to achieve the functionality of a particular product function, and also the requirements with respect to the spatial relationships of a form feature instantiation with adjacent form feature instantiations. The structure of functional constraints data enables the functionality evaluation of the independent form features set identified in Chapter 6. The evaluation of functional constraints is carried out with respect to a specification defined by the designer, supported by the specification requirements data in the Product Range Model. Specification requirements data captures the information that a designer must provide to instantiate a specification for a particular product function in the Product model. The instantiation is subsequently associated with a chosen form feature instantiation to create a functional feature. Specification requirements data also captures the relationships between specifications provided by the designer for different product functions on the same product in order to provide feedback advice where one specification creates a conflict with another in respect of the overall product functionality. Adjacency data simply provides the type of form feature that should be identified as adjacent geometry when evaluating the functionality of a form feature with respect to a particular product function, so that if there are several adjacent form features in a Product Model only the type that is relevant to the functionality evaluation will be identified. The above structured methodology solves the problem of supporting evaluation of the functionality of associated function and form data in the Product model. A formal representation of functional requirements data is presented in section 8.2.3 using the Booch methodology, and product examples are described in section 8.2.4.

When building up the geometry of a product in the Product model, it has been identified by the author that in order to ensure the instantiation of fully functional geometry some product functions must be addressed before others. For example, the destack function of a pot cannot be evaluated before the geometry has been instantiated for the containment function as the relevant dimensions of the pot for analysis of the destack function are unknown. Therefore in this thesis the *initial product definition data* captures those product functions that if addressed lead to the instantiation of geometry that is a prerequisite for evaluation of other product functions. In this way as the designer builds up the geometry of a particular product he/she must address first those functions identified in the Product Range Model as a part of the initial product definition, before the remaining product functions can be addressed in any order chosen. A formal representation of initial product definition data is presented in section 8.2.3 using the Booch methodology, and product examples are described in section 8.2.4.

8.2.3. The representation of product range data.

This section argues for a representation that captures the product range functional data as explained in the previous section. The author has proposed that functional data based on product ranges can be captured in a Product Range Model. From the previous section the areas to be considered are form/function relations data, functional requirements and initial product definition data. A formal representation of the structure of captured functional data in Booch is presented. The full Booch representation of the Product Range Model is shown in Appendix 5.

From section 8.2.2, *form/function relations data* captures a set of relations between the product functions and the geometric forms that can be used to achieve each function. For each product range that exists in the Product Range Model there must be a choice of form features that can be used to achieve each product function. Figure 8.8 shows how the captured relationships between product functions and form can be represented in the Booch methodology. The figure shows that the form/function relations data is captured for each product range in the Product Range Model.

Functional requirements data captures i) the functional constraints of a product function to provide functional feedback advice, ii) data to support an instantiation of a specification in the Product model and to capture the relationships between the specifications for different product functions, and iii) the adjacent feature type that is significant for the functionality of any given product function. Figure 8.9 shows how the structure of the captured functional requirements data, minus the functional constraints can be represented in the Booch methodology. Figure 8.10 shows the representation of the captured functional constraints in the Booch methodology. Figure 8.9 shows that the functional requirements data is captured for each product range in the Product Range Model.

Initial product definition data captures those product functions that if addressed lead to the instantiation of geometry that is a prerequisite for evaluation of other product functions. Figure 8.11 shows how the captured 'initial product definition' product functions be represented in the Booch methodology. The figure shows that the identity of initial product definition functions is captured plus any precedence between initial product definition functions. It can be seen that this data is captured for each product range in the Product Range Model.

8.2.4. Product range data on example product ranges.

This section presents product examples of the data structures considered to be required in the Product Range Model, as described in section 8.2.3. Example data structures are presented using the Booch methodology and EXPRESS. The full Booch representation of pro–

Project: ron
Title: Class Diagram: Form_function_relations_data
Printed on: Wednesday, November 29, 1995
Printed by: enrjvl

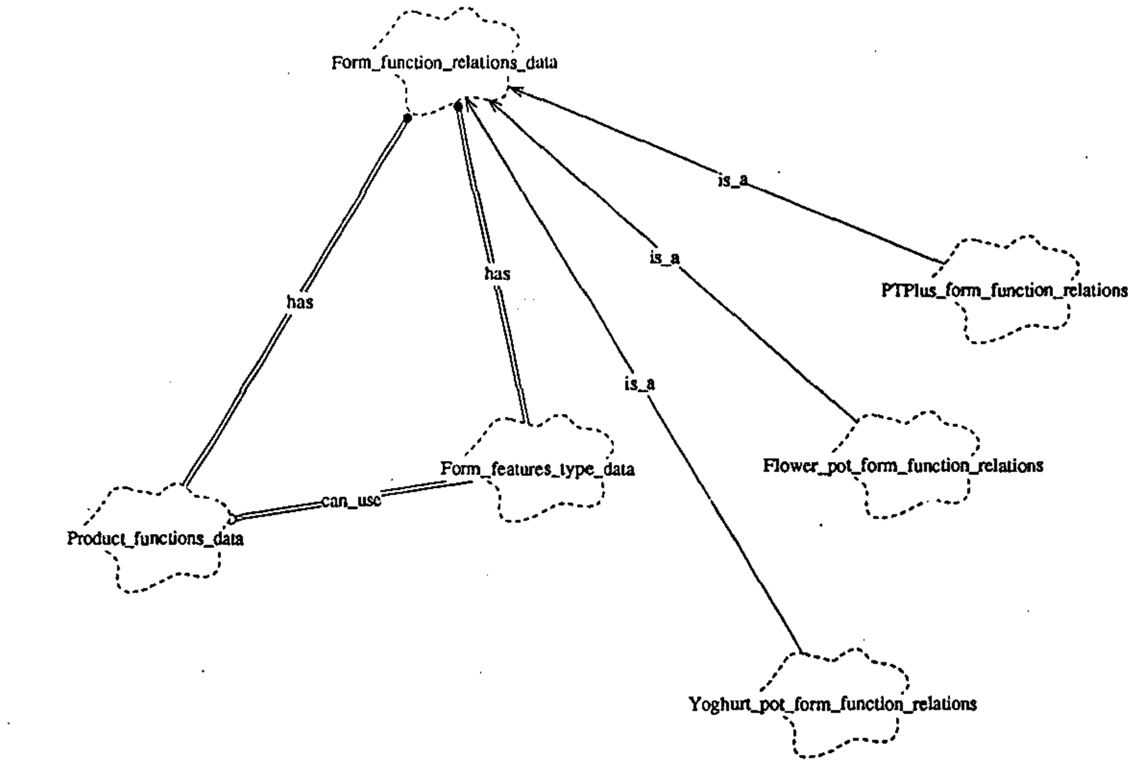


Figure 8.8. The Booch representation of the general structure of form/function relations data.

Project: ron
Title: Class Diagram: Functional_requirements
Printed on: Tuesday, November 28, 1995
Printed by: enrjvl

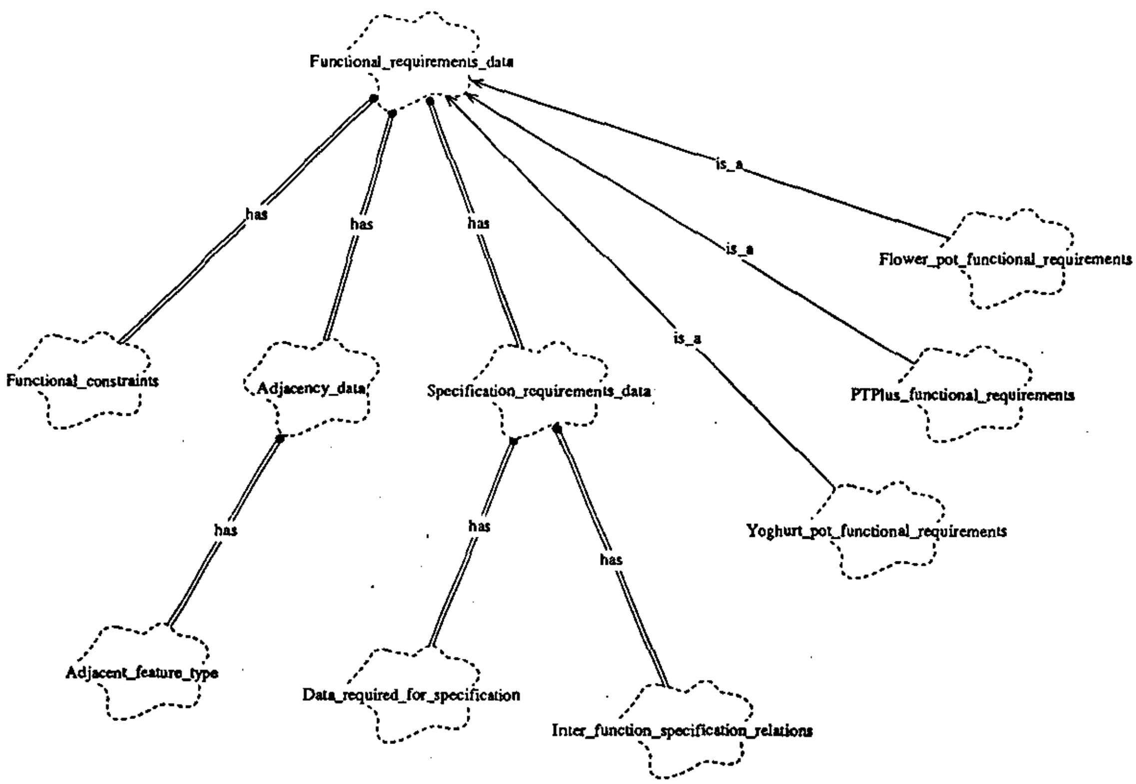


Figure 8.9. The Booch representation of the general structure of functional requirements data, minus functional constraints.

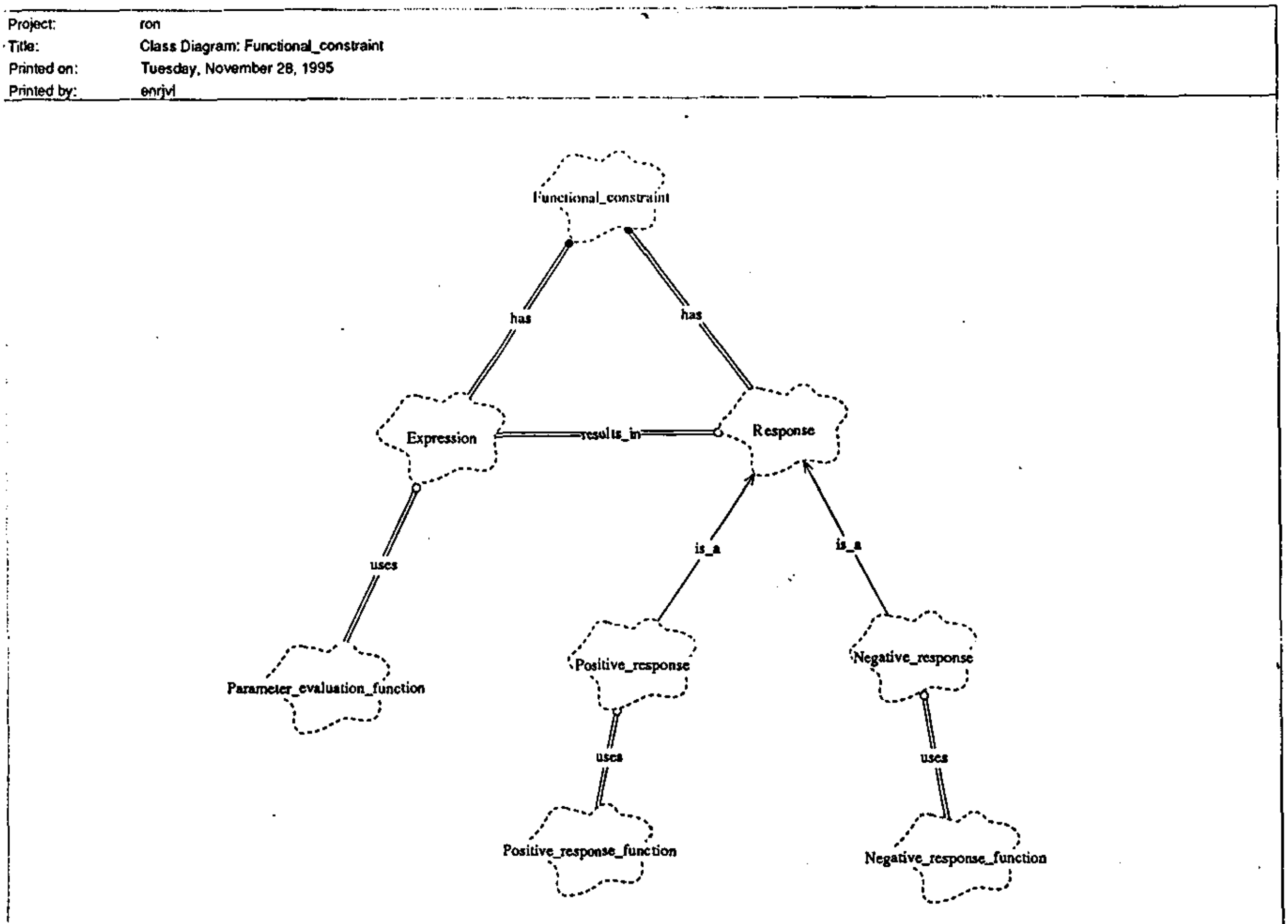


Figure 8.10. The Booch representation of the general structure of functional constraints data.

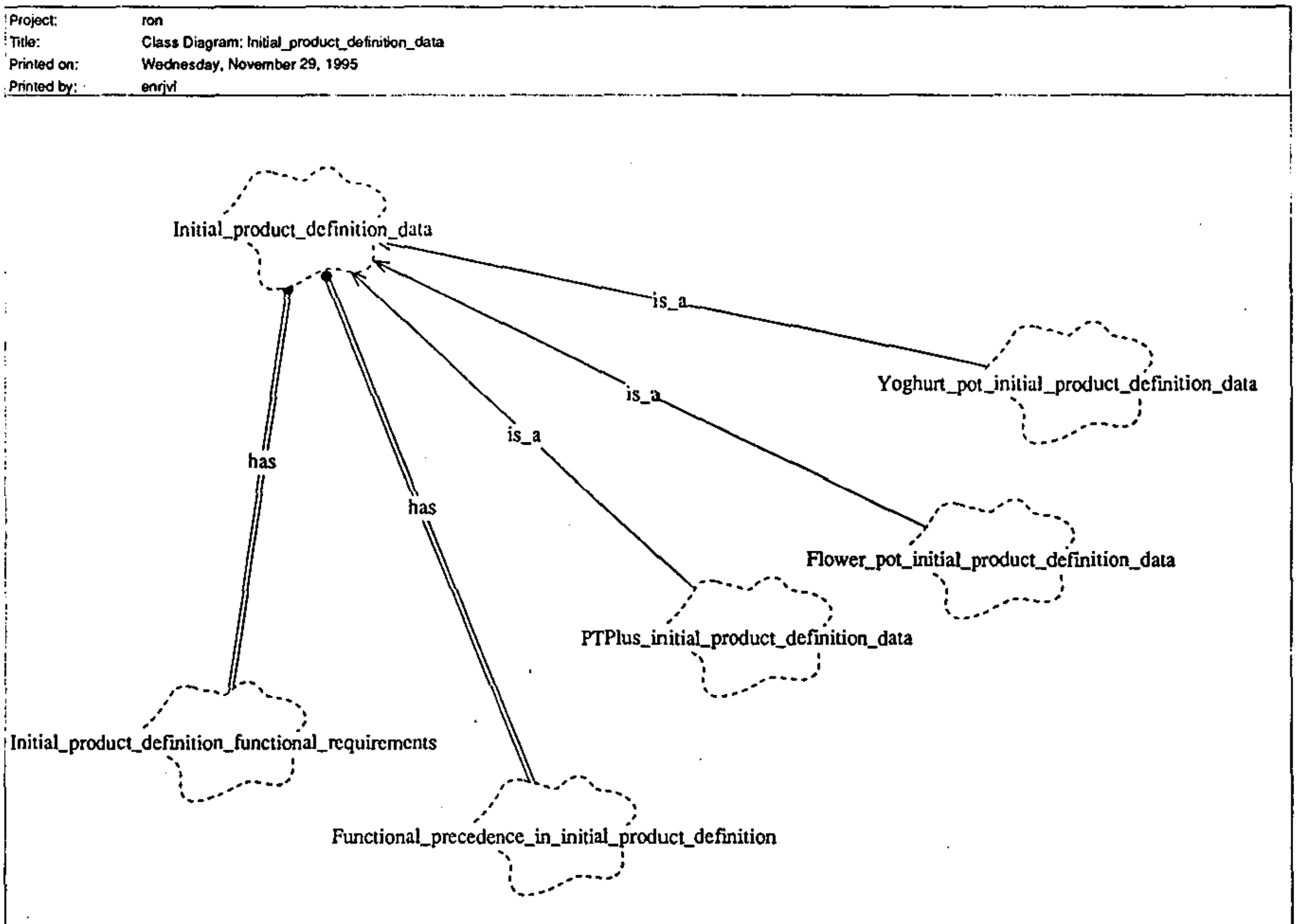


Figure 8.11. The Booch representation of the general structure of initial product definition data.

duct range data in the Product Range Model is shown in Appendix 5, and the full EXPRESS representation is shown in Appendix 6.

Examples of *form/function relations data* can be seen in Figure 8.7. It can be seen from the figure that for example a PTPlus product has seven product functions, each if which has one choice of form feature type which can be used to achieve the function. A flower pot has eight product functions of which the drainage clearance function has a choice of two form feature types which can be used to achieve the function. Similarly the insert destack function of a yoghurt pot has a choice of two form feature types that can be used to achieve the function. This captured form/function relations data can be represented using the Booch methodology as shown in Figures 8.12 and 8.13 for the examples of the PTPlus and flower pot product ranges. Figure 8.14 shows a detailed representation of the form/function relations data for a PTPlus and Flower pot product range in the EXPRESS language.

An example of *functional requirements data* is the 'break in torsion' product function of a PTPlus. Figure 8.15 shows example product functions including 'break in torsion' of the PTPlus tamper evident ring and their specifications. The break in torsion function relates to the breakage of the bridges between the top and bottom side walls if the seal is broken on the jar, thereby indicating air contamination of the contents. The specification data shown for break in torsion relates to the torsion at which the bridges should fail. If the torque is too small tampering may be erroneously indicated, and too large a torque means that the bridges fail in elongation rather than shear. In the latter case the seal may be broken without detection. The functional constraints of the break in torsion product function are shown in Figure 8.16. Obviously the adjacent form feature type that is relevant to functionality is a side wall. Figure 8.16 a) shows that there is a functional constraint on the position of spaced bosses used to fulfil the break in torsion function. The position must be such that full contact is maintained between a side wall and the spaced bosses, and such that a second side wall can have full contact with the spaced bosses without the two side walls touching. Figure 8.16 b) shows that a functional constraint exists on the group diameter of a spaced bosses feature to avoid a drastic reduction in the torque at failure. The group diameter of

Project:	ron
Title:	Class Diagram: PT_functional_requirements_form
Printed on:	Wednesday, November 29, 1995
Printed by:	enrjvl

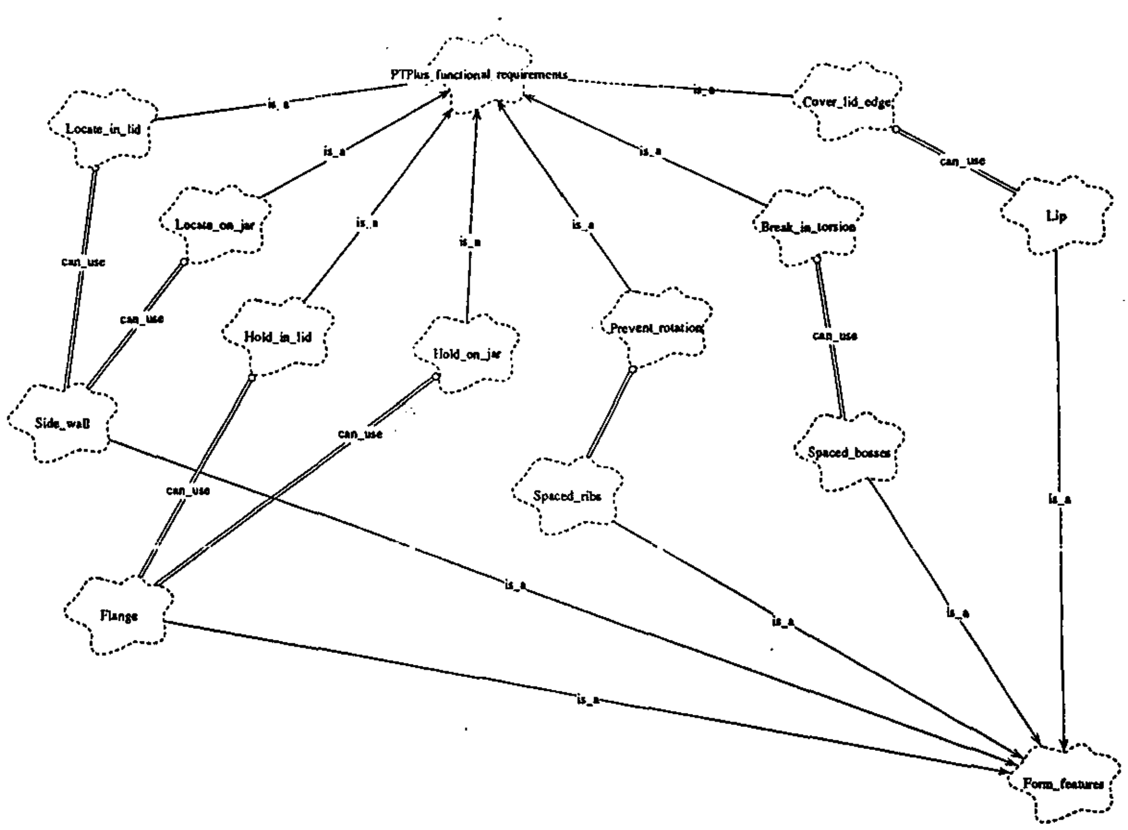


Figure 8.12. The Booch representation of form/function relations data for PTPlus example product.

Project:	ron
Title:	Class Diagram: FP_functional_requirements_form
Printed on:	Wednesday, November 29, 1995
Printed by:	enrjvl

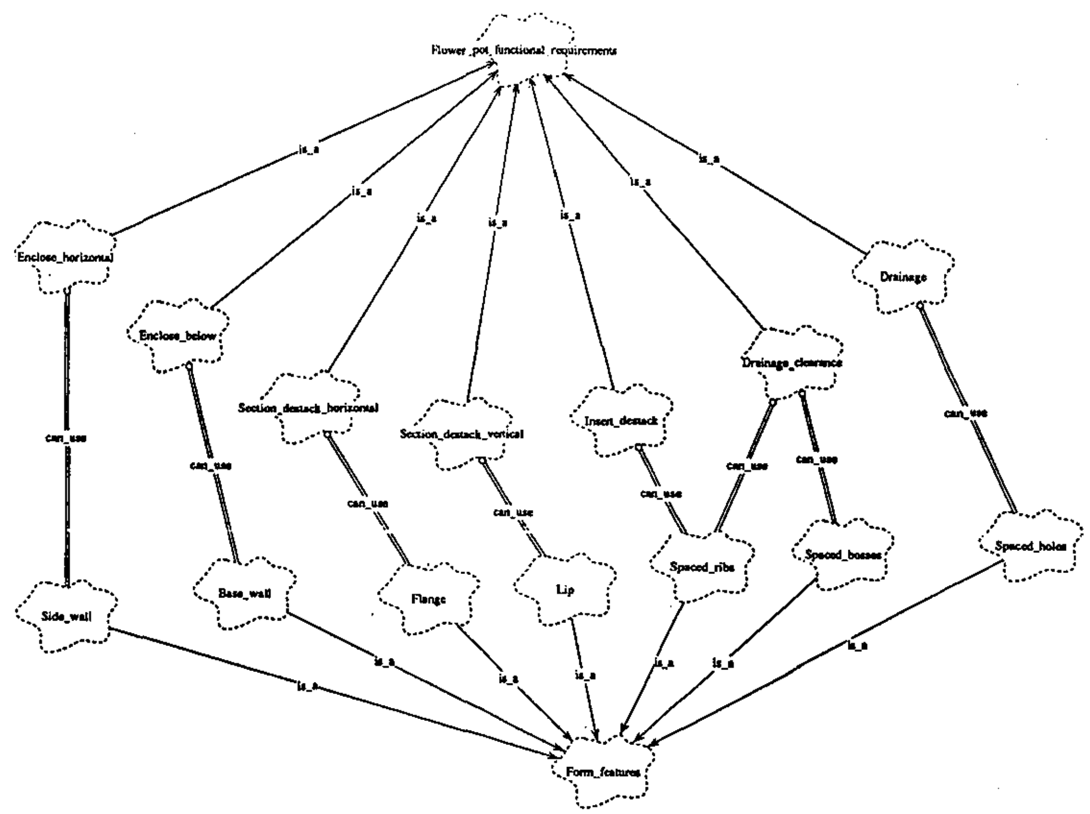


Figure 8.13. The Booch representation of form/function relations data for flower pot example product.

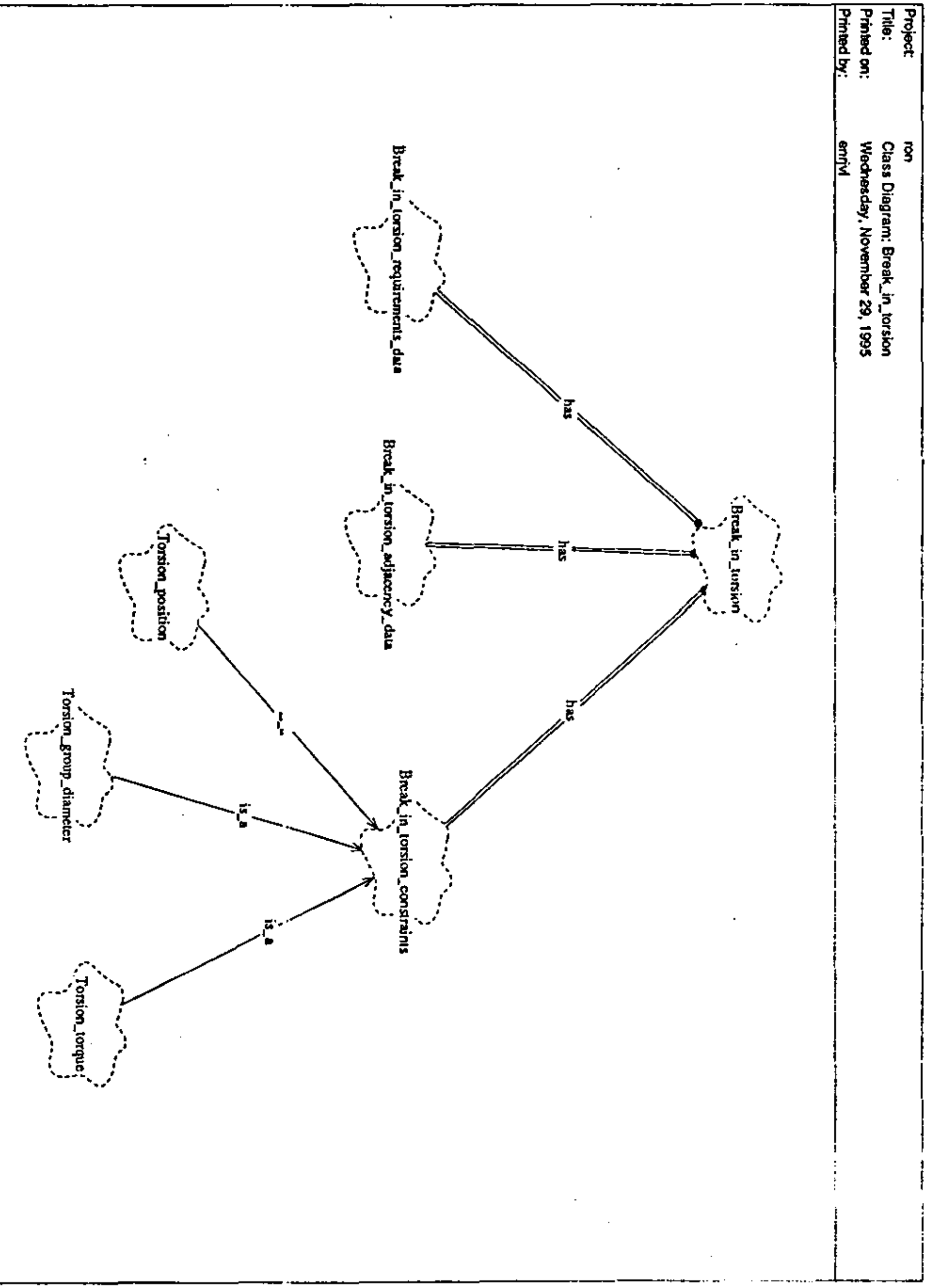


Figure 8.17. The Booch representation of functional constraints data for 'break in torsion' function on PTPlus example product.

ENTITY PTPlus_form_function_relations
SUBTYPE OF (Form_function_relations_data);
RULE PTPlus_form_function_relations FOR
(PTPlus_form_function_relations);
IF functional_requirement == Locate_in_lid THEN
choice of form == Side_wall.
IF functional_requirement == Break_in_torsion THEN
choice of form == Spaced_bosses.
IF functional_requirement == Locate_on_jar THEN
choice of form == Side_wall.
IF functional_requirement == Hold_in_lid THEN
choice of form == Flange.
IF functional_requirement == Hold_on_jar THEN
choice of form == Flange.
IF functional_requirement == Prevent_rotation THEN
choice of form == Spaced_ribs.
IF functional_requirement == Cover_lid_edge THEN
choice of form == Lip.
END_RULE;
END_ENTITY;

ENTITY Flower_pot_form_function_relations
SUBTYPE OF (Form_function_relations_data);
RULE Flower_pot_form_function_relation
FOR (Flower_pot_form_function_relations);
IF functional_requirement == Enclose_horizontal THEN
choice of form == Side_wall.
IF functional_requirement == Enclose_below THEN
choice of form == Base_wall.
IF functional_requirement == Insert_destack THEN
choice of form == Spaced_ribs.
IF functional_requirement == Section_destack_horizontal THEN
choice of form == Flange.
IF functional_requirement == Section_destack_vertical THEN
choice of form == Lip.
IF functional_requirement == Drainage THEN
choice of form == Spaced_holes.
IF functional_requirement == Drainage_clearance THEN
choice of form == Spaced_ribs or Spaced_bosses.
END_RULE;
END_ENTITY;

Figure 8.14. EXPRESS representation of form/function relations data for case study Yogurt pot and PTPlus.

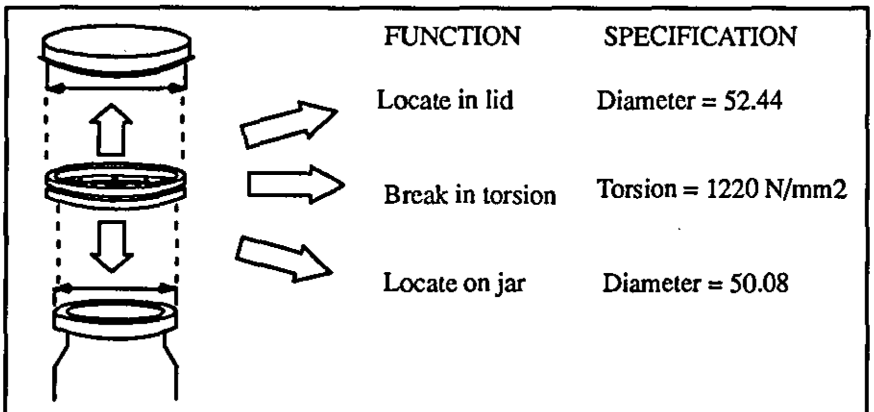


Figure 8.15. Example product functions and specifications of PTPlus product range.

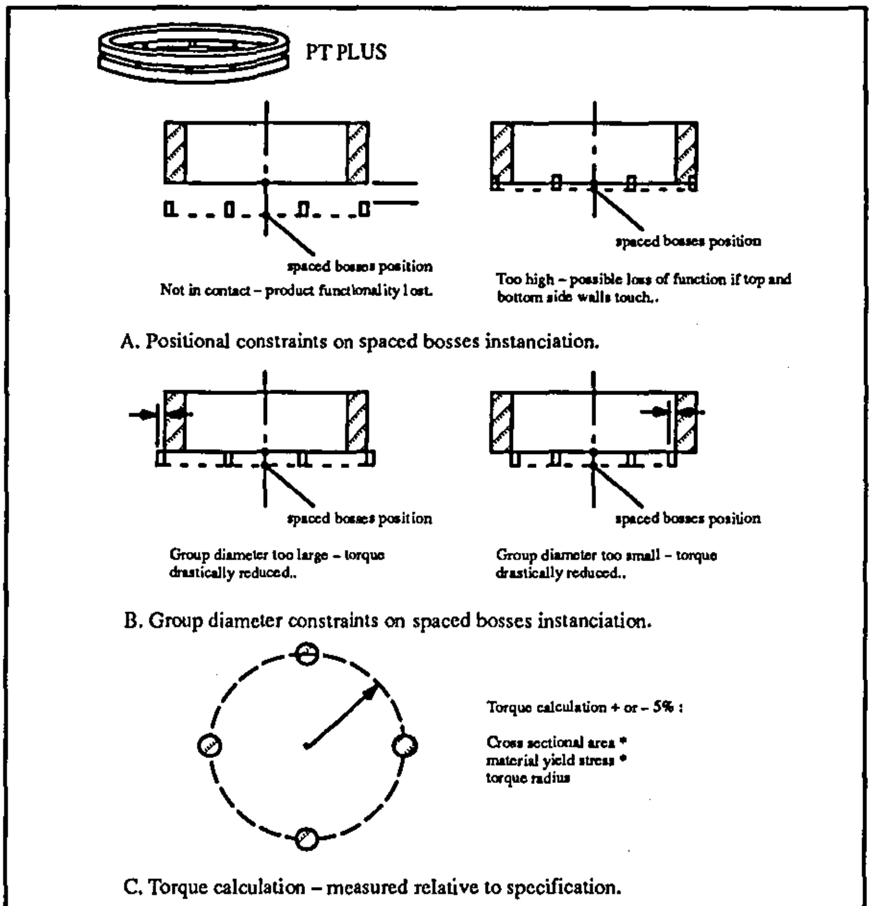


Figure 8.16. Example 'break in torsion' functional constraints on PTPlus product range.

spaced holes must be such that contact is maintained between a side wall and the full cross section of each group member. Figure 8.16 c) shows the breakage torque calculation based on the cross sectional area of the spaced bosses. The target is that a breakage torque be within + or – five percent of that specified by the designer.

The captured functional requirements data of the break in torsion function on a PTPlus can be represented using the Booch methodology as shown in Figure 8.17, which shows the constraints on position, group diameter and torque. Figure 8.18 shows a detailed representation of the captured functional requirements data for the break in torsion function, using the EXPRESS language. RULE Break_in_torsion_adjacency_data FOR (Break_in_torsion) captures the adjacent form feature type that is relevant to functionality as a side wall. RULE Torsion_position FOR (Break_in_torsion) captures the functional problems of erroneously positioned spaced bosses. RULE Torsion_group_diameter FOR (Break_in_torsion) captures the functional problems of spaced bosses where the group diameter prevents full contact of group members with a side wall. RULE Torsion_torque FOR (Break_in_torsion) captures the functional problems of a spaced bosses cross sectional area that provides too much or too little torque for the break in torsion function.

An example of functional requirements data where *inter function specification relations data* is important is the 'prevent rotation' product function on the PTPlus product range. As shown in Figure 8.4 a prevent rotation function aims to ensure that the spaced bosses fulfilling the break in torsion function break in shear, rather than elongation via rotation in the lid. The specification data for the prevent rotation product function relates to the minimum torque in order to compensate for the torque applied to the spaced bosses fulfilling the break in torsion function during shear. In order to achieve the overall functionality of a PTPlus, the minimum torque specification for the prevent rotation function must be greater than the torque specified for the break in torsion function. If this is not the case, fulfilment of the prevent rotation function may not achieve the overall product functionality. A safety margin of twice the torque specified for the break in torsion function is recommended in the prevent rotation function. Figure 8.9 shows the captured *inter function specification relations data* as a part of the specifications requirements data using the Booch methodology, and Figure 8.19 shows part of a detailed representation of the captured prevent rotation product function of a PTPlus product range using the EXPRESS language. RULE Prevent_rotation_requirements_data FOR (Prevent_rotation) captures the problems of a lesser torque specification than that for the break in torsion product function, and captures the recommended safety margin of two times the break in torsion specification.

```
ENTITY Break_in_torsion
SUBTYPE OF ( PTPlus_functional_requirements, IPD_functional_requirements);
adjacent_feature_type : STRING;
torsion_req : dimension;
connect_boss : Spaced_bosses;
connect_wall : Side_wall;
WHERE
min_group_dia : = connect_wall.inner_dia + connect_boss.boss_dia;
max_group_dia : = connect_wall.inner_dia + connect_wall.thickness – connect_boss.boss_dia;
yield : = 29.0;
boss_area : = 3.1416*SQ(connect_boss.boss_dia/2.0);
total_area : = boss_area*connect_boss.boss_numb;
Force : = yield*total_area;
Torque_calculation : = Force*connect_boss.axis_dia/2.0;
Reverse_force : = Torque_calculation/(connect_boss.boss_dia/2.0);
Reverse_total_area : = Reverse_force/yield;
Reverse_boss_area : = Reverse_total_area/connect_boss.boss_numb;
recom_diameter : = SQRT(Reverse_boss_area/3.1416);
recom_boss_no : = FLOOR(Reverse_total_area/boss_area);
RULE Break_in_torsion_requirements_data FOR (Break_in_torsion);
Require breakage torsion (Nmm) == torsion_req.
END_RULE;
RULE Break_in_torsion_adjacency_data FOR (Break_in_torsion):
adjacent_feature_type == Side_wall.
END_RULE;
RULE Torsion_position FOR (Break_in_torsion);
IF (connect_boss.position[2] + connect_boss.height) < connect_wall.position[2] THEN
Top of boss grouping not in contact with wall.
Lost product functionality. Advise reposition
connect_boss.position[2] to (connect_wall.position[2] –
connect_boss.height).
IF (connect_boss.position[2] + connect_boss.height) > connect_wall.position[2] THEN
Top of boss grouping is higher than underside of
side wall. Possible loss of function if boss grouping
is contained in side wall. Advise relocate connect_boss.
position[2] to connect_wall.position[2] – connect_insert.height.
END_RULE;
RULE Torsion_group_diameter FOR (Break_in_torsion);
IF (connect_boss.axis_dia + connect_boss.boss_dia) > ( connect_wall.inner_dia +
2.0*connect_wall.thickness) THEN
Group outer diameter greater than that of supporting
wall. Breakage torsion drastically reduced from that
intended and cannot be evaluated. Advise decrease
group diameter to be in full contact with supporting
wall ie min_group_dia <= connect_boss.axis_dia <=
max_group_dia.
IF (connect_boss.axis_dia – connect_boss.boss_dia) < connect_wall.inner_dia THEN
Group inner diameter smaller than that of supporting
wall. Breakage torsion drastically reduced from that
intended and cannot be evaluated. Advise increase
group diameter to be in full contact with supporting
wall ie min_group_dia <= connect_boss.axis_dia <=
max_group_dia.
END_RULE;
RULE Torsion_torque FOR (Break_in_torsion);
IF Torque_calculation < torsion_req THEN
Feature cross sectional area not large enough to
provide failure at the torque specified. Tampering
with the jar may be indicated erroneously. The lid
can be removed from the jar too easily. Advise
increase connect_boss.boss_numb to recom_boss_no
or increase connect_boss.boss_dia to recom_diameter.
IF Torque_calculation > torsion_req THEN
Feature cross sectional area too large to provide
failure at the torque specified. Difficulty in removing
the lid from the jar. Advise decrease connect_boss.boss_numb
to recom_boss_no or decrease connect_boss.boss_dia
to recom_diameter.
END_RULE;
END_ENTITY;
```

Figure 8.18. EXPRESS representation of break in torsion function on PTPlus.

```
ENTITY Prevent_rotation
SUBTYPE OF ( PTPlus_functional_requirements, IPM_functional_requirements);
adjacent_feature_type : STRING;
min_torque : dimension;
connect_rib : Spaced_ribs;
connect_wall : Flange;
connect_torsion : Break_in_torsion;
WHERE
yield : = 29.0;
rib_length : = (connect_rib.outer_dia - connect_rib.inner_dia)/2.0;
torque_distance : = connect_rib.inner_dia + (rib_length/2.0);
rib_area : = rib_length*connect_rib.width;
total_area : = rib_area*connect_rib.rib_numb;
force : = yield*total_area;
torque_calculation : = force*torque_distance;
reverse_force : = min_torque/torque_distance;
reverse_total_area : = reverse_force/yield;
reverse_rib_area : = reverse_total_area/connect_rib.rib_numb;
recom_width : = reverse_rib_area/rib_height;
recom_rib_no : = FLOOR(reverse_total_area/reverse_rib_area);
RULE Prevent_rotation_requirements_data FOR (Prevent_rotation);
Require minimum torque at which rotation
can occur (Nmm) == min_torque.
IF min_torque < connect_torsion.torsion_req THEN
min_torque smaller than torque for break_in_torsion.
Rotation of component will occur - bridges will fail by
elongation. Seal can be broken without detection. Advise
specify min_torque higher than that for break_in_torsion
function, recommended safety margin is times 2 ==
(2.0*connect_torsion.torsion_req).
IF (min_torque > connect_torsion.torsion_req) && (min_torque <
2.0*connect_torsion.torsion_req)THEN
min_torque smaller than recommended safety margin above
torque for break_in_torsion function. Possible rotation of
component - bridges will fail by elongation. Possible non
indication of broken seal. Advise specify min_torque higher
than that for break_in_torsion function, recommended safety
margin is times 2 == (2.0*connect_torsion.torsion_req).
END_RULE;
```

Figure 8.19. Part of EXPRESS representation of prevent rotation function on PTPlus.

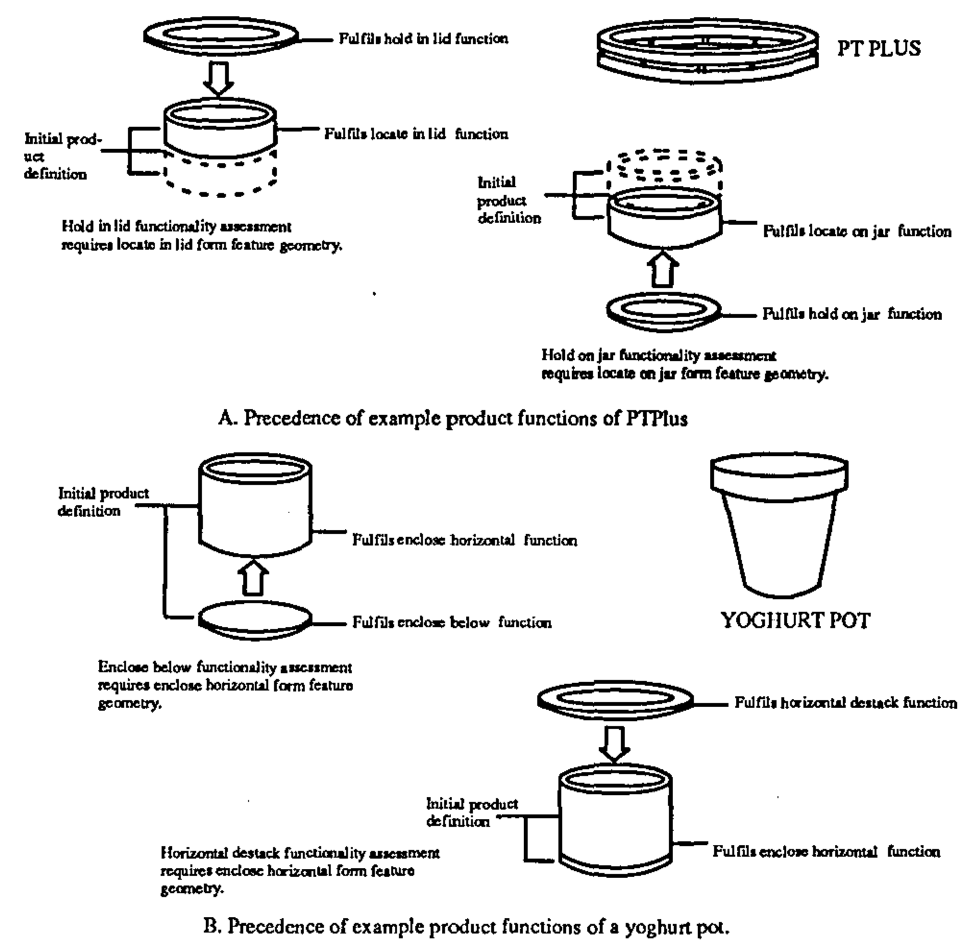


Figure 8.20. Examples of precedence between product functions on PTPlus and Yoghurt pot product ranges.

An example of *initial product definition data* is the three product functions that must be addressed on a PTPlus product range as a prerequisite to the others. The initial product definition functions of a PTPlus are shown in Figure 8.15 and are locate in lid, break in torsion, locate on jar. The other functions associated with PTPlus are hold in lid, hold on jar, prevent rotation and cover lid edge. Figure 8.20 a) shows examples of the product functions on a PTPlus other than those of the initial product definition, and it can be seen that the instantiation of form features fulfilling the initial product definition functions are a prerequisite to the functionality evaluation of the remaining product functions because the spatial relationships necessary to fulfil the non initial product definition functions cannot be evaluated.

Furthermore there may be product functions within the initial product definition that are a prerequisite to other initial product definition functions. For example as shown in Figure

```
ENTITY PTPlus_initial_product_definition_data
SUBTYPE OF (Initial_product_definition_data);
RULE Initial_product_definition FOR (PTPlus);
Initial product definition =
Locate_in_lid THEN
Break_in_torsion THEN
Locate_on_jar
END_RULE;
END_ENTITY;

ENTITY Yoghurt_pot_initial_product_definition_data
SUBTYPE OF (Initial_product_definition_data);
RULE Initial_product_definition FOR (Yoghurt_pot);
Initial product definition =
Enclose_horizontal THEN
Enclose_below
END_RULE;
END_ENTITY;
```

Figure 8.21. Initial product definition data of PTPlus and Yoghurt pot product ranges.

8.20 b), the initial product definition functions of a yoghurt pot are 'enclose horizontal' and 'enclose below', and product functions such as 'destack horizontal' cannot be evaluated without the forms fulfilling the initial product definition functions. However, it can be seen from the figure that in the absence of a form feature instantiation for the enclose horizontal function, a form feature fulfilling the enclose below function cannot be evaluated for functionality because the spatial relationship necessary to fulfil the enclose below function cannot be evaluated. Initial product definition data therefore captures the order of addressing product functions required as part of the initial product definition as well as their identity. Figure 8.11 shows the general structure of initial product definition data using the Booch methodology, and Figure 8.21 shows a detailed representation of the captured initial product definition data of the PTPlus product range and the yoghurt pot product range using the EXPRESS language. RULE Initial_product_definition FOR (PTPlus product range) and RULE Initial_product_definition FOR (Yoghurt pot product range) capture the product functions that must be included in the initial product definition and the order in which these functions should be addressed.

8.3. MANUFACTURING MODEL.

A Manufacturing model is required to capture injection moulding process constraints independent of any specific product. This section describes the authors investigation into the detailed structure of a Manufacturing model to support the operation of an injection moulding strategist. In particular a Manufacturing model structure has been constructed which can support the three dimensional spatial relationships of the injection moulding process. This section relates to the work in Chapter 6 which identified the feature types for each viewpoint in injection moulding design, and investigates how the process constraints associated with each viewpoint can be applied to the feature types therein and the way they can be captured in a Manufacturing model structure.

8.3.1. General Manufacturing model structure.

Figure 8.22 shows the conceptual structure of a Manufacturing model as defined by Al–Ashaab (1994). The Figure shows the injection moulding representation as one process within a range of processes represented in a Manufacturing model. The injection moulding process representation captures information describing the capabilities and constraints of the process relating to three main classes of object: mouldability, injection moulds and machine elements. The general structure of Al–Ashaab has been used by the author. However this structure was not good enough in detail to provide information support to interacting injection moulding strategist applications. To explore an injection moulding strategist fully there was a need to re–evaluate the structure of Al–Ashaab in order to provide the appropriate information structure. The full Booch representation of the detailed Manufacturing model structure identified in this thesis is shown in Appendix 3.

The mouldability class captures the general constraints imposed on any product to be manufactured using injection moulding. The injection mould elements capture all the different ways that injection moulds can be configured. The injection mould elements are shown in Figure 8.23. This representation has all the elements of a typical injection mould but it should be noted that this is not a mould for a specific product.

The machine elements have not been considered in the work reported in this thesis, as this was not considered necessary to explore information support for an injection moulding strategist.

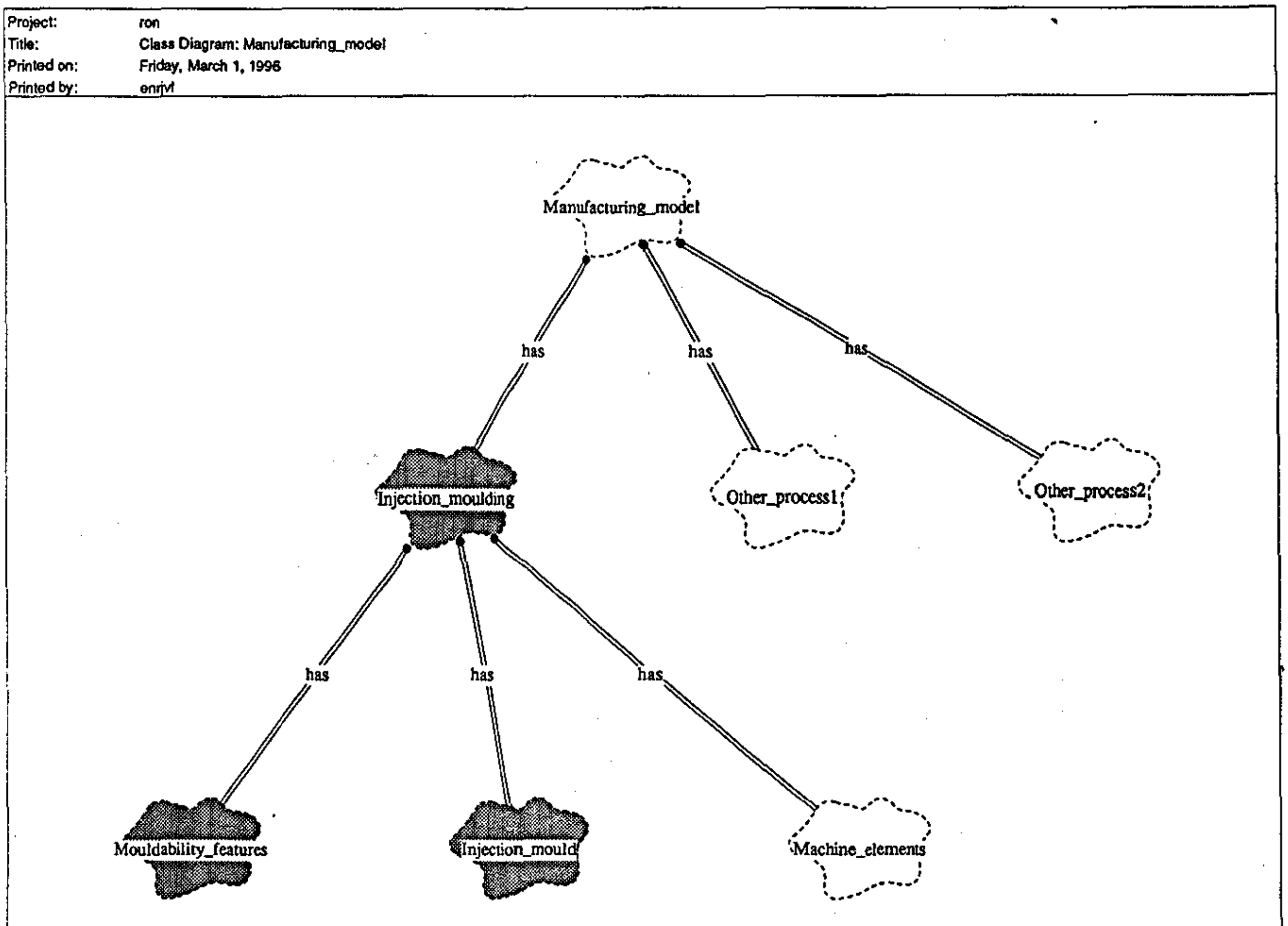


Figure 8.22. Conceptual structure of a Manufacturing model in Booch.

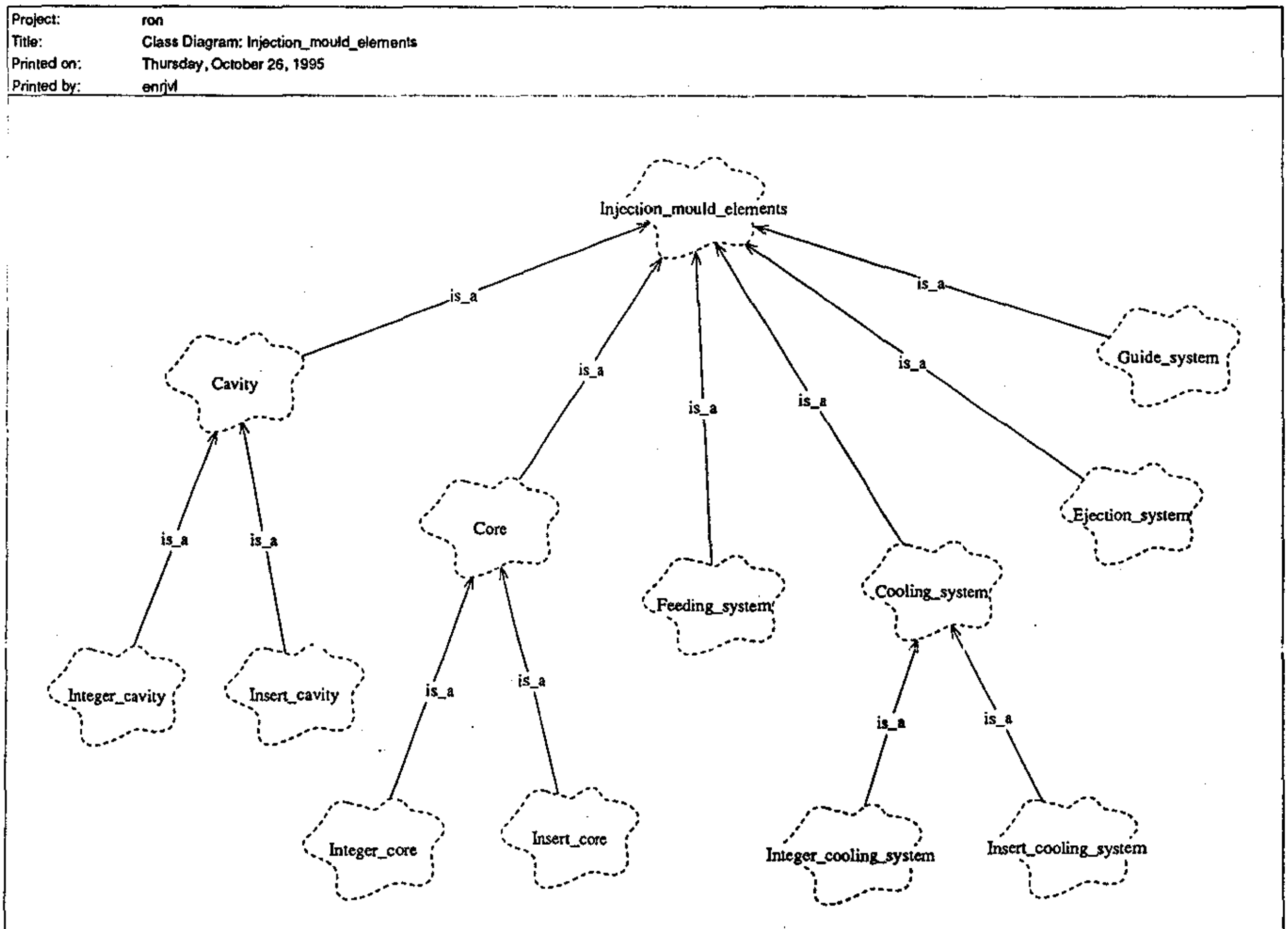


Figure 8.23. The Booch representation of the injection mould elements.

8.3.2. Mouldability.

8.3.2.1. Mouldability process constraints.

The process constraints that apply to mouldability are well known and can be listed as; i) Shrinkage and warping, ii) Ejection, iii) Stress concentrations, iv) Weld lines, v) Surface defects. The types of problem which can occur and the significant factors which need to be considered for each of these five are described below:

i) Shrinkage – A major characteristic of the injection moulding process is that it is capable of producing only thin walled products. A wall thickness should therefore be less than or equal to a maximum value which is recommended by the material suppliers. Too thick a wall may cause shrinkage to occur as the plastic solidifies in the mould, or cause warpage or bending of the product. Uniform thickness of product walls is recommended where possible, otherwise a gradual change of wall thickness is very important to avoid warping due to shrinkage.

ii) Ejection – All parts of a product require draft angles parallel to the axis of the injection machine in order to facilitate removal of the product from the mould. A component with no draft angles will become stuck in the mould.

iii) Stress concentrations – These occur in a mould where sharp corners exist or due to sudden changes in section thickness which cause differential shrinkage rates. This can lead to failure of the product. Stress concentrations can be avoided by the provision of radii on the product to avoid sharp corners and by ensuring uniform thickness of product walls where possible.

iv) Weld lines – These occur where interference with the material flow exists, such as a core to make a hole. Molten plastic material in a mould flows around an obstruction by splitting into two streams and then enveloping it and rejoining at the other side. Where the two material streams meet they may not properly fuse together due to cooling, leading to the occurrence of a weld line which is a weak point in the product. To avoid weld lines the position of such obstructions in relation to the gate must be considered, ie the hotter the material at the location the less likely a weld line is to occur. Also thin streams around an obstruction should be avoided, as these accelerate the cooling process.

iv) Surface defects – Too large a radius applied to a product should be avoided as this can cause a surface defect due to turbulent flow of material in a mould. If too thick a wall section exists or the size of a protrusion produces a thick section at the intersection with a wall, sink marks can appear on the surface of a product due to shrinkage.

The types of mouldability features defined by the author have been described in Chapter 6. The following section describes how the process constraints can be applied to these features and the way in which they can be captured in a Manufacturing model structure.

8.3.2.2. Process constraints applied to mouldability features.

Typical ways of avoiding mouldability problems are indicated in Figure 8.24, which shows design rules linked to mouldability features, Dixon (1988), Al–Ashaab (1994). The set of mouldability features defined in Chapter 6 are wall, rib, solid boss, hollow boss, hole, gate, taper, protrusion blend and corner blend. Examination of each of these features in turn has enabled the author to ascertain how process constraints can be applied. Process constraints can be feature specific or can be feature interaction constraints. The attributes of example mouldability feature types are now described:

<u>Wall feature</u>: The attributes of a *wall* feature that relate to the wall feature alone are wall thickness, wall taper, wall gating. These attributes are described below:

i) Wall thickness – A wall thickness should be less than or equal to a maximum value which is recommended by the material suppliers. Too thick a wall may cause shrinkage to occur

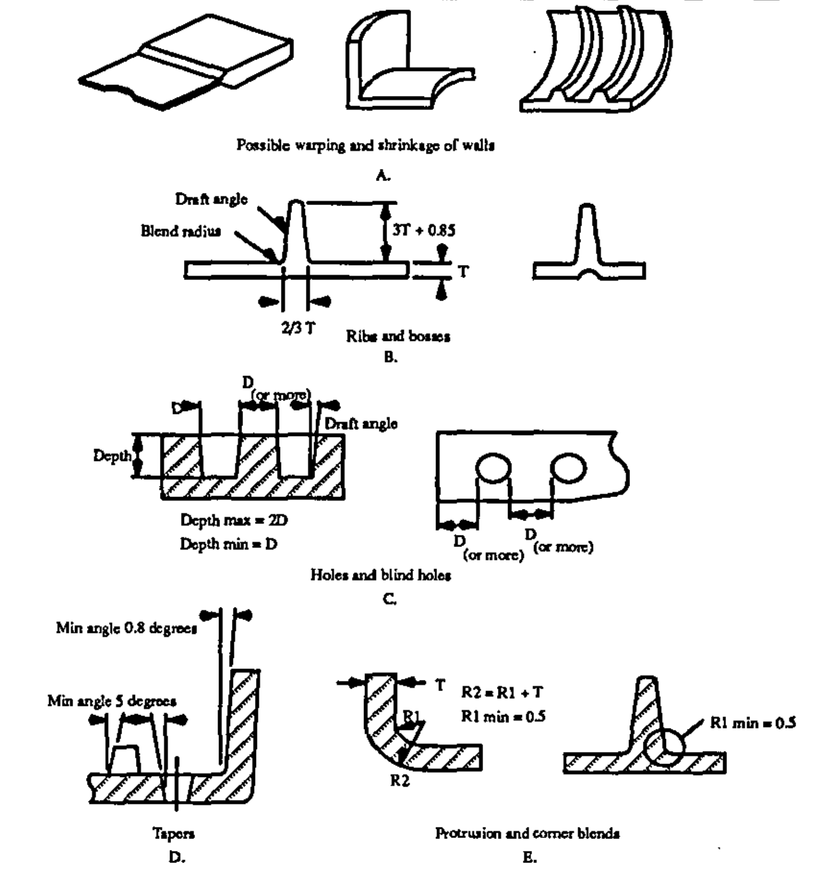


Figure 8.24. Recommended design rules to avoid mouldability problems.

as the plastic solidifies in the mould, causing sink marks on the surface or causing warpage or bending of the product, WEKA (1991) Examples of these problems are shown in Figure 8.24. a). A designer may use ribs to enable a reduction in wall thickness without losing strength.

ii) Wall taper – Wall features require a taper parallel to the axis of the injection machine in order to facilitate removal of the product from the mould.

iii) Wall gating – In order to consider the gating of a product the designer must have the opportunity of instantiating a gate on a wall.

With respect to the attributes relating to *wall feature interactions* with other features, there is a significant difference in the approach of the author and that of previous work, eg Al–Ashaab (1994), Dixon (1988). Previously wall features were considered the 'base' feature of injection moulded products, ie all wall features are instantiated on the product before

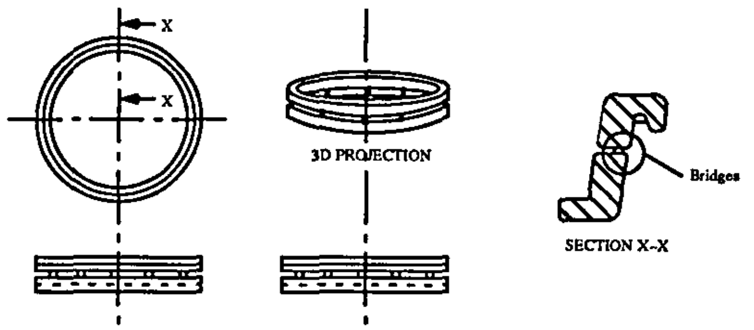


Figure 8.25. The case study component PTPlus.

'add ons' such as protrusions and holes are considered. Consequently the attributes of a wall feature with respect to relationships with other features, have only been considered for interaction with adjacent walls. However due to the interaction of functional design with mouldability this approach is no longer appropriate. An example is the PTPlus product shown in Figure 8.25. A critical aspect of this product is the bridges illustrated in the Figure, which would be overlooked by the approach of using walls as 'base' features. In this thesis a translation process is used to instantiate mouldability features. Therefore the order in which mouldability types are instantiated is driven by the alternative design viewpoint of functionality, and cannot be predicted. Thus the design rules of a wall feature have to be captured not only with respect to adjacent walls, but also with respect to adjacent protrusions.

The attributes of a wall feature which provide a relationship to adjacent features are:

i) Relative wall thickness – Uniform thickness of product walls is recommended where possible, otherwise a gradual change of wall thickness is very important to avoid warping or surface finish problems due to shrinkage. Also stress concentrations can occur in the product due to differential shrinkage.

ii) Wall blends – The intersection between a wall and another wall or between a wall and a protrusion requires a blend radius or radii to avoid stress concentrations at the sharp corners and possible surface defects due to turbulent flow in the mould.

The following example shows how the EXPRESS language complements the Booch representation in capturing the Manufacturing model structure. Figure 8.26 shows how the mouldability constraints of a wall feature can be represented in the Booch methodology, showing a 'Use_rib' attribute. This is a recommendation for the possible use of ribs to alleviate the strength problems of a reduced wall thickness.

The Booch representation captures the attribute but cannot capture when a recommendation for rib usage should be made. As shown in Figure 8.27 a detailed representation of the mouldability of a wall entity in the EXPRESS language captures this data in RULE Use_rib FOR (Wall). Also shown in the figure are two constraints on the thickness attributes of a wall entity; in RULE wall_thickness FOR (Wall) and RULE Relative_thickness FOR (Wall, against adjacent wall). The first rule represents the mouldability problems of a wall feature whose thickness attribute is greater than the max_thickness value, which is defined in the WHERE clause as 5.0 mm. The relative thickness rule represents the mouldability problems of a wall thickness attribute which has a value other than that of the adjoining wall, where the difference in the thickness attributes of the two walls is identified in the DERIVE clause.

Figure 8.27 shows three constraints relating to the requirement of secondary mouldability features to be added to a wall: RULE Wall_taper FOR (Wall, untapered) captures the mouldability requirement for a taper on a wall to avoid problems in removal of the part from the mould. RULE Wall_blend FOR (Wall, against adjacent wall or reinforcement) captures the mouldability requirement for blend radius or radii between a wall feature and an adjacent wall or reinforcement. If the adjacent mouldability type is a wall then a corner blend is required, otherwise a protrusion blend. RULE Wall_gating FOR (Wall, ungated) captures the mouldability requirement for a gate on a wall. A gate is not necessary on every wall in a product. Sufficient gating should be provided to ensure that adequate feeding is attained throughout the mould. The EXPRESS representation also shows that numbers are being put into the constraints for evaluation, which cannot be captured in Booch.

```
ENTITY Wall
SUBTYPE OF (Primary_mouldability_features);
Feature_name : STRING;
Feature_type : STRING;
Associated_form_name : STRING;
Associated_form_type : STRING;
thickness : dimension;
length : dimension;
height : dimension;
position : POINT3D;
orientation : POINT3D;
gated : STRING;
connect_wall : wall;
WHERE
max_thickness : thickness<=5.0;
DERIVE
thickness_difference : BOOLEAN := (thickness.NE.connect_wall.thickness);
RULE wall_thickness FOR (Wall);
IF thickness>max_thickness THEN
Possible shrinkage marks and component
warpage. Advise reduction in wall thickness
to a maximum of max_thickness.
END_RULE;
RULE Use_rib FOR (Wall);
IF thickness>max_thickness THEN
Advise the use of ribs to allow a
reduction in wall thickness.
END_RULE;
RULE Wall_taper FOR (Wall, untapered);
IF wall has no taper THEN
Problems with removal of part from
mould. Request that a Taper be created.
END_RULE;
RULE Relative thickness FOR (Wall, against adjacent wall);
IF thickness difference THEN
Possible feeding problems, component
warpage or surface finish problems,
stress concentrations in the component.
Advise change of thickness to
connect_wall.thickness.
END_RULE;
RULE Wall_blend FOR (Wall, against adjacent wall or reinforcement);
IF wall has no blend THEN
Stress concentrations in the component,
possible surface defects. Request that a
Blend be created.
IF adjacent mouldability type == Wall THEN
Blend type = corner.
IF adjacent mouldability type == Solid_boss OR Rib OR Hollow_boss THEN
Blend type = protrusion.
END_RULE;
RULE Wall_gating FOR (Wall, ungated);
IF wall has no gate THEN
Ask if a gate is to be created
on the wall.
END_RULE;
END_ENTITY;
```

Figure 8.27. EXPRESS representation of mouldability wall feature.

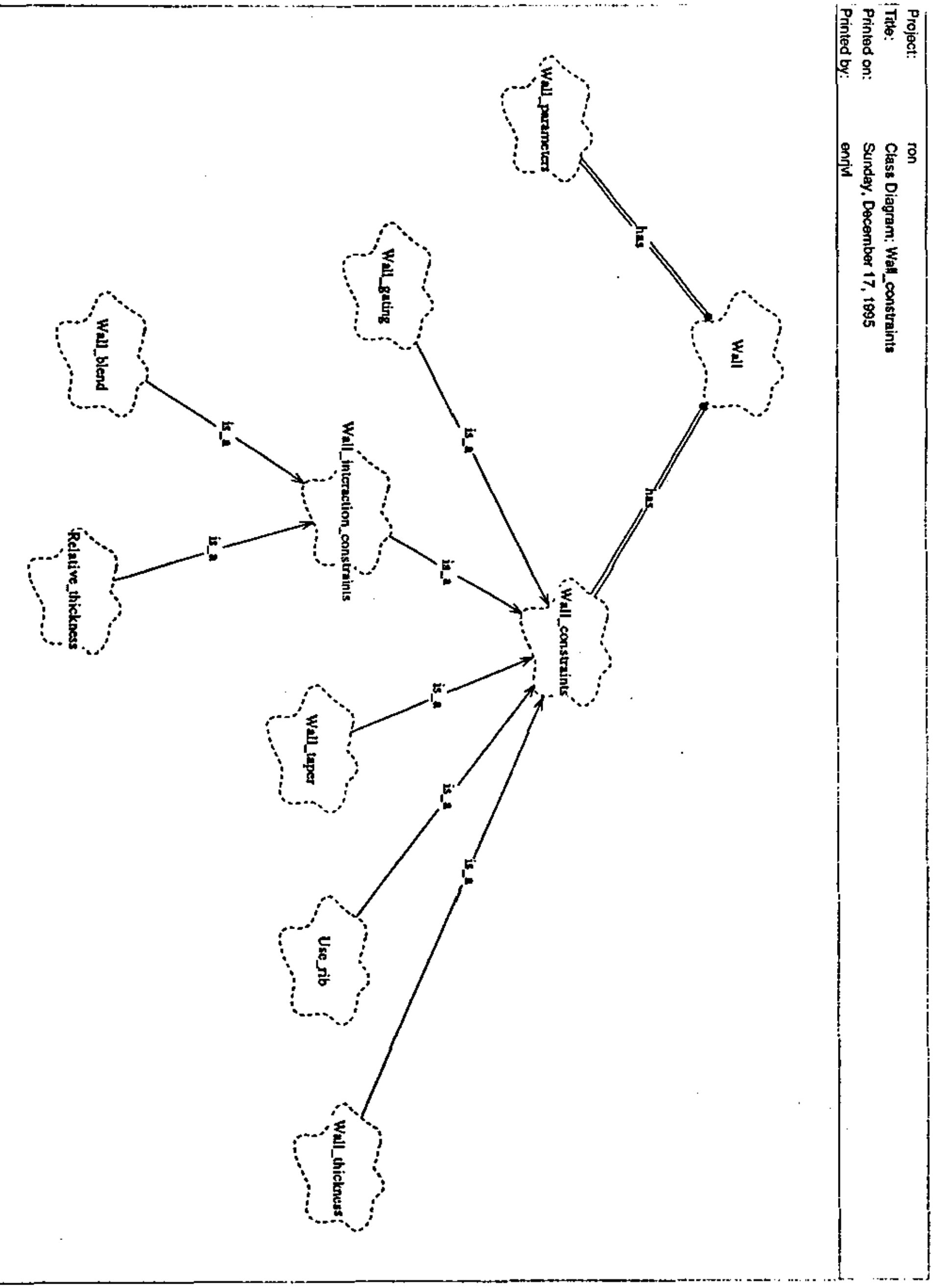


Figure 8.26. The Booch representation of wall feature constraints.

The above example has shown how the EXPRESS representation complements the Booch methodology. The remaining EXPRESS representations can be seen in Appendix 4.

Reinforcement features:

Mouldability ribs, solid bosses and hollow bosses can collectively be termed 'reinforcement' features, since they can all be used to reinforce product geometry to facilitate a reduction in wall thickness and the design rules are also identical. The attributes that relate to a mouldability *reinforcement* feature alone are:

i) Reinforcement orientation – The orientation of a reinforcement must be such that its height is measured parallel to the machine axis to avoid an overhang that may require a collapsing core or a split mould for ejection.

ii) Reinforcement taper – In order to facilitate ejection of the product from the mould a generous taper is essential.

The attributes relating to *reinforcement feature interactions* with adjacent wall features are reinforcement width, reinforcement height and reinforcement blend, as follows:

i) and ii) Reinforcement width and Reinforcement height – The width and height dimensions of a reinforcement are limited by those of the host wall. If the dimensions of a reinforcement exceed limits directly related to the thickness of the wall, then sink marks may occur on the opposite side of the wall to the reinforcement and component warpage may occur.

iii) Reinforcement blend – The intersection between a reinforcement and a wall requires a blend radius to avoid stress concentrations at the sharp corners and possible surface defects due to turbulent flow in the mould. Examples of the mouldability problems of reinforcement features are shown in Figure 8.24. b).

Figure 8.28 shows how the captured mouldability of a reinforcement feature can be represented in the Booch methodology. The detailed representation of the mouldability reinforcement entity in the EXPRESS language is shown in Appendix 4.

Taper feature:

Previous work has not considered a *taper* feature as separate entity but as an attribute of a wall, reinforcement or hole feature. Due to the requirements of considering 3D geometry described in Chapter 6, the author has defined the taper as a separate mouldability type. The attributes of a taper are described below:

i) Wall draft angle – From the literature, in order to avoid difficulty in removing the component from the mould, tapers on wall features should have a minimum angle of 0.8 degrees, Pye (1989), WEKA (1991).

ii) and iii) Reinforcement draft angle and Hole draft angle –From the literature, in order to avoid difficulty in removing the component from the mould, tapers on reinforcement or hole features should have a minimum angle of 5 degrees, Pye (1989), WEKA (1991). Examples of the mouldability design rules of taper features are shown in Figure 8.24. d). Figure 8.29 shows how the captured mouldability of a taper feature can be represented in the Booch methodology. The detailed representation of the mouldability taper entity in the EXPRESS language is shown in Appendix 4.

8.3.2.3. The representation of the mouldability features hierarchy.

Mouldability features can be categorised as either primary mouldability features which follow on from form, or secondary mouldability features which arise from the mouldability requirements of the primary mouldability features. A primary mouldability feature is either a wall, a reinforcement or a hole feature type, and a reinforcement is shown to be either a

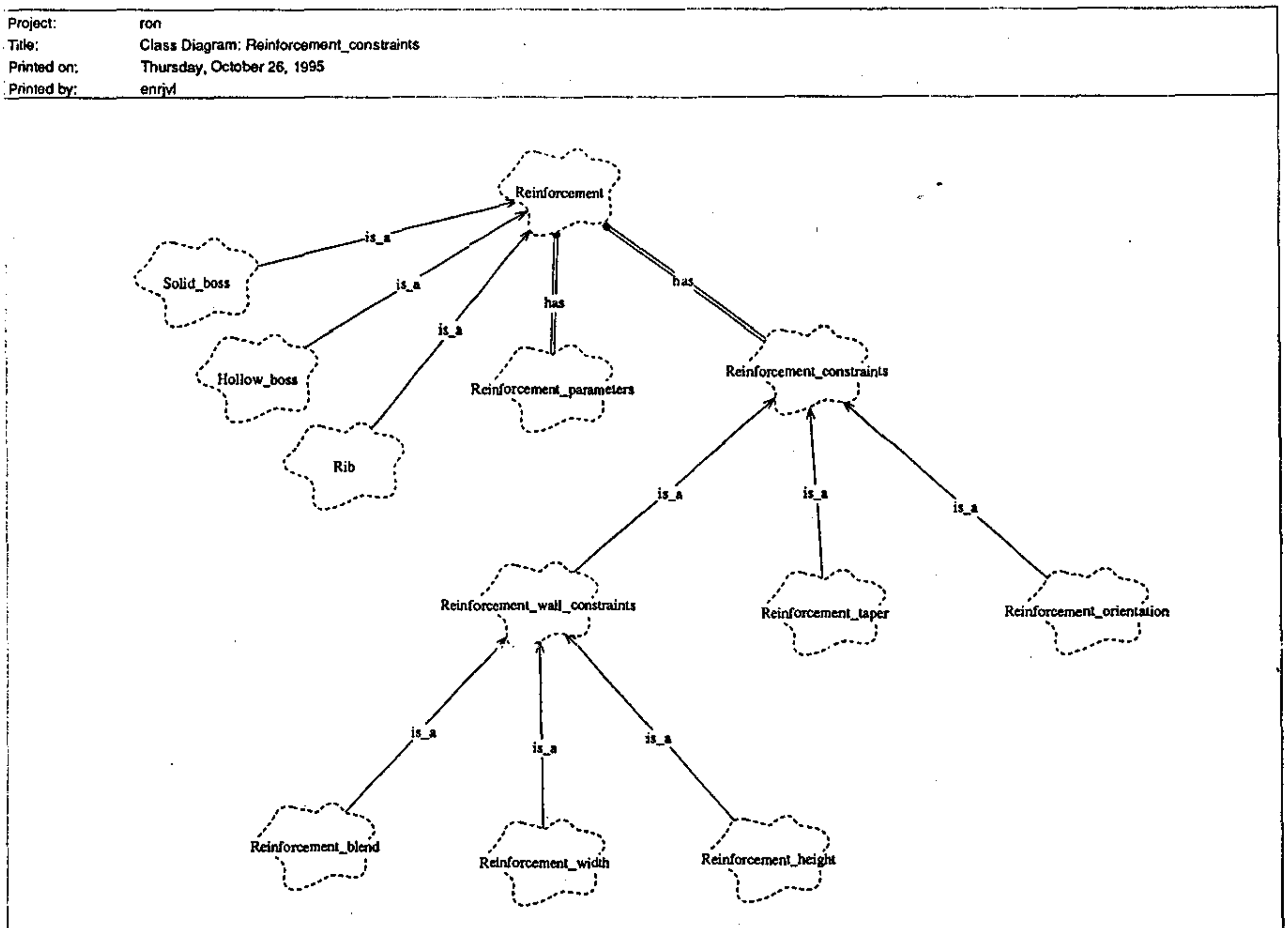


Figure 8.28. The Booch representation of reinforcement feature constraints.

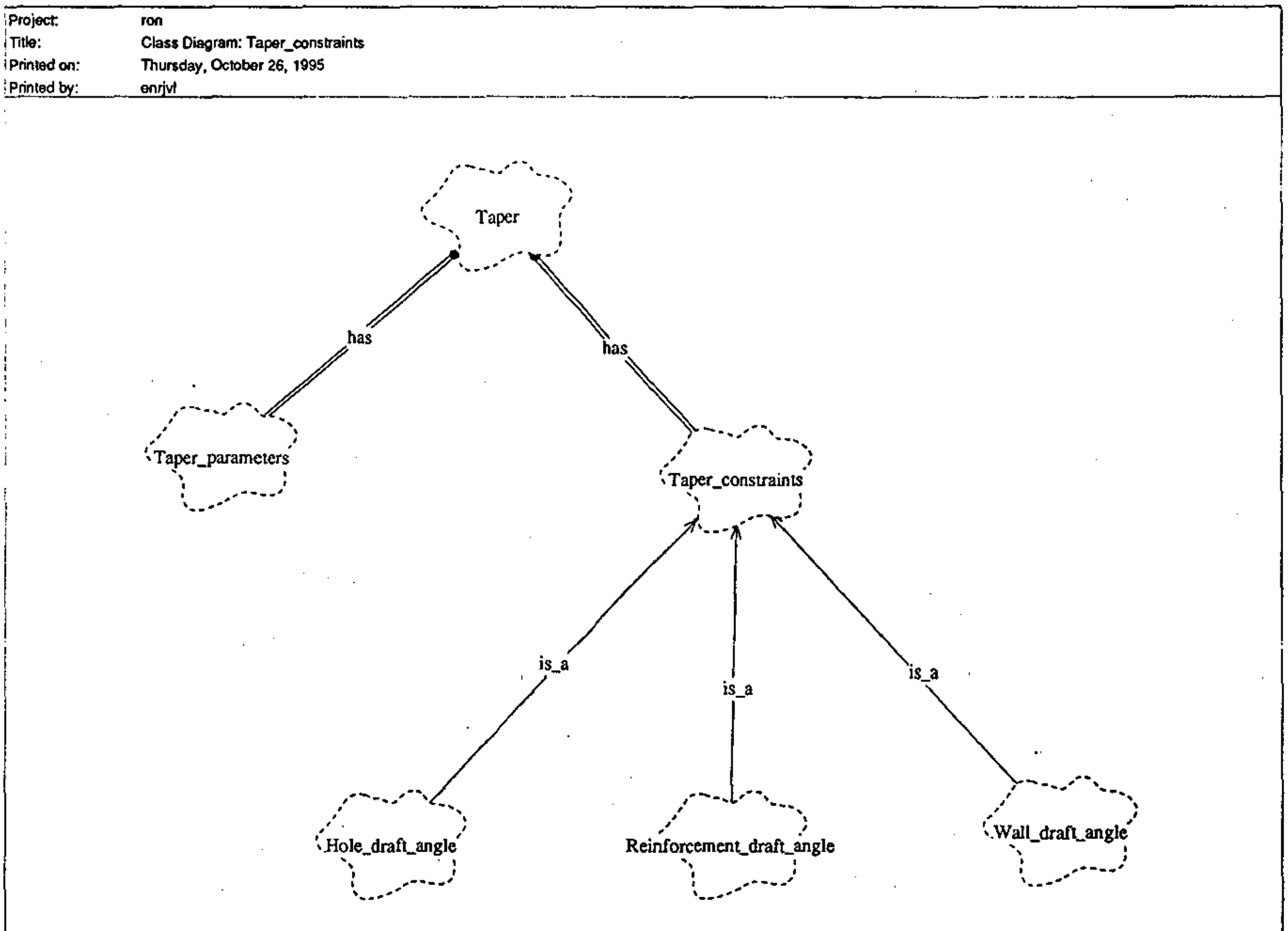


Figure 8.29. The Booch representation of taper feature constraints.

solid boss, a hollow boss or a rib. A secondary mouldability feature is either a taper, a blend or a gate feature type, and a blend is shown to be either a protrusion blend or a corner blend. As described in chapter 7, an association is required between secondary mouldability features and primary mouldability features. Figure 8.30 represents the mouldability features hierarchy.

8.3.3. Mould cavities and cores.

There has been no previous published consideration of an equivalent representation either of the cavity or the core impression capable of concurrent interaction with the geometry of the product, to allow the designer to examine the process constraints of the cavity or core as he/she builds up the product geometry, and enabling product geometry to be re–examined with respect to the effects of design decisions. This thesis describes the first investigation of the capture of the process constraints of the cavity and core impression to support the concurrent design of a product and a mould.

8.3.3.1. Process constraints of the cavity and core impression.

The process constraints that apply to the cavity impression and core impression are similar and have been identified by discussions with people in industry about their design methods for injection moulding, and by the use of literature about the methodology and considerations in injection mould design. Three items need consideration for each: In the cavity, i) overhangs, ii) cavity tapers, iii) cavity blends. In the core, i) overhangs, ii) core blends, iii) core tapers. The types of problem arising and the significant factors are:

i) Overhangs – If an overhang exists in the mould *cavity* a split mould is required for removal of a product from a mould, If a rim exists on a rotational product this renders the product non–mouldable, as the rim acts as an obstruction to the opening of the second split line. If an all round overhang exists in the *core* of 1.5 mm or less stripping of the component is required from the core. For a greater overhang than 1.5 mm a collapsing core is required

Title: Class Diagram: Mouldability_features
Printed on: Sunday, December 17, 1995
Printed by: enrjvl

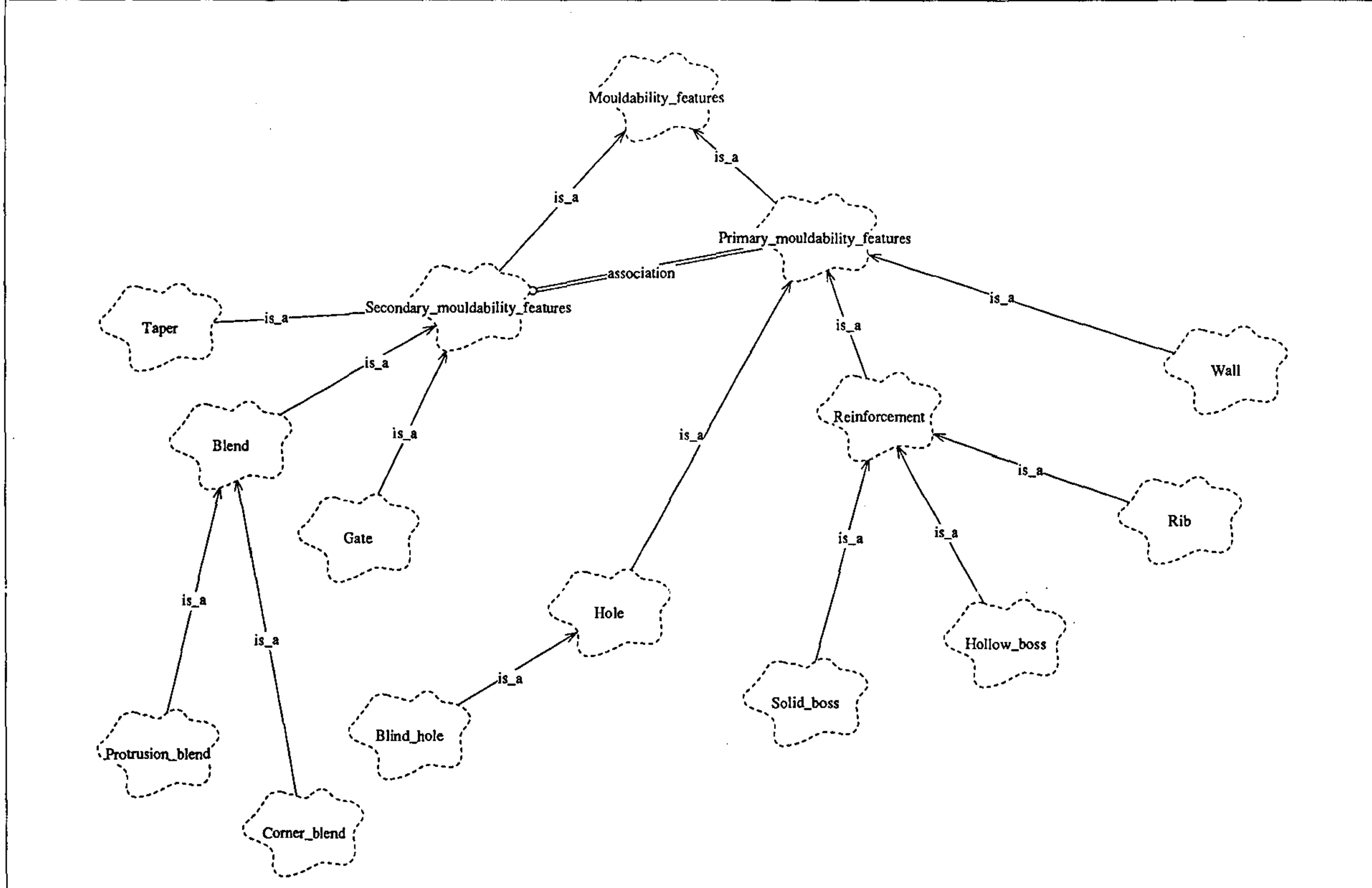


Figure 8.30. The Booch representation of the mouldability features hierarchy.

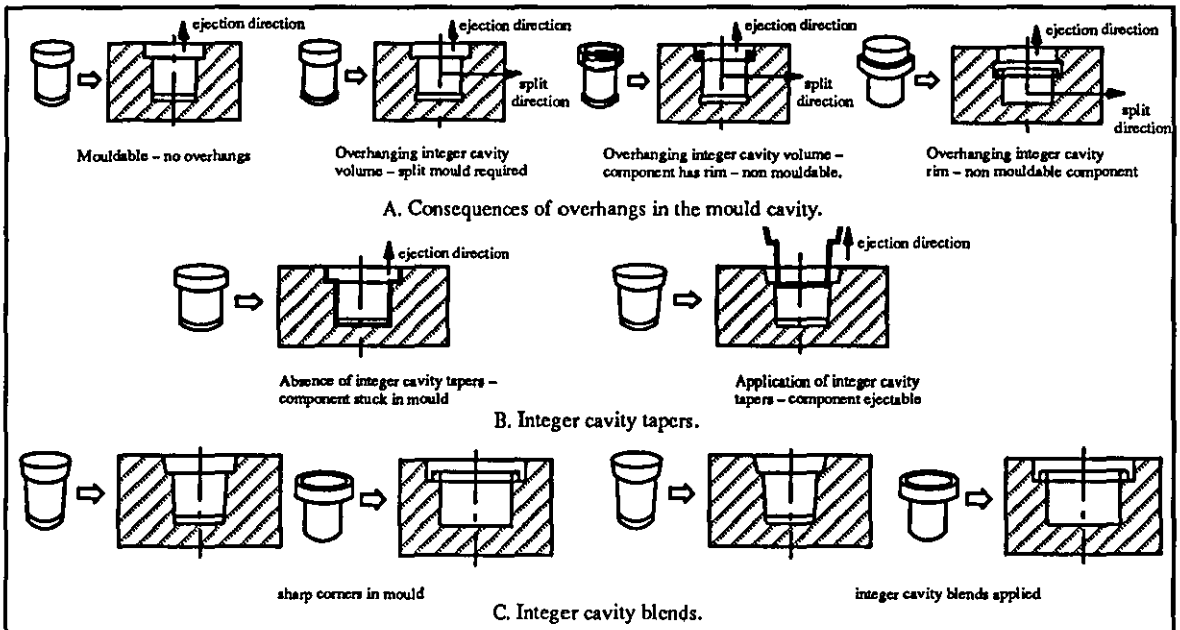


Figure 8.31. Process constraints of integer cavity rims and integer cavity volumes.

for removal of the product from the mould, Pye (1989).

ii) Tapers – In order to reproduce the outside and inside shapes of a product, both cavity and core tapers are required in the impression to match those on the outside and inside of the corresponding walls on a product. If there are no tapers in an impression components will become stuck in the mould.

iii) Blends – In order to produce the outside and inside shapes of a moulded product, both cavity and core blends are required to match those on the outside and inside of the corresponding walls on a product. The absence of blends leaves sharp corners which are difficult to produce and can wear during operation of a mould, Pye (1989).

8.3.3.2. Process constraints applied to cavity/core impression features.

The ways of avoiding problems in cavity and core impressions have been identified by the author in Figures 8.31, 8.32 and 8.33, which show design rules linked to cavity impression features, and Figures 8.34 to 8.36 for core features. The features defined as part of the cavity and core impression in Chapter 6 are: integer cavity/core volume, rim, slot, hole, boss,

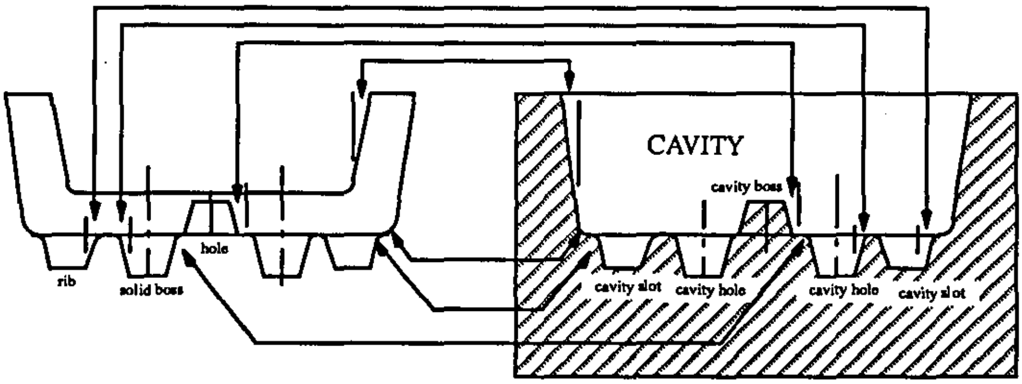


Figure 8.32. Integer cavity tapers and blends on integer cavity slots, bosses and holes.

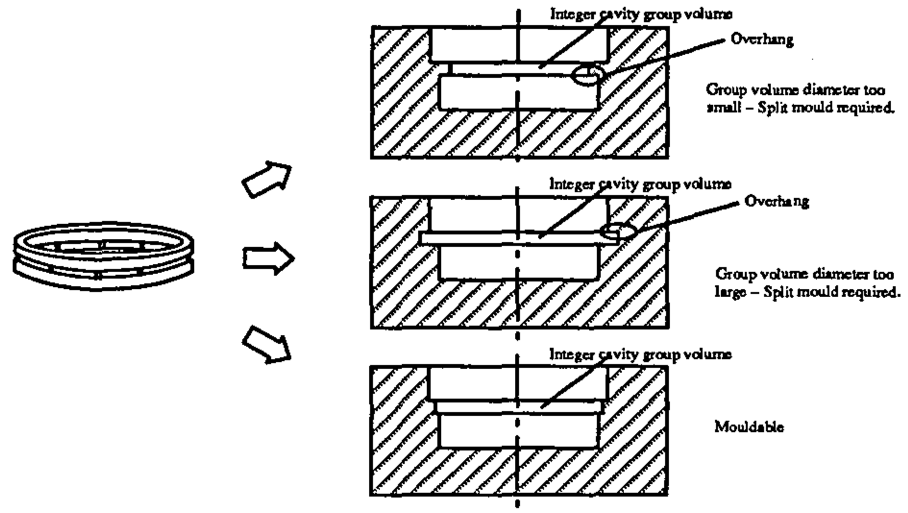


Figure 8.33. Process constraints of integer cavity group volumes.

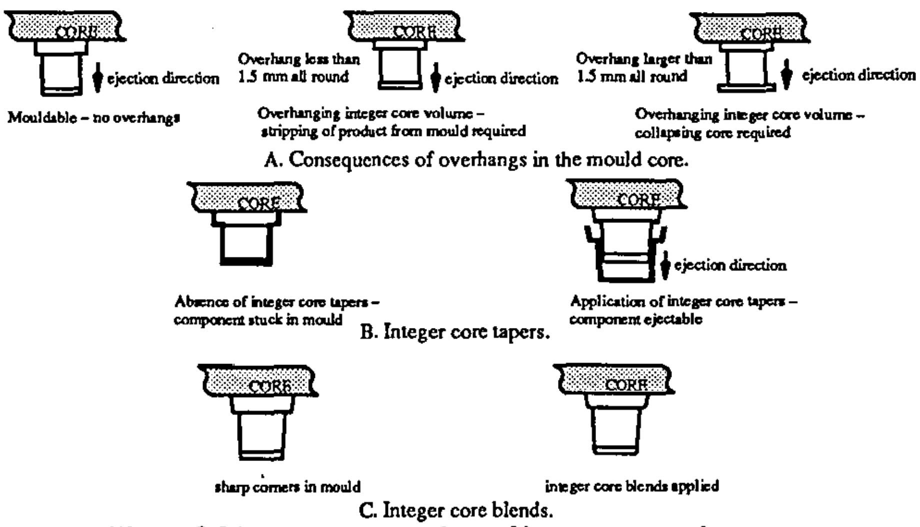


Figure 8.34. Process constraints of integer core volumes.

blend, taper and group volume. Examination of each of these features in turn has enabled the author to ascertain how process constraints can be applied. Examples of the application of process constraints to these impression features are now presented.

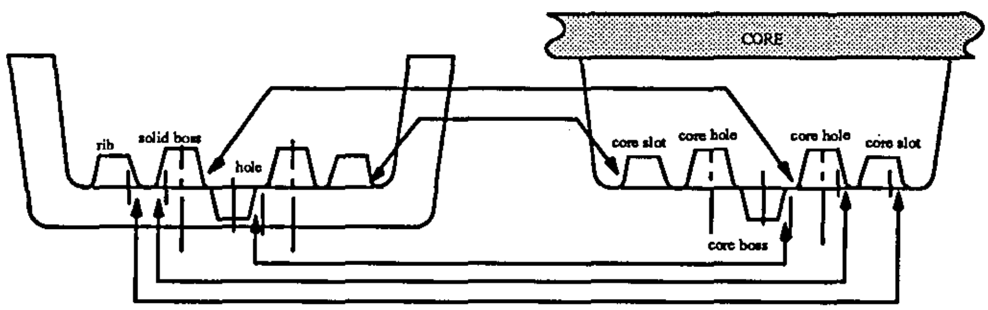


Figure 8.35. Integer core tapers and blends on integer core slots, bosses and holes.

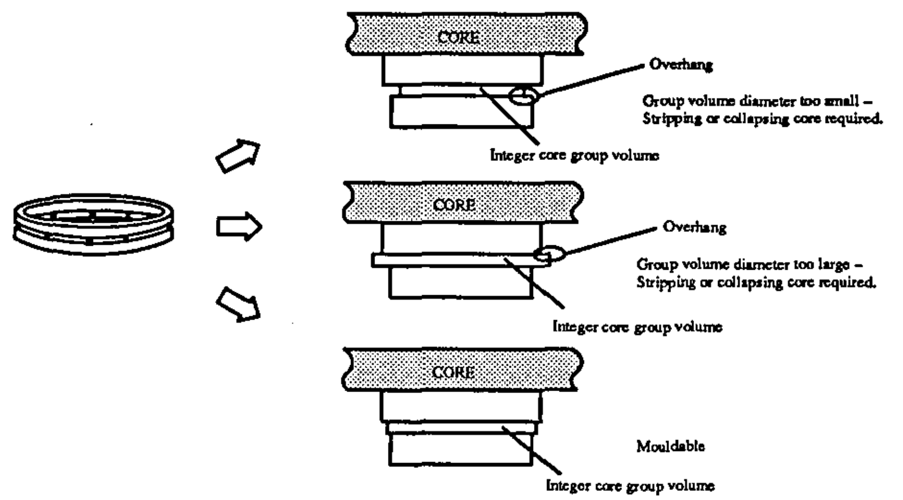


Figure 8.36. Process constraints of integer core group volumes.

Integer cavity volume, Integer core volume:

The attributes that relate to an *integer cavity volume* and *integer core volume* alone are cavity/core volume tapers:

i) Cavity and core volume tapers – To produce the outside or inside shape of a moulded product, the integer cavity or core volumes require a taper to match those on the corresponding walls. If there are no tapers the component will become stuck in the mould

The attributes relating to *integer cavity or core volume interactions* with adjacent features are the volume overhang and volume blend:

i) Cavity or core volume overhang – If the diameter of an integer *cavity* volume is larger than that of another integer cavity volume or an integer cavity rim that is closer to the parting line, an overhang exists in the mould cavity and a split mould is required for removal of the product from the mould. If a rim exists on a rotational product this renders the product non–mouldable, as the rim acts as an obstruction to the opening of a second split line. In order to avoid a split mould and a possible non–mouldable component the diameter of each integer cavity volume should be progressively reduced away from the parting line or remain the same. If the diameter of an integer *core* volume is larger than that of another integer core volume that is closer to the parting line, an overhang exists in the mould core. For an all round overhang of 1.5 mm or less stripping of the component is required from the core. For a greater overhang than 1.5 mm a collapsing core is required for removal of the product from the mould, Pye (1989). In order to avoid stripping of the component or a collapsing core the diameter of each integer core volume should be progressively reduced away from the parting line or remain the same.

ii) Cavity or core volume blends – In order to produce the outside shape of a moulded product in the *cavity*, an integer cavity volume requires a blend to match that on the outside of a corresponding wall. An integer cavity blend can be between two abutting integer cavity volumes or between an integer cavity volume and an integer cavity rim. In order to produce the inside shape of a moulded product in the *core*, an integer core volume requires a blend to match that on the inside of a corresponding wall. An integer core blend is between two abutting integer core volumes. The absence of blends in the core or cavity leaves difficult to produce sharp corners which can wear during the operation of the mould, Pye (1989).

Figures 8.37 and 8.38 show how the captured design rules of an integer cavity and core volume feature can be represented in the Booch methodology. The detailed representation of an integer cavity and core volume in the EXPRESS language are shown in Appendix 4.

Integer group volumes:

The attributes that relate to an *integer cavity/core group volume* are group volume overhang

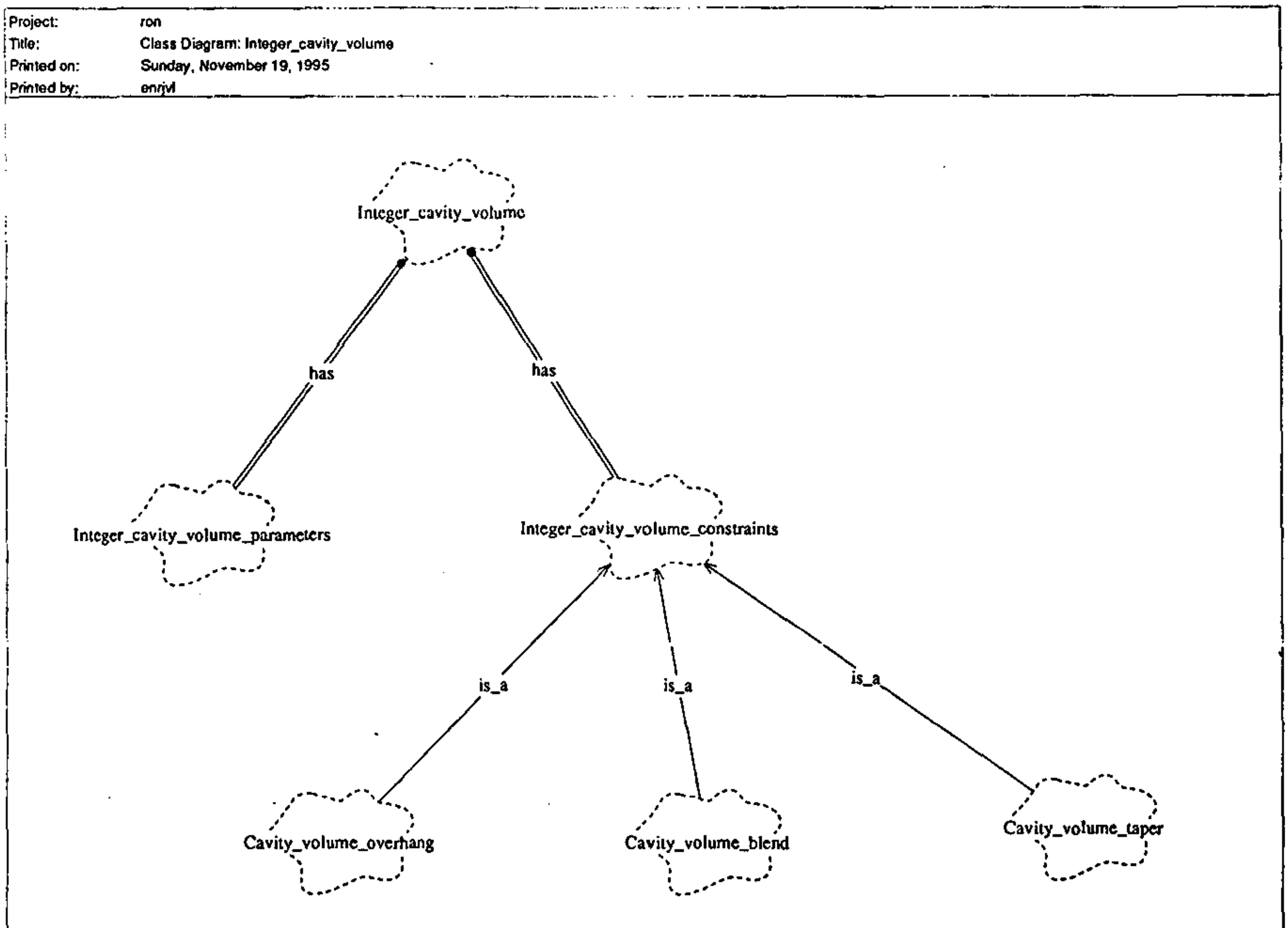


Figure 8.37. The Booch representation of integer cavity volume feature constraints.

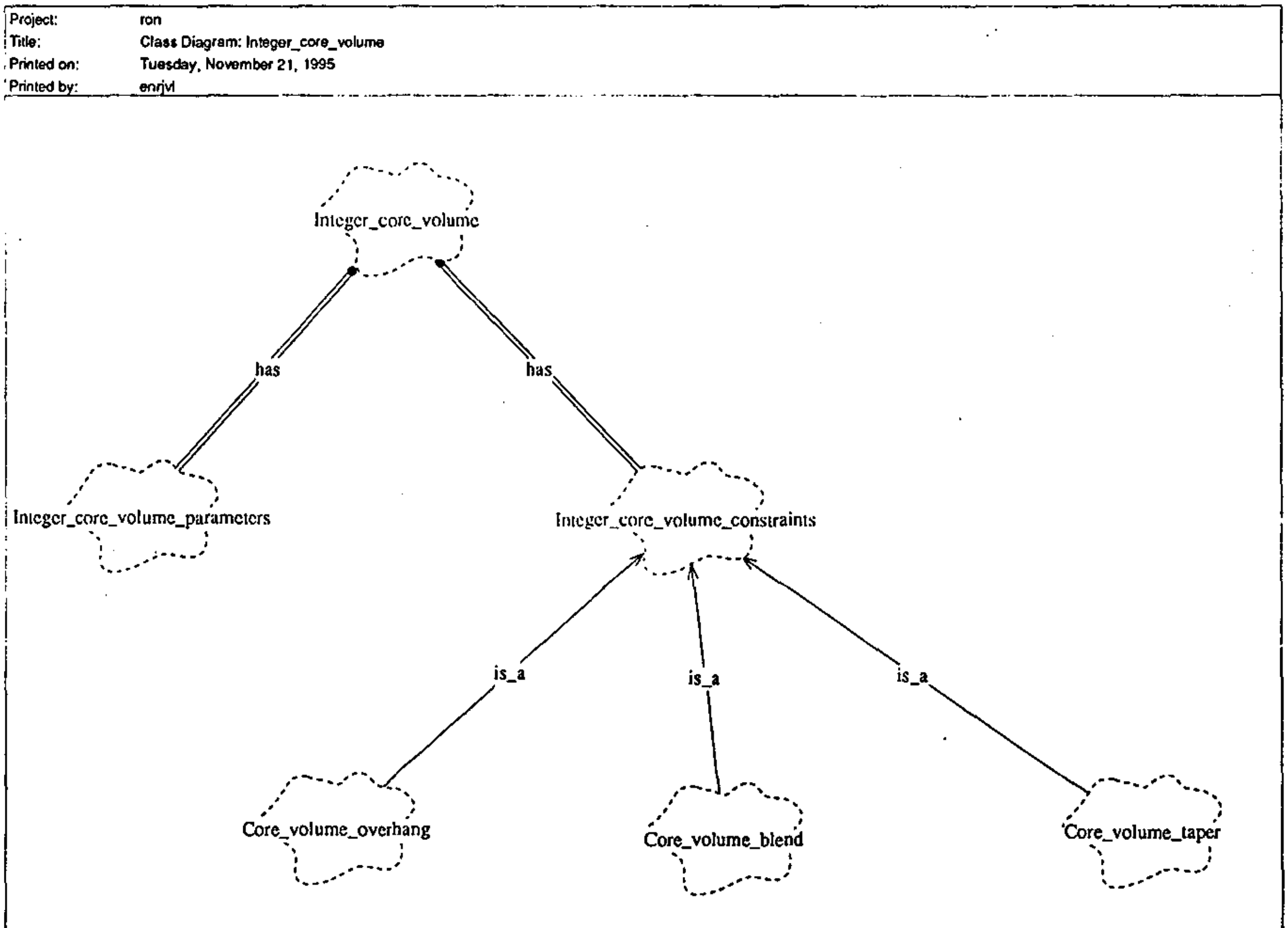


Figure 8.38. The Booch representation of integer core volume feature constraints.

above and group volume overhang below. These attributes are described below:

i) Group volume overhang above – If the diameter of an integer cavity or core group volume is larger than that of an integer cavity or core volume that is closer to the parting line, an overhang exists in the mould cavity or core respectively. In the former case a split mould is required for removal of the product from the mould. If a rim exists on a rotational product this renders the product non–mouldable, as the rim acts as an obstruction to the opening of a second split line. In order to avoid a split mould and a possible non–mouldable component the diameter of an integer cavity group volume should be the same or lower than the diameter of the integer cavity volume on the parting line side. In the case of the core, for an all round overhang of 1.5 mm or less stripping of the component is required from the core. For a greater overhang than 1.5 mm a collapsing core is required for removal of the product from the mould, Pye (1989). In order to avoid stripping of the component or a collapsing core the diameter of an integer core group volume should be the same or lower than the diameter of the integer core volume on the parting line side.

ii) Group volume overhang below – If the diameter of an integer cavity group volume is smaller than that of an integer cavity volume that is further from the parting line, an overhang exists in the mould cavity and a split mould is required for removal of the product from the mould. If a rim exists on a rotational product this renders the product non–mouldable, as the rim acts as an obstruction to the opening of a second split line. In order to avoid a split mould and a possible non–mouldable component the diameter of an integer cavity group volume should be the same or higher than the diameter of the integer cavity group volume on the furthest side from the parting line. If the diameter of an integer core group volume is smaller than that of an integer core volume that is further from the parting line, an overhang exists in the mould core. For an all round overhang of 1.5 mm or less stripping of the component is required from the core. For a greater overhang than 1.5 mm a collapsing core is required for removal of the product from the mould, Pye (1989). In order to avoid stripping of the component or a collapsing core the diameter of an integer core group volume should be the same or higher than the diameter of the integer core volume on the furthest side from the parting line

Examples of the problems of integer group volumes are shown in Figure 8.33 (cavity) and 8.36 (Core). Figures 8.39 and 8.40 show how the captured design rules can be represented in the Booch methodology. The figures show that the existence of overhangs is considered both above and below, ie nearer the parting line and further away. The detailed representation of integer cavity and core group volume entities in the EXPRESS language is shown in Appendix 4.

8.3.3.3. Process constraints of the cavity plate and core plate.

The cavity plate and core plate are those parts of the cavity/core that contain the impression, the feeding system and the cooling system. Previous work has not considered the geometry in the mould and work was restricted to identifying the components of an insert type mould in relation to the overall shape of the insert block containing the impression. The parameters of the mould elements were considered as abstract entities. We now consider the capture of the process constraints of the cavity plate needed to provide feedback support for the designer as he/she builds up the mould.

The features defined as part of the cavity plate or core plate in Chapter 6 are integer cavity block, integer core block, integer cavity or core inner land, integer cavity or core peripheral land, integer cavity backing plate, and integer cavity nozzle recess. Two types of inner land have been considered here which are integer cavity or core rectangular land and integer cavity or core circular land. Examination of each of these features in turn has shown that there are certain process constraints relating to feature types and also to their interaction with other features. The process constraints that apply to the cavity plate have been identified as: i) Accommodation of impression (cavity) or supporting the impression (core), ii) Accommodation of the feeding system, iii) Accommodation of the cooling system, iv) Accommodation of the guide system. The following are the types of problem that can occur and the significant factors which need to be considered:

Project:	ron
Title:	Class Diagram: Integer_cavity_group_volume
Printed on:	Sunday, November 19, 1995
Printed by:	enrjvl

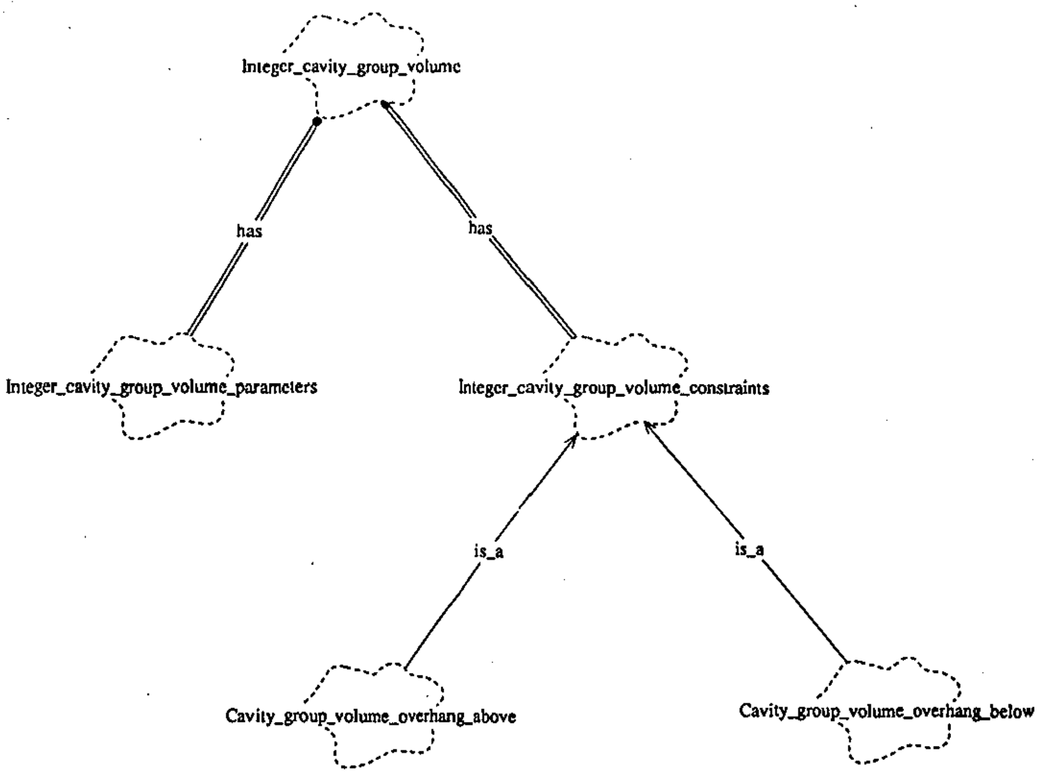


Figure 8.39. The Booch representation of integer cavity group volume feature constraints.

Project:	ron
Title:	Class Diagram: Integer_core_group_volume
Printed on:	Tuesday, November 21, 1995
Printed by:	enrjvl

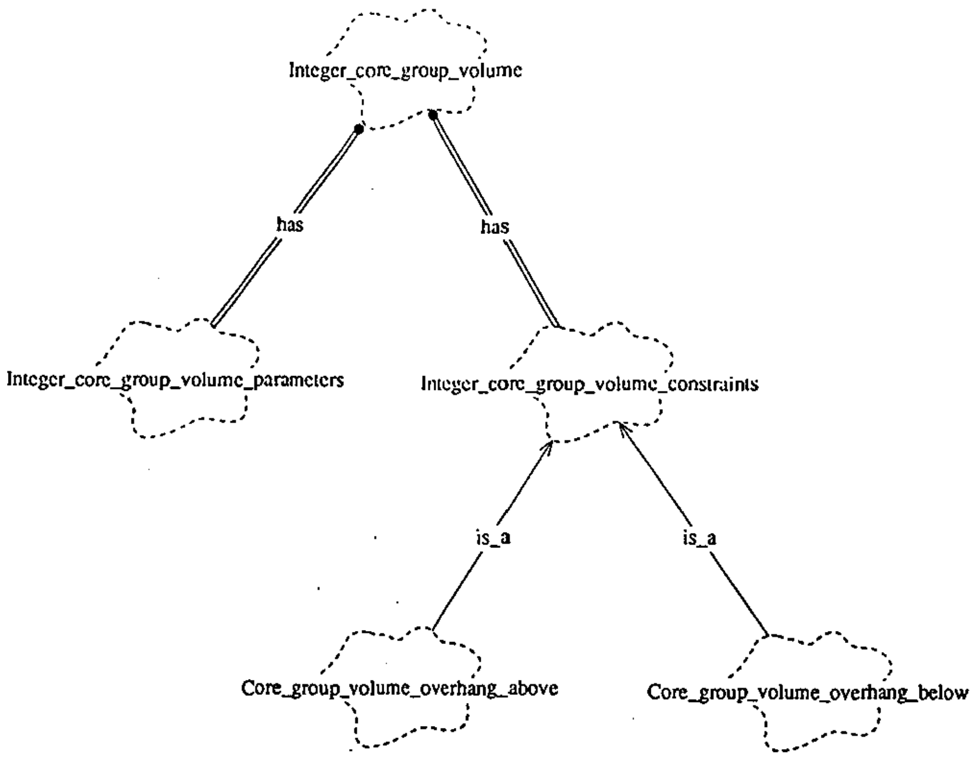


Figure 8.40. The Booch representation of integer core group volume feature constraints.

i) Accommodation of impression – A *cavity* plate must accommodate the cavity impression such that the opening of the impression is at the widest part in the draw direction, allowing removal of products from a mould. The spatial relationship of the cavity plate and the impression and the configuration of the cavity plate must provide balanced forces in the mould due to clamping forces and minimum strength requirements to avoid distortion of the mould and the consequent flash formation on the products. Supporting the impression – A *core* plate must provide support for the core impression such that the opening of the impression is at the widest part in the draw direction, allowing removal of products from a mould. The spatial relationship of the core plate and the impression and the configuration of the core plate must provide balanced forces in the mould due to clamping forces and minimum strength requirements to avoid distortion of the mould and the consequent flash formation on the products.

ii) Accommodation of feeding system – The configuration and dimensions of the cavity and core plates must enable the accommodation of a feeding system in a manner that facilitates provision of minimum venting requirements, allows removal of the feeding system from the mould and facilitates the interfacing of the feeding system with the injection machine nozzle.

iii) Accommodation of the cooling system – The dimensions of the cavity and core plates must enable the accommodation of a cooling system of the appropriate configuration to provide optimum cooling to the cavity impression, and to the core respectively.

iv) Accommodation of the guide system – Allowance must be made in the dimensions of the cavity plate for the accommodation of the appropriate size of guide system to ensure the mating of the two mould halves. Likewise the configuration and dimensions of the core plate must ensure the mating of the two mould halves.

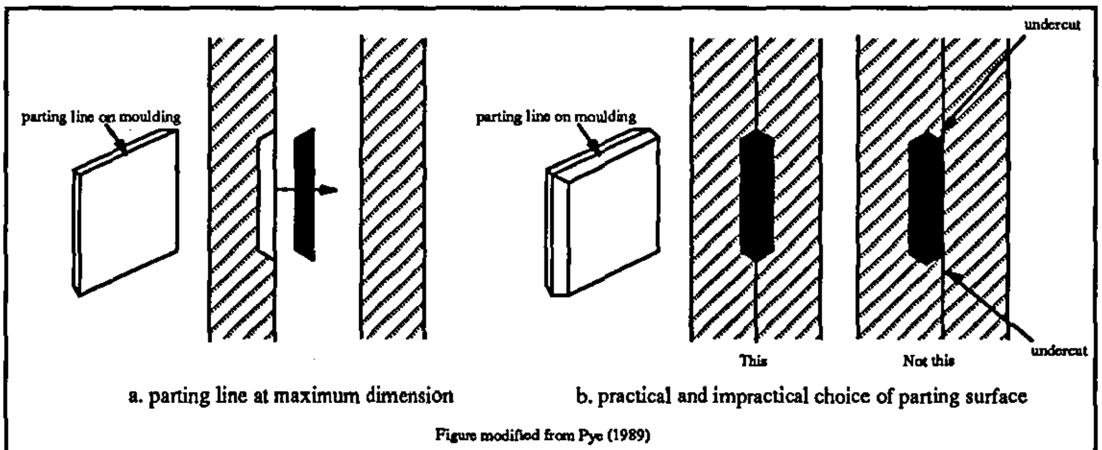


Figure 8.41. Process constraints of integer cavity parting line.

8.3.3.4. Process constraints applied to the cavity plate features, and core plate features.

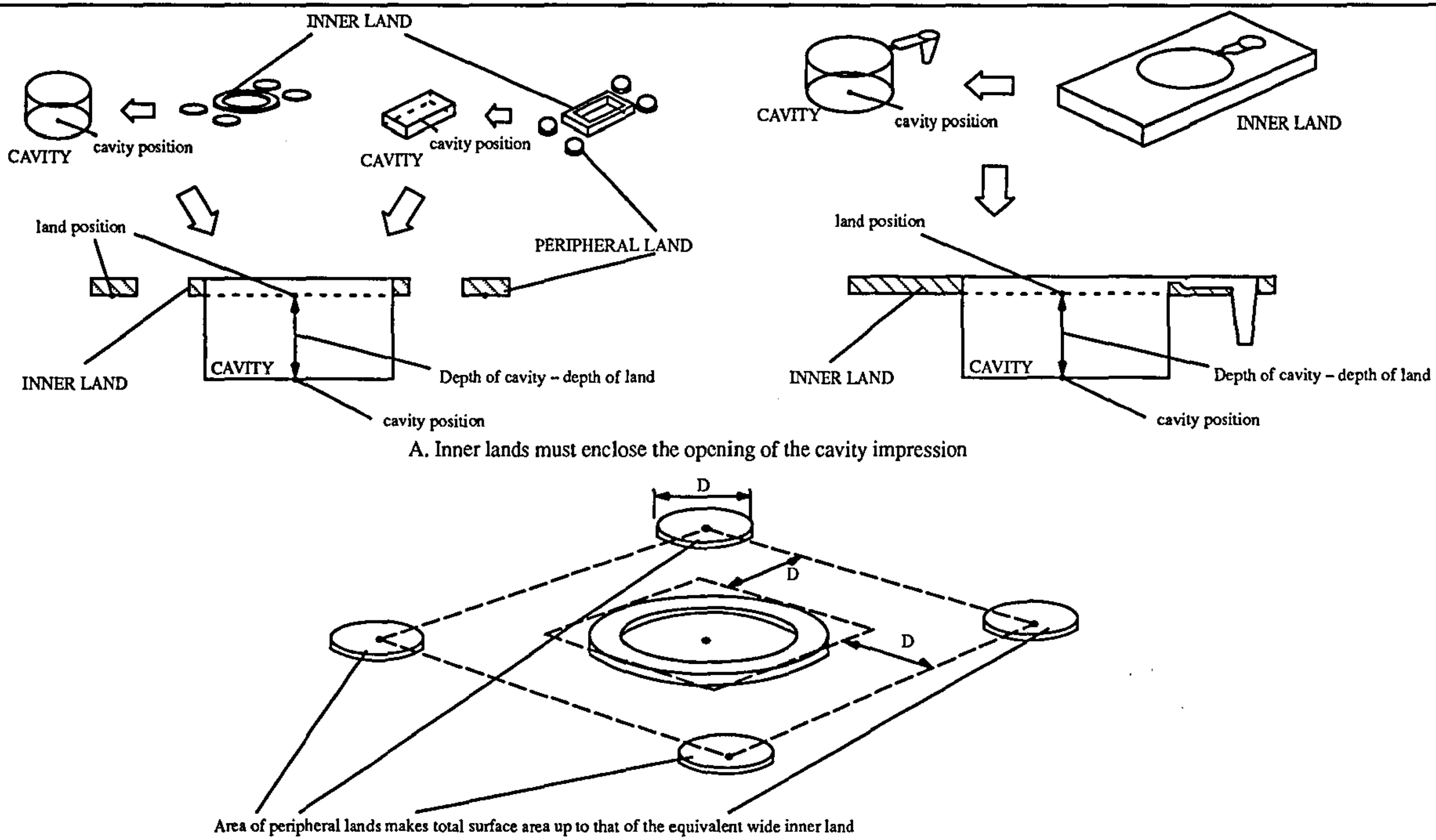
The ways of avoiding the problems in the cavity and core plates have been identified in Figures 8.41 to 8.44, and 8.45 respectively which show design rules linked to these features. Each of the plate features has been examined in turn in order to ascertain how process constraints can be applied. Examples are described below:

Integer cavity rectangular land:

The attributes that relate to an *integer cavity rectangular land* feature are cavity rectangular land position, cavity rectangular land depth, cavity rectangular land width and cavity rectangular land length. These attributes are described below:

i) Cavity rectangular land position – There is a positional constraint on an integer cavity inner land arising from the requirement to enclose the opening of the cavity impression. The land is a small bedded down area adjacent to the impression, Pye (1989). If a cavity impression is edge gated, the integer cavity inner land must provide a parting surface for a gate and that part of a runner system that is on the parting line as well as for a cavity impression.

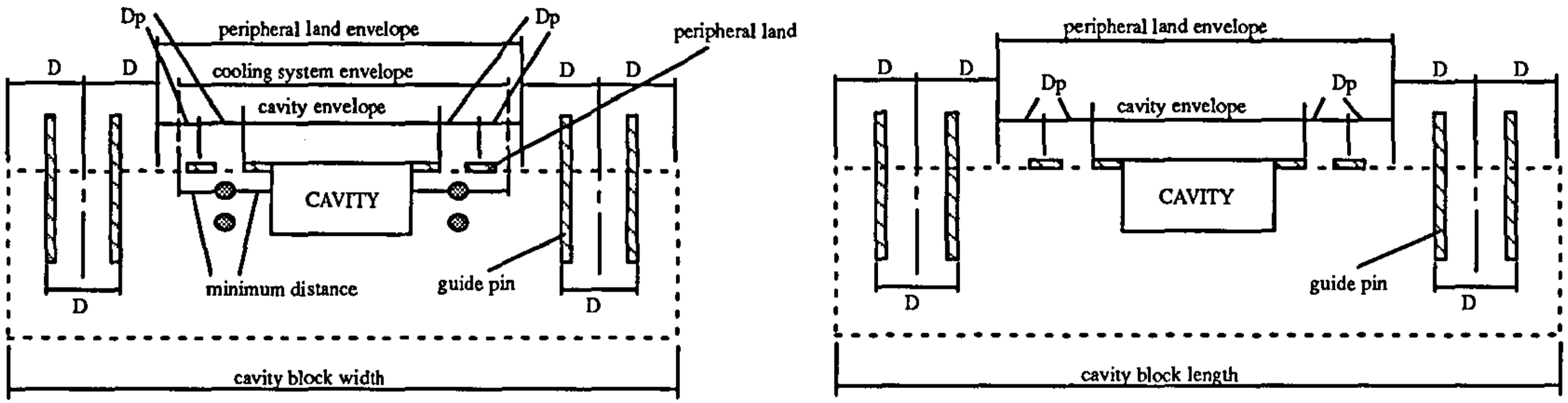
ii) Cavity rectangular land depth – There is a recommended minimum value which places



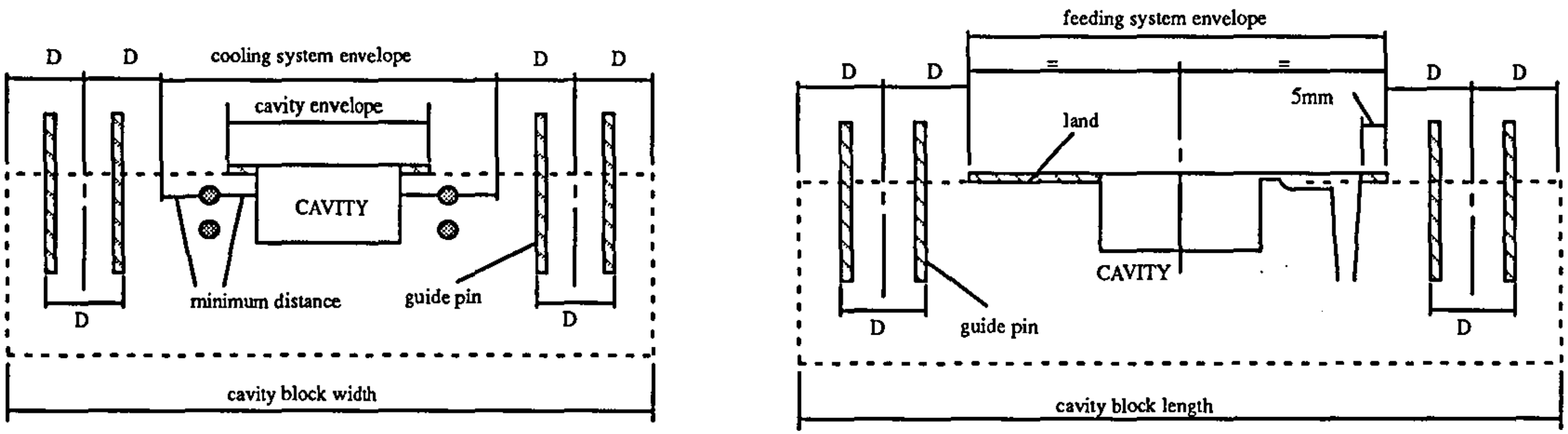
A. Inner lands must enclose the opening of the cavity impression

B. Peripheral lands are required to make up the surface area of narrow inner lands

Figure 8.42. Process constraints of integer cavity lands.



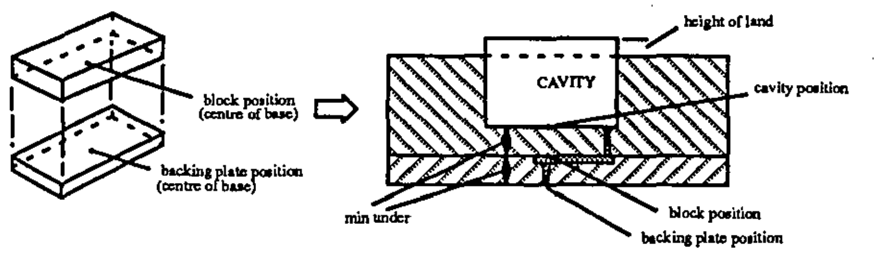
A. Block size calculation with narrow land



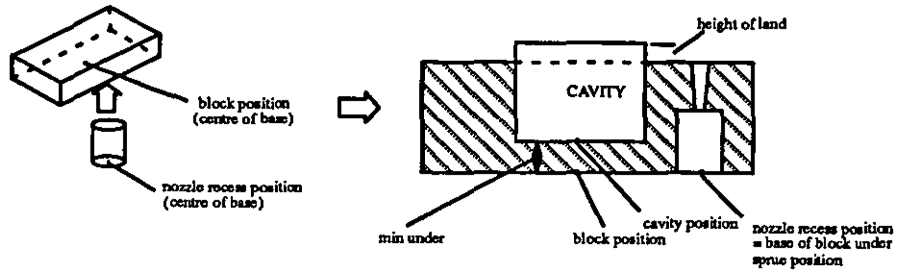
B. Block size calculation with wide land

● = cooling tube
Dp = diameter of peripheral land

Figure 8.43. Process constraints of integer cavity block.



A. Backing plate parameters are dependant upon those of the block and the feeding system



B. Nozzle recess parameters are dependant upon those of the block and feeding system

Figure 8.44. Process constraints of backing plate and nozzle recess.

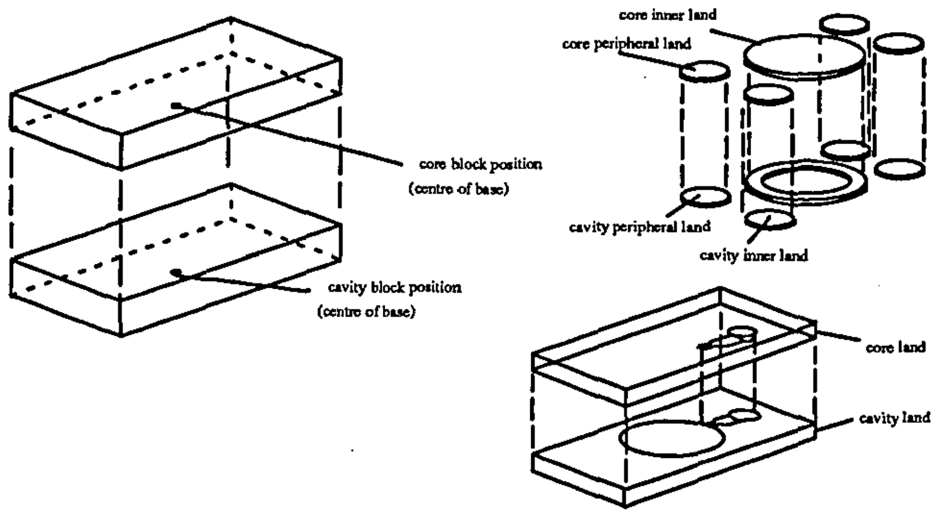


Figure 8.45. Relationship between cavity and core elements other than the impression

a constraint on the depth attribute of an integer cavity rectangular land. Apart from the surface of a land, which is used for bedding down in order to seal the impression, the surrounding area is relieved to a depth of a least 2.4 mm, Pye (1989).

iii) and iv) Cavity rectangular land width and Cavity rectangular land length– The distance between the opening of an impression and the edge of an integer cavity inner land is normally between 5 mm and 25 mm. The small distance permits venting where required to be added easily, by scribing fine grooves across the surface of the land from the cavity impression to the relief area, Pye (1989). A distance above 25 mm causes problems of achieving a perfect parting surface over a large area.

Examples of inner lands are shown in Figure 8.42. Figure 8.46 shows how the the captured design rules of an integer cavity rectangular land feature can be represented in the Booch methodology, The detailed representation in the EXPRESS language is shown in Appendix 4.

Integer core circular and peripheral lands:

An integer core circular land is used where an impression is undergated to give the thinnest possible land around the impression opening, and thereby the best possible venting. Due to large clamping forces in a mould, if a small inner land is used the land area may be insufficient to withstand the applied force and may deform. Therefore the effective land area is made up by using integer core peripheral lands, normally at the corners of the mould. The attributes that relate to an *integer core circular land* are core circular land position, core circular land depth and core circular land diameter, as follows:

i) Core circular land position – There is a positional constraint on an integer core circular land arising from the requirement to enclose the opening of the impression. In order to ensure this and the mating and sealing of the two mould halves the position of an integer core circular land must be directly above an integer cavity circular land.

ii) Core circular land depth – There is a minimum value which places a constraint on the depth attribute of an integer core circular land. As in the cavity, apart from the surface of a land, which is bedded down in order to seal the impression, the surrounding area is relieved to a depth of at least 2.4 mm, Pye (1989).

iii) Core circular land diameter – The dimensions of an integer core circular land should match those of an integer cavity circular land so that the two halves of the mould can mate and seal.

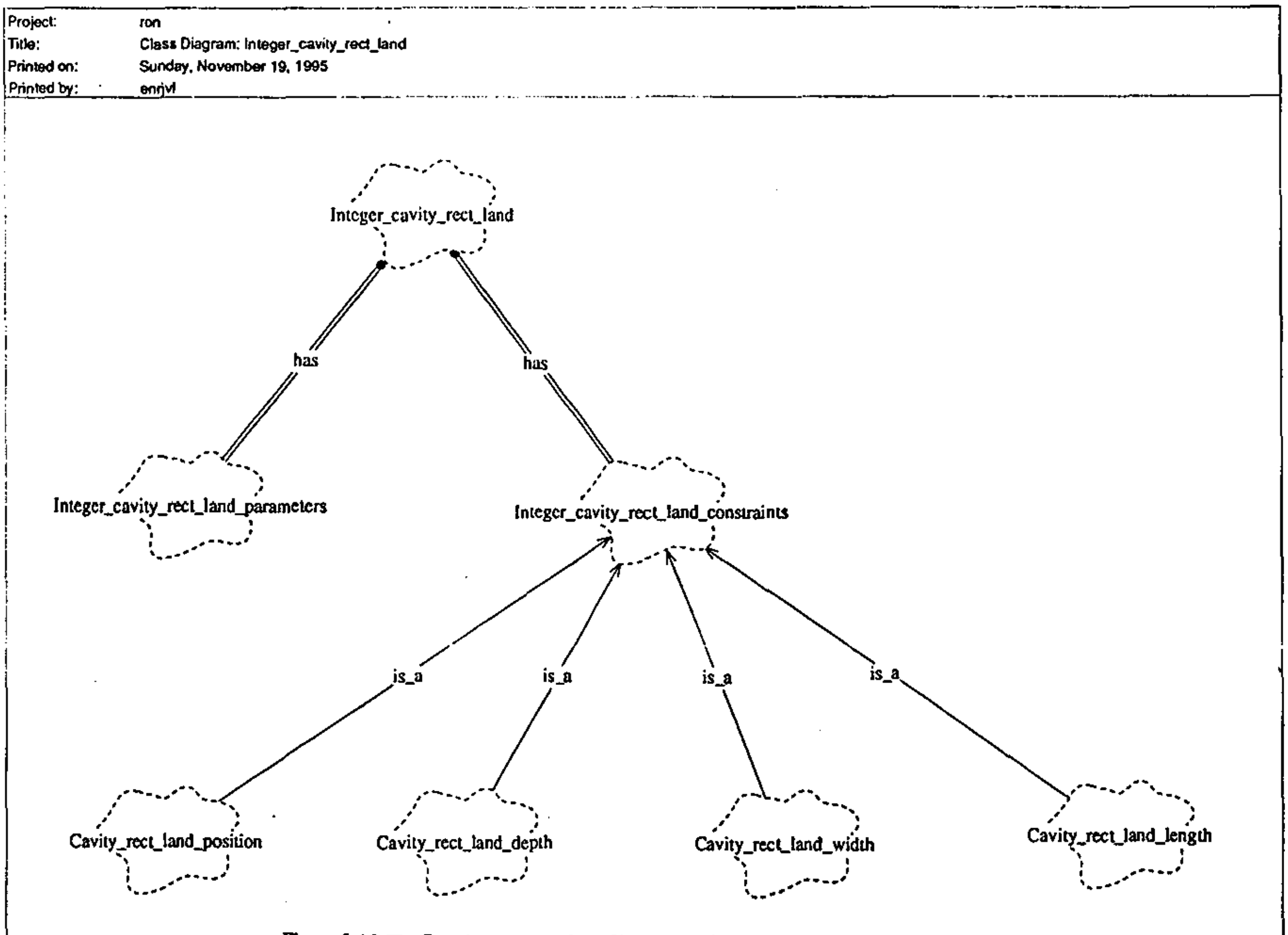


Figure 8.46. The Booch representation of integer cavity rectangular land feature constraints.

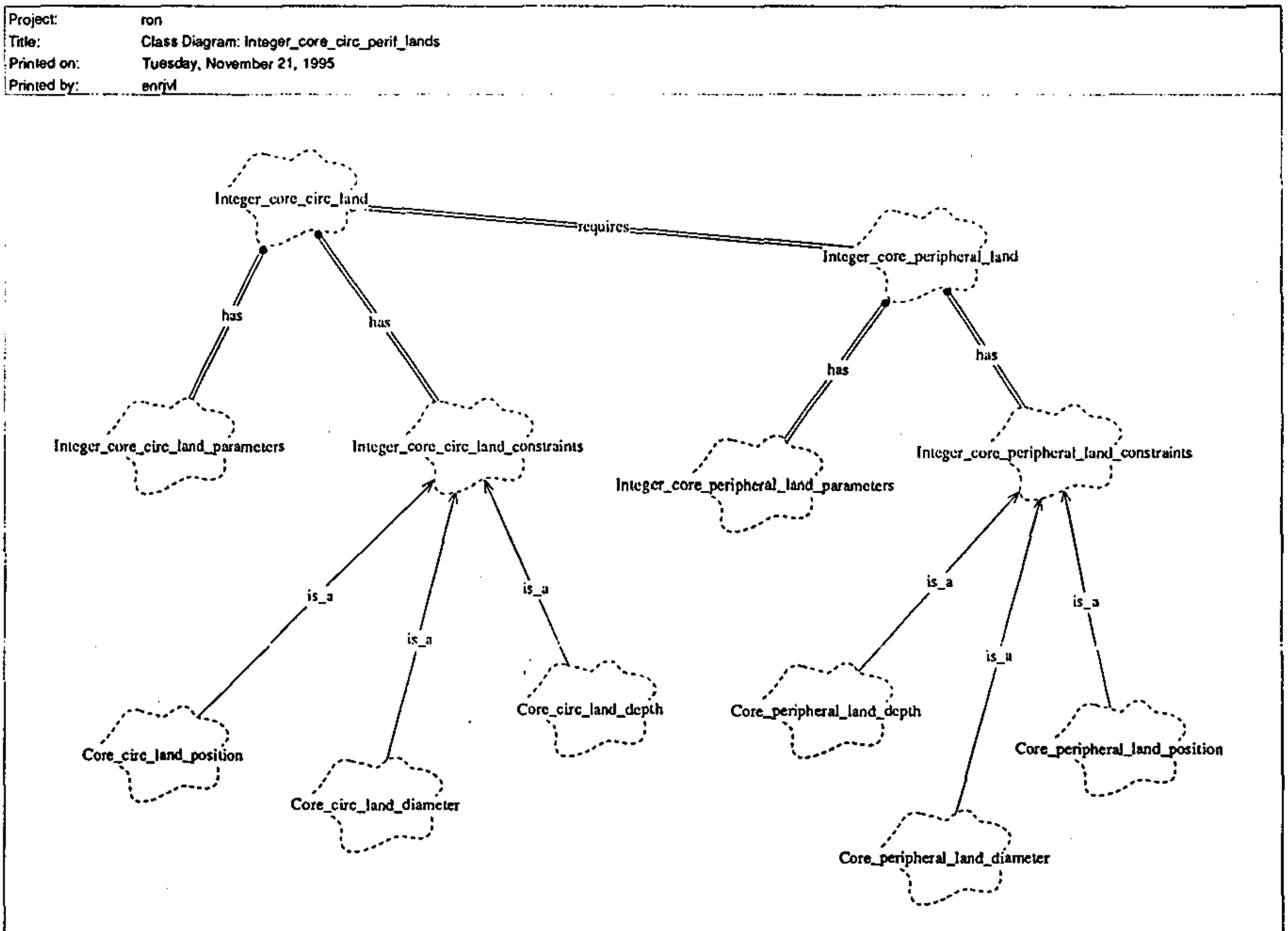


Figure 8.47. The Booch representation of integer core circular and peripheral land feature constraints.

The attributes that relate to an *integer core peripheral land* are core peripheral land position, core peripheral land depth and core peripheral land diameter:

i) Core peripheral land position – There is a positional constraint on an integer core peripheral land arising from the requirement for the mating of the two mould halves in a manner that provides extra land area to withstand clamping forces. The position of an integer core peripheral land must be directly above an integer cavity peripheral land.

ii) Core peripheral land depth – To increase the land area to withstand clamping forces in the mould the depth of the peripheral lands must be identical to that of the core inner land.

iii) Core peripheral land diameter – The dimensions of an integer core peripheral land should match those of an integer cavity peripheral land so that the two halves of the mould can mate and the maximum extra land area is provided.

Figure 8.47 shows how the captured design rules of integer core circular land and peripheral land features can be represented in the Booch methodology. The detailed representation in the EXPRESS language is shown in Appendix 4.

<u>Integer cavity mould block and integer core mould block:</u>

The attributes that relate to an *integer cavity and core mould block* are block position, block depth, block width, block length and guide system parameters. Consider each of these:

i) Block position – Integer cavity and core mould blocks must be centred on a cavity/core impression to avoid unbalanced clamping forces in the mould and must make allowance for depth requirements below the cavity impression or above the core impression for mould system elements.

ii) Block depth – An integer mould block must be deep enough to accommodate the depth as appropriate of a cavity impression below the lands, and any parts of a feeding system or cooling system that may be below a cavity, above a core, impression. In order to avoid distortion of the mould due to the injection forces, a minimum thickness of metal must exist below a cavity, and above a core, impression, Pye (1989). The minimum thickness is related to the width of the impression when viewed in the draw direction.

iii) and iv) Block width and Block length – The length and width of a mould block must be sufficient to accommodate an impression, a feeding system, cooling system, lands, and a guide system. Beyond the above requirements the dimensions of a block should be kept

to a minimum to avoid cost and weight. As shown in Figure 8.45 the length and width of an integer core block would normally match those of an integer cavity block since the largest of the mould elements, the guide system, must be accommodated by both blocks and must mate between the cavity and the core. Therefore the minimum length and width calculations of the two blocks will be the same. Furthermore in an integer mould the cavity and core may be made from the same block of steel.

v) Guide system parameters – In order to ascertain the minimum dimensions of an integer cavity mould block the size of the guide system it must accommodate must be ascertained. The diameter of the pins in a guide system can be related to the size of the mould and whether side forces are likely to occur.

Example problems of an integer cavity block are shown in Figure 8.43. Figures 8.48 and 8.49 show how the captured design rules of an integer cavity or core block respectively can be represented in the Booch methodology. Figure 8.48 shows that due to the requirement of an integer cavity block to accommodate the guide system, one of the design rules on the dimensions of an integer cavity block is the size of the guide system. The detailed representation of the integer cavity and core block entities in the EXPRESS language is shown in Appendix 4.

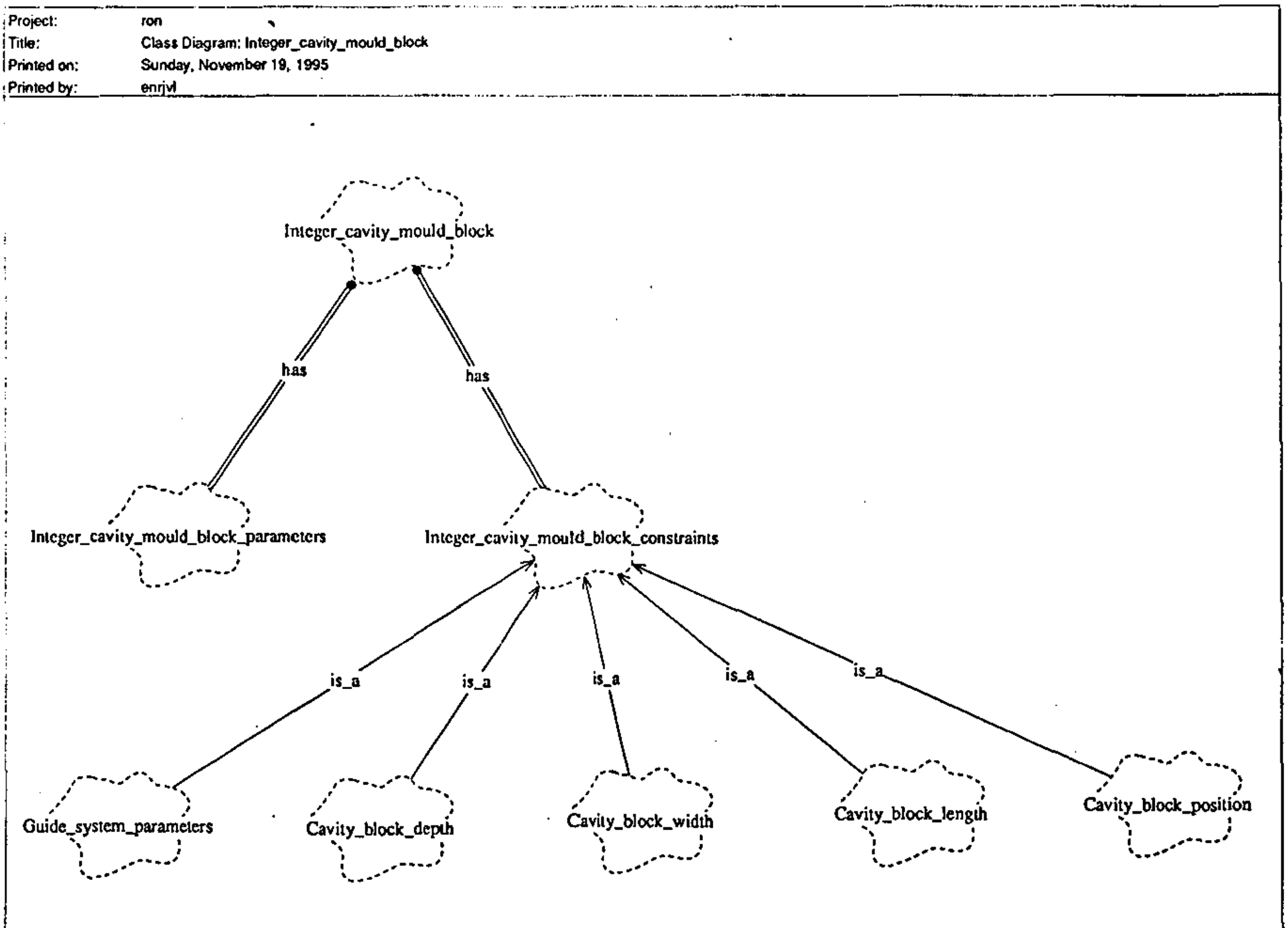


Figure 8.48. The Booch representation of integer cavity block feature constraints.

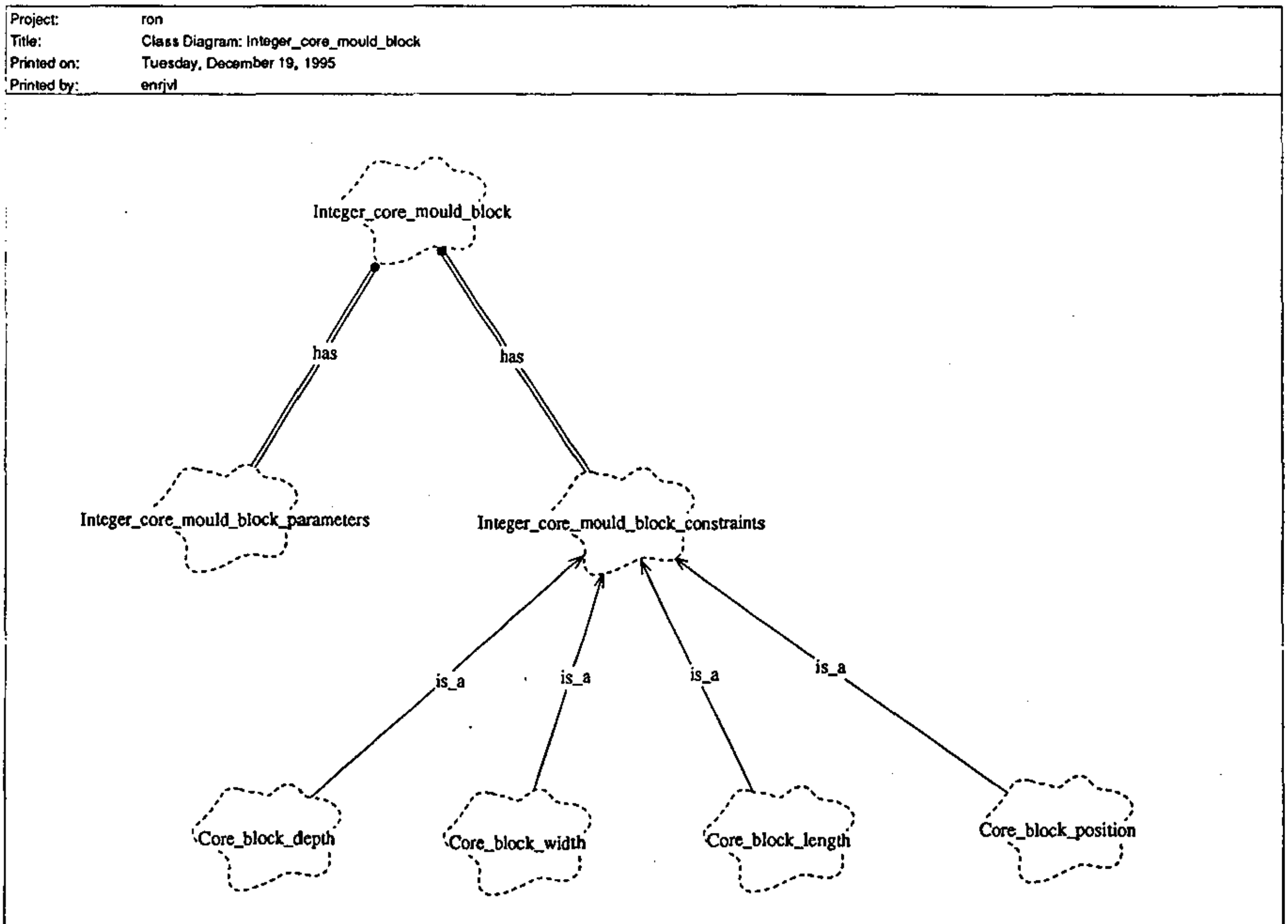


Figure 8.49. The Booch representation of integer core block feature constraints.

8.3.3.5. The representation of the cavity and core features hierarchy.

Cavity and core features can be categorised as either impression or plate features. There is an association between cavity or core impression features and the mouldability viewpoint as the impression features represent the corresponding part of a cavity or core. The cavity plate features have relations with the cavity impression and the other mould system elements, similarly for the core. Of the impression features, the rim, boss, hole and slot are all local inserts which means that a separate plate or 'local insert' is placed in the impression to create the geometry. In the plate features an integer cavity or core land can be an inner land or a peripheral land. An integer cavity or core inner land is either a rectangular land or a circular land. These relations are reflected in the features hierarchies as shown in Figure 8.50 (cavity) and 8.51 (core).

8.3.4. Feeding systems.

The capture of feeding system behaviour is very important as it has a significant influence on a part performance and the moulding process. For example the gate size affects the cycle time in a mould because when it freezes it acts as insulation between the impression and the feeding system. Whether or not the gate provides adequate feeding to the impression determines whether weld lines appear in the product etc. No previous work has considered the issue of feeding system design modification as the geometry of the product evolves. To support the designer in consideration of the feeding system design during the build up of product geometry it is necessary to provide the designer with the choices with respect to the configuration and geometry of a feeding system at each juncture in the development of a product. In this thesis a representation has been investigated to capture the process constraints of a feeding system, in order to support the designer in making such choices as the geometry of the product is built up.

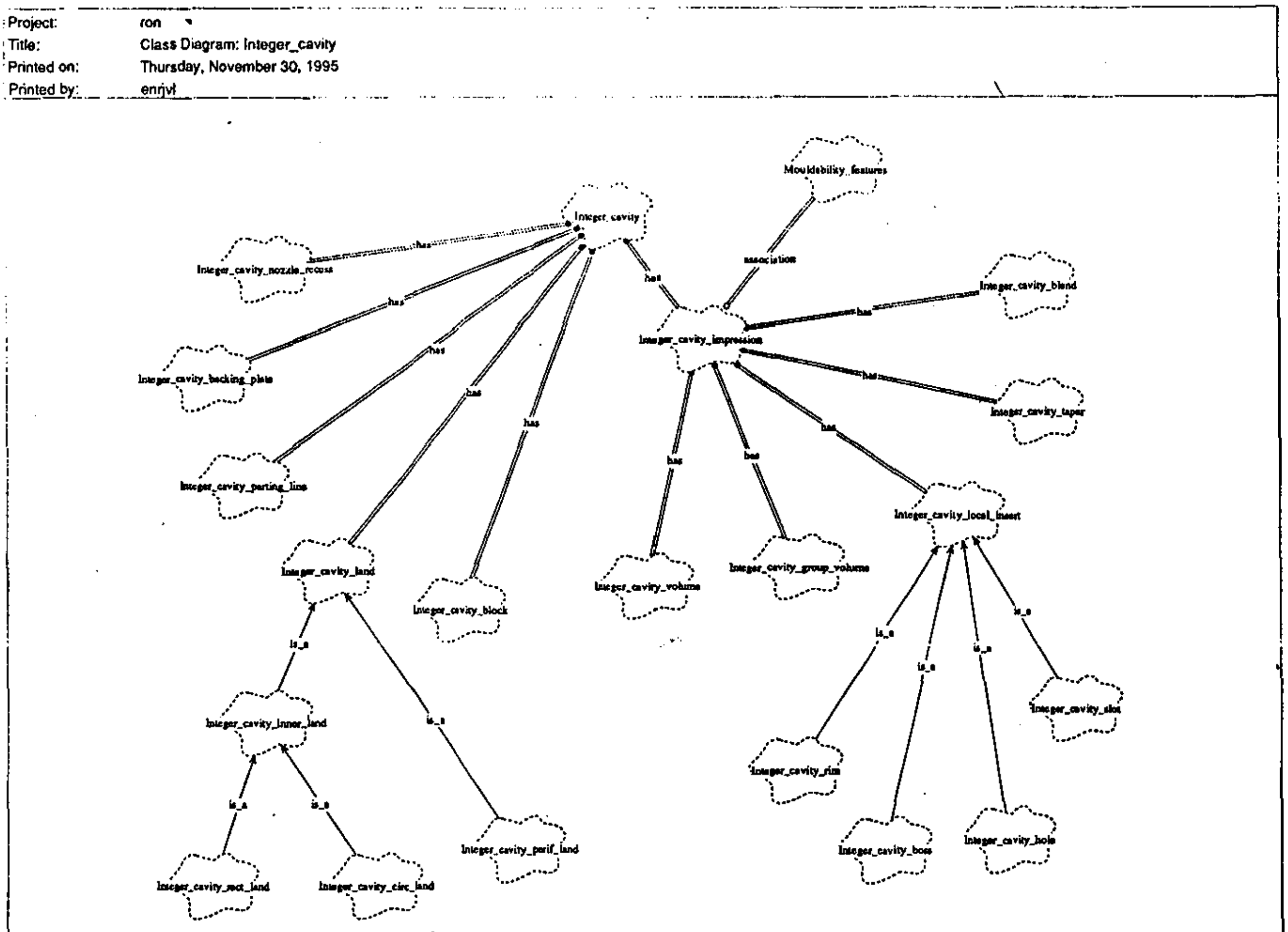


Figure 8.50. The Booch representation of integer cavity features hierarchy.

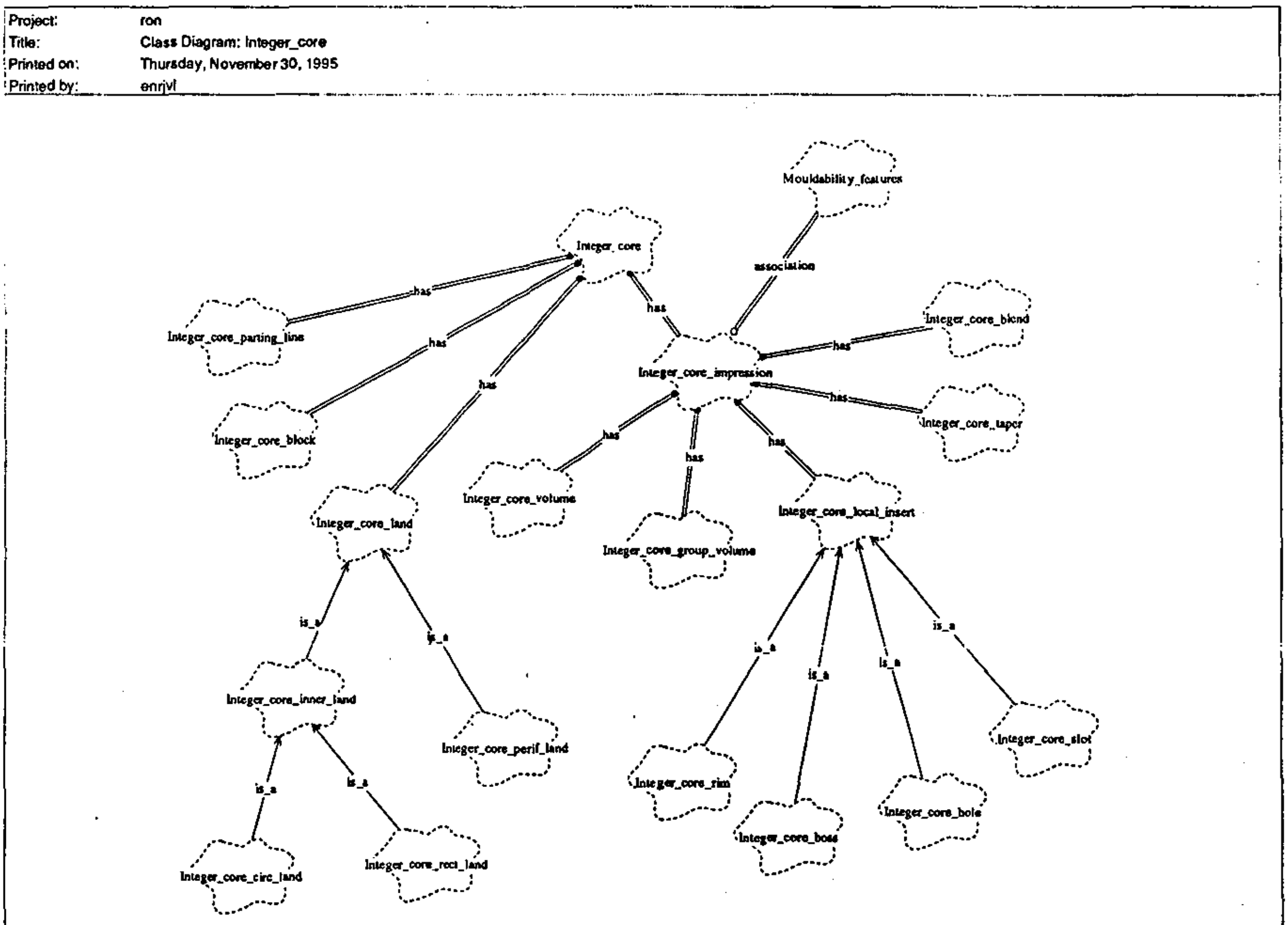


Figure 8.51. The Booch representation of integer core features hierarchy.

8.3.4.1. Process constraints of feeding system elements.

The process constraints that apply to a feeding system are: i) Adequate feeding, ii) Removal from mould, iii) Venting, iv) Mould manufacture considerations, and these are now considered.

i) Adequate feeding – A feeding system should provide adequate flow of molten material into the mould impression to ensure material reaches the extremities of the impression. This prevents weld lines or shrinkage occurring in a moulding. The rate of flow of material into the mould should be controlled to ensure that surface defects do not occur, for example as a result of turbulent flow in the mould.

ii) Removal from the mould – The geometry and configuration of a feeding system plus the spatial relationships with other mould elements should ensure it can be removed from the mould upon solidification at the same time as a moulding without extra removal operations being required.

iii) Venting – The geometry and configuration of a feeding system plus the spatial relationships with other mould elements should facilitate venting to avoid gas pockets forming.

iv) Mould manufacture considerations – The geometry and configuration of a feeding system plus the spatial relationships with other mould elements should not be such that it is impossible or significantly more difficult to manufacture a mould or a weakness in mould construction is enforced that may fail during mould operation.

8.3.4.2. Process constraints applied to feeding system features.

The ways of avoiding problems in the cavity impression have been identified from the literature using empirical formulae and by the author in Figures 7.57 to 7.62, which show design rules linked to feeding system features. The feeding system features are: ring gate, tab gate, sprue gate, sub–surface gate, diaphragm gate, rectangular edge gate, overlap gate, fan gate, pin gate, film gate, trapezoidal runner, modified trapezoidal runner, circular runner, hexagonal runner and main feeding sprue. Examination of each of these features in turn has enabled the author to ascertain how process constraints can be applied. Examples of the application of process constraints to feeding system features are described below:

Rectangular edge gate:

The attributes that relate to a *rectangular edge gate* are rectangular edge gate position, rectangular edge gate land length, rectangular edge gate width and rectangular edge gate depth.

i) Rectangular edge gate position – It can be seen from Figure 7.57 that the position of a rectangular edge gate is constrained by the needs of feeding the cavity impression, the strength requirements to cope with mould operation, and the requirements of ejection and mould manufacture. From Figure 7.57, the problems that can arise with respect to the position of a rectangular edge gate are i) a gate position is inside the edge of the cavity, ie too near the centre of the impression, ii) a gate position is on the parting line but away from the edge of the impression, iii) a gate position is away from the parting line. In the first scenario the gate land length (see Figure 7.58) is reduced and this can lead to a weakness in the mould construction, resulting in wear and/or eventual failure during mould operation, Pye (1989). Obviously in scenario ii) the gate is not attached to the component and no feeding can occur. In scenario iii) the gate and runner system cannot be ejected and so the component is non–mouldable. Moreover such a gating and runner system could not be machined into the mould block, making the mould non–manufacturable.

ii) Rectangular edge gate land length – The pressure drop across a gate is determined by the land length, Dym (1987), and too large a land length causes an excessive pressure drop across the gate and inadequate filling of the impression. The land length should therefore be kept to a minimum. However too small a land length causes a weakness in the mould, as described above for position scenario i). The recommended size of land length for a rectangular edge gate is between 0.5 mm and 0.75 mm, Pye (1989).

iii) Rectangular edge gate depth – The depth of a gate controls the time for which a gate remains open. This must be long enough for material to reach the extremities of the impression. To identify the recommended depth of a rectangular edge gate the empirical formula $h = nt$ can be used, where h is the depth of gate (mm), t is the wall section thickness (mm) and n is a material constant, Pye (1989).

iv) Rectangular edge gate width – The cross sectional area of the gate controls the rate at which material enters the impression, and if the depth is established as above the width is the controlling dimension for the flow rate. The gate width is related to the surface area of the cavity (mm) using the empirical formula W = n*SQRT(cavity_area)/30, where W is the width dimension of a rectangular edge gate and n is a material constant, Pye (1989).

Figure 8.52 shows how the captured design rules of a rectangular edge gate can be represented in the Booch methodology. The detailed representation of the design rules of a rectangular edge gate feature in the EXPRESS language is shown in Appendix 4.

<u>Circular runner:</u>

The attributes that relate to a *circular runner* are circular runner position, circular runner orientation, circular runner length and circular runner diameter. These attributes are described below:

i) and ii) Circular runner position and Circular runner orientation – As shown in Figure 7.59 the position and orientation of a circular runner is constrained by the required geometric relationship with a gating system. The required geometric relationship depends on the type of gating system.

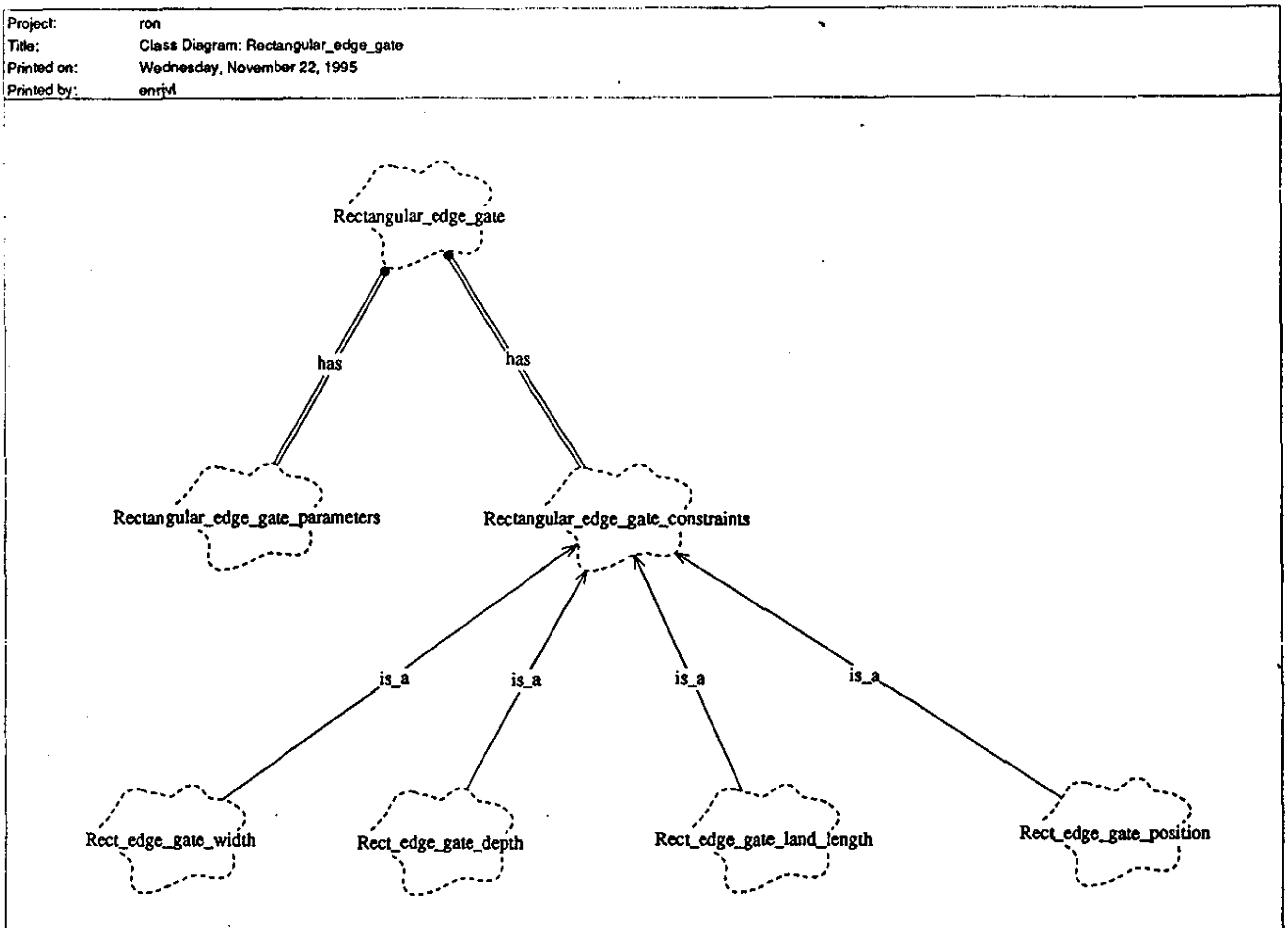


Figure 8.52. The Booch representation of rectangular edge gate feature constraints.

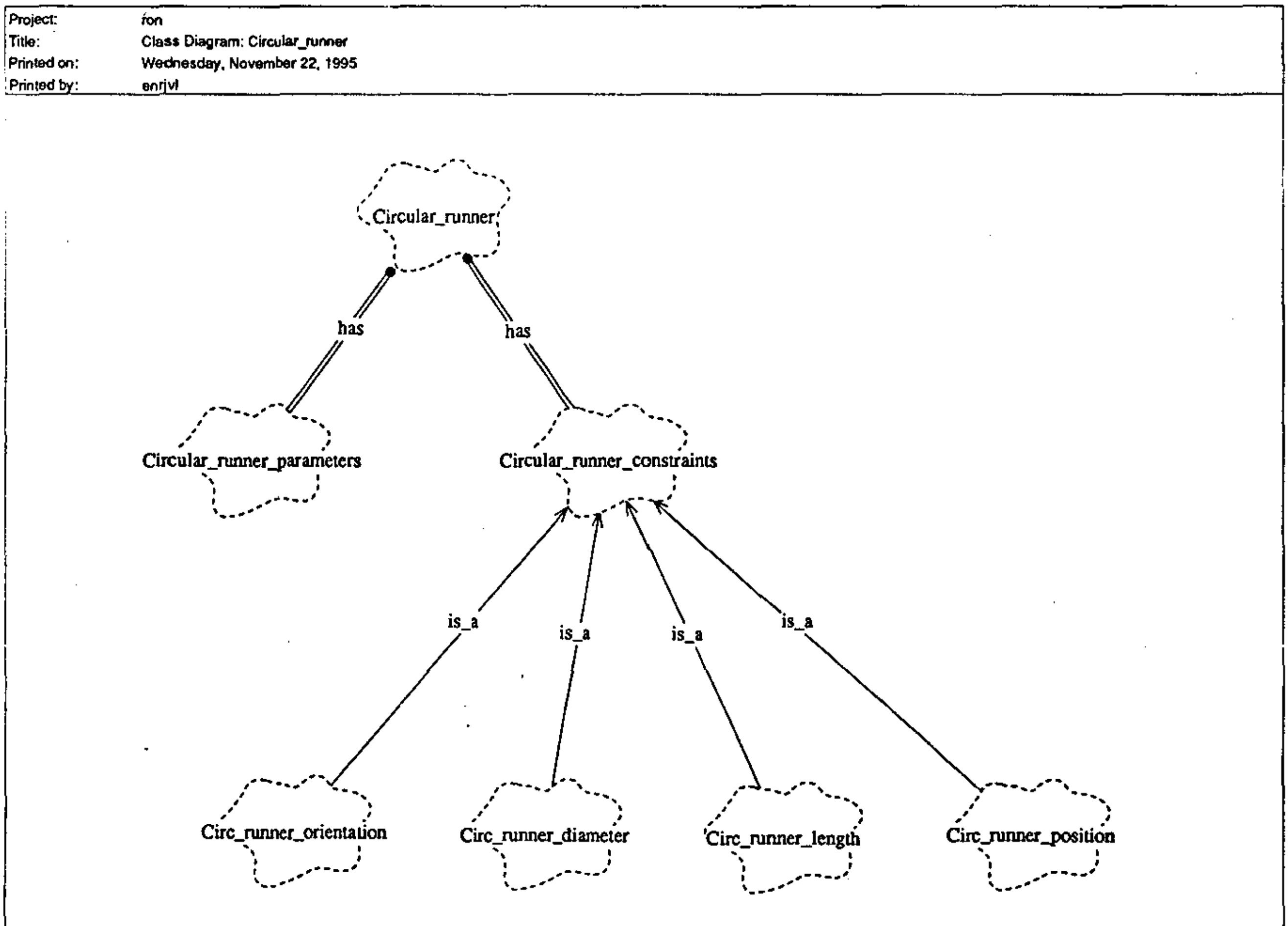


Figure 8.53. The Booch representation of circular runner feature constraints.

iii) Circular runner length – As shown in Figure 7.60, the length of a circular runner is determined by the distance between the runner position and the centre of a main feeding sprue. A main feeding sprue must be a minimum distance from the cavity impression to avoid a hot sprue causing local heating effects in the cavity.

iv) Circular runner diameter – The diameter of a circular runner can be related to the weight of a moulding and the runner length using the empirical formula *D = (SQRT(part_weight)*POW(runner_length,0.25))3.7*, where D is the runner diameter and POW(runner_length,0.25) is the fourth root of the runner length., Pye (1989). Additionally a minimum runner diameter of 2 mm is recommended in order that a runner does not freeze before the cavity is filled, and a maximum diameter of 10 mm is recommended so that the solidification of a runner does not control the injection cycle time.

Figure 8.53 shows how the captured design rules of a circular runner can be represented in the Booch methodology. The detailed representation of the design rules of a circular runner feature in the EXPRESS language is shown in Appendix 4.

8.3.4.3. The representation of the feeding system features hierarchy.

An element in a feeding system is either a gating system, a runner system or a main feeding sprue. A gating system can be a tab gate, sub–surface gate, overlap gate, fan gate, sprue gate, film gate, ring gate, rectangular edge gate, pin gate or a diaphragm gate. A runner system can be a circular runner, a hexagonal runner, a trapezoidal runner or a modified trapezoidal runner. As identified in Chapter 7, a gating system has an association with a mouldability gate feature. The above is reflected in the structure of the feeding system features hierarchy shown in Figure 8.54.

Project:	ron
Title:	Class Diagram: Feeding_system
Printed on:	Thursday, November 23, 1995
Printed by:	enrjvl

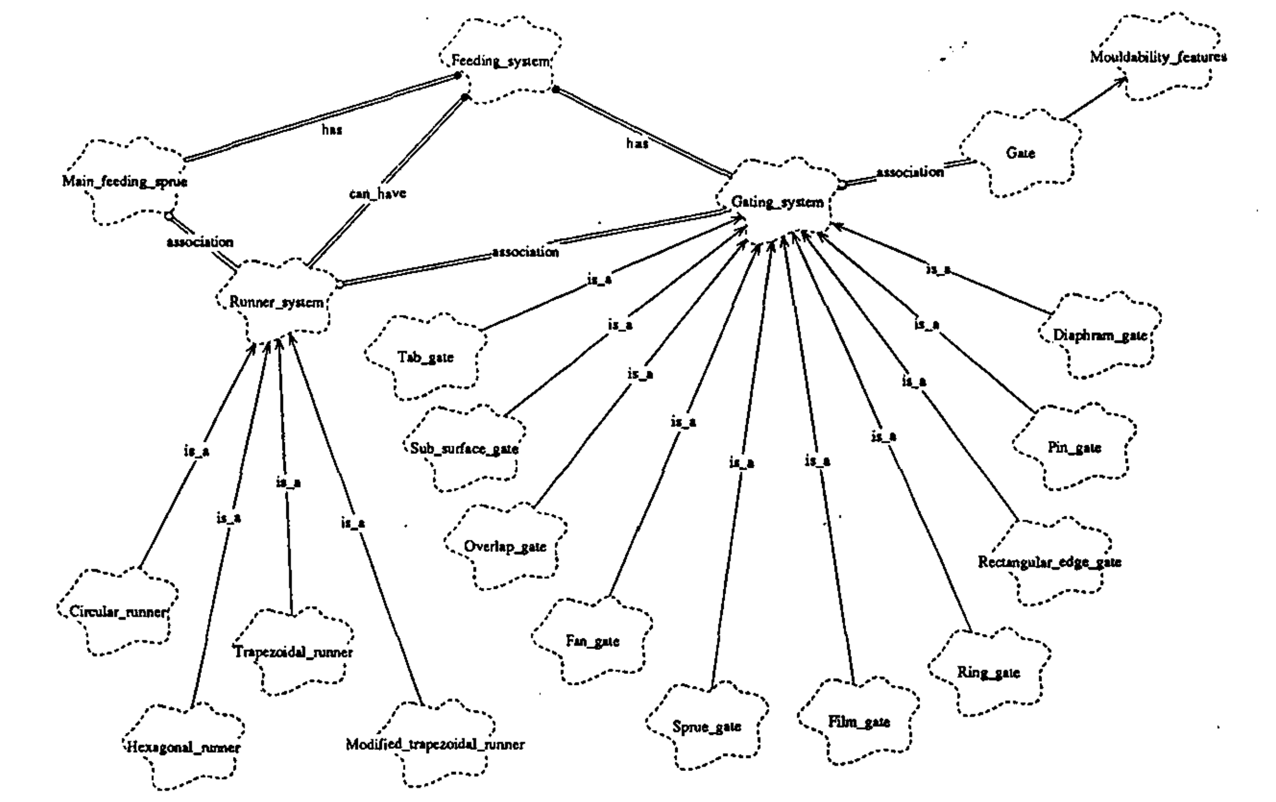


Figure 8.54. The Booch representation of the feeding system features hierarchy.

Project:	ron
Title:	Class Diagram: Integer_cavity_cooling_system
Printed on:	Thursday, November 23, 1995
Printed by:	enrjvl

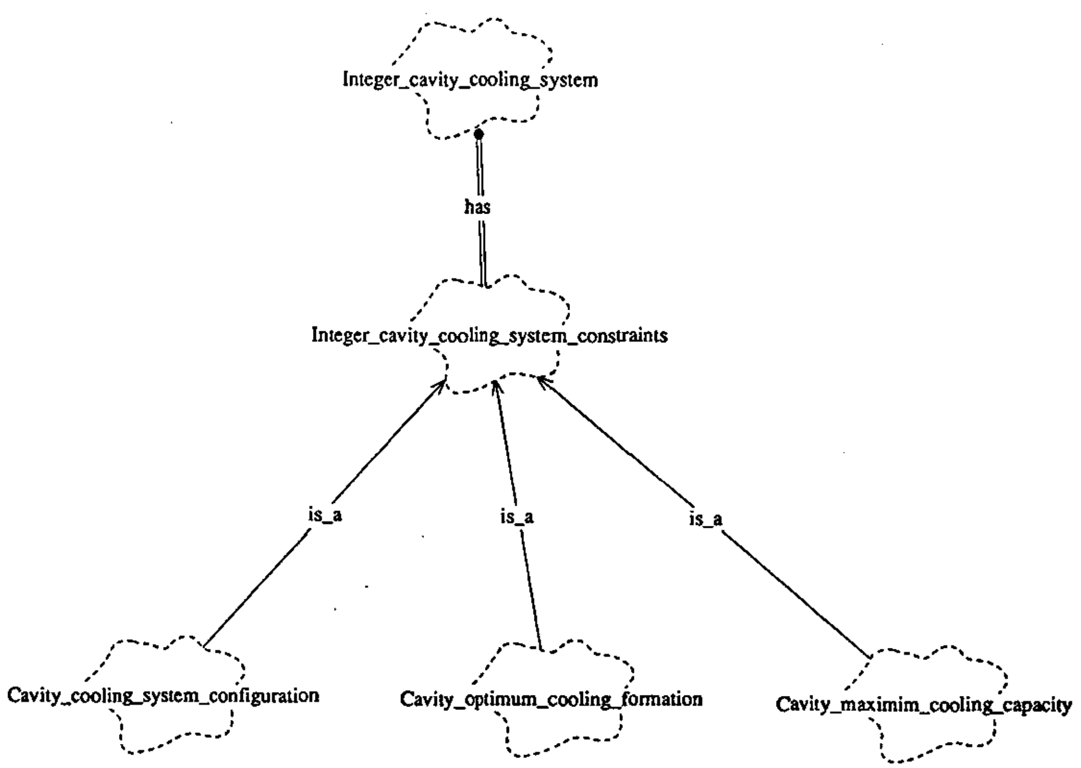


Figure 8.55. The Booch representation of integer cavity cooling system constraints.

8.3.5. Cooling systems.

No previous research has been found that considers the issue of cooling system design modification as the geometry of the product evolves. To support the designer in consideration of the cooling system design during the build up of product geometry it is necessary to provide the designer with the choices with respect to the configuration and geometry of a cooling system at each juncture in the development of a product. In this thesis a representation has been investigated to capture the process constraints of a cooling system, in order to provide the designer with such choices.

8.3.5.1. Process constraints of a cooling system.

Three process constraints apply to a cooling system: i) Even cooling, ii) Maximum cooling, iii) Non localised cooling, and the types of problems that can occur and the significant factors which need to be considered are as follows:

i) Even cooling – As far as possible an even cooling effect should be provided over the impression. This involves providing extra cooling to 'hot' areas of the moulding, eg at a gate location, and less cooling to 'cooler' areas, eg at the extremities. Uneven cooling of a moulding can cause a differential thickness over a moulding or differential shrinkage resulting in warpage.

ii) Maximum cooling – Providing the maximum cooling capacity of a cooling system in a mould provides the maximum capability to reduce the cycle time for each moulding.

iii) Non localised cooling – The optimum cooling formation of a cooling system provides maximum cooling capability while avoiding getting too close to the impression and causing local cooling effects.

8.3.5.2. Process constraints applied to cooling system elements.

From the work described in Chapter 7 relating to cooling systems, and the investigation described in this chapter it has been identified by the author that most design rules to capture process constraints in the cooling system must be applied in a Manufacturing model to higher level entities as they relate to the overall cooling effect of a system. The design rules in the lower level entities, ie the cooling system features described in Chapter 6, concern the relationship of an individual lower level entity with the design rules in the higher level entities. This relationship is dependant upon which part of a cooling system a lower level entity is used for in an instantiation. The higher level entities are cavity cooling system, core cooling system, deep core cooling system and shallow core cooling system. The ways of providing the optimum cooling configuration and avoiding problems have been investigated by the author as illustrated in Figures 7.63 to 7.69, which show design rules linked to higher and lower level cooling system elements. The cooling system features are standard flow way, baffle flow way and baffle blade. Examination of each of these features in turn and the higher level entities has enabled the author to ascertain how process constraints can be applied. Examples of the application of process constraints to the higher level cooling system entities are described below:

<u>Integer cavity cooling system:</u>

Attributes that relate to an *integer cavity cooling system* are described below:

i) Cavity cooling system configuration – As shown in Figure 7.64 the appropriate configuration of an integer cavity cooling system is related to the type of gating system and the number of gates, whereby the best cooling system configuration promotes an even cooling over the moulding. If a cavity impression is side gated using a single gating system or undergated acentrally, a 'U' tube configuration is recommended. If a cavity impression is undergated centrally or multigated a 'paired' tube configuration is recommended.

ii) Cavity maximum cooling capacity – The diameter of the tubes in a cavity cooling system is constrained by the requirement to provide the maximum cooling capacity. A diameter less than 7 mm provides insufficient surface area for heat exchange in a mould, one greater than 10 mm requires too much coolant to be pumped around a mould, Pye (1989). As shown in Figure 7.67 a), subject to the above limits the recommended diameter is that which provides the highest total surface area for heat exchange. This recommended diameter is identified based on the maximum number of tubes of a given diameter that can fit alongside the cavity impression, multiplied by the surface area per unit length of that diameter.

iii) Cavity optimum cooling formation – As shown in Figure 7.68 a) in order to avoid local cooling effects in the impression there is a constraint on the tubes in a cooling system to provide a clearance of 16 mm around the nearest parts of the cavity impression, Pye (1989).

Figure 8.55 shows how the captured design rules can be represented in the Booch methodology. The detailed representation of the design rules in the EXPRESS language is in Appendix 4.

<u>Integer core deep cooling system:</u>

The attributes that relate to an *integer core deep cooling system* are type of deep cooling, deep core maximum cooling capacity and deep core optimum cooling formation. These attributes are described below:

i) Type of deep cooling – As shown in Figure 7.65 a) of the three common configurations of deep core cooling systems the stepped circuit system has openings into the impression which require re–sealing and the angled hole system does not cool the periphery of the core base. Both of these configurations have a lesser cooling capacity than the baffle tube system and therefore this is the only type of deep core cooling system considered in the present work.

ii) Deep core maximum cooling capacity – The diameter of the baffle tubes in a deep core cooling system is constrained by the requirement to provide the maximum cooling capacity. Below a diameter of 12 mm there is insufficient surface area for heat exchange in a mould, while above a diameter of 16 mm too much coolant needs to be pumped around a mould, Pye (1989). As shown in Figure 7.67 b), subject to the above limits the recommended diameter is that which provides the highest total surface area for heat exchange. This recommended diameter is identified based on the maximum number of tubes of a given diameter that can fit inside the core impression multiplied by the surface area per unit length of that diameter.

iii) Deep core optimum cooling formation – As shown in Figure 7.68 b) in order to avoid local cooling effects in the impression there is a constraint on the tubes in a cooling system to provide a clearance of 16 mm around the nearest parts of the impression, Pye (1989).

Figure 8.56 shows how the captured process constraints of an integer core cooling system can be represented in the Booch methodology. The detailed representation of the design rules of an integer core deep cooling system in the EXPRESS language is shown in Appendix 4.

8.3.5.3. The representation of the cooling system features hierarchy.

A cooling system is either an integer cavity cooling system or an integer core cooling system. An integer cavity cooling system has either a pair tube configuration or a U tube configuration. An integer core cooling system can be an integer core deep cooling system or an integer core shallow cooling system. An integer core shallow cooling system has a paired tube configuration, a U tube configuration, a Z tube configuration or a single tube configuration. An integer core deep cooling system is either a stepped circuit system, an angle hole system or a baffle system. The above is reflected in the structure of the cooling system features hierarchy shown in Figure 8.57.

Project:	ron
Title:	Class Diagram: Integer_core_deep_cooling
Printed on:	Saturday, November 25, 1995
Printed by:	enrjvl

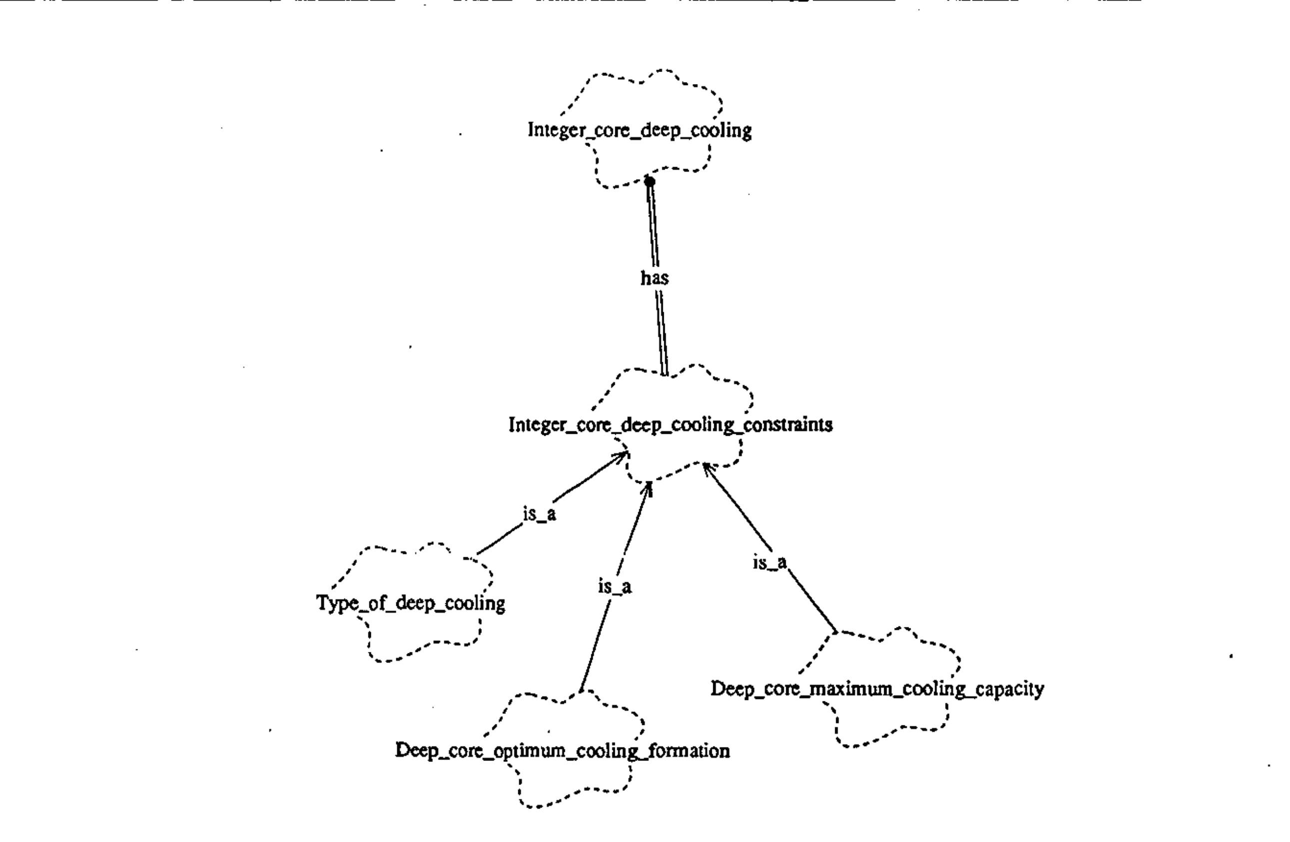


Figure 8.56. The Booch representation of integer core deep cooling system constraints.

Project:	ron
Title:	Class Diagram: Cooling_system
Printed on:	Thursday, November 23, 1995
Printed by:	enrjvl

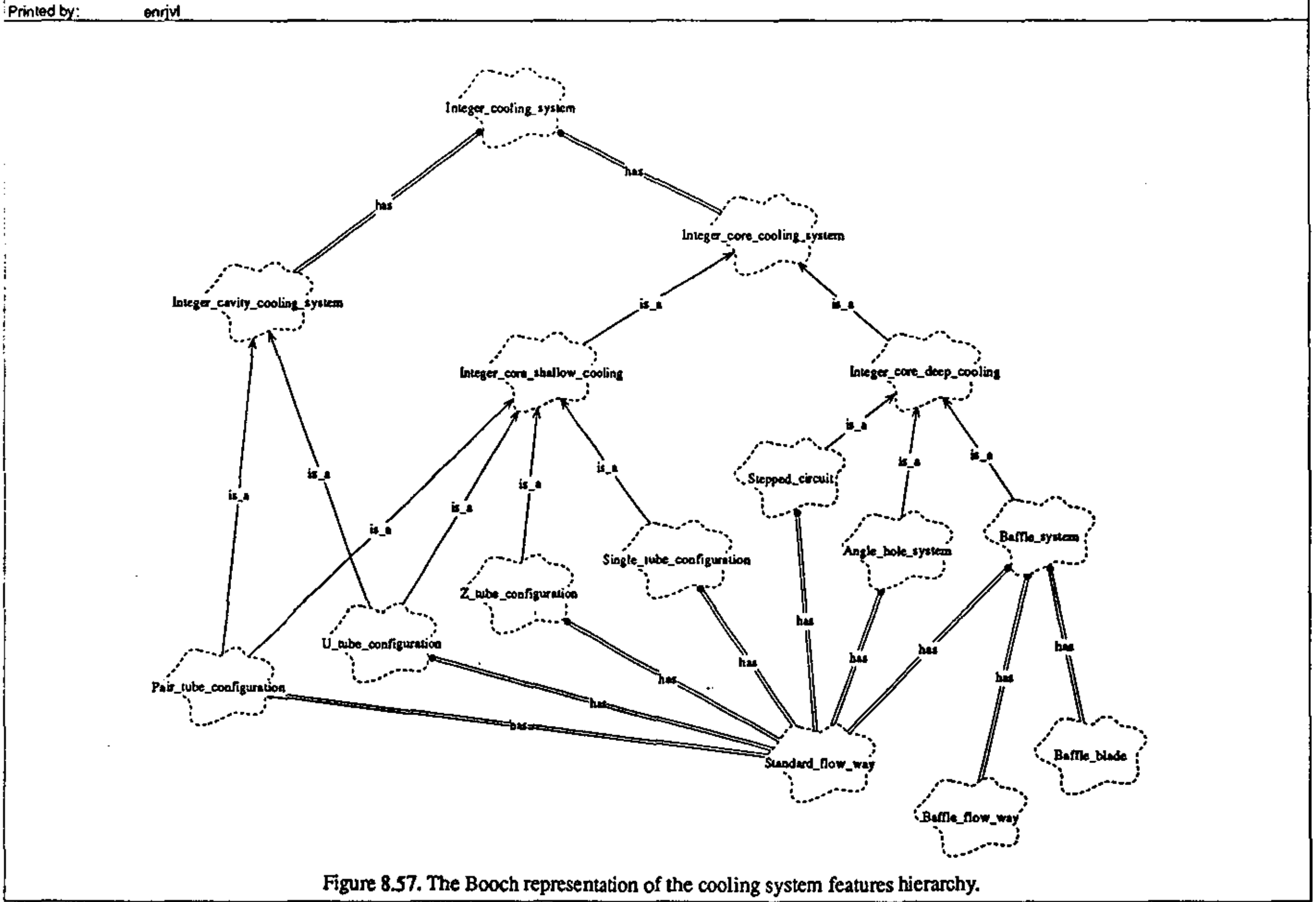


Figure 8.57. The Booch representation of the cooling system features hierarchy.

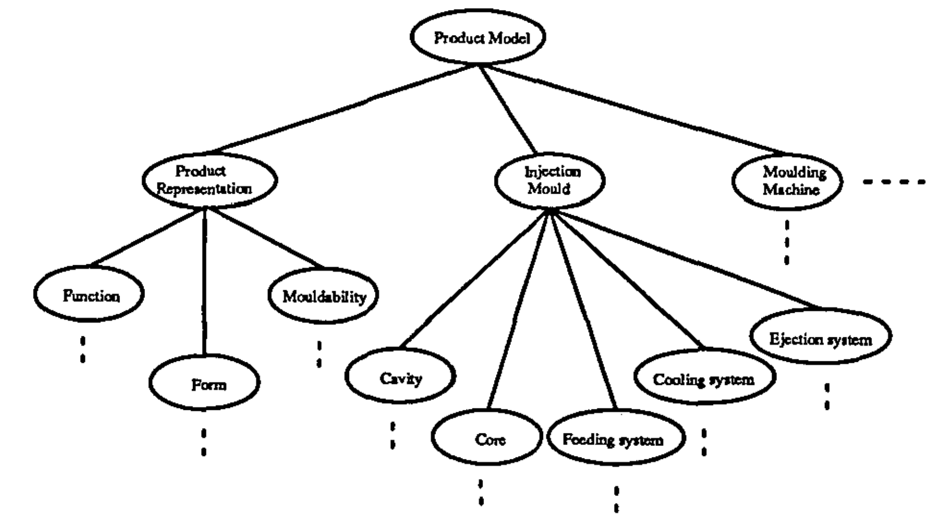


Figure 8.58. Conceptual data structure of the Product Model.

8.4. PRODUCT MODEL.

The Product model is required to represent the product and mould from multiple design viewpoints as they are built up by the designer. The appropriate structures must exist in the Product model to facilitate analysis and re–analysis of the current state of a product and a mould for multiple design viewpoints. To enable design support the model must be populated in such a way as to facilitate the association of the product specific data therein with the general injection moulding process capabilities data in the Manufacturing model and with the functional constraints in the Product Range Model.

8.4.1. General Product model structure.

Figure 8.58 shows the conceptual structure of a Product model based on the features sets identified by the author in Chapter 6. This work has not considered the ejection system or the moulding machine. Under each viewpoint in the Product model, product specific data is instantiated using the translation processes described in Chapter 7. In this way, with the appropriate links between the viewpoints, the Product model can represent the current state of a product and mould in a way which facilitates analysis and re–analysis of the current state of multiple viewpoints. The Booch methodology has been used to create a formal representation of the Product model structure and Figure 8.59 shows the scope of the auth–

Title:	Class Diagram: Product_model
Printed on:	Thursday, November 30, 1995
Printed by:	enrjvl

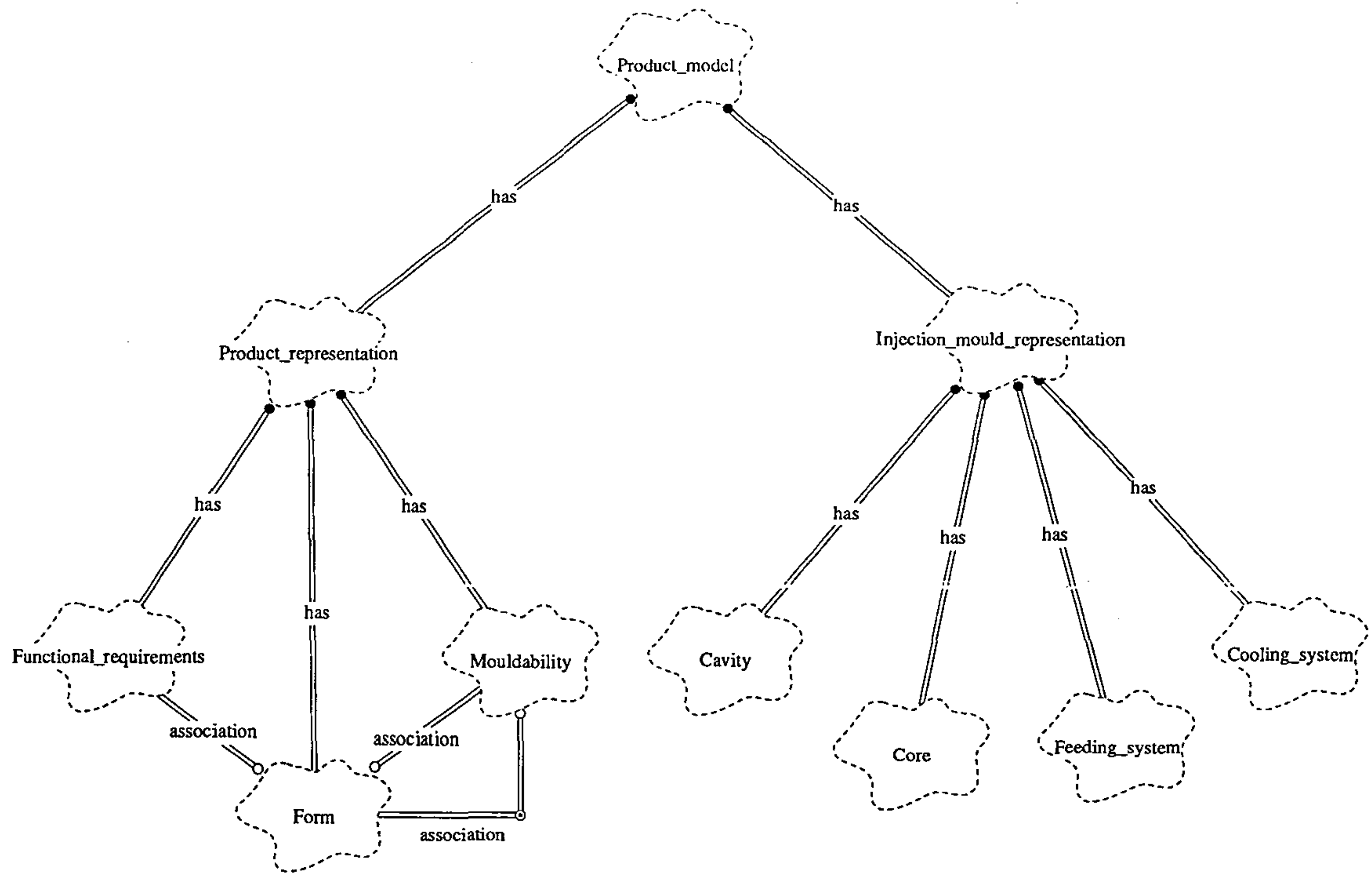


Figure 8.59. The Booch representation of the general top level Product model structure.

ors investigation into the Product model, by showing the top level structure.

8.4.2. Capturing the structure of multiple viewpoint data in the Product model.

The entities that are instantiated in the Product model for each viewpoint have been identified by the author in Chapter 6. Further structural requirements for the Product model were identified in Chapter 7 to enable the translation processes. This section reviews the identified structural requirements of the Product model and discusses any further requirements.

As described in section 7.3 the approach of this work is to represent *functionality* using a combination of function and form data in the Product model. The association of function and form is driven by the designer as he/she builds up the product geometry to achieve a function on a particular product. However a backward link is required to enable evaluation of functionality in response to changes from other design viewpoints. This link cannot be driven by the designer and therefore a software provision is required to check functionality. This link is created at the time the forward link is being driven by the designer, and is shown in Figure 7.3. This link is represented using the Booch methodology in the top level of the Product model structure, as shown in Figure 8.59.

From section 7.4, in order to evaluate the *mouldability* of a specific product a knowledge is required of adjacent mouldability geometry. In order to identify adjacent mouldability instantiations it is necessary to go via the equivalent form feature instantiation to the associated functional data to get information on the type of adjacent form feature instantiation to identify. The adjacent form feature is identified and the equivalent mouldability instantiation can then be identified. Thus a two way link is required in the Product model between mouldability feature instantiations and the equivalent form feature instantiations. A backward link is required to get from a new mouldability instantiation, via form to the functional data, and a forward link is required to get from an identified adjacent form feature instantiation to the equivalent mouldability instantiation. These links are shown in Figure 7.8.

These links are represented using the Booch methodology in the top level of the Product model structure, as shown in Figure 8.59.

Secondary mouldability features have no form equivalent and therefore no direct link with the form is possible. However to evaluate the effects of secondary mouldability instantiations from other viewpoints in injection moulding design, and for the mouldability evaluation of instances of blends which involves a knowledge of adjacent mouldability geometry, a link is still required to the form. Therefore a secondary mouldability instantiation can be associated with the form viewpoint via a backward link to the associated primary mouldability instantiation, as shown in Figure 7.22. This link is represented using the Booch methodology as shown in Figure 8.30.

After translation of primary mouldability instantiations to the *cavity and core* design viewpoints, consideration of process constraints may change the geometry of the cavity or core feature instantiations. To enable concurrency these changes must be reflected in the corresponding mouldability geometry in the Product model. This is necessary in order that the consequences of design decisions in the cavity and core can be considered from the viewpoint of the product, thus achieving a link between the geometry of the product and that of the cavity/core. Therefore an association is required in the Product model between the cavity and core feature instantiations and the corresponding mouldability feature instantiations, as shown in Figures 7.36 and 7.54. Associations between the primary mouldability instantiations and the equivalent cavity/core feature instantiations in the Product model are also a prerequisite to the instantiation of integer cavity tapers and integer cavity blends or integer core tapers and integer core blends, as described in sections 7.5 and 7.6. The links between cavity and core features translated from the mouldability viewpoint and the mouldability features is represented using the Booch methodology as shown in Figures 8.50 and 8.51.

From section 7.7 the first part of a *feeding system* to be instantiated is a gating system. The type instantiated depends on the data in an equivalent mouldability gate instantiation. Once instantiated the process constraints of a gating system are evaluated, these being the appropriate position in relation to the geometry of the cavity/core. Adjustment of a gating system in response to feedback advice means that for consistency the equivalent mouldability gate instantiation must also be adjusted, so that a backward link is required from a gating system instantiation in the Product Model to an equivalent mouldability gate feature. Other individual elements of a feeding system are instantiated in relation to existing instantiations in a feeding system and the whole geometry of the impression and mould plates. Therefore individual instantiations do not require links to those of other viewpoints in the Product model. The link between a gating system and a mouldability gate feature is represented using the Booch methodology as shown in Figure 8.54.

A *cooling system* is instantiated in a mould in relation to the whole geometry of the impression and feeding system (section 7.8) and therefore no links are required between individual elements of a cooling system instantiation in the Product model and those of another viewpoint.

The above links, together with the entities in the features sets identified in Chapter 6 provide a representation that facilitates the association of product specific data therein with the general injection moulding capabilities data in the Manufacturing model and with the functional constraints in the Product Range Model. The Product model structure also enables interaction between applications in an injection moulding strategist application. The use of the above Product model structure is described in Chapter 9 for an experimental injection moulding strategist application.

Chapter 9

An experimental injection moulding strategist.

9.1. INTRODUCTION.

This chapter describes the design of an experimental injection moulding strategist application to fulfil the requirements of supporting concurrent design for injection moulding. The general structure of an injection moulding strategist is shown in Figures 5.10 and 5.11. Based on the features sets identified in Chapter 6 applications have been implemented for the functional interface, mouldability, cavity/ core, feeding system and cooling system, together with higher level strategies for their interaction. The cavity and core applications include design support for the cavity and core plate.

Section 9.2 describes the overall strategy for interaction of applications within the injection moulding strategist. Sections 9.3 and 9.4 describe the authors investigation into the design of the functional interface and mouldability applications. Section 9.5 describes the strategy for interaction of mould design applications. Sections 9.6 and 9.7 describe the authors investigation into the design of the cavity and core applications. Sections 9.8 and 9.9 describe the authors investigation into the design of the feeding system and cooling system applications.

9.2. OVERALL STRATEGY.

In order for the strategist applications to interact such that design support is provided in a concurrent manner, an overall strategy is required to control these interactions. The interactions must take account of the requirements of each viewpoint in injection moulding design. From Chapter 8, the functional viewpoint requires an initial product definition phase to evaluate those product functions with precedence over the others. The functional viewpoint drives the design process, and therefore for concurrency in product design the mouldability strategist must be triggered by any activity of a functional interface that results in the creation or changing of geometry in the Product model. Equally mouldability changes may effect the functionality viewpoint, eg the creation of tapers. Thus there is a requirement for a functional interface to support re–analysis of the functional viewpoint in response to mouldability changes (backtracking).

In order to support concurrency between product and mould design an interaction between strategist applications is required to allow the designer to analyse mould design as the product evolves. Therefore after each change in product geometry it must be possible to analyse the consequences for mould design. Furthermore, it must be possible to analyse all the options for mould design at any juncture in the design of the product.

The overall strategy to support concurrent product and mould design is shown in Figure 9.1. If a product already exists in the Product Model the functional strategist is triggered to support the designer in addressing remaining product functions in any order he wishes. If a new product is created the functional strategist is triggered to support the designer in instantiating the initial product definition. During these phases any changes to geometry as a result of functional interface activity triggers the mouldability strategist application. After the initial product definition is instantiated or a product function is addressed the designer has a choice of addressing further product functions, re–analysis of functionality to asses the consequences of mouldability changes, going onto mould design or displaying geometry from the Product model. Each of these can be done in any order and a designer may go into and out of a particular application as many times as he wishes without using

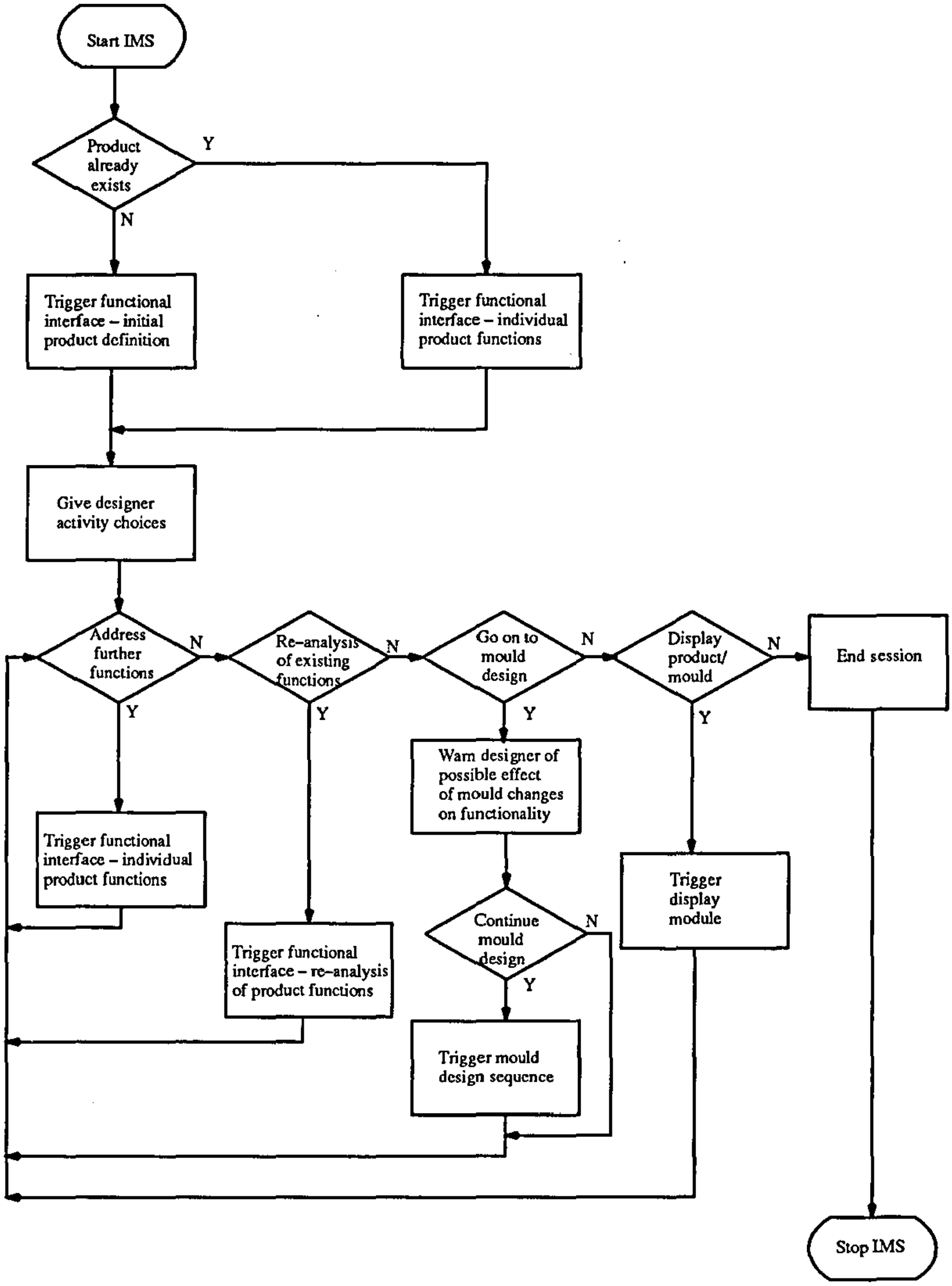


Figure 9.1. Overall strategy to support concurrent product and mould design.

the other applications. For example at a particular juncture in the product evolution a designer may try various different options in mould design by going through several cycles before building any more product geometry. In the mould design cycle the design is warned that changes to the mould design viewpoint will also change the geometry of the functional viewpoint and may therefore effect functionality. The mould design sequence is started by triggering the cavity design strategist and the applications within mould design then interact to support the designer in producing a complete mould design as described in section 9.5.

9.3. DESIGN FOR FUNCTION INTERFACE.

The functional interface application has three strategies; i) to support the designer in instantiating an initial product definition, ii) to support the designer in addressing subsequent product functions individually, and iii) to support re–analysis of functionality due to changes from other viewpoints, ie mouldability or mould design. The strategy for initial product definition is shown in Figure 9.2. It can be seen that the functional interface identifies the product ranges for which data is held in the Product Range Model and provides the designer with the choice. Once the choice is made the application identifies the number of product functions in the initial product definition and their order of precedence. For each of these functions in turn the designer is supported in defining a specification in the Product model, using data from the Product Range Model. In this way any conflicts between individual specifications can be evaluated before creation of any geometry. Subsequently, form/function data in the Product Range Model is used to support the designer in a choice of form to achieve each product function. Each form feature is evaluated for functionality upon instantiation, using adjacent geometry data from the Product model and functional constraints data in the Product Range Model. Taper applications on adjacent geometry are

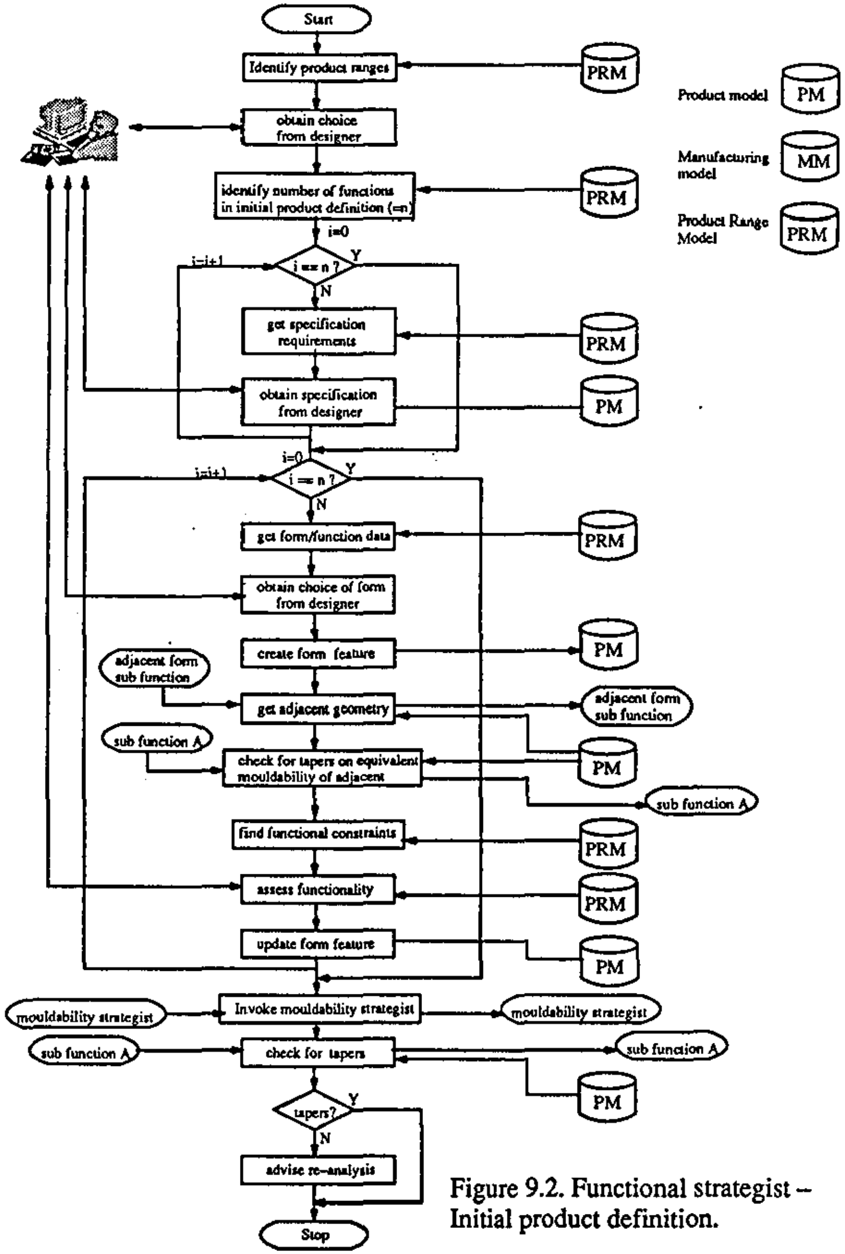


Figure 9.2. Functional strategist – Initial product definition.

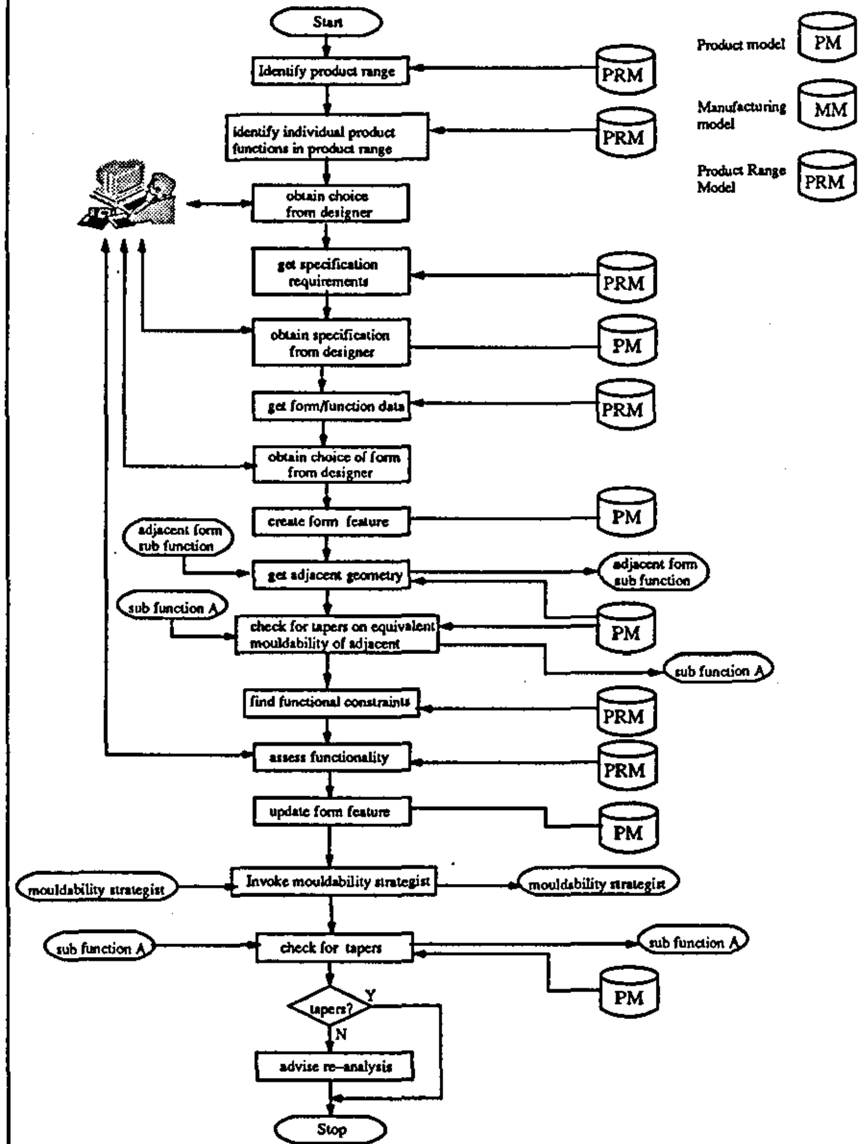


Figure 9.3. Functional strategist – Individual product functions.

included in the evaluation. A method of identifying adjacent form features in the Product model has been described in Chapter 7 using the links shown in Figure 7.3.

It can be seen that after functionality evaluation the mouldability strategist is triggered to evaluate the new form feature instantiation. The mouldability application is discussed in the next section. After mouldability analysis, if tapers have been applied to the product the designer is advised of the possible effect on functionality and which form features should therefore be re–analysed. The order of re–analysis is also advised. This should be the same as the order of instantiation, otherwise the design intent is lost and a re–analysis may not identify changes in functionality. An example of this situation is shown in the experimental chapter 10.

The strategy for addressing individual product functions is shown in Figure 9.3. This shows that the functional interface identifies those functions that must be individually addressed using data from the Product Range Model and gives the designer the choice of which order to address the functions. Once a product function is selected it can be seen that the strategy is identical to that in the initial product definition.

Figure 9.4 shows the strategy for re–analysis of the functional viewpoint. It can be seen that the designer can choose any form feature instantiation to be re–evaluated for functionality. A choice of form feature to re–evaluate does not have to be based on feedback advice and a designer can re–evaluate any part of a product for functionality at any time. The associated functional data to a form feature is identified by the links created by the functional interface during form feature instantiation, as shown in Figure 7.3. It can be seen that the strategy for re–evaluation is identical to the post form feature instantiation stages of strategies i) and ii), including the triggering of the mouldability application.

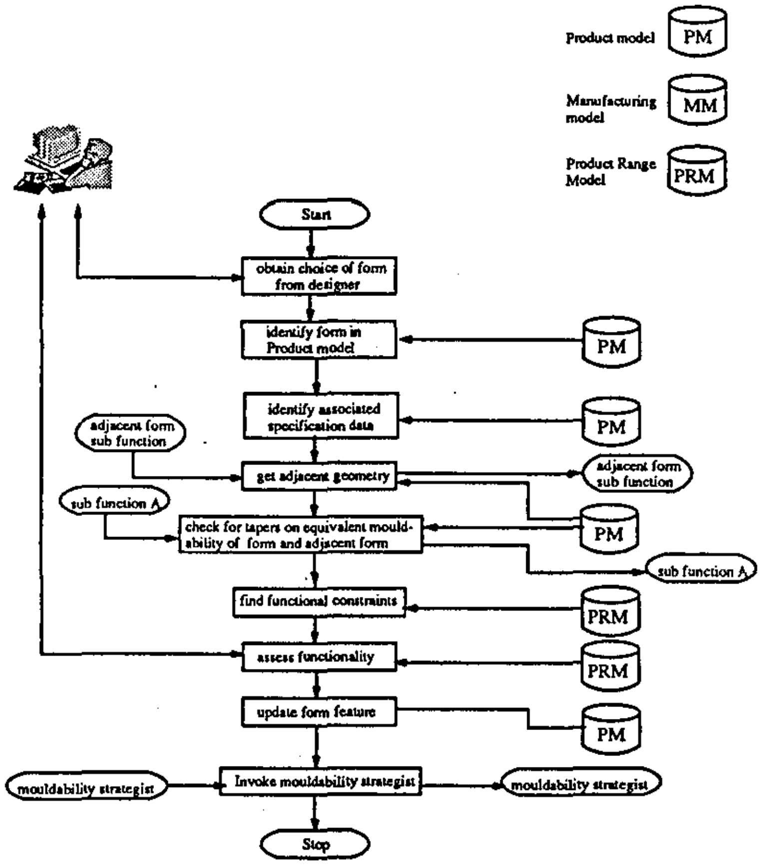


Figure 9.4. Functional strategist – Product function re–analysis.

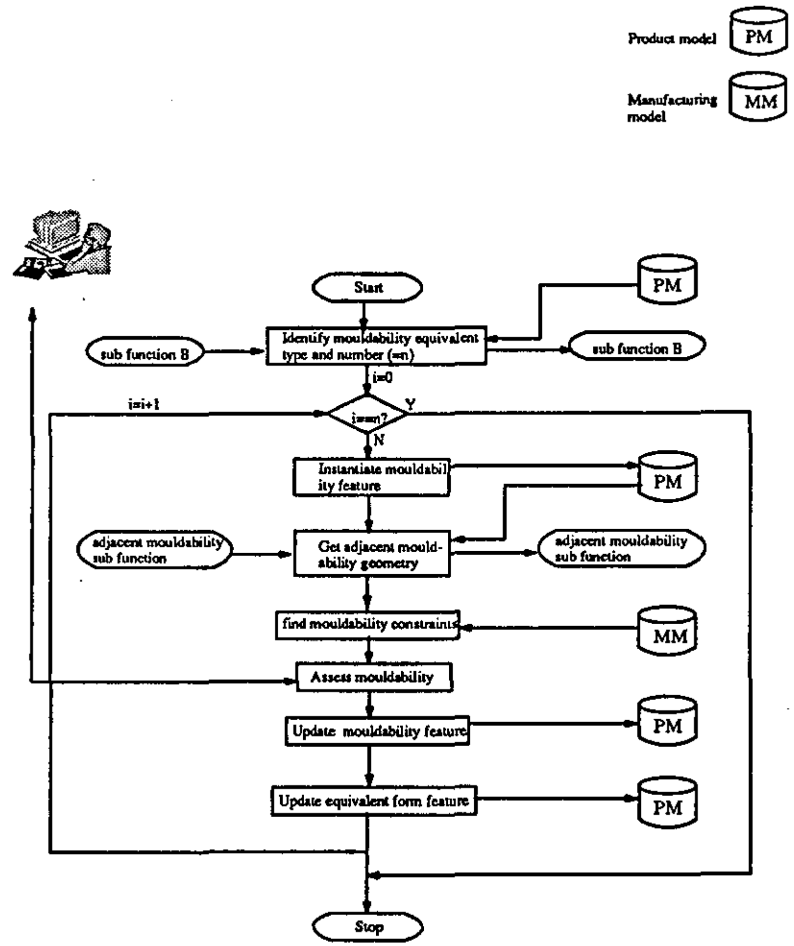


Figure 9.5. Mouldability strategist – Instantiation strategy.

9.4. MOULDABILITY STRATEGIST.

The mouldability strategist application has three strategies; i) for instantiation of mouldability features in response to the instantiation of a new form feature in the Product model, and ii) for evaluation of existing mouldability instantiations in response to changes to existing form feature instantiations in the Product model, and iii) for requested instantiations by the cavity, core or feeding system applications. The mouldability application is triggered by occurrence of one of the above. Figure 9.5 shows the strategy in response to the former occurrence. The first activity is to translate from form to mouldability. The translation process has been described in detail in Chapter 7. From Chapter 7, it may be that a one–to–many relationship exists between a form feature instantiation and the mouldability viewpoint. It can be seen that the mouldability application instantiates each mouldability equivalent in the Product model and evaluates each for mouldability upon instantiation. It can be seen that evaluation of mouldability is carried out using data on adjacent mouldability geometry from the Product model and mouldability constraints from the Manufacturing model. A method of identifying adjacent mouldability geometry in the Product model has been described in Chapter 7, using the links shown in Figure 7.8 and 7.22. It can be seen that any changes to a mouldability instantiation due to feedback advice is reflected in the equivalent form feature instantiation by updating of the form viewpoint by the mouldability application.

Figure 9.6 shows the strategy for re–evaluation of the mouldability viewpoint due to changes to an existing form feature instantiation. The mouldability equivalent of an existing form feature instantiation is identified using the links in the product model created by the mouldability application during mouldability feature instantiation, as shown in Figure 7.8 and 7.22. Thereafter, apart from the lack of an instantiation, the strategy is identical to

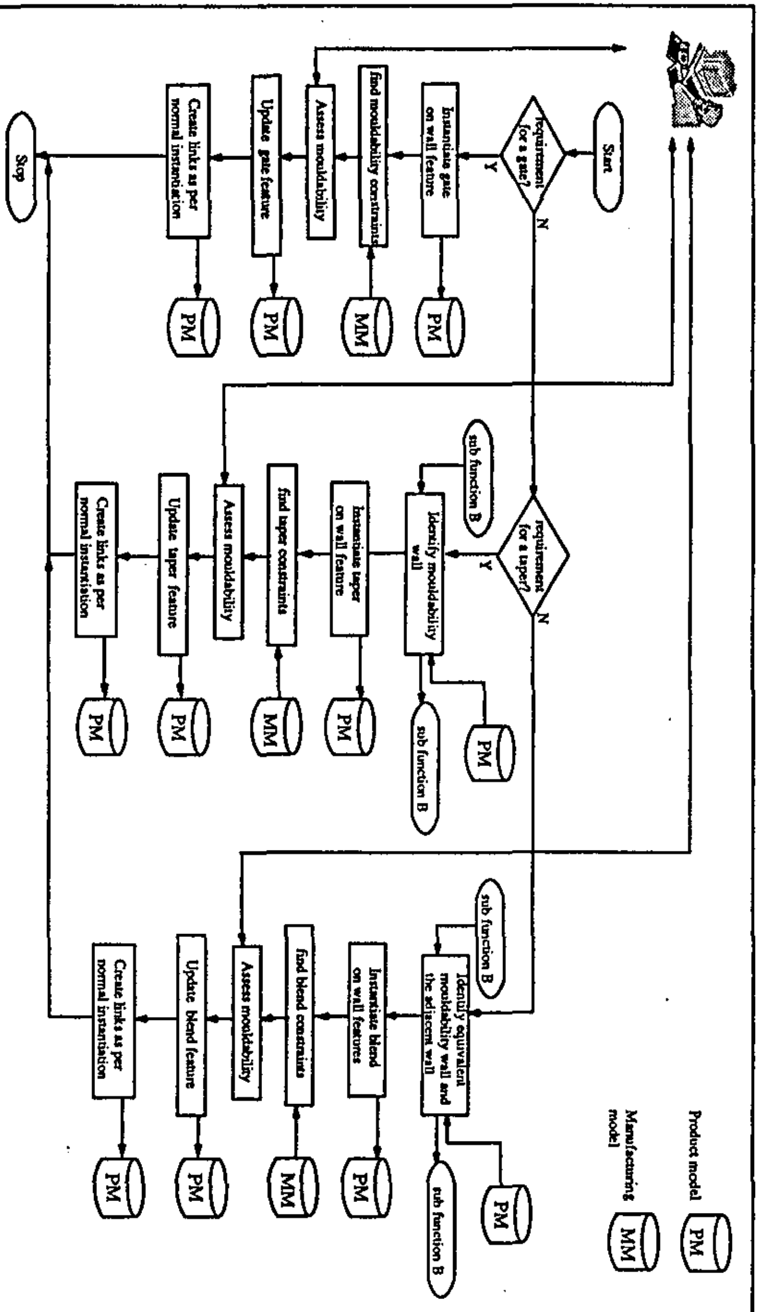


Figure 9.7. Mouldability strategist – Retrospective instantiation strategy.

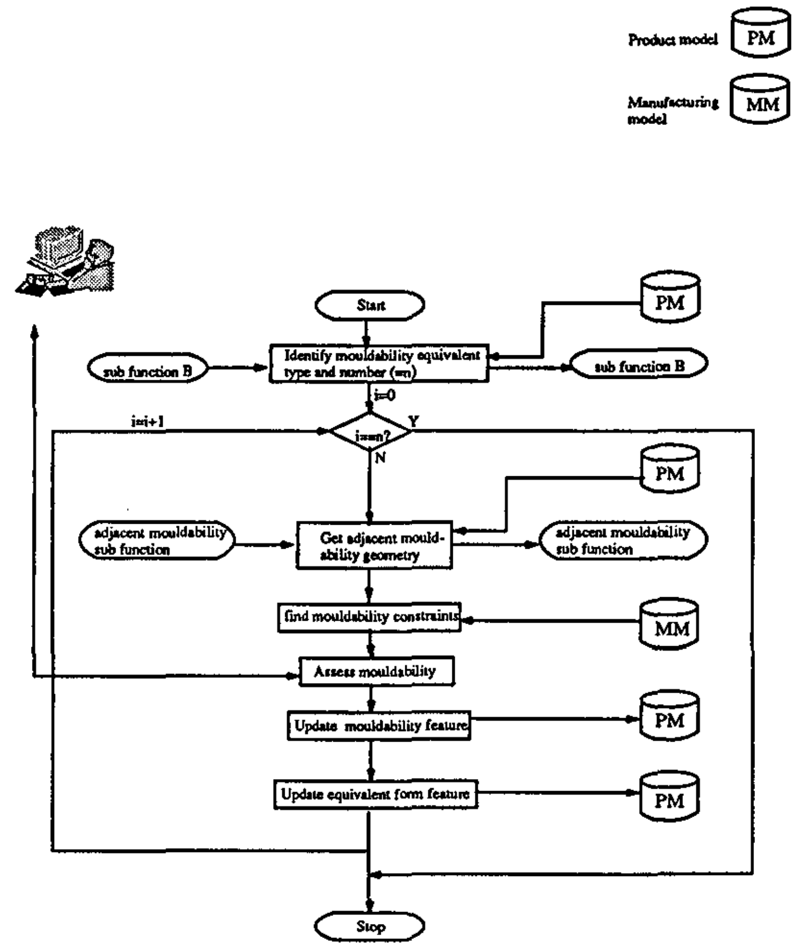


Figure 9.6. Mouldability strategist – Re-evaluation strategy.

the instantiation strategy in Figure 9.5.

Figure 9.7 shows the strategy for requested instantiations from other viewpoints, ie feeding or cavity/core. As described in section 9.8, the feeding system design strategist may request a gate instantiation if none exists on a product, and as described in section 9.6 and 9.7 the cavity or core design strategist may request instantiation of a blend or taper in response to a designer request to instantiate the equivalent entities in the cavity or core impression. From Figure 9.7, if a gate instantiation is requested the gated wall is chosen by the designer, and is therefore already known. Thus a gate is instantiated on the wall and the process constraints are considered using data from the Manufacturing model. In order to enable re–analysis or backtracking the link between the gate instantiation and that of the wall is created as in a forward instantiation (ie driven by the functional interface). This link is shown in Figure 7.22.

It can be seen that if a taper or a blend instantiation is requested, the first activity is to identify the equivalent wall instantiations of the cavity or core viewpoint, using the links shown in Figure 7.36 (cavity) and Figure 7.54 (core). A taper or blend is instantiated on the equivalent wall instantiation(s) and the process constraints are evaluated using data in the Manufacturing model. As for a new gate instantiation, to enable re–analysis or backtracking the link between the taper or blend instantiation and that of the wall(s) is created as in a forward instantiation (ie driven by the functional interface). These links are shown in Figure 7.22.

9.5. MOULD DESIGN STRATEGY.

In order for the strategist applications to interact within the area of mould design such that design support is provided in a concurrent manner, a mould design strategy is required to ensure the appropriate interactions. The strategy in the present work is based on the precedence relationships identified in Chapter 5 and Chapter 7, and is shown in Figure 9.8. The first major part of a mould to be generated is the cavity impression. However, as described in Chapter 7 a prerequisite to instantiation of the cavity impression is the instantiation of the parting line, so this is identified and instantiated first. Afterwards support is provided to the designer for instantiation of the cavity impression by the cavity design strategist. Next the core design strategist is triggered to provide design support for instantiation of the main core impression geometry. Once the main geometry in the cavity and core impression has been instantiated it is possible to examine the effects of gaps in the cavity/core if they exist by supporting the designer in instantiation of cavity and core group volumes and consideration of their process constraints. This requires interaction between the cavity and core design strategists as described in sections 9.6 and 9.7. After instantiation of group volumes, or if no gaps exist in the cavity/core the cavity design strategist considers the spatial relationship between mouldability reinforcement and hole instantiations and the geometry of the cavity and core impression to see if they translate to local inserts in the cavity or the core. Instantiation of those that translate to local inserts in the cavity is supported by the cavity design strategist and those that translate to local inserts in the core result in triggering of the core design strategist to support their instantiation.

Having identified the cavity and core impression geometry, the feeding system design strategist can support instantiation of the feeding system, and this requires some interaction with the cooling system design strategist as described in section 9.8. Once the feeding system geometry and the cavity and core impression geometry exists in the Product model the cooling system can be instantiated. The cooling system design strategist supports instantiation of the cavity cooling system first. This provides the prerequisite data for the cavity design strategist to support the designer in instantiation of the cavity plate. Some dimensions

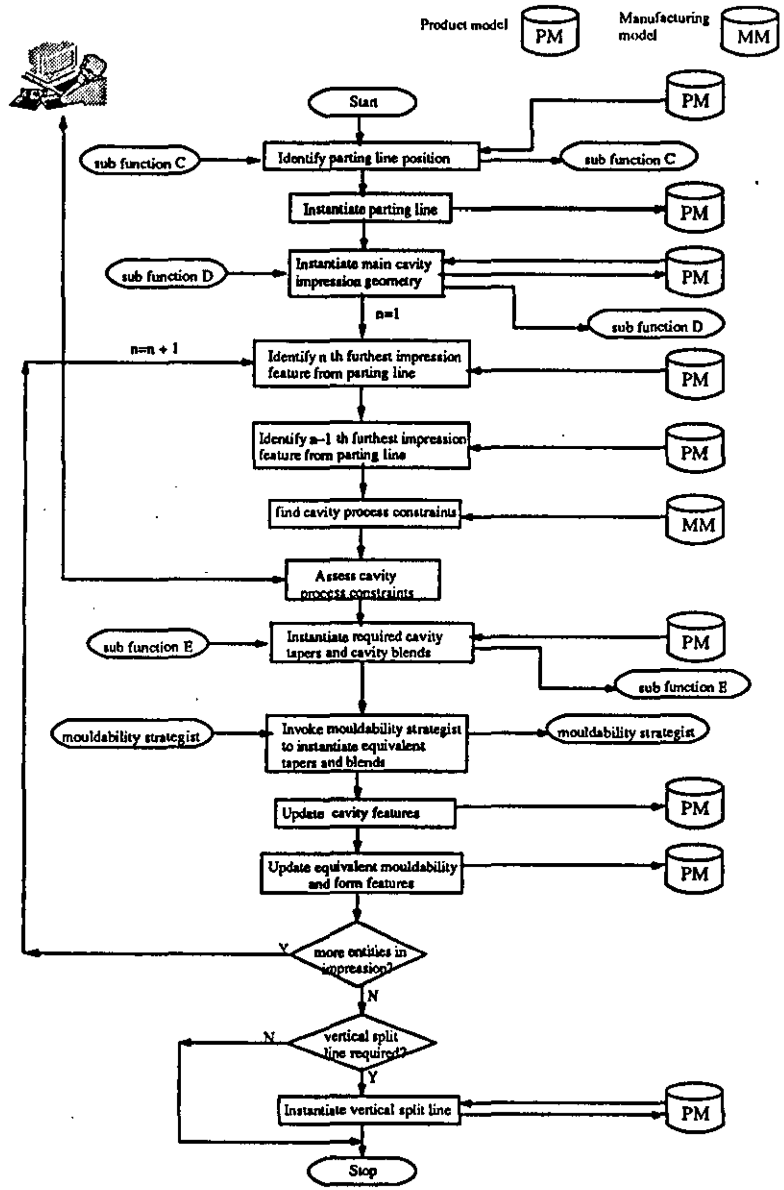


Figure 9.9. Cavity strategist – Instantiation of main impression geometry.

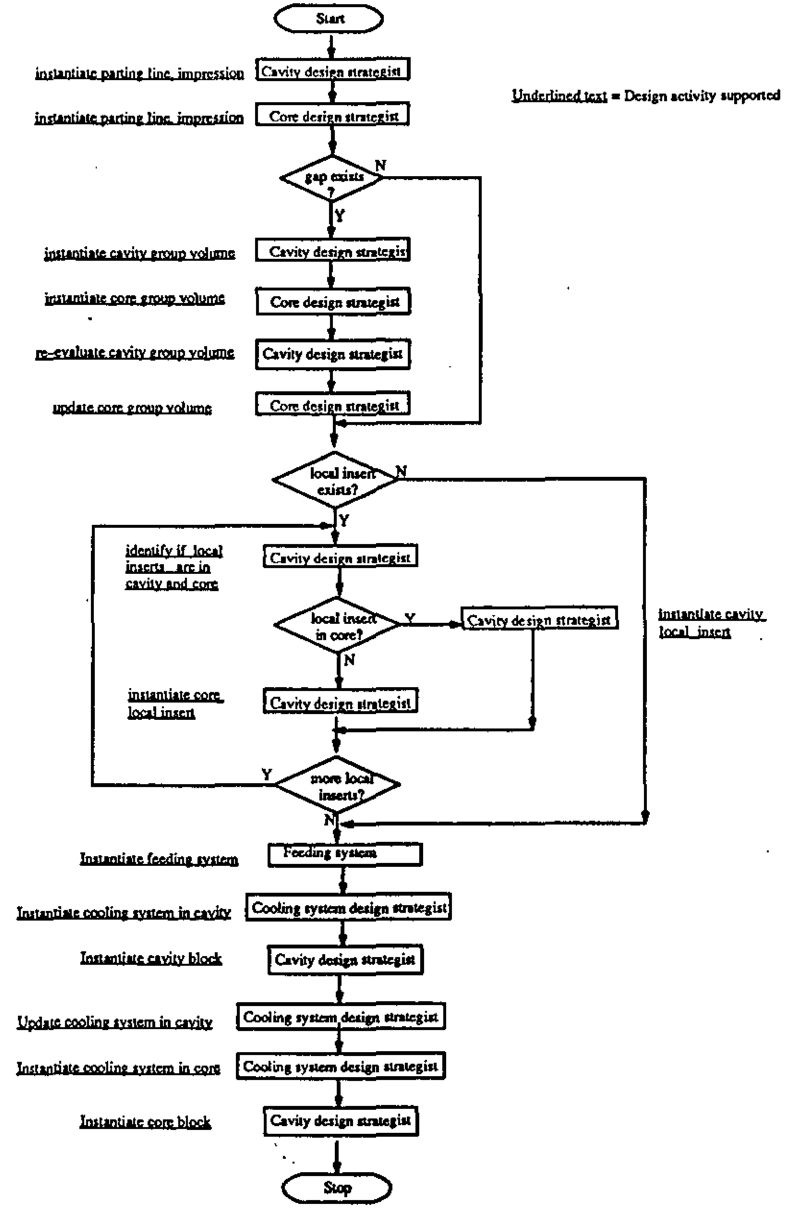


Figure 9.8. Mould design strategy.

of the cavity cooling system are dependant upon the length and width of the cavity block, and therefore the cooling system design strategist must be triggered again to complete the cavity cooling system geometry. Subsequently the cooling system design strategist supports instantiation of the core cooling system. As with the cavity cooling system, some dimensions of the core cooling system are dependant upon the length and width dimensions of the block, but in this work these dimensions can be taken from the cavity block as the dimensions are the same for the cavity and core blocks. Finally the core design strategist supports instantiation of the core plate, completing the mould design strategy. The following sub sections describe the part each strategist plays in the mould design strategy, describing their strategies in order of utilisation.

9.6. CAVITY DESIGN STRATEGIST.

The cavity design strategist has four strategies; i) for instantiation of the main geometry of the cavity impression, ii) for dealing with gaps in the main geometry of the cavity/core impression, iii) for instantiation of local inserts in the cavity, and iv) for instantiation of the cavity plate. The first cavity design support strategy is shown in Figure 9.9. It can be seen that the position of an integer cavity parting line is identified and a parting line instantiated as a prerequisite to instantiation of the cavity impression (see Chapter 7). The next activity is the translation from the mouldability viewpoint to the main cavity impression geometry, which has been described in detail in Chapter 7. From Chapter 7, at the time of each instantiation not all adjacent geometry is known. Therefore the process constraints of the main cavity impression geometry are considered retrospectively of instantiation, shown by the loop in Figure 9.9. Consideration of process constraints may result in the instantiation of integer cavity tapers and integer cavity blends, in which case the equivalent mouldability instantiations must be generated on the equivalent walls in the product. instantiation of mouldability types is performed by triggering the mouldability strategist. After consideration of the process constraints of the main cavity impression a vertical split line is instantiated if required. After instantiation of the main cavity impression the core design strategist is triggered to instantiate the main core impression geometry before returning to the cavity design strategist for the second strategy below:

The second strategy of the cavity design strategist is shown in Figure 9.10. This strategy identifies gaps in the main cavity/core geometry. If none exist the cavity design strategist goes on to the third strategy. If a gap exists an integer cavity group volume is instantiated, as described in Chapter 7 and the process constraints are considered in relation to the main cavity and core impression geometry as described in Chapter 8. The core design strategist is triggered to instantiate a core group volume of identical dimensions in the core. After process constraints have been examined in the core (see section 9.7) the cavity design strategist matches the dimensions of the cavity group volume with those of the core group volume and re–evaluates the process constraints to check the effects of core design decisions. Finally the core design strategist is triggered to match the core group volume to the final dimensions of the cavity group volume. After consideration of the process constraints of the integer cavity group volume a vertical split line is instantiated if required.

The third strategy of the cavity design strategist, to instantiate local inserts in the cavity is shown in Figure 9.11. Using information on the main cavity and core impression geometry from the Product model the cavity design strategist identifies whether a mouldability reinforcement or hole instantiation translates to a local insert in the cavity or the core, as described in Chapter 7. If it is in the cavity the instantiation is made, but if it is in the core the core design strategist is triggered to instantiate a local insert in the core. Consideration of process constraints in the cavity may result in the instantiation of integer cavity

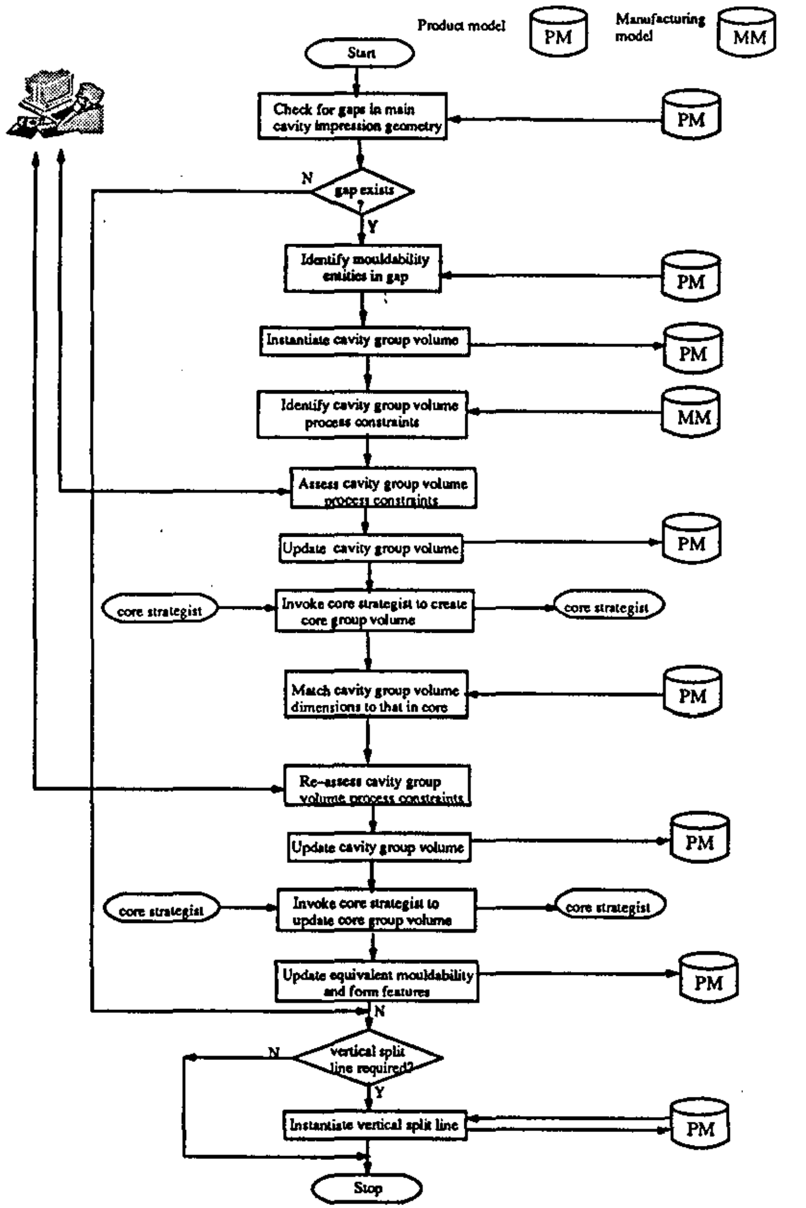


Figure 9.10. Cavity strategist – Instantiation of group volume.

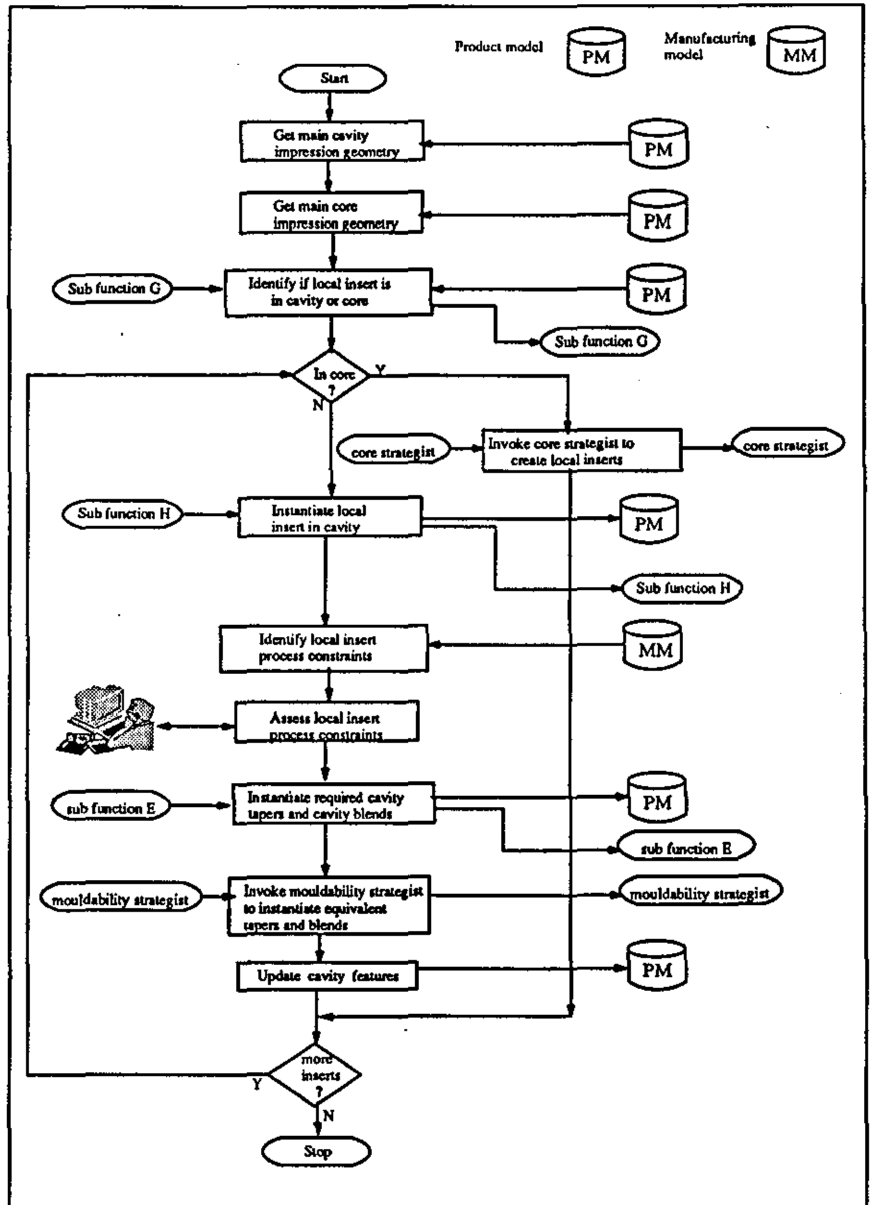


Figure 9.11. Cavity strategist – Instantiation of local inserts.

tapers and integer cavity blends, in which case the equivalent mouldability instantiations must be generated on the equivalent walls in the product. instantiation of mouldability types is performed by triggering the mouldability strategist

The final strategy of the cavity design strategist is to instantiate the cavity plate, shown in Figure 9.12. It can be seen that instantiation of the cavity plate is carried out as described in Chapter 7, using a knowledge of the parting line, the dimensions of the cavity impression and the dimensions of the feeding system from the Product model, and a knowledge of cavity block process constraints from the Manufacturing model. The requirement for a nozzle recess is identified by the dimensions of the feeding system compared to those of the cavity block. If a nozzle recess is required it is instantiated using a knowledge of the cavity block and feeding system geometry plus the dimensions of the injection machine nozzle. The type of inner land is identified based on the configuration of the feeding system. Either a rectangular inner land or a circular inner land is instantiated. If a circular inner land is instantiated, four peripheral lands are subsequently instantiated. If a pin gate has been used in the feeding system a backing plate is instantiated to enclose all parts of the feeding system below the gate, as described in Chapter 7.

9.7. CORE DESIGN STRATEGIST.

The core design strategist has four strategies; i) for instantiation of the main geometry of the core impression, ii) for dealing with gaps in the main geometry of the cavity/core impression, iii) for instantiation of local inserts in the core, and iv) for instantiation of the core plate. Each strategy is invoked by a call from the cavity design strategist, as described in the previous section, and all but the last end with triggering of the cavity design strategist. The last strategy ends with triggering of the feeding system design strategist. The first core

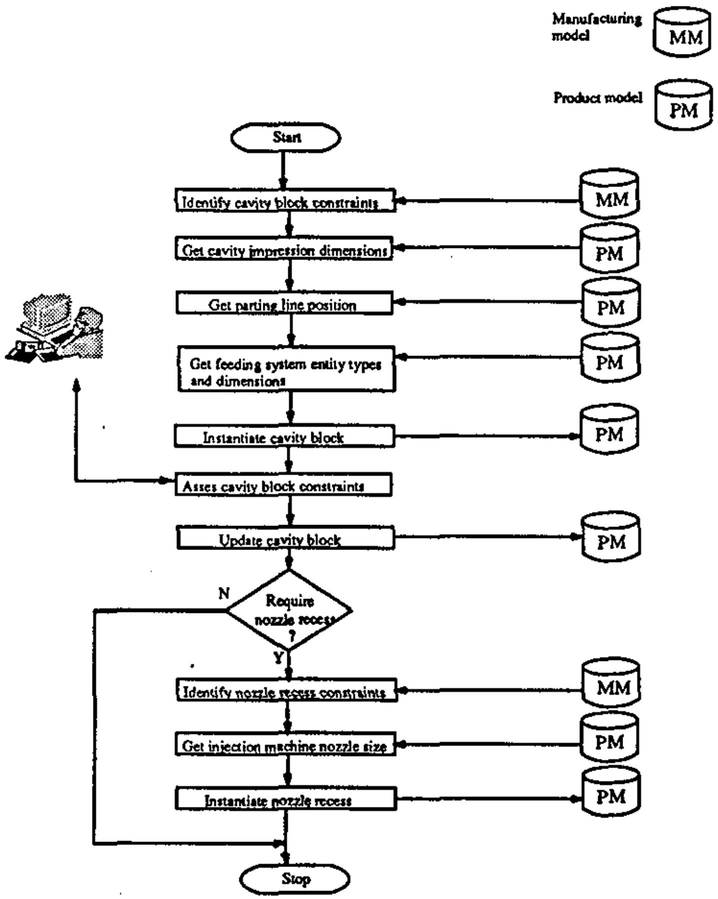


Figure 9.12a). Cavity design strategist –
Instantiating cavity block and nozzle recess.

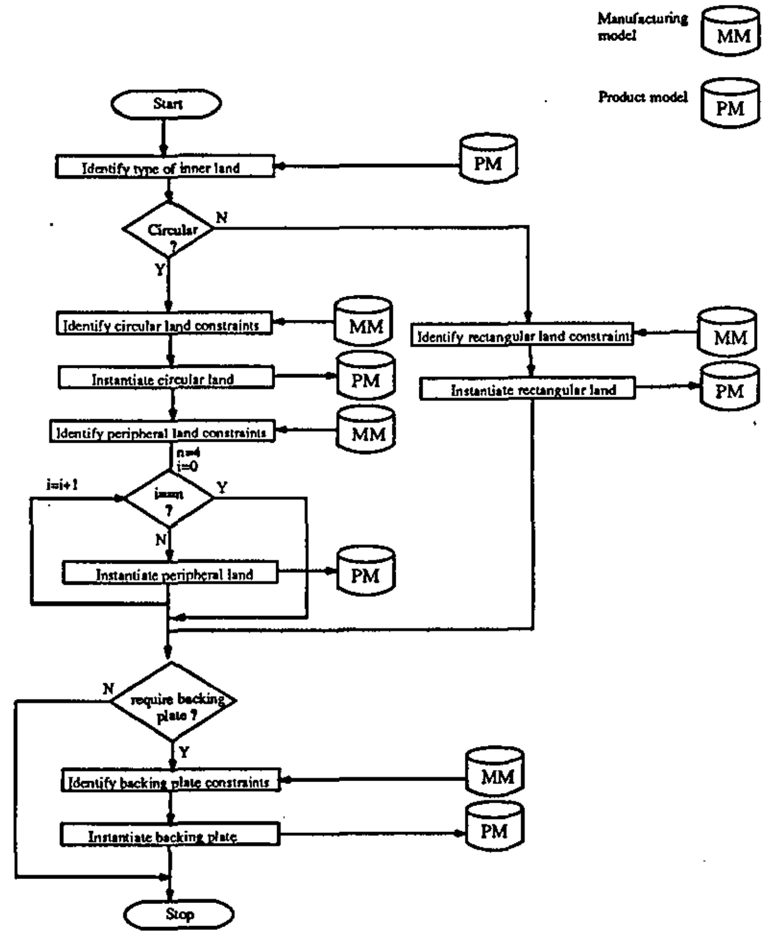


Figure 9.12b). Cavity design strategist –
Instantiating cavity lands and backing plate.

design support strategy is shown in Figure 9.13. It can be seen that the position of an integer core parting line is identified and a parting line instantiated as a prerequisite to instantiation of the core impression (see Chapter 7). The next activity is the translation from the mouldability viewpoint to the main core impression geometry, which has been described in detail in Chapter 7. As with the cavity, at the time of each instantiation not all adjacent geometry is known. Therefore the process constraints of the main core impression geometry are considered retrospectively of instantiation, shown by the loop in Figure 9.13. Consideration of process constraints may result in the instantiation of integer core tapers and integer core blends, in which case the equivalent mouldability instantiations must be generated on the equivalent walls in the product. instantiation of mouldability types is performed by triggering the mouldability strategist. After instantiation of the main core impression the cavity design strategist is triggered to instantiate an integer cavity group volume if a gap exists in the main cavity/core geometry before returning to the core design strategist for the second strategy below. If no gap exists the cavity design strategist goes on to the third strategy, which ends by triggering the third core design strategy.

The second strategy of the core design strategist is shown in Figure 9.14. If an integer cavity group volume is instantiated, the core design strategist instantiates an integer core group volume of identical dimensions. Using a knowledge of the main cavity and core impression geometry and a knowledge of process constraints from the Manufacturing model the process constraints of an integer core group volume are evaluated. The cavity design strategist is then triggered to reconsider the cavity group volume process constraints in respect of any changes, and the core design strategist is triggered again to update the dimensions of the integer core group volume to be identical to those of the re–evaluated integer cavity group volume.

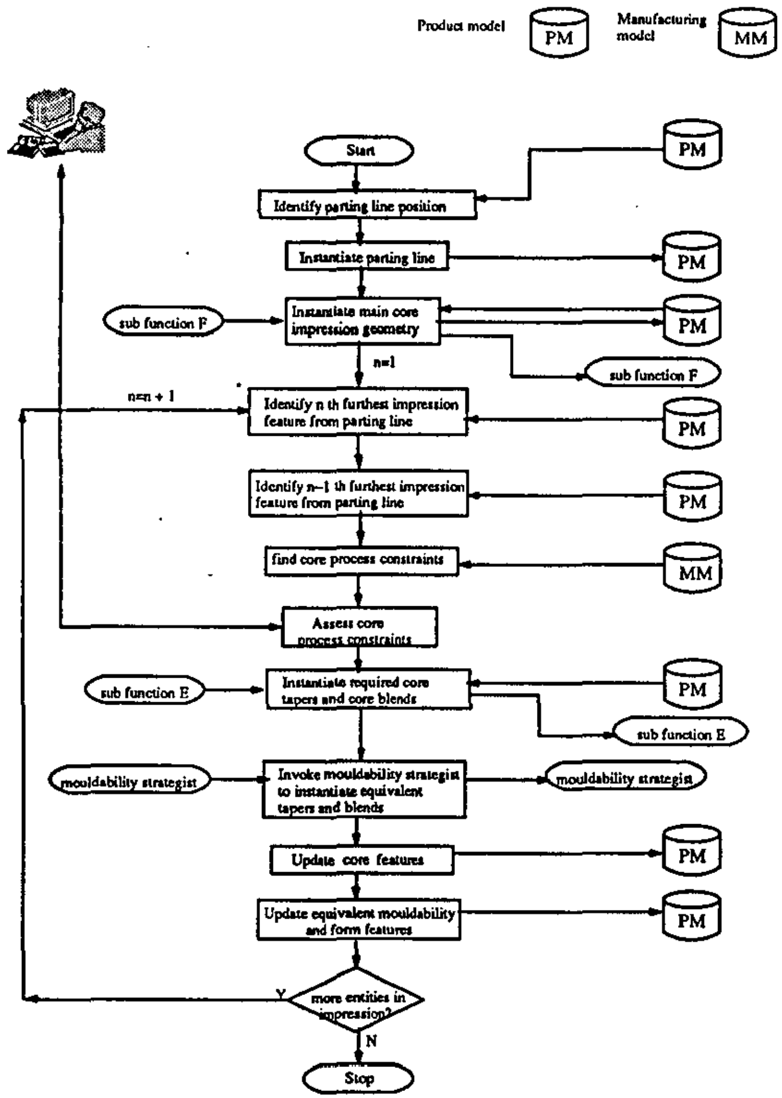


Figure 9.13. Core strategist – Instantiation of main impression geometry.

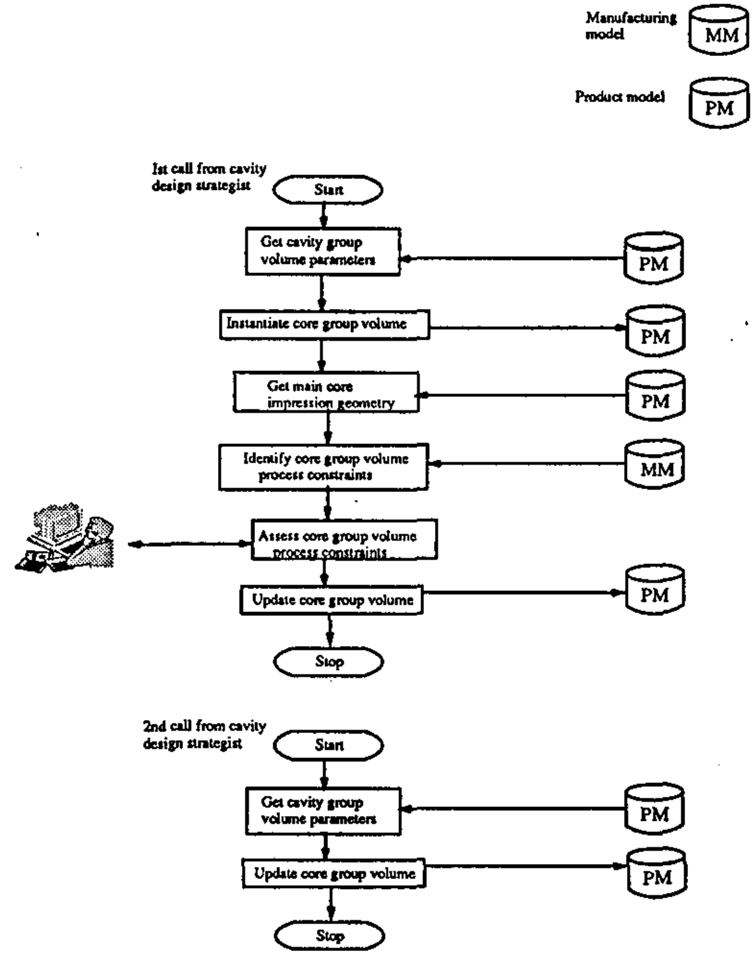


Figure 9.14. Core strategist – Instanciation and updation of group volume.

The third strategy of the core design strategist is to instantiate local inserts in the core and is shown in Figure 9.15. When the cavity design strategist identifies that a local insert is in the core, a local insert is instantiated in the core by the core design strategist, as described in Chapter 7, and the process constraints evaluated. Consideration of process constraints in the core may result in the instantiation of integer core tapers and integer core blends, in which case the equivalent mouldability instantiations must be generated on the equivalent walls in the product. instantiation of mouldability types is performed by triggering the mouldability strategist.

The final strategy of the core design strategist is to instantiate the core plate, and is shown in Figure 9.16. It can be seen that instantiation of the core plate is carried out as described in Chapter 7, using a knowledge of the parting line, the dimensions of the cavity block and the dimensions of the core cooling system from the Product model and a knowledge of core block process constraints from the Manufacturing model. As described in Chapter 7, the type and dimensions of lands are identified from the cavity plate. If a circular inner land is instantiated, four peripheral lands are subsequently instantiated.

9.8. FEEDING SYSTEM DESIGN STRATEGIST.

The feeding system design strategist has only one strategy, shown in Figure 9.17. The first activity is to identify a gate or gates on the product in order to instantiate the equivalent gating system. If a mouldability gate has not been instantiated on a product the designer is asked to indicate a wall and a gate is instantiated as described in Chapter 7. instantiation of the mouldability gate is performed by triggering the mouldability strategist. Before many of the process constraints of the feeding system can be considered the minimum depth in the cavity block to contain the cooling system must be known. Thus the cooling design

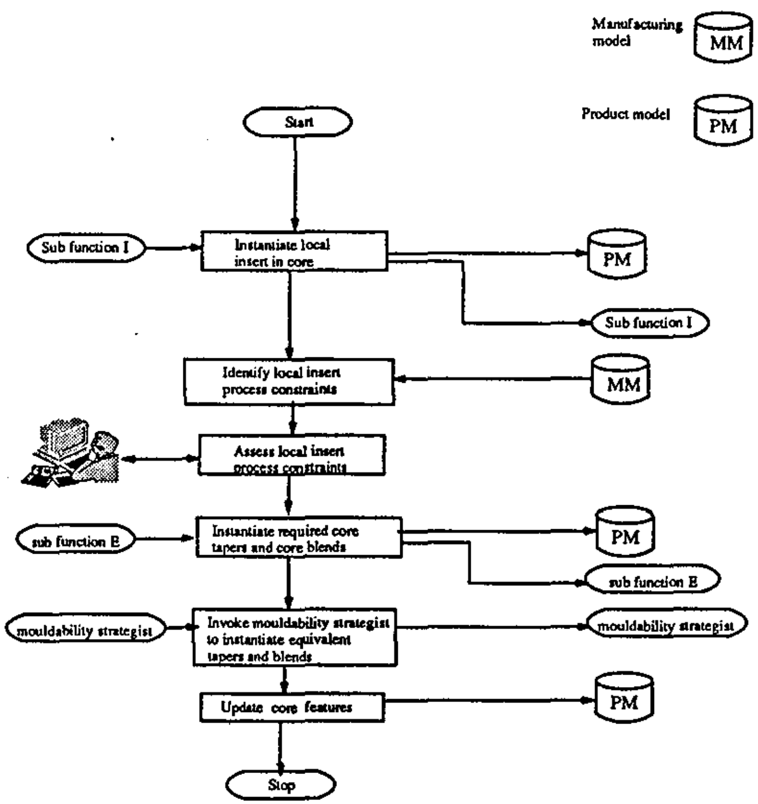


Figure 9.15. Core strategist – Instantiation of local inserts.

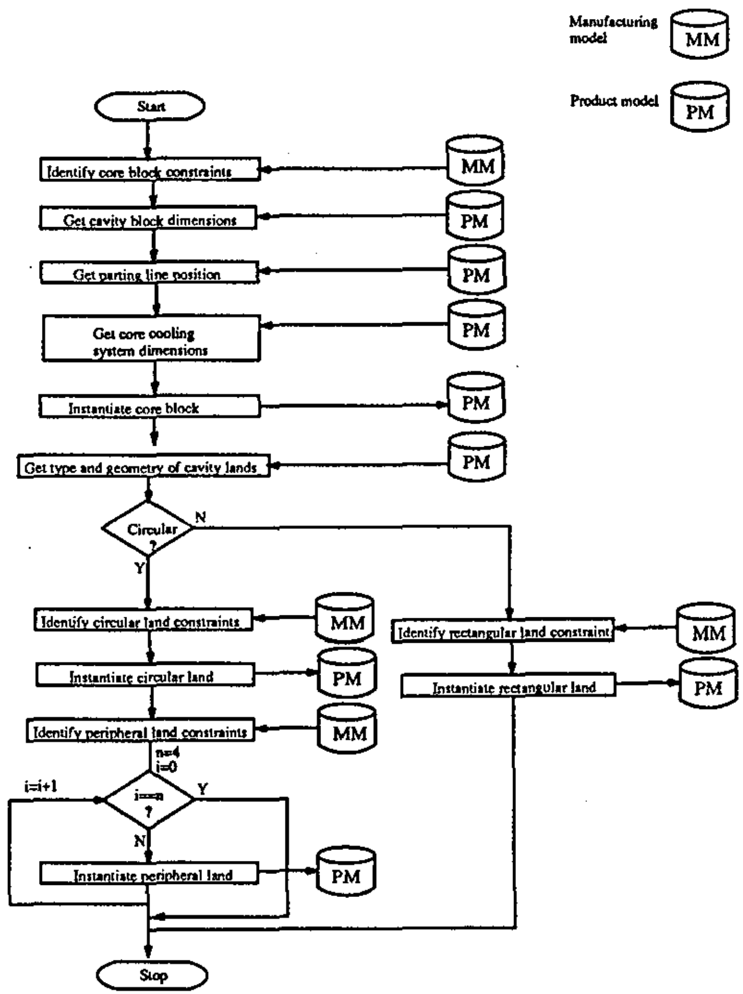


Figure 9.16. Core design strategist – Instantiating core block and lands.

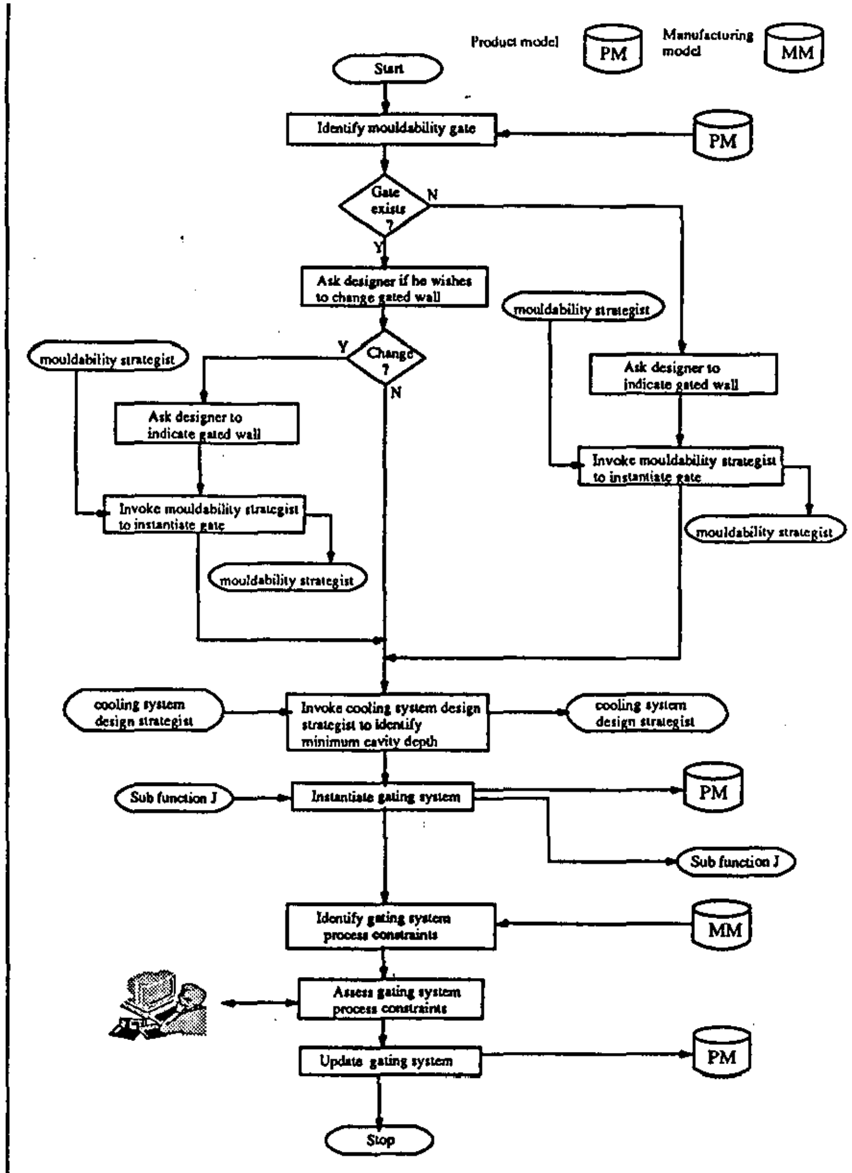


Figure 9.17a). Feeding system design strategist – Instantiation of gating system.

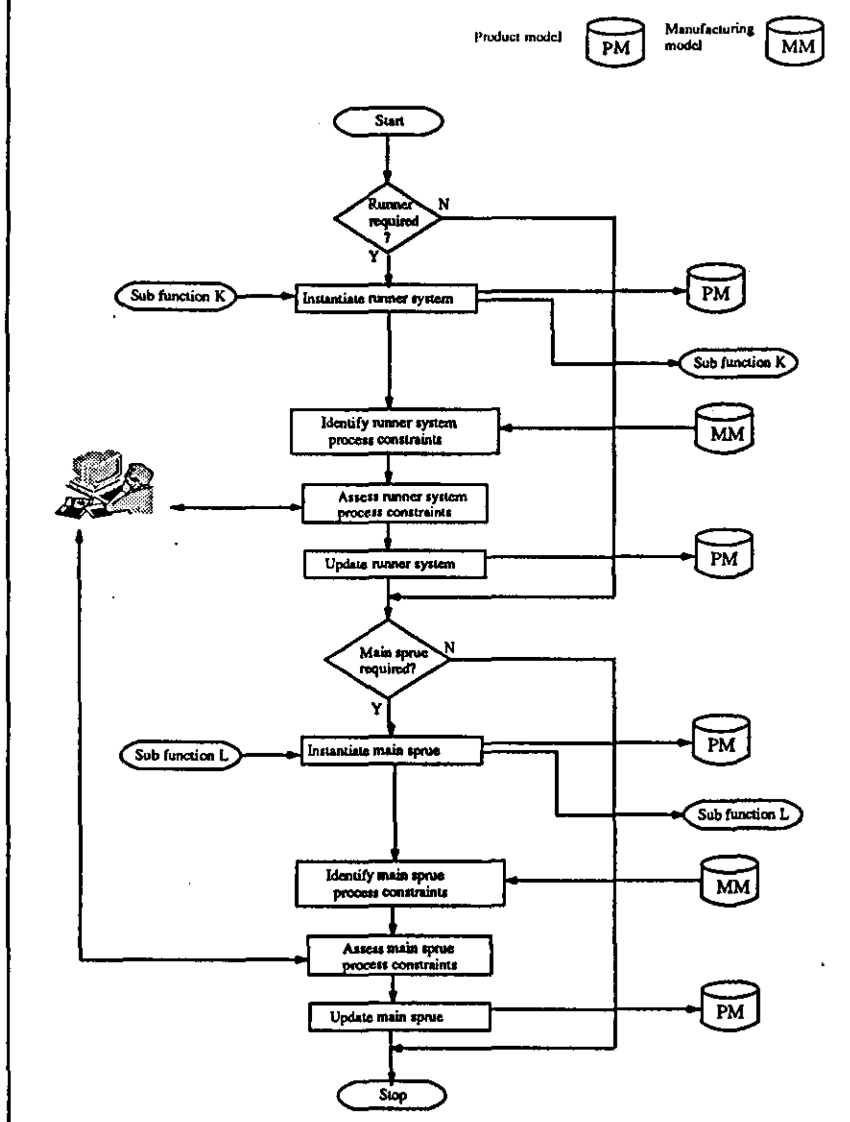


Figure 9.17b). Feeding system design strategist – Instantiation of runner system and main feeding sprue.

strategist is called to support the designer in establishing this depth before returning to the feeding system design strategist. This support is described in section 9.9.

Once the minimum depth of a cavity block is known a gating system is instantiated of the type indicated during mouldability gate instantiation (see Chapter 7) and the process constraints evaluated using data from the Manufacturing model. The type of gating system is used to identify the elements of the rest of the feeding system, eg what type of runner system. As shown in Figure 9.17, if a runner system is required (depending upon the gating system type) the appropriate type is instantiated in the Product model and the process constraints evaluated using data from the Manufacturing model. Finally (if required) a main feeding sprue is instantiated in the Product model and the process constraints evaluated using data from the Manufacturing model.

9.9. COOLING SYSTEM DESIGN STRATEGIST.

The cooling system design strategist has four strategies; i) for identifying the minimum depth of the cavity block to contain the cooling system, ii) for supporting instantiation of most of the cavity cooling system, iii) for completing the cavity cooling system geometry after instantiation of the cavity block, and iv) for supporting instantiation of the core cooling system.

The first strategy is shown in Figure 9.18. This strategy is triggered by a call from the feeding system design strategist for information. Using a knowledge of the cavity impression geometry the maximum cooling capacity of a cavity cooling system is identified. The minimum depth of a cavity block to contain such a cooling system is then identified based on the minimum distances to avoid directional cooling, as described in Chapter 7. The second

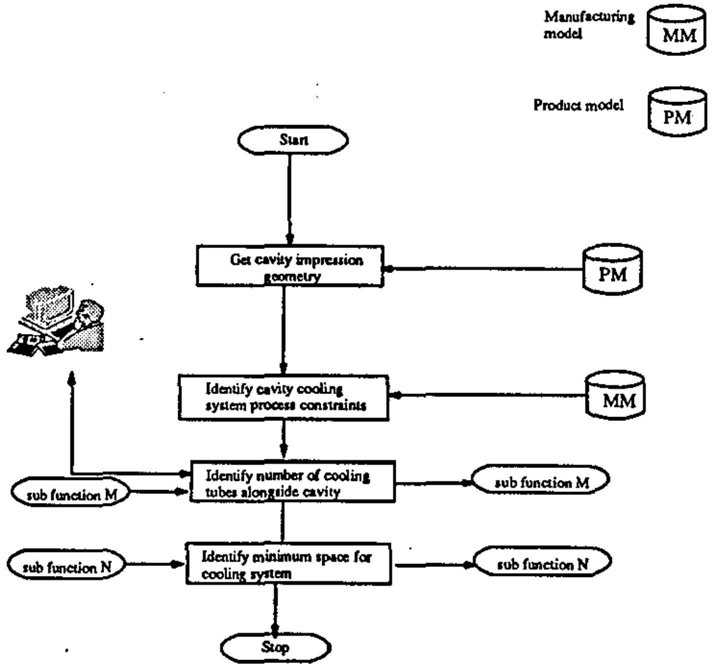


Figure 9.18. Cooling system design strategist –
Identifying minimum depth in block for cooling system.

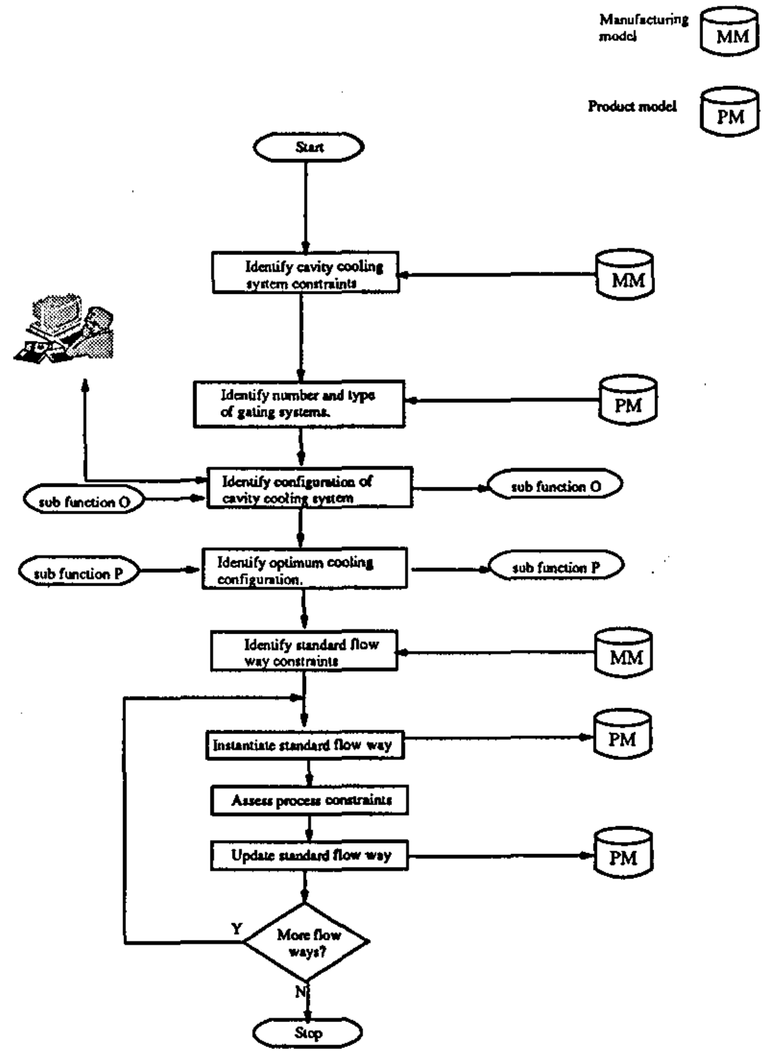


Figure 9.19. Cooling system design strategist –
Instantiating cavity cooling system.

strategy, shown in Figure 9.19 is triggered by completion of the feeding system design. Using a knowledge of the cavity cooling system constraints from the Manufacturing model and of the type and numbers of gating systems in the Product model the designer is supported in identifying the optimum configuration of cooling system for balanced cooling of the mould. Using a knowledge of cavity cooling system constraints from the Manufacturing model the optimum cooling formation is identified based on minimum distances between the cooling system and the closest parts of the impression and the feeding system. The spacing between layers of tubes is identified based on minimum distances between the tubes in a cooling system, and the standard flow ways are instantiated in the appropriate positions and orientation to produce the required configuration of cavity cooling system in the optimum cooling formation. All of the above is as described in Chapter 7.

The third strategy, shown in Figure 9.20 is triggered by the instantiation of the cavity block. Using a knowledge of cavity cooling system constraints from the Manufacturing model, and based on the dimensions of the new cavity block instantiation, the geometry of the cavity cooling system in the Product model is completed. The final strategy is shown in Figure 9.21. It can be seen that using a knowledge of the core impression geometry and a knowledge of core cooling system constraints it is identified whether a deep or a shallow core cooling system should be used. If a shallow core cooling system is used, then using a knowledge of shallow core cooling system constraints from the Manufacturing model and the geometry of the cavity impression, the number of tubes across the core is identified (maximum cooling capacity), as described in Chapter 7. With the additional knowledge of the feeding system geometry the designer is supported in identifying the optimum shallow cooling system configuration for even cooling of the mould core. Using a knowledge of the shallow core cooling system constraints from the Manufacturing model the optimum cooling formation is identified based on minimum distances between the cooling system

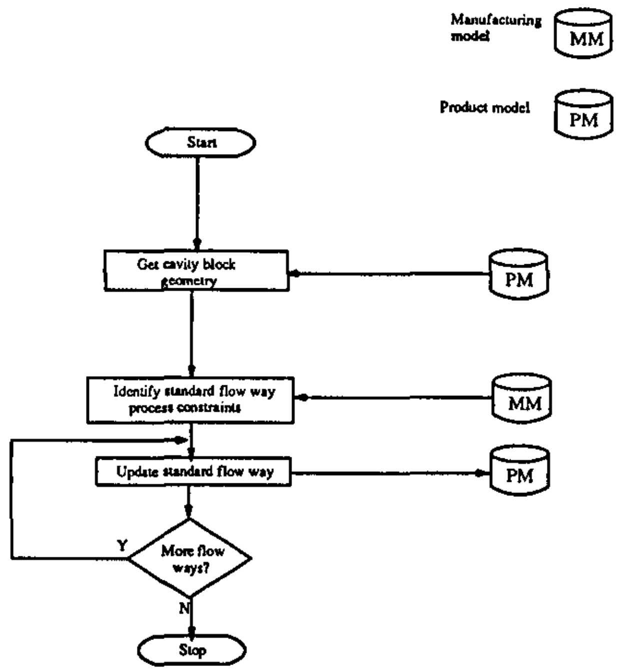


Figure 9.20. Cooling system design strategist – Completing cavity cooling system geometry after cavity block instantiation.

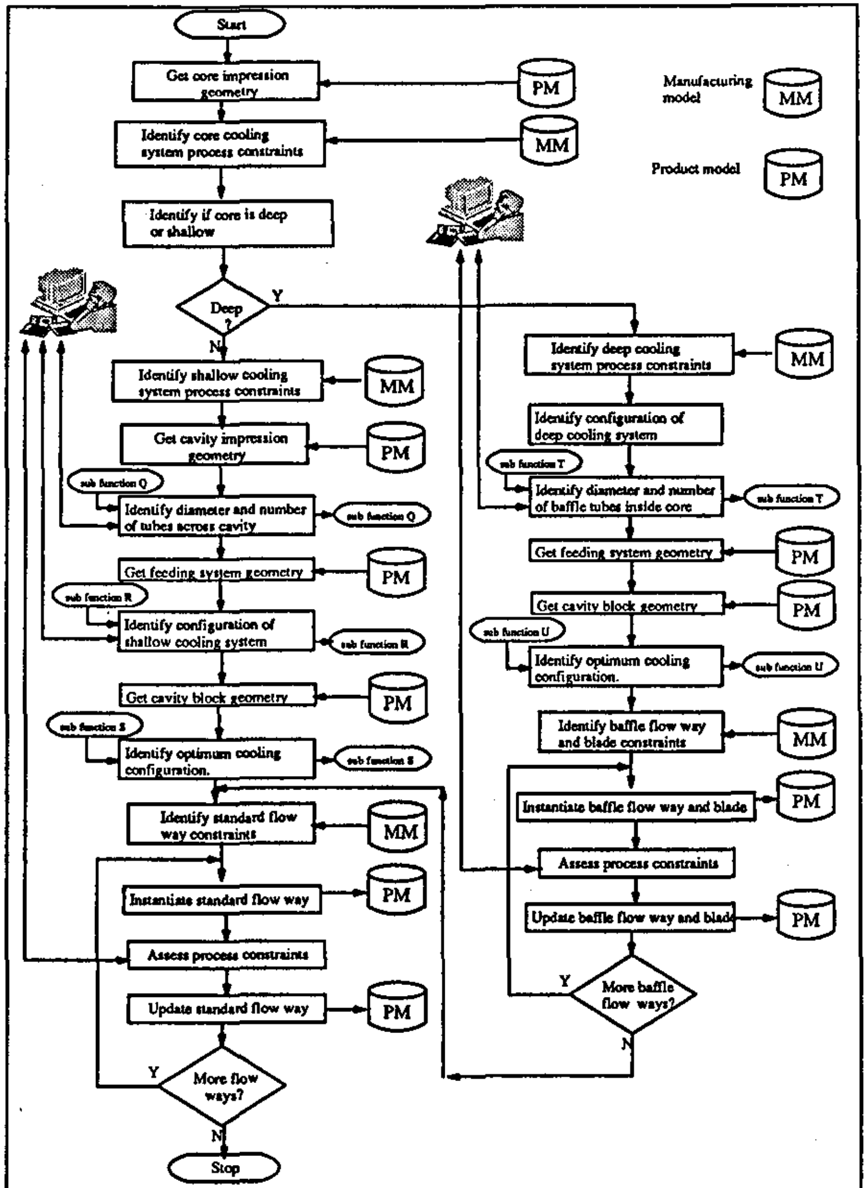


Figure 9.21. Cooling system design strategist – Instantiation of core cooling system.

and the closest parts of the impression and the feeding system. Finally the standard flow ways are instantiated in the appropriate positions and orientation to produce the required configuration of core cooling system in the optimum cooling formation. All of the above is as described in Chapter 7.

If a deep core cooling system is used, then using a knowledge of deep core cooling system constraints from the Manufacturing model and the geometry of the core impression the number of tubes inside the core is identified (maximum cooling capacity), as described in Chapter 7. Using a knowledge of the deep core cooling system constraints from the Manufacturing model the optimum cooling formation is identified based on minimum distances between the cooling system and the closest parts of the impression and the feeding system. Finally the baffle flow ways and standard flow ways are instantiated in the appropriate positions and orientation to produce the required configuration of core cooling system in the optimum cooling formation. All of the above is as described in Chapter 7.

Chapter 10

The experimental investigations performed.

10.1. INTRODUCTION.

This chapter explains the experimental work performed by the author to explore support for concurrent design for injection moulding by an injection moulding strategist and its use of the representations in the Manufacturing model, Product model and Product Range Model, when supporting the instantiation of 3D geometry. Section 10.2. presents the scope of an experimental injection moulding strategist implementation. This covers the implementation objectives and the implementation of the injection moulding strategist itself, the Manufacturing model, the Product Range Model and the interface to the user. Section 10.3 presents the experimental exploitation of an injection moulding strategist to support concurrent design for manufacture of example products. Section 10.4 discusses the experimental results.

The experimental design support system is based on the definition of the injection moulding strategist functionality and structure as described in Chapters 5, 7 and 9, and the definition of the Manufacturing model and Product Range Model in Chapters 6 and 8.

10.2. EXPERIMENTAL EVALUATION OF THE STRATEGIST

10.2.1. An implementation of feature types.

Figure 10.1 shows a screen dump of the *primary mouldability features* class and *secondary mouldability features* classes in the Object DB browser. The primary mouldability features

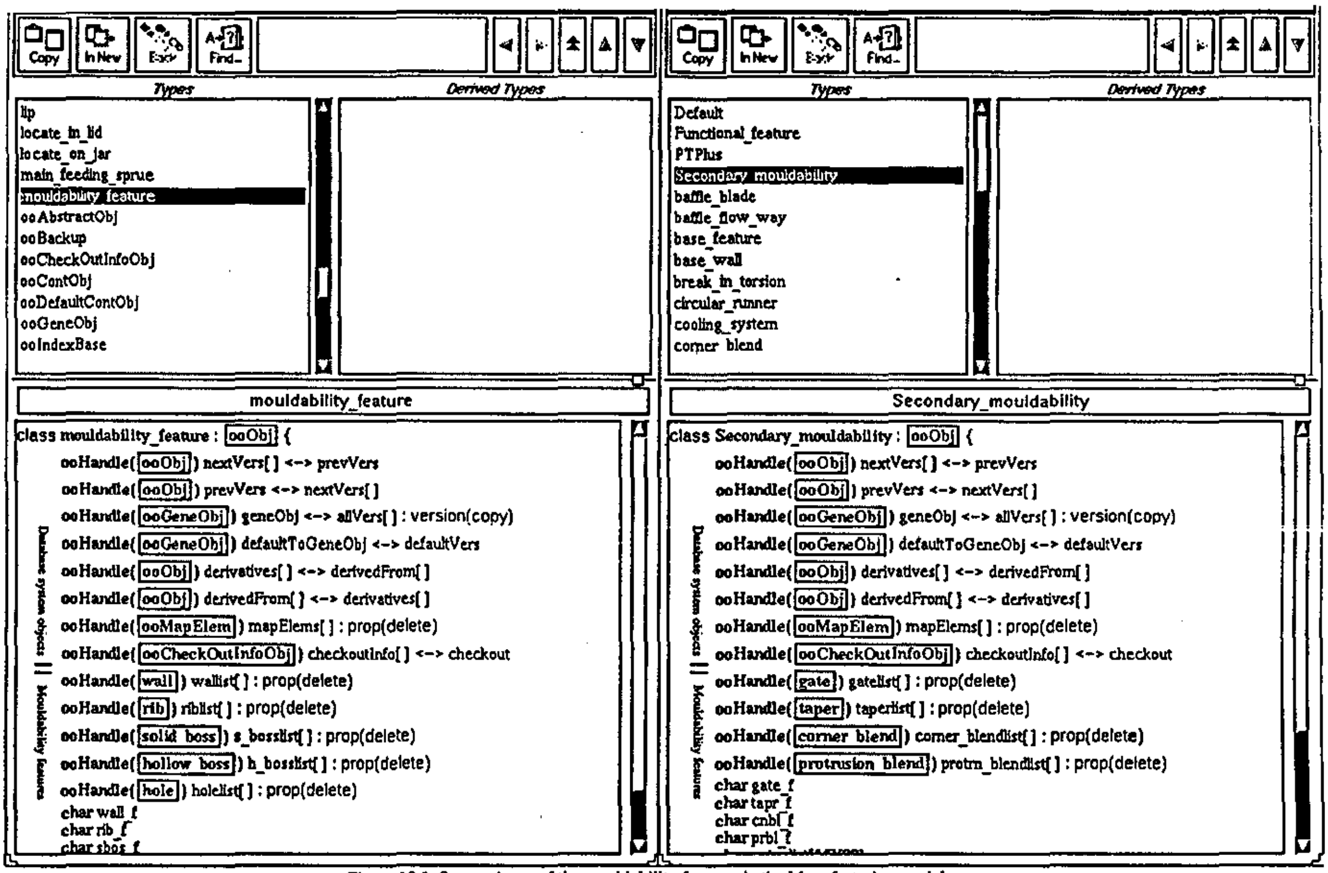


Figure 10.1. Screen dump of the mouldability features in the Manufacturing model.

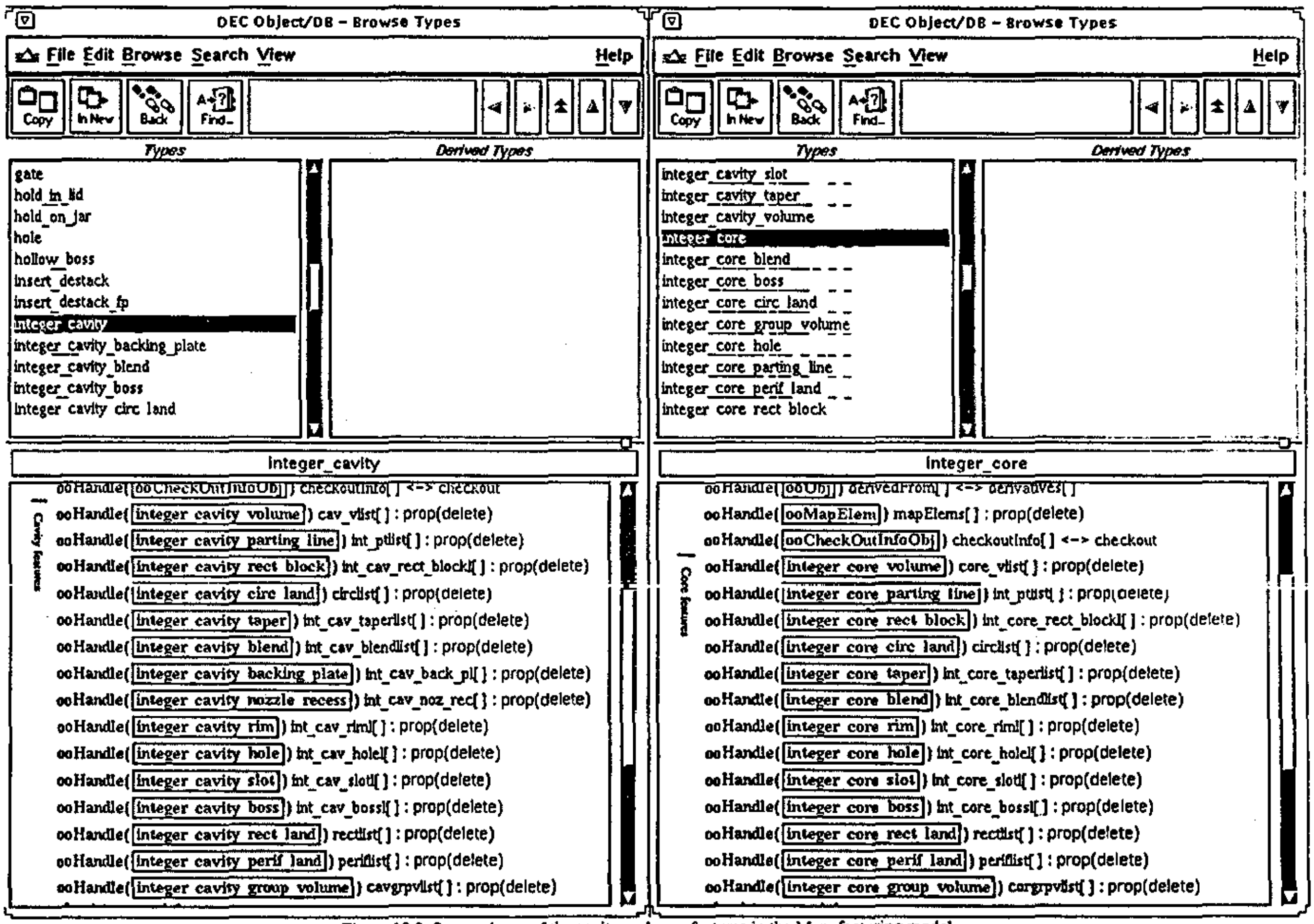


Figure 10.2. Screen dump of the cavity and core features in the Manufacturing model.

are wall, rib, hollow boss, solid boss, hole. The secondary mouldability features, which have been implemented as a separate class, are corner blend, protrusion blend, taper and gate. Figure 10.2 shows a screen dump of the *integer cavity* and *integer core* classes in the Object DB browser. Figure 10.3 shows a screen dump of the implemented feeding system features, which are pin gate, sprue gate, rectangular edge gate and main feeding sprue, and the implemented cooling system features, which are standard flow way, baffle flow way and baffle blade. The process constraints of the cavity cooling system, the shallow core cooling system and the deep core cooling system are stored in the cooling system class. For all the above feature types methods have been defined in each class to capture the process constraints of the feeding system or cooling system and their detailed representation in the EXPRESS language is shown in Appendix 4..

Figure 10.4 shows a screen dump of the example product ranges of PTPlus, Yoghurt pot and Flower pot in the Object DB browser. Note that Object DB does not permit double inheritance, and thus those product functions common to the yoghurt pot and flower pot product ranges have had to be implemented as separate product functions for the flower pot with the appendage '_fp'.

10.2.2. Interfacing the experimental software program.

The injection moulding strategist is linked to and drives the Object DB software. It provides an interface menu for the designer to drive the application and receive feedback advice. The menu also allows the designer to select display options. The product design or parts of the mould can be displayed from the current state of the Product model in Object DB. The display of the current state of a product or mould in the Product model is provided via a linker to the Unigraphics V10 solid modelling package.

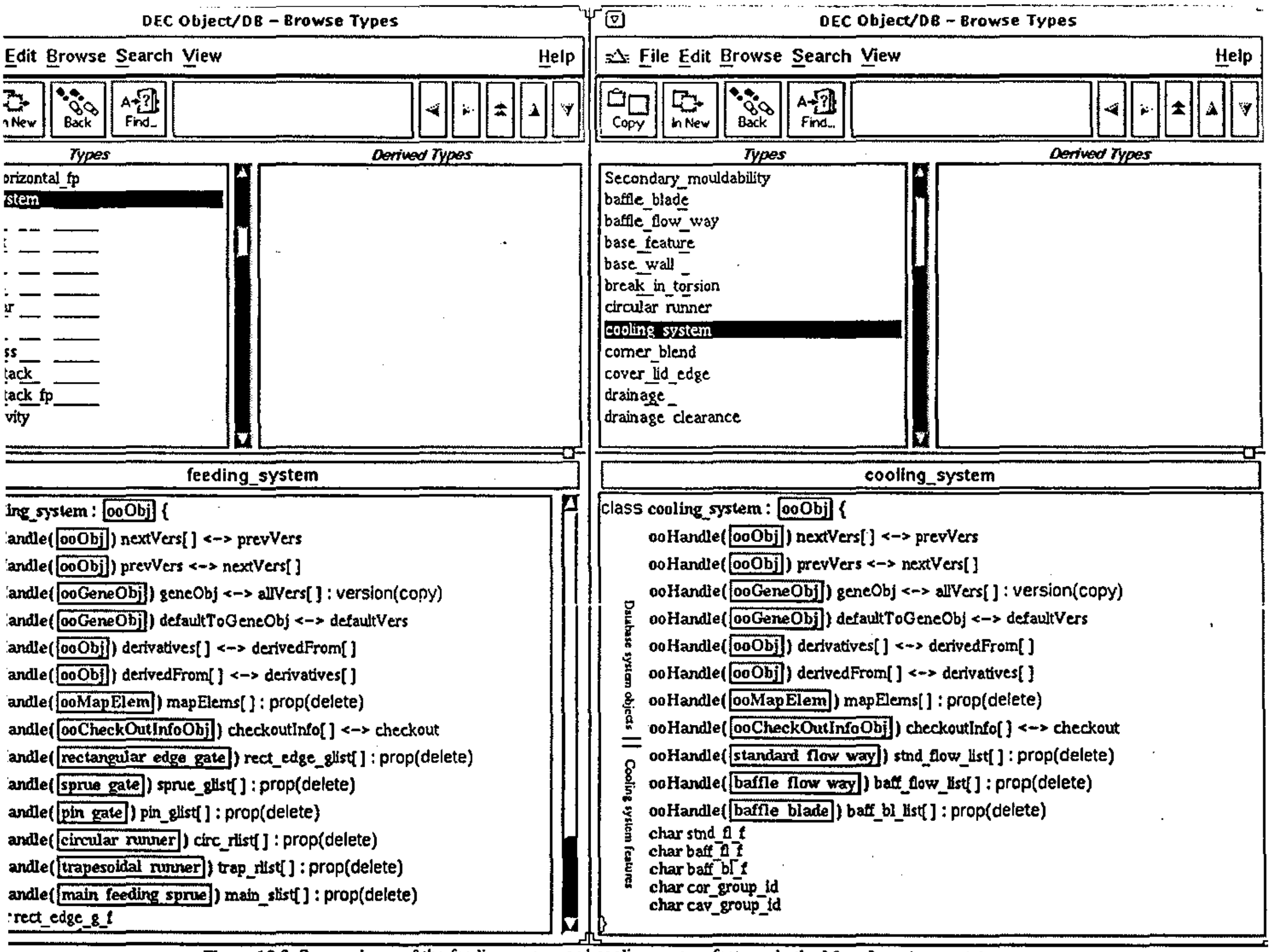


Figure 10.3. Screen dump of the feeding system and cooling system features in the Manufacturing model.

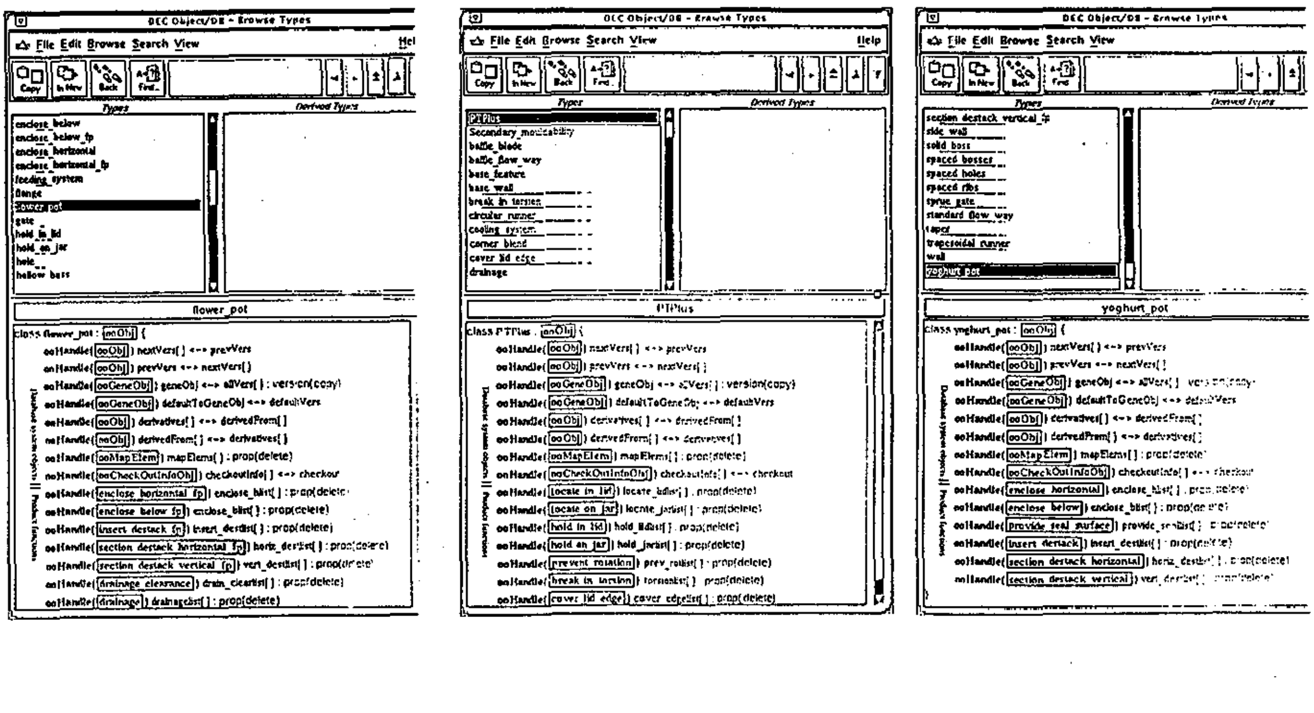


Figure 10.4. Screen dump of the product ranges in the Product Range Model.

10.2.3. Supporting concurrent design for function and mouldability on a yoghurt pot.

The experiment described in this section has been performed to demonstrate how an injection moulding strategist supports the designer in considering the interaction of functional and mouldability constraints on a yoghurt pot. The objectives can be listed below:

i. To show how the designer is supported in the association of function and form using information from the Product Range Model.

ii. To show how the designer is supported in consideration of mouldability using a translation process from form to mouldability and information from the Manufacturing model.

In this experiment the designer creates a new product in the yoghurt pot product range. Therefore the functional interface strategy for supporting the instantiation of an initial product definition is invoked (Figure 9.2). The designer is provided with feedback advice with respect to functional and manufacturing constraints as he/she builds up 3D geometry in the Product model. The plastic part that is built up in the experiment is shown in Figure 10.5. The initial product definition of a yoghurt pot comprises two product function instantiations; enclose horizontal and enclose below, and two associated form feature instantiations; a side wall to achieve the former function and a base wall to achieve the latter. Also there are two mouldability instantiations which are the alternative of the form viewpoint and associated secondary mouldability instantiations such as tapers and blends.

The functional interface application first asks the designer to choose a product range from

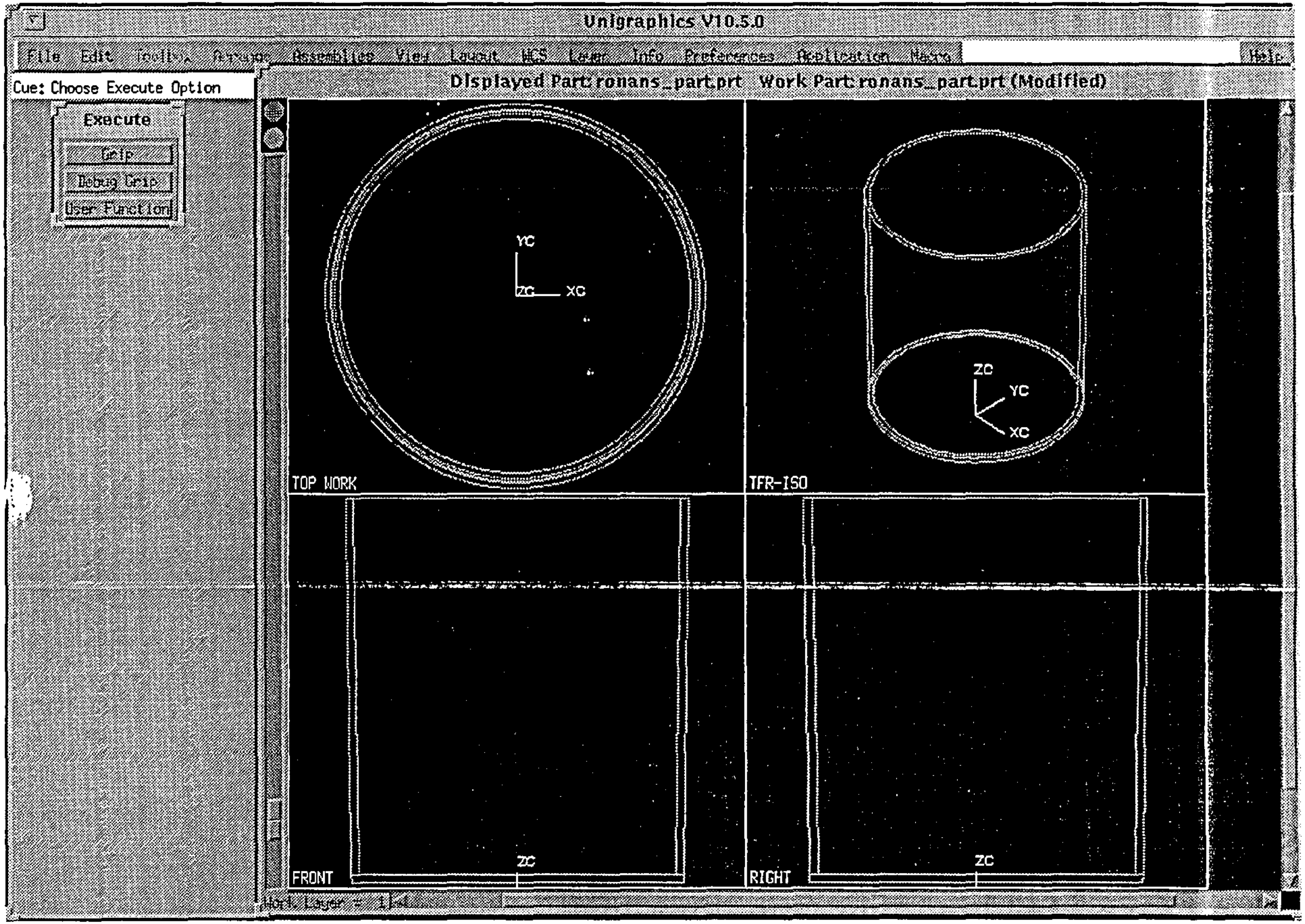


Figure 10.5. Screen dump of built up yoghurt pot geometry in a solid model.

those for which data is stored in the Product Range Model.

Product ranges where data is available in the Product Range Model are:
1. Yoghurt pot range of products
2. Flower pot range of products
3. PTPlus range of products
Enter choice.
1

After the designer chooses a product range and indicates to create a new product the functional interface application identifies the product functions that must be addressed and asks the designer to provide a specification for those product functions that have been identified as within the initial product definition. The data required for each specification is relayed to the designer using information in the product function class in the Product Range Model, eg enclose horizontal.

Functional requirements:

enclose_horizontal
enclose_below
provide_seal_surface
Section_destack_horizontal
Section_destack_vertical
Insert_destack

Specify functional requirements for initial product definition:

FUNCTION – Enclose horizontal

What is the enclosed volume? (mm3)500000

What is the diameter of the enclosure? (mm)80

FUNCTION – Enclose below

What is the enclosed diameter? (mm)70

This diameter specification is smaller than that for adjacent 'Enclose horizontal' function

Consequences:
1. If form matches specification, enclosure not achieved, product functionality lost

Remedial options:
1. Increase diameter specification to a minimum of 80 mm
– No further options
Change specification? y/ny

Enter new specification:80

In this experiment it can be seen that the specification given for the second product function creates a conflict with that for the first. This is because fulfilment of the second specification would not necessarily result in attainment of the overall product functionality. Therefore the designer is asked to change the latter specification. After providing the specification the designer is a supported in associating each product function in the initial product definition with a form feature using the form/function data in the Product Range Model. First the designer is provided with a choice of forms to achieve the product function. In this experiment only one type of form feature is identified as usable to achieve the enclose horizontal function. The choice is a side wall. Therefore the designer is supported in providing the dimensions of a side wall form feature instantiation.

FUNCTION:enclose_horizontal

Forms available for use:
1.side_wall
No more forms available.

FORM FEATURE- SIDE_WALL: Ronan_Fsid_w0

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (base of central axis of rotation):0 0 10

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

Specify inner diameter:80

Specify side_wall thickness:2

Specify side_wall height:70

After instantiation of the form feature in the Product model, the association with the functional viewpoint (Figure 7.3) is used by the functional interface application to identify the functional constraints in the Product Range Model. Using data from the Yoghurt Pot – enclose horizontal class in the Product Range Model, functional feedback advice is provided to the designer with respect to the functionality of the form feature. In this experiment the height of the side wall instantiation has to be increased to satisfy the enclose horizontal specification.

FORM FEATURE Ronan_Fsid_w0
FUNCTIONALITY ASSESSMENT – enclose_horizontal function:

Inner diameter satisfactory for enclose horizontal function

Present wall dimensions mean that the enclosed volume is lower than that specified
Specified volume: 500000 mm3

Consequences:
Enclose horizontal function specification not achieved

Remedial options:
1. Increase inner diameter to: 93 mm
NOTE: Enclose horizontal specification for inner diameter: 80 mm
2. Increase wall height to: 94.5 mm
– No further options
Change feature inner diameter? y/nn

Present diameter recorded
Change height?y/ny

New height: 94.5

After functionality has been evaluated the mouldability strategist is invoked and a translation is made from the form viewpoint to that of mouldability. The designer is provided with mouldability feedback advice using data from the wall class in the Manufacturing model. This application strategy is shown in Figure 9.5. The dimensions of the wall instantiation are acceptable for mouldability. However a taper is required on the wall instantiation to facilitate removal of the product from a mould. The designer is supported in instantiation of a taper feature and provided with feedback advice with respect to taper constraints using data from the wall class and taper class of the Manufacturing model respectively.

MOULDABILITY WALL FEATURE: Ronan_Mwall0

Wall thickness ok

Wall features require a taper

Consequences of non–inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/ny

Creating taper on wall Ronan_Mwall0
Enter taper angle:
Recommended minimum draft angle = 0.8 degrees0.8

Taper angle ok

Having addressed the first product function in the initial product definition and considered the functional and manufacturing constraints, the designer is supported in addressing the second (and last) product function in the initial product definition. Using the form/function data in the Product Range Model it is identified that only one type of form feature as usable to achieve the enclose below function. The choice is a base wall. Therefore the designer is supported in providing the dimensions of a base wall form feature instantiation.

FUNCTION:enclose_below

Forms available for use:
1.base_wall
No more forms available.

FORM FEATURE- BASE_WALL: Ronan_Fbs_wl0

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (centre of base):0 0 8

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

Specify base_wall diameter:84

Specify base_wall thickness:3

After instantiation of the second form feature in the Product model, the association with the functional viewpoint (Figure 7.3) is used by the functional interface application to identify the functional constraints in the Product Range Model. Using data from the Yoghurt Pot – enclose below class in the Product Range Model, functional feedback advice is provided to the designer with respect to the functionality of the form feature. The form feature has to be repositioned to avoid a reduction in the internal volume of the yoghurt pot.

FORM FEATURE Ronan_Fbs_wl0
FUNCTIONALITY ASSESSMENT – enclose_below function:

Top of base wall is higher than the base of horizontal enclosure wall

Consequences:
1. Base wall encroaching on 'Enclose horizontal' surface– Loss of functionality

Remedial options:
1. Lower base wall to z position 7

–No further options

Reposition the feature? y/ny

New feature position: 0 0 7
Diameter satisfactory for enclose below specification
Feature outer diameter satisfactory for enclose below function

Next the mouldability strategist is invoked and a translation is made from the form viewpoint to mouldability. As before, the designer is provided with feedback advice using data from the wall class in the Manufacturing model and the taper class as a taper is instantiated on the wall. With more than one mouldability instantiation in the Product model, feedback advice is also required on the relationships between mouldability instantiations. In this experiment the thickness of the new mouldability instantiation is acceptable for the individual instantiation, but it is not acceptable for the relationship with other mouldability instantiations, as it is not the same as that of the adjacent wall feature instantiation. The designer changes the thickness to be the same.

MOULDABILITY WALL FEATURE: Ronan_Mwall1

Wall thickness ok

Wall features require a taper

Consequences of non–inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/ny

Creating taper on wall Ronan_Mwall1
Enter taper angle:
Recommended minimum draft angle = 0.8 degrees.8

Taper angle ok

Wall thickness is not the same as adjacent wall

Possible consequences:
1. Feeding problems if a thick section is fed by a thin section
2. Stress concentrations at abrupt section changes
3. Abrupt section changes can interfere with the flow of material in the mould causing surface defects
4. Component warpage

Remedial options:
1. Make wall thickness the same or near to that of adjacent wall (2)
2. If the difference in thickness must remain make sure the change is not abrupt

–No further options

Change wall thickness? y/ny

Enter new wall thickness (mm):2.0

New thickness ok

As well as the wall thickness comparison a blend is required between adjoining mouldability wall instantiations. The designer is supported in instantiating a corner blend using feedback advice from the wall class of the Manufacturing model and from the corner blend class with respect to corner blend constraints. A constant section thickness around a corner section is required to avoid mouldability problems, and the designer has to modify the outer radius of the corner blend.

Wall features require a blend

Possible consequences of non inclusion:
1. Stress concentrations in the component
2. Turbulent flow around the corner can cause surface defects

Do you wish to create a blend?y/ny

Creating blend on wall Ronan_Mwall1
Enter inside radius:
Recommended inside radius is between 0.8 and 1.2 mm
0.5 mm is the recommended minimum radius.8

Inside blend radius ok

Enter outside radius:
Recommended outside radius is 2.8 mm2.7

This blend radius is less than 2.8 causing thickening corner section

Possible consequences:
1. Shrinkage marks or surface depressions in the corner
2. Widening of the corner angle
3. Curvature of the wall sections either side of the corner

Remedial options:
1. Increase blend radius to 2.8

–No further options

Increase the blend radius?y/ny

Enter new blend radius:2.8

New radius recorded

The above completes the initial product definition. However the designer is advised that the application of tapers for mouldability may effect the functionality of the product. Re–analysis of the functional viewpoint is advised (backtracking). This should be carried out in the order of functional feature instantiation if the design intent is not to be lost.

WARNING:Application of tapers for manufacturing objectives may invalidate the functional relationships within the product

Advise re–analysis of functional features in the given order before proceeding:

0. Ronan_Fsid_w0
1. Ronan_Fbs_wl0

Select modification/re–analysis option on main menu

Re–analysis of the functional viewpoint is carried out by the designer, as advised, in the order shown. After functional re–analysis, mouldability re–analysis is automatically performed. The functional and mouldability re–analysis strategies are shown in Figures 9.4 and 9.6 respectively. For re–analysis the designer is asked for the identity of the form feature to re–analyse. No problems exist for form feature Ronan_Fsid_w0 (analysis not shown). The functionality of form feature Ronan_Fbs_wl0 has been affected by mouldability changes, but not because of the taper application. Instead the reduction of the wall thickness in response to mouldability advice has destroyed the functionality of the product, and the form feature must be repositioned. After functionality re–analysis and adjustments no further problems exist for mouldability.

FORM FEATURE Ronan_Fbs_wl0
FUNCTIONALITY ASSESSMENT – enclose_below function:

Top of base wall is not in contact with base of horizontal enclosure wall

Consequences:
1. Product functionality lost

Remedial options:
1. Relocate base wall to z position 8

–No further options

Reposition the feature? y/ny

New feature position: 0 0 8
Diameter satisfactory for enclose below specification
Feature outer diameter satisfactory for enclose below function

MOULDABILITY WALL FEATURE: Ronan_Mwall1

Wall thickness ok

Wall thickness relative to adjacent wall ok

Figure 10.6 shows the results of the above experiment in the Product model. The Figure shows the two product function instantiations, those of the form features and the primary and secondary mouldability instantiations.

The above experiment has shown:

1. That an injection moulding strategist of the structure described in Chapter 9 can support the designer in the concurrent design of an injection moulded product.

2. A functional interface application facilitates the build up of a product definition by supporting the designer in the association of function and form.

3. A translation process allows the interaction of the functional and mouldability viewpoints.

4. The functional interface application invokes functional data from the Product Range Model and relates this to individual feature instantiations in the Product model.

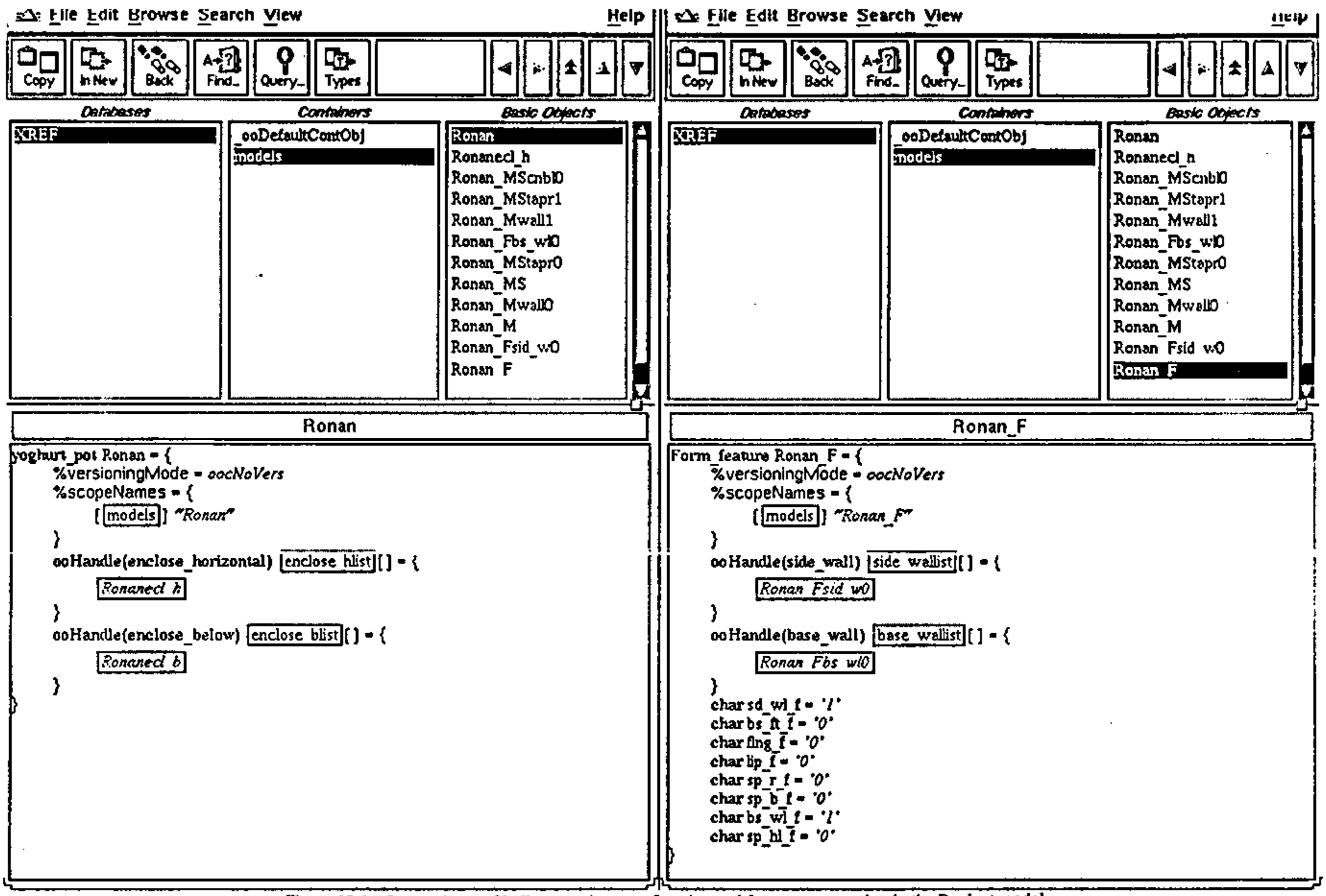


Figure 10.6a. Screen dump of built up yoghurt pot function and form representation in the Product model.

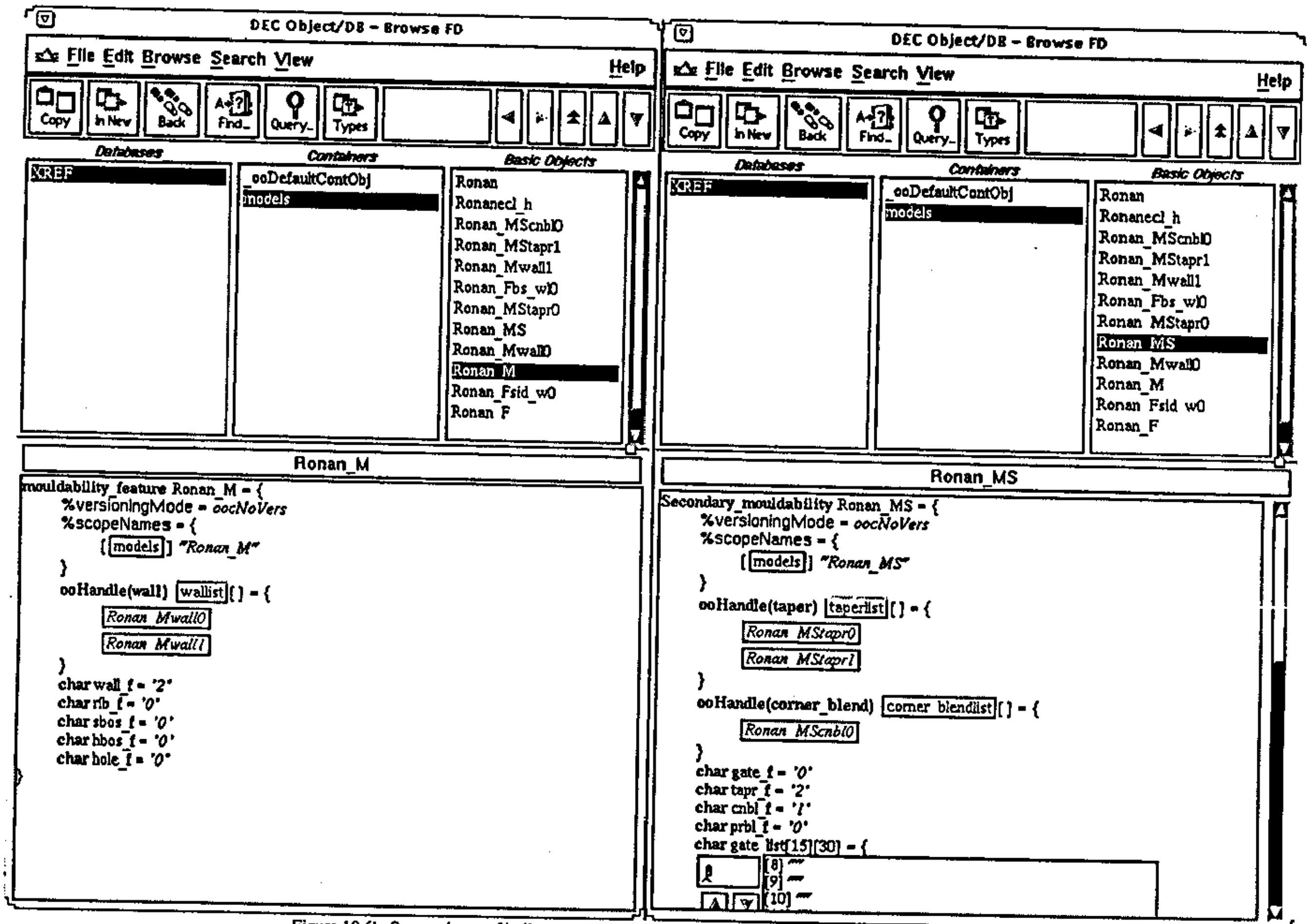


Figure 10.6b. Screen dump of built up yoghurt pot mouldability representation in the Product model.

5. The mouldability strategist application invokes mouldability data from the Manufacturing model and relates this to individual feature instantiations in the Product model.

6. The Product Range Model provides feedback advice to the designer about product functionality.

7. The Manufacturing model provides feedback advice to the designer about product mouldability.

The entire interaction between the designer and the injection moulding strategist in above experiment is shown in Appendix 7.

10.2.4. Supporting concurrent design for function and mouldability on a PTPlus.

The experiment described in this section has been performed to demonstrate how an injection moulding strategist supports the designer in considering the interaction of functional and mouldability constraints on a PTPlus. The objectives are identical to those listed for the previous experiment, in order to show that different product types can be considered. Additional objectives can be listed below:

i. To show how a taper application can affect the functionality viewpoint.

ii. To show how an injection moulding strategist deals with gaps in wall geometry and with one–to–many relationships between form and mouldability.

In this experiment the designer creates a new product of the PTPlus product range. Therefore the functional interface strategy for supporting the instantiation of an initial product definition is invoked. (Figure 9.2). The plastic part that is built up in the experiment is shown in Figure 10.7. The initial product definition of a PTPlus comprises three product

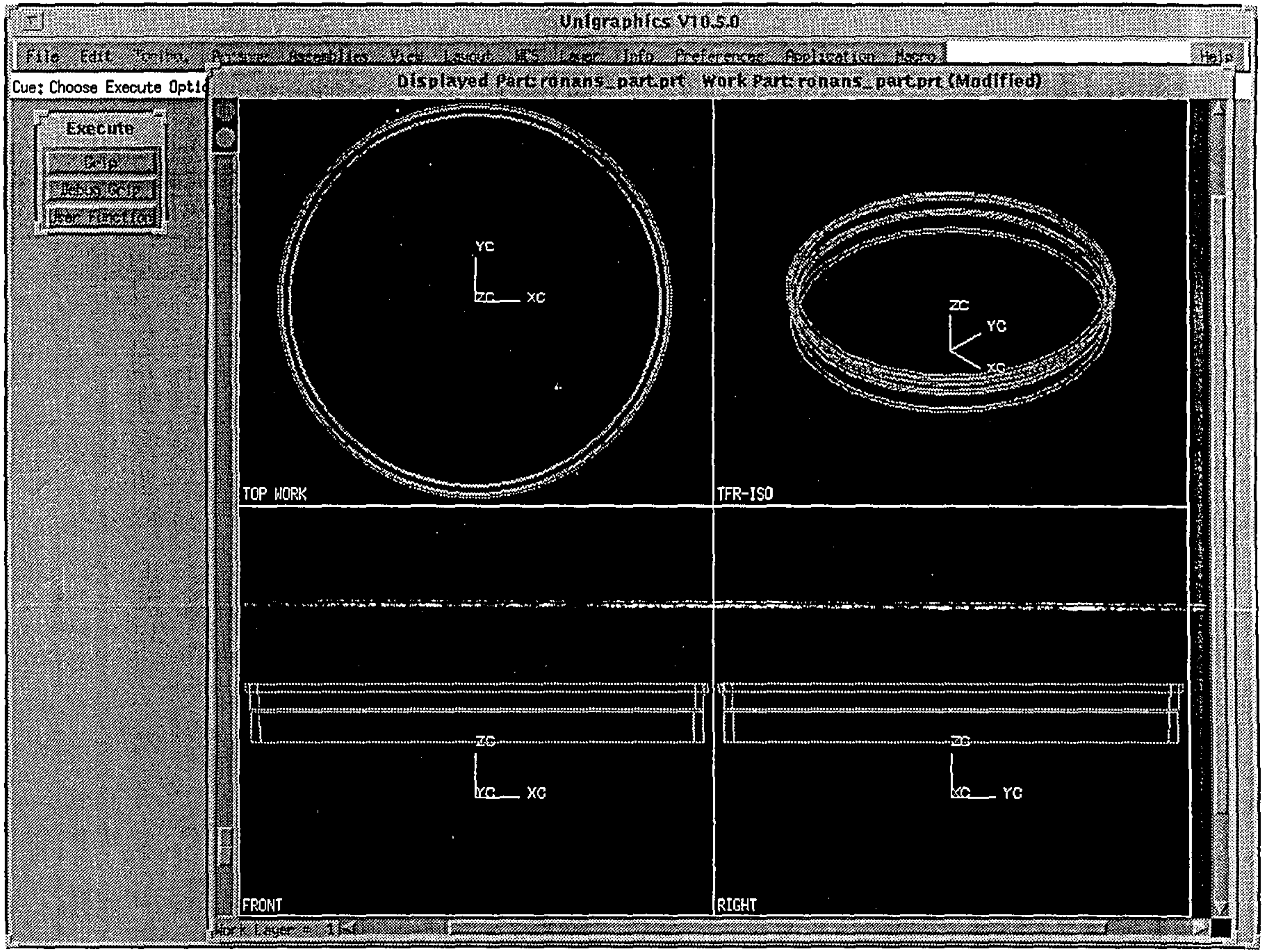


Figure 10.7. Screen dump of built up PTPlus geometry in a solid model.

function instantiations; locate in lid, break in torsion and locate on jar, and three associated form feature instantiations; a side wall to achieve the first and the third, and spaced bosses to achieve the second. Also there are the five mouldability instantiations which are the equivalent of the form viewpoint and associated secondary mouldability instantiations such as tapers and blends. Subsequently the designer goes on to address an individual product function outside the initial product definition.

Having identified the product range as PTPlus by asking the designer to choose, the functional interface application identifies the product functions that must be addressed and asks the designer to provide a specification for those product functions that have been identified as within the initial product definition. The data required for each specification is relayed to the designer using information in the product function class in the Product Range Model, eg locate in lid.

Functional requirements:

locate_in_lid
locate_on_jar
break_in_torsion
hold_in_lid
hold_on_jar
prevent_rotation
cover_lid_edge

Specify functional requirements for initial product definition:

FUNCTION – locate_in_lid

What is the inner diameter of the metal lid? (mm)52.44

Note: location surface contains flange for 'hold_in_lid' function as well as mating with inside lid surface
What is the height of the location surface? (mm)3

FUNCTION – break_in_torsion

What is the breakage torsion required? (Nmm)1220

FUNCTION – locate_on_jar

What is the outer diameter of the jar neck? (mm)50.08

What is the height of the location surface? (mm)3.5

After providing the specification the designer is supported in associating each product function in the initial product definition with a form feature using the form/function data in the Product Range Model. In this experiment only one type of form feature is identified as usable to achieve the locate in lid function. The choice is a side wall. Therefore the designer is supported in providing the dimensions of a side wall form feature instantiation.

FUNCTION:locate_in_lid

Forms available for use:
1.side_wall
No more forms available.

FORM FEATURE- SIDE_WALL: Ronan_Fsid_w0

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (base of central axis of rotation):0 0 10

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

specify inner diameter:50.44

specify side_wall thickness:1.3

specify side_wall height:2

After instantiation of the form feature in the Product model, the association with the functional viewpoint (Figure 7.3) is used by the functional interface application to identify the functional constraints in the Product Range Model. Using data from the PTPlus – locate in lid class in the Product Range Model, functional feedback advice is provided to the designer with respect to the functionality of the form feature. In this experiment the diameter of the side wall must be reduced to satisfy the locate in lid function. and the height must be increased.

FORM FEATURE Ronan_Fsid_w0
FUNCTIONALITY ASSESSMENT- locate_in_lid function:

Feature height is smaller than the height of the lid location surface
The location surface height is 3 mm.

Consequence:
The location surface is under utilised
Possible problems with 'hold_in_lid' function
Change feature height?y/ny

Enter new height:3

Feature position is currently: 0 0 10
Do you wish to adjust the position for the new location?y/nn

Feature outer diameter is greater than the inside diameter of the lid
The location diameter is 52.44 mm.

Consequence:
This is an interference fit
Change feature diameter?y/ny

New outer diameter: 52.44

To maintain constant inner diameter new thickness should be: 1
Do you wish to adjust the thickness?y/ny

New thickness: 1

After functionality has been evaluated the mouldability strategist is invoked and a translation is made from the form viewpoint to that of mouldability. The designer is provided with mouldability feedback advice using data from the wall class in the Manufacturing model. This application strategy is shown in Figure 9.5. The dimensions of the wall instantiation are acceptable for mouldability. However a taper is required on the wall instantiation to facilitate removal of the product from a mould. The designer is supported in instantiation of a taper feature and provided with feedback advice with respect to taper constraints using data from the wall class and taper class in the Manufacturing model respectively. Additionally the designer decides to gate on the wall and is supported in instantiating a gate feature and provided with feedback advice with respect to the feeding distance and the gate type in respect of the product geometry using data from the wall class and gate class in the Manufacturing model respectively.

MOULDABILITY WALL FEATURE: Ronan_Mwall0

Wall thickness ok

Wall features require a taper

Consequences of non–inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/ny

Creating taper on wall Ronan_Mwall0
Enter taper angle:
Recommended minimum draft angle = 0.8 degrees.8

Taper angle ok
Do you wish to create a new gate on this wall?y/ny

Creating gate on wall Ronan_Mwall0

Enter gate position X Y Z26.22 0 12

Feeding distance ok

This product is tubular:

Possible choices of gate type:
1. Rectangular edge gate
2. Pin gate

3. Diaphram gate
4. Ring gate

Enter choice:1

Gate type is rectangular edge gate

Having addressed the first product function in the initial product definition and considered the functional and manufacturing constraints, the designer is supported in addressing the second product function in the initial product definition. Using the form/function data in the Product Range Model it is identified that only one type of form feature as usable to achieve the break in torsion function. The choice is spaced bosses. Therefore the designer is supported in providing the dimensions of a spaced bosses form feature instantiation.

FUNCTION:break_in_torsion

Forms available for use:
1.spaced_bosses
No more forms available.

FORM FEATURE- SPACED_BOSSES: Ronan_Fsp_bs0

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (base of group central axis):0 0 9.75

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

Specify number of bosses:5

Specify boss diameters:.17

Specify diameter between boss axes:52

Specify bosses height:.25

After instantiation of the second form feature in the Product model, the association with the functional viewpoint (Figure 7.3) is used by the functional interface application to identify the functional constraints in the Product Range Model. Using data from the PTPlus – Break in torsion class in the Product Range Model, functional feedback advice is provided to the designer with respect to the functionality of the form feature. The number of bosses in the form feature has to be reduced to avoid difficulty in removing the lid from the jar.

FORM FEATURE Ronan_Fsp_bs0
FUNCTIONALITY ASSESSMENT- break_in_torsion function:

Torque specification: 1220
Feature position satisfactory for break in torsion function
Group diameter satisfactory for break in torsion function
Torque calculation2013.45

Feature cross sectional area is too large to provide failure
at the torque specified

Possible consequences:
1. Difficulty in removing the lid from the jar

Remedial options:
1. Decrease the number of bosses
2. Decrease the diameter of the bosses
Recommended number of bosses: 3
Necessary boss diameter at present numbers: 0.103007

–No further options

Decrease the number of bosses? y/ny

Enter number of bosses:3

Recommended boss diameter: 0.171679

Change the diameter of bosses? y/nn

Present diameter recorded
New torque = 1208.07 Nm

Next the mouldability strategist is invoked and a translation is made from the form viewpoint to mouldability. The spaced bosses form feature translates to three separate mouldability solid boss instantiations. The designer is provided with feedback advice using data from the solid boss class in the Manufacturing model. A taper is required on a solid boss instantiation to facilitate removal of the product from a mould. However, from Figure 10.7 the solid bosses are eventually sandwiched between two walls. Therefore the advice to create a taper is ignored. With more than one mouldability instantiation in the Product model, feedback advice is also required on the relationships between mouldability instantiations. In this experiment the width and height of the solid bosses, the maximum dimension of which is related to the adjacent wall thickness, are acceptable. As well as the width and height evaluation a blend is required between adjoining mouldability wall instantiations. However the height of the solid bosses is only .25 mm and the minimum recommended blend radius is 0.5 mm. Therefore the advice is ignored.

MOULDABILITY SOLID BOSS FEATURE: Ronan_Msbos0

Solid boss orientation ok

Solid boss features require a taper

Consequences of non–inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/nn

Solid boss height ok

Solid boss width ok

Solid boss features require a blend

Possible consequences of non inclusion:
1. Stress concentrations in the component
2. Turbulent flow around the corner can cause surface defects

Do you wish to create a blend?y/nn

MOULDABILITY SOLID BOSS FEATURE: Ronan_Msbos1

Solid boss orientation ok

Solid boss features require a taper

Consequences of non-inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/nn

Solid boss height ok

Solid boss width ok

Solid boss features require a blend

Possible consequences of non inclusion:
1. Stress concentrations in the component
2. Turbulent flow around the corner can cause surface defects

Do you wish to create a blend?y/nn

MOULDABILITY SOLID BOSS FEATURE: Ronan_Msbos2

Solid boss orientation ok

Solid boss features require a taper

Consequences of non-inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/nn

Solid boss height ok

Solid boss width ok

Solid boss features require a blend

Possible consequences of non inclusion:
1. Stress concentrations in the component
2. Turbulent flow around the corner can cause surface defects

Do you wish to create a blend?y/nn

Having addressed the second product function in the initial product definition and considered the functional and manufacturing constraints, the designer is supported in addressing the third product function in the initial product definition. Using the form/function data in the Product Range Model it is identified that only one type of form feature is usable to achieve the locate on jar function. The choice is a side wall. Therefore the designer is supported in providing the dimensions of a side wall form feature instantiation.

FUNCTION:locate_on_jar

Forms available for use:
1.side_wall
No more forms available.

FORM FEATURE- SIDE_WALL: Ronan_Fsid_w1

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (base of central axis of rotation):0 0 6.25

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

Specify inner diameter:50.08

Specify side_wall thickness:1

The functional and mouldability evaluation is carried out as for the first and second product functions, but is not shown in this section. The full PTPlus initial product definition is shown in Appendix 8. The above completes the initial product definition. However the designer is advised that the application of tapers for mouldability may affect the functionality of the product. Re–analysis of the functional viewpoint is advised (backtracking). This should be carried out in the order of functional feature instantiation if the design intent is not to be lost.

WARNING:Application of tapers for manufacturing objectives may invalidate the functional relationships within the product

Advise re–analysis of functional features in the given order before proceeding:

0. Ronan_Fsid_w0
1. Ronan_Fsp_bs0
2. Ronan_Fsid_w1

Select modification/re–analysis option on main menu

As advised, re–analysis of the functional viewpoint is carried out by the designer in the order shown. After functional re–analysis, mouldability re–analysis is automatically performed. The functional and mouldability re–analysis strategies are shown in Figures 9.4 and 9.6 respectively. For re–analysis the designer is asked for the identity of the form feature to re–analyse. Only re–analysis of the first product function is shown below, and the remainder can be seen in Appendix 8. The functionality of the form feature Ronan_Fsid_w0 has been affected by the application of a mouldability taper. At the beginning of the experiment the diameter of the form feature had to be reduced to achieve the locate in lid function. However the application of a mouldability taper requires that the diameter be reduced again. After functionality analysis no further problems exist for mouldability.

FORM FEATURE Ronan_Fsid_w0
FUNCTIONALITY ASSESSMENT– locate_in_lid function:

Feature height satisfactory for locate in lid function

Feature outer diameter is greater than the inside diameter of the lid
The location diameter is 52.44 mm.

Consequence:
This is an interference fit
Change feature diameter?y/ny

New outer diameter: 52.44

To maintain constant inner diameter new thickness should be: 0.979055
Do you wish to adjust the thickness?y/nn

MOULDABILITY WALL FEATURE: Ronan_Mwall0

Wall thickness ok

Wall thickness relative to adjacent wall ok

After completion of the initial product definition the designer is given the opportunity of addressing the remaining product functions in any order he wishes as one of the menu options. In this experiment the designer chooses to go on to address the hold in lid function.

1. Go on to Interactive product Modification design phase
2. Modification/re–analysis of existing forms
3. Display options
4. Go on to mould design
5. End session1

Functional requirements:
1. hold_in_lid
2. hold_on_jar
3. prevent_rotation
4. cover_lid_edge
Select a product function.1

When addressing individual product functions the designer is asked to provide a a specification for the chosen function. The data required for the specification is relayed to the designer using information in the hold in lid class in the Product Range Model. After providing the specification the designer is supported in associating the product function with a form feature using the form/function data in the Product Range Model. First the designer is provided with a choice of forms to fulfil the product function. In this experiment only one form feature is identified as usable to achieve the hold in lid function. The choice is a flange. Therefore the designer is supported in providing the dimensions of a flange form feature instantiation.

FUNCTION:hold_in_lid

Forms available for use:
1.flange
No more forms available.

FORM FEATURE- FLANGE: Ronan_Fflang0

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (base of axis of rotation):0 0 12

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

Specify inner diameter:53

Specify flange width:.5

Specify flange thickness:1

The designer goes on to consider the interaction of the functional and mouldability constraints on the form feature Ronan_Fflang0. The remainder of the experiment can be seen in Appendix 8. Figure 10.8 shows the results of the above experiment in the Product model. The Figure shows four product function instantiations, ie three from the initial product definition and the individual hold in lid function. Also shown are the instantiations of four form features to achieve the product functions and the equivalent primary and secondary mouldability instantiations.

The above experiment has shown:

1. That an injection moulding strategist of the structure described in Chapter 9 can support the designer in the concurrent design of an injection moulded product.

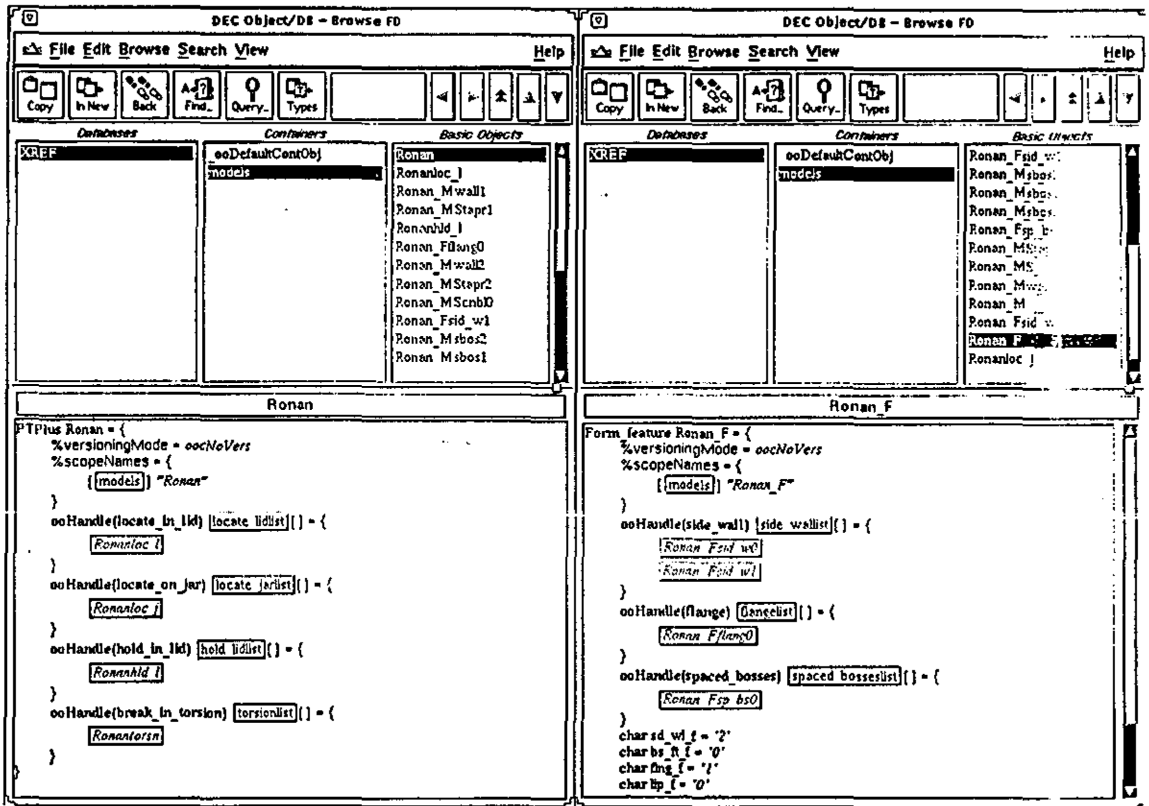


Figure 10.8a. Screen dump of built up PTPlus function and form representation in the Product model.

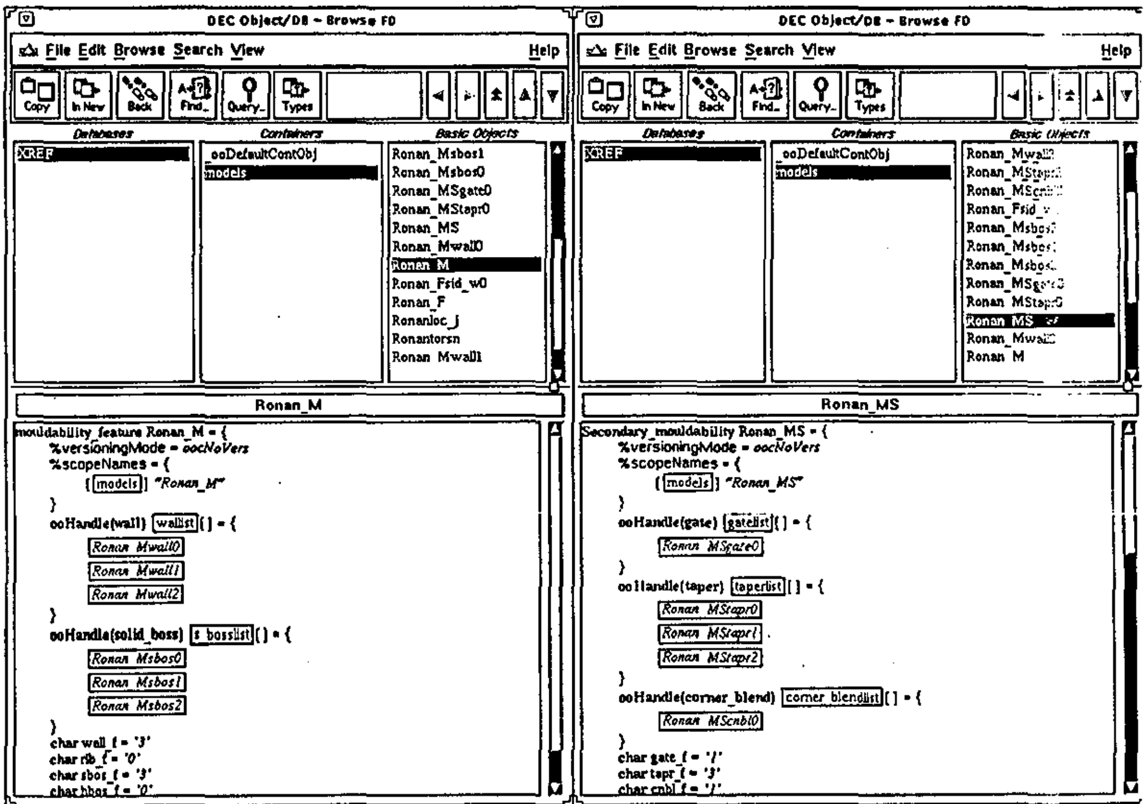


Figure 10.8b. Screen dump of built up PTPlus mouldability representation in the Product model.

2. A functional interface application facilitates the build up of a product definition by supporting the designer in the association of function and form.

3. A translation process allows the interaction of the functional and mouldability viewpoints.

4. The functional interface application invokes functional data form the Product Range Model and relates this to individual feature instantiations in the Product model.

5. The mouldability strategist application invokes mouldability data from the Manufacturing model and relates this to individual feature instantiations in the Product model.

6. The Product Range Model provides feedback advice to the designer about product functionality for 3D geometry.

7. The Manufacturing model provides feedback advice to the designer about product mouldability for 3D geometry.

8. The injection moulding strategist can deal with gaps in wall geometry and with one–to–many relationships between form and mouldability.

9. The injection moulding strategist can support the designer in considering the effect of mouldability taper applications on the functional viewpoint.

10. The designer can progress from an initial product definition to address remaining product functions individually.

10.2.5. Supporting concurrent design for function and mouldability on a Flower pot.

The experiment described in this section has been performed to demonstrate how an injection moulding strategist supports the designer in considering the interaction of functional and mouldability constraints on a Flower pot. The objectives are identical to those shared by the yoghurt pot and PTPlus experiments. However in addition to further emphasising that a variety of product types may be considered, this experiment shows support for developing a product that already exists in the product model, as opposed to creating a new product. Additional objectives are to show consideration of a more complex relationship between function and form than existed in previous experiments, and can be listed below:

i. To show how the designer is supported in the association of function and form where a choice of form features exists.

ii. To show how the designer is supported in the association of function and form where a form can be used to achieve more than one product function on a product range.

In this experiment the designer modifies an existing flower pot design. Therefore the functional interface strategy for supporting the instantiation of individual product functions is invoked (Figure 9.3). The plastic part that is built up in this experiment is shown in Figure 10.9. The designer enters the system and chooses a product function to address. In this experiment the drainage clearance function is chosen which provides space under the holes to allow water to flow out. For the flower pot product range the same form feature instantiation can be used to achieve the insert destack function. Therefore the functional interface application asks the designer for a specification for the chosen product function and informs him/her that the insert destack function can also be achieved, warning that this is conditional on instantiating the form feature below the base of the product and not on the inside. In this experiment the designer indicates to continue. Therefore the functional interface application asks the designer for a specification for the second product function. Note

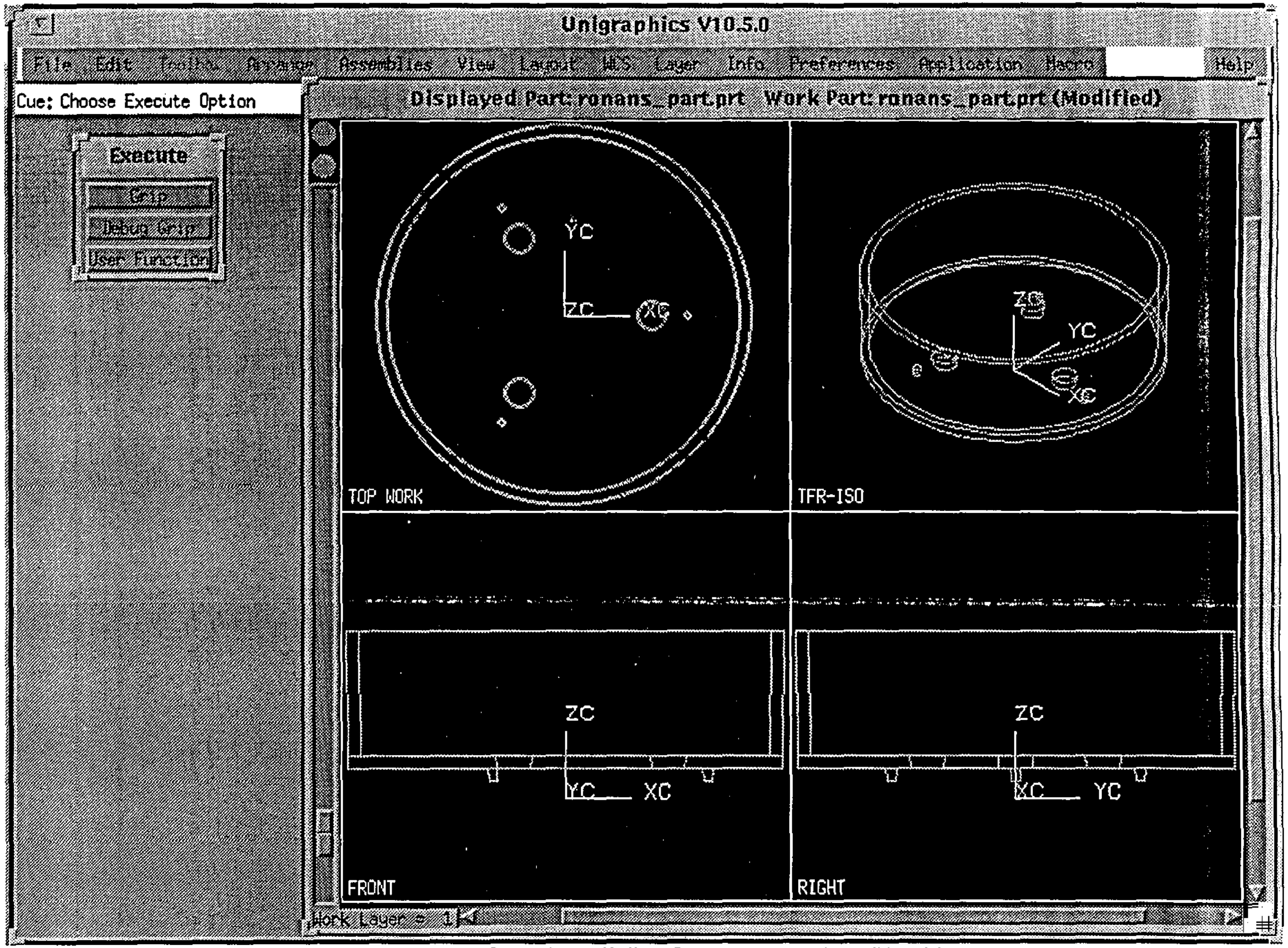


Figure 10.9. Screen dump of built up flower pot geometry in a solid model.

that within the insert destack specification the designer is informed that fulfilment of this product function negates the requirement for two others; section destack horizontal and section destack vertical, ie these product functions are a different way of achieving the same objective.

1.Create a new product (Initial product definition).
2.Modify existing product (Interactive product modification).
3.Delete a product and its mould
Enter choice.
2

Enter name of existing product.
Ronan

Functional requirements:
1. insert_destack
2. section_destack_horizontal
3. section_destack_vertical
4. drainage
5. drainage_clearance
Select a product function.5

FUNCTION – Drainage_clearance

What is the drainage clearance height (mm)1.5

SAME FORM:'Drainage_clearance' and 'Insert_destack' functions can be performed using the same form on flower pot type products:

Do you wish to use same form for 'Insert_destack' function?y/ny

NOTE: For 'Drainage_clearance' the feature MUST be below the base wall
Do you still wish to use the same form for 'Insert_destack'? y/ny

FUNCTION – Insert_destack

NOTE: Using 'insert_destack' function:
1. 'Section_destack_horizontal' function – no longer required.
2. 'Section_destack_vertical' function – no longer required.

What is the required protruding height of each (stacked) product? (mm)3

After providing the specification the designer is supported in associating the product functions with a form feature using the form/function data in the Product Range Model. First the designer is provided with a choice of forms to fulfil the product function. In this experiment two form features are identified as usable to achieve the product functions. The choice is either spaced bosses or spaced ribs. The designer chooses spaced bosses and is supported in providing the dimensions of the spaced bosses form feature instantiation.

FUNCTION:drainage_clearance

Forms available for use:
1.spaced_bosses
2.spaced_ribs

Select form.1

FORM FEATURE- SPACED_BOSSES: Ronan_Fsp_bs0

Do you wish to see feature dimensioning instructions?y/nn

Specify feature position (base of group central axis):0 0 3

Specify feature orientation 1 O 0 major axis is X direction
0 1 0 major axis is Y direction, 0 0 1 major axis is Z direction:0 0 1

Specify number of bosses:3

Specify boss diameters:2

Specify diameter between boss axes:70

Specify bosses height:4

After instantiation of the form feature in the Product model, the association with the functional viewpoint (Figure 7.3) is used by the functional interface application to identify the functional constraints in the Product Range Model. Using data from the Flower pot – drainage clearance and Flower pot – insert destack classes in the Product Range Model, functional feedback is provided to the designer with respect to the functionality of the functional feature instantiation. In this experiment feedback advice to reduce the spaced bosses height for the drainage clearance function is ignored but is subsequently taken for the insert destack function.

FORM FEATURE Ronan_Fsp_bs0
FUNCTIONALITY ASSESSMENT- drainage_clearance function:

Position of boss grouping satisfactory for 'drainage_clearance' function

Boss height is higher than specified to achieve 'drainage_clearance' function

Consequences:
1. Drainage clearance is greater than specified – unecessary material in product

Remedial options:
1. Reposition feature to z position 5.5
Change position? y/nn

Present height recorded

FORM FEATURE Ronan_Fsp_bs0
FUNCTIONALITY ASSESSMENT- insert_destack function:

Position of boss grouping satisfactory for insert destack function

Boss height is higher than specified to achieve 'insert_destack' function

Consequences:
1. Destack height is greater than specified – unecessary material in product

Remedial options:
1. Reposition feature to z position 4
Change position? y/ny

New position: 0 0 4

After functionality has been evaluated the mouldability strategist is invoked and a translation is made from the form viewpoint to mouldability. The designer is provided with mouldability feedback advice using the solid boss class in the Manufacturing model. This application strategy is shown in Figure 9.5. The dimensions of the three solid boss instantiations which are the equivalent of the spaced bosses instantiation are acceptable for mouldability. However a taper is required on each to facilitate removal of the product from a mould and a blend is required on each to avoid stress concentrations in the component or surface defects. Only one of the mouldability instantiation evaluations is shown here and the remainder of the experiment can be seen in Appendix 9.

MOULDABILITY SOLID BOSS FEATURE: Ronan_Msbos0

Solid boss orientation ok

Solid boss features require a taper

Consequences of non–inclusion of a taper can be difficulty in
removal of the component from the mould

Do you wish to create a taper?y/ny

Creating taper on solid boss Ronan_Msbos0
Enter taper angle:
Recommended minimum draft angle = 5.0 degrees5.0

Taper angle ok

Solid boss height ok

Solid boss width ok

Solid boss features require a blend

Possible consequences of non inclusion:
1. Stress concentrations in the component
2. Turbulent flow around the corner can cause surface defects

Do you wish to create a blend?y/ny

Creating blend on solid boss Ronan_Msbos0
Enter blend radius:
Recommended minimum radius = 0.5 mm.5

Blend radius ok

Figure 10.10 shows the results of the above experiment in the Product model. The figure shows four product function instantiations, two from the initial product definition and the individual drainage and drainage clearance functions. Note that a separate insert destack specification is **not** instantiated but this data is contained in the drainage clearance instantiation. This must be so in order that if functional feedback shows that the two specification objectives cannot be reconciled, the designer has the option of ignoring one of the product functions and making a subsequent second form feature instantiation to achieve the ignored product function.

The above experiment has shown:

1. That an injection moulding strategist of the structure described in Chapter 9 can support the designer in the concurrent design of an injection moulded product.

2. A functional interface application facilitates the build up of a product definition by supporting the designer in the association of function and form.

3. A translation process allows the interaction of the functional and mouldability viewpoints.

4. The functional interface application invokes functional data form the Product Range Model and relates this to individual feature instantiations in the Product model.

5. The mouldability strategist application invokes mouldability data from the Manufacturing model and relates this to individual feature instantiations in the Product model.

6. The Product Range Model provides feedback advice to the designer about product functionality.

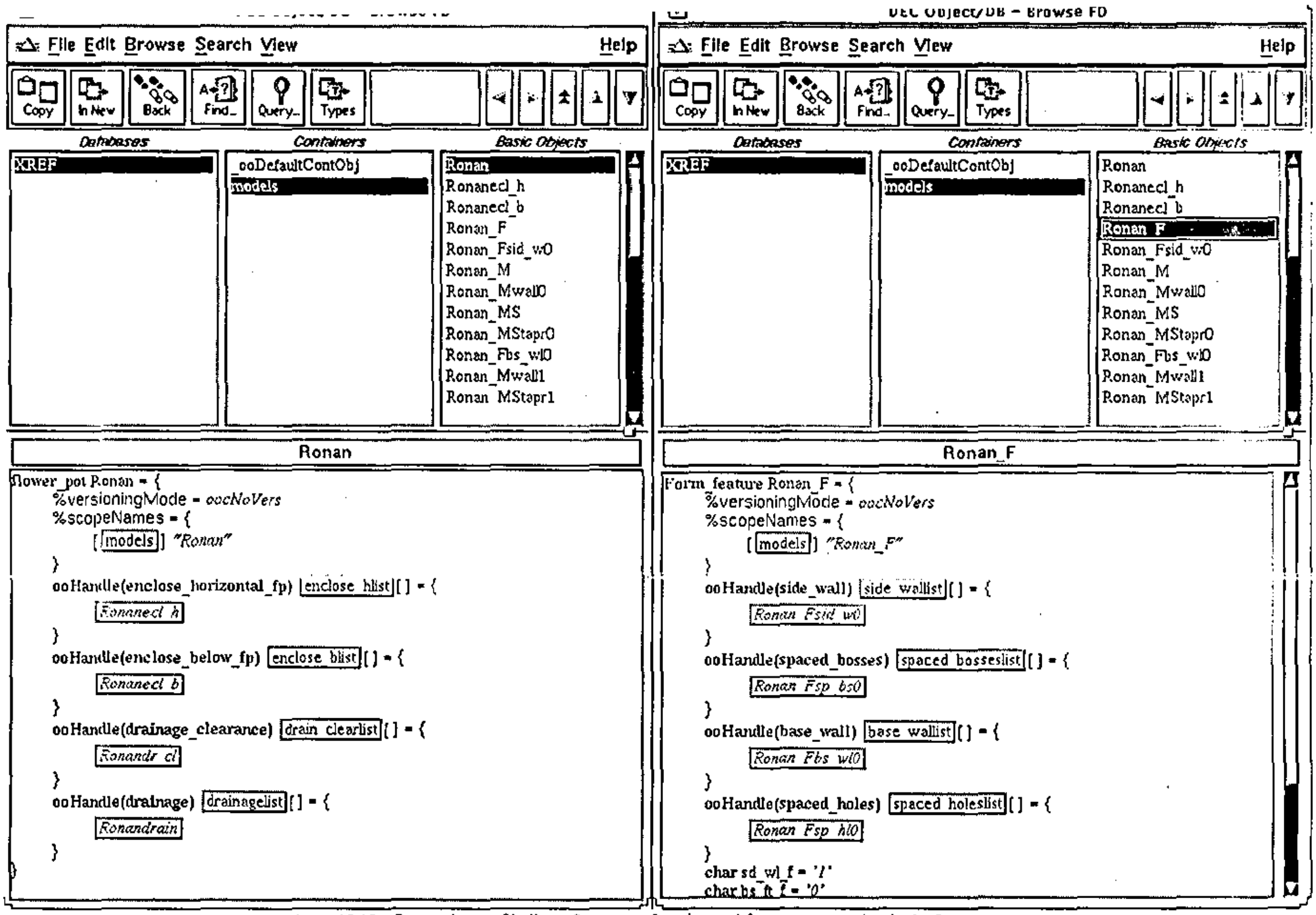


Figure 10.10a. Screen dump of built up flower pot function and form representation in the Product model.

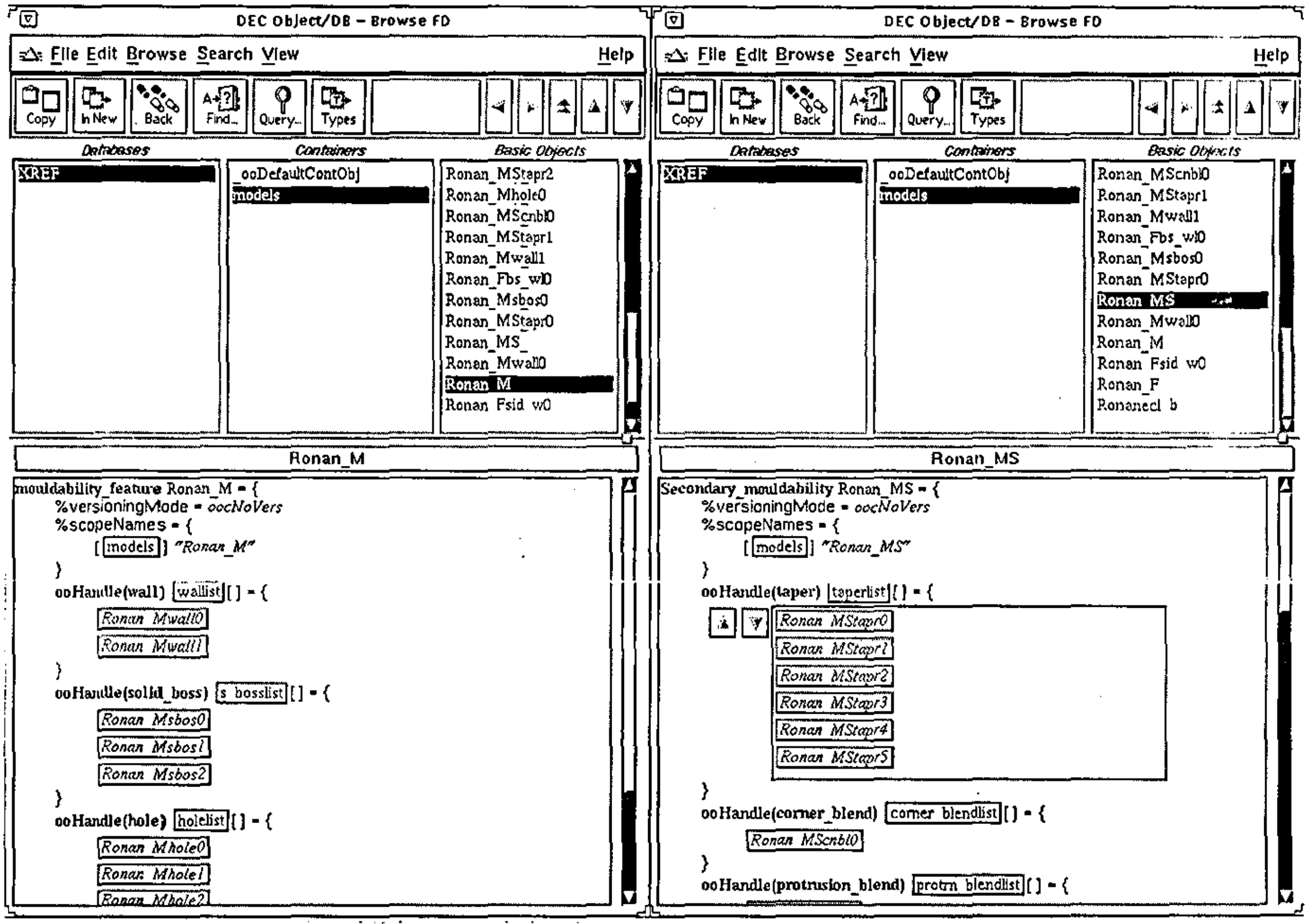


Figure 10.10b. Screen dump of built up flower pot mouldability representation in the Product model.

7. The Manufacturing model provides feedback advice to the designer about product mouldability.

8. The designer can be supported in the association of function and form where a choice of form features exists.

9. The designer can be supported in the association of function and form where a form can be used to achieve more than one product function on a product range.

10.2.6. Supporting concurrent product and mould design for a flower pot.

The experiment described in this section has been performed to demonstrate how an injection moulding strategist supports the designer in considering the interaction of constraints for the design viewpoints of the product and mould when building up a flower pot design. The objectives can be listed below:

i. To show how the designer is supported in considering the interaction of product and mould process constraints.

ii. To demonstrate the link between the geometry of a product and that of a mould using a set of translations between design viewpoints.

iii. To show that the designer can be supported in considering the options for the configuration of a mould design.

In this experiment the designer considers the options for design of the mould during build up of a flower pot product design. Therefore the cavity and core design strategists are invoked as well as the feeding system design strategist and the cooling system design strategist. The starting point for the product is shown in Figure 10.11. This experiment shows

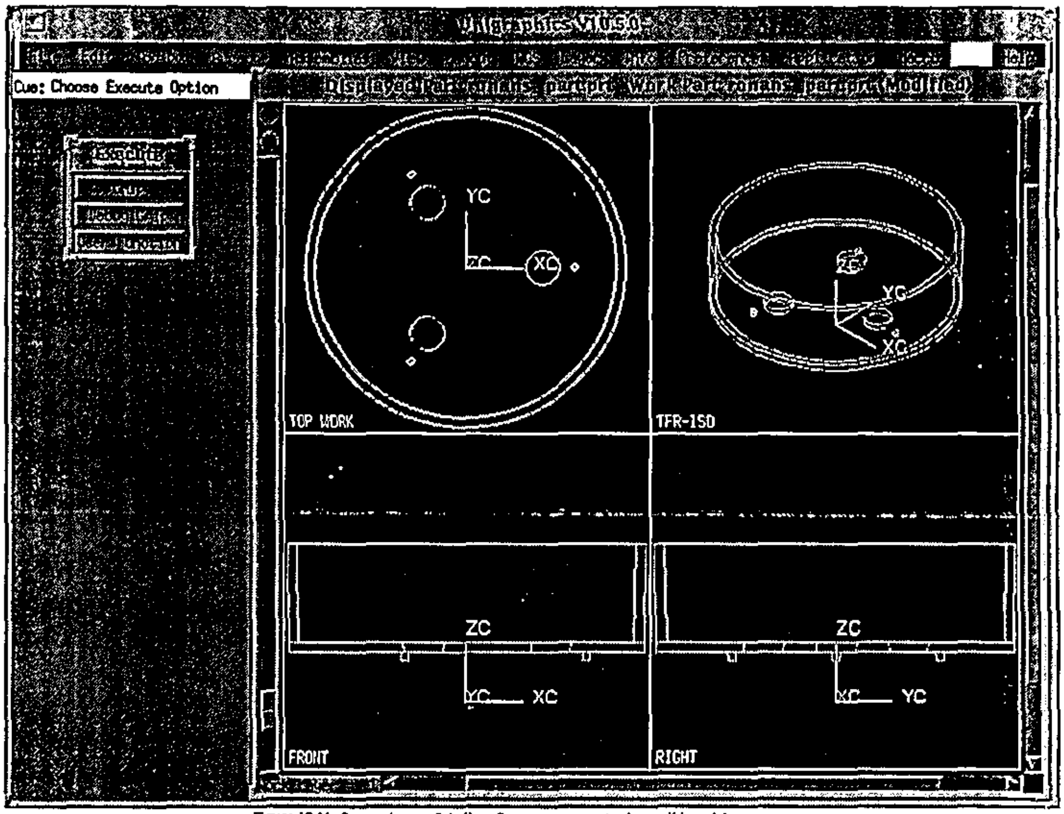


Figure 10.11. Screen dump of shallow flower pot geometry in a solid model.

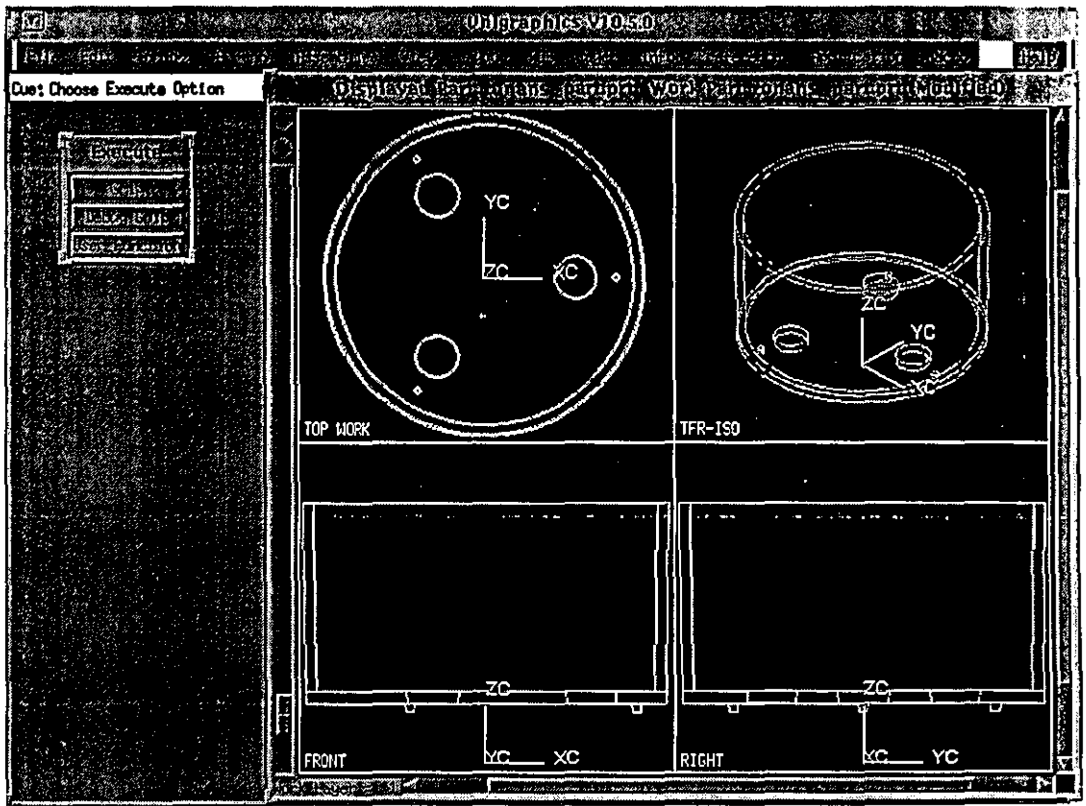


Figure 10.12. Screen dump of deep flower pot geometry in a solid model.

how the designer is supported in considering the options for mould design with respect to the flower pot design in Figure 10.11, when the designer has opted for a sprue gate. Subsequently the designer changes the dimensions of the flower pot to make it deeper and smaller in diameter, as shown in Figure 10.12. The designer then analyses the options for mould design with respect to an edge gated component. The entire experiment can be seen in Appendix 10.

Having built up the product geometry to be as in Figure 10.11, the designer selects the menu option for mould design, is warned that changes to mould geometry also change the product geometry, and opts to continue with the mould design.

1. Go on to Interactive product Modification design phase
2. Modification/re-analysis of existing forms
3. Display options
4. Go on to mould design
5. End session4

WARNING: Dimensional changes to cavity/core elements and application of tapers during mould design will result in dimensional changes and creation of tapers on corresponding product features

As a consequence of the above, functional relationships within the product may be invalidated

Re-analysis of all functional features in the product must be undertaken in original order of creation

Are you sure you want to go on to mould design?y/ny

The first strategist triggered in the mould design cycle is the cavity design strategist, as shown in Figure 9.8. This strategist identifies the parting line and translates from the mouldability viewpoint to the cavity, starting with the walls. The walls are translated to the cavity design viewpoint as described in section 7.5. The strategist identifies tapers and blends on walls and generates the equivalent cavity tapers and blends, providing feedback to the designer with respect to their parameters. The strategist identifies whether it is the smaller or larger radius of a corner blend that should be instantiated in a cavity blend. Overhangs are also checked for in the cavity. This application strategy is shown in Figure 9.9.

INTEGER CAVITY VOLUME FEATURE: Ronan_CAVitcv_v10

taper has angle 0.8

Volume diameter ok

Blend has inner 0.8 and outer 2.8

Radius for blend= 2.8

INTEGER CAVITY VOLUME FEATURE: Ronan_CAVitcv_vl1

taper has angle 0.8

Volume diameter ok, evaluation completed by analysis of previous feature

Blend ok, evaluation completed by analysis of previous feature

Having instantiated the parting line and the cavity equivalent of the mouldability wall geometry, the core design strategist is invoked to do the same in the core. The walls are translated to the core design viewpoint as described in section 7.6. As per the cavity, the core design strategist identifies tapers and blends on walls and generates the equivalent core tapers and blends, providing feedback to the designer with respect to their parameters. The strategist identifies whether it is the smaller or larger radius of a corner blend that should be instantiated in a core blend and overhangs are also checked for in the core. This application strategy is shown in Figure 9.13.

INTEGER CORE VOLUME FEATURE: Ronan_CORitcr_vl0

taper has angle 0.8

Volume diameter ok

Blend has inner 0.8 and outer 2.8

Radius for blend= 0.8

INTEGER CORE VOLUME FEATURE: Ronan_CORitcr_vl1

taper has angle 0.8

Volume diameter ok, evaluation completed by analysis of previous feature

Blend ok, evaluation completed by analysis of previous feature

As shown in Figure 9.11, having instantiated the main geometry of the cavity/core impression the cavity design strategist looks for local inserts in the cavity, ie mouldability solid and hollow bosses, ribs and holes. Three integer cavity hole features are instantiated as the cavity impression equivalent of the solid bosses on the base of the flower pot. When the cavity strategist identifies that a local insert is actually in the core, the core design strategist is invoked to instantiate the local insert in the core. Thus three integer core boss features are instantiated as the equivalent of the mouldability holes. However, as they are through holes, integer cavity hole features are also instantiated to provide a location hole in the cavity for an extended core pin. No feedback is provided for the cavity holes as their dimensions are governed by those of the integer core bosses. The strategy of the core design strategist is shown in Figure 9.15.

INTEGER CAVITY HOLE FEATURE: Ronan_CAVitcv_hl0

Blend has radius 0.5

taper has angle 5

INTEGER CAVITY HOLE FEATURE: Ronan_CAVitcv_hl0

⋮

Creating core boss and cavity hole

INTEGER CORE BOSS FEATURE: Ronan_CORitcr_bs0

Blend has radius 0.5

taper has angle 5

Creating core boss and cavity hole

INTEGER CORE BOSS FEATURE: Ronan_CORitcr_bs1

⋮

The geometry of the cavity and core impression is now complete. The feeding system strategist is now invoked which gives the designer a chance to change the gated wall and type of gating system. Subsequently the cooling system design strategist is invoked to support the designer in considering an aspect of the cooling system that is a prerequisite to building a feeding system design. The cooling system design strategist supports the designer in identifying the number of cooling tubes alongside the cavity as described in section 7.8, using the strategy in Figure 9.18.

Do you wish to change the gated wall? y/nn

CREATING FEEDING SYSTEM:

In order to design the feeding system it is necessary to first identify some parameters of the cooling system:

CAVITY COOLING SYSTEM: Ronan_CS

Number of cooling layers using 7 mm flow ways: 1

Number of cooling layers using 8 mm flow ways: 1

Number of cooling layers using 9 mm flow ways: 1

Number of cooling layers using 10 mm flow ways: 1

Choice of cooling tube diameter for cavity cooling system:
1. 7mm
2. 8mm
3. 9mm
4. 10mm

Lower than 7 mm – insufficient cooling effect – difficulty drilling deep holes

Higher than 10mm – high volume of water to be pumped around the mould for cooling

Maximum cooling effect for Ronan_CS cavity dimensions: 10 mm diameter

Enter choice (1–4):4

The feeding system design strategist now supports the designer in consideration of the process constraints of the feeding system. In this experiment the designer has chosen to use a sprue gate and therefore no runner system or main feeding sprue is required. Thus the designer is supported only in consideration of the gate dimensions. The feeding system application strategy is shown in Figure 9.17.

CREATING FEEDING SYSTEM:

FEEDING SYSTEM – SPRUE GATE:Ronan_FSsprue_g0

Gate position ok

Moulding machine nozzle inner diameter is 3mm

Gate diameter should be slightly larger then nozzle to allow for misalignment

Lower gate diameter has been calculated as 3.1 mm
Do you wish to change the diameter?y/nn

Sprue length has been calculated as 29 mm

This allows the minimum cavity block depth (below cavity) to avoid mould distorsion, and allows space for the cavity cooling system
Do you wish to change the sprue length?y/nn

Enter gate taper angle:
Recommended minimum angle = 4.0 degrees (minimum recommended)4

Taper angle ok

Having instantiated the feeding system the cooling system design strategist supports the designer in considering the appropriate configuration of cavity cooling system, as described in section 7.8. Having identified the appropriate cooling system configuration the standard flow ways are instantiated to build up the required cooling system geometry in the cavity In this experiment a paired tube configuration is chosen requiring two parallel standard flow ways. This cooling system application strategy is shown in Figure 9.19.

CAVITY COOLING SYSTEM:Ronan_CS

Choice of cooling system configurations for mould cavity:
1. paired tube configuration
2. U tube configuration

U tube configuration is not recommended when using sprue gate:

- Cooling flow ways on three sides can cause uneven cooling of the moulding

Possible consequences:
1. Differential section thickness over the moulding
2. Differential shrinkage causing component warping

Remedial options:
1. Used paired tube configuration

-No further options

Enter choice:1

STANDARD FLOW WAY:Ronan_CSstd_fl_w0

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_CAV

Configuration: pair

Vertical coordinate: 11 mm

⋮

Not all the dimensions of the cavity cooling system can be identified and the integer cavity plate features must be instantiated to enable completion of the cavity cooling system geometry. The designer is therefore supported in instantiating the cavity plate features by the cavity design strategist using the strategy in Figure 9.12. In this experiment the impression is undergated using a sprue gating system. Therefore as described in section 7.5 the mould plate includes an integer cavity mould block, a narrow (circular) inner land and four peripheral lands to increase the land surface area. It can be seen that although the guide system is not instantiated in the present work the size of the guide system is identified to determine the dimensions of the integer cavity block. As described in section 7.5 the recommended size of the guide system pins is based on the area of the cavity.

CREATING CAVITY BLOCK FOR: Ronan_CAV

INTEGER CAVITY RECTANGULAR MOULD BLOCK :Ronan_CAVitcv_rbl0

Cavity block position: 0 0 -19

Depth of cavity block: 48.6 mm

Choice of standard guide pin diameters:
1. 10 mm
2. 13 mm

3. 16 mm
4. 19 mm
5. 22 mm
6. 25 mm
7. 32 mm
8. 38 mm

Recommendation: Use smallest suitable guide pin diameter to minimise size and weight of mould assembly

Recommended size for current mould parameters: 3. 16 mm

Enter choice:(1–8):3

Guide pin diameter is 16 mm

Cavity block length: 178.659 mm

Cavity block width: 222 mm

INTEGER CAVITY CIRCULAR LAND:Ronan_CAVitcv_crl0

Circular land position: 0 0 29.6

Circular land depth:2.4 mm

Circular land diameter:84.5584 mm

INTEGER CAVITY PERIPHERAL LAND:Ronan_CAVitcv_pf0

Peripheral land position: 57.3296 57.3296 29.6

Peripheral land depth:2.4 mm

Peripheral land diameter:15.3297 mm

INTEGER CAVITY PERIPHERAL LAND:Ronan_CAVitcv_pf1

⋮

The remaining dimensions of the cavity cooling system are then identified using the strategy shown in Figure 9.20. The cooling system design strategist then looks at the cooling system in the core using the strategy in Figure 9.21. The core is identified as shallow and therefore the designer is supported in considering the appropriate configuration of a shallow core cooling system. The number of cooling tubes across the core is established using the maximum cooling capability calculation (section 7.8). Subsequently, the possible configurations of shallow core cooling system are identifed from the number of tubes that can fit across the core. In this experiment three tubes can fit across the core and the choice is either a Z tube or single tube configuration. Having identified the appropriate cooling system configuration the standard flow ways are instantiated to build up the required cooling system geometry in the core. In this experiment a single tube configuration is chosen requiring three parallel standard flow ways.

CORE COOLING SYSTEM:Ronan_CS

Number of standard flow ways using 7 mm flow ways: 3

Number of standard flow ways using 8 mm flow ways: 3

Number of standard flow ways using 9 mm flow ways: 3

Number of standard flow ways using 10 mm flow ways: 3

Choice of cooling tube diameter for shallow core cooling system:
1. 7mm
2. 8mm
3. 9mm
4. 10mm

Maximum cooling effect for Ronan_CS core dimensions: 10 mm diameter

Enter choice (1–4):4

Choice of cooling system configurations for shallow mould core at diameter 10 mm:

1. Z_tube configuration

2. single tube configuration

Using Z tube configuration 'Cooler' water entering the mould at the gated end provides uneven cooling of the moulding

Possible consequences:
1. Differential section thickness over the moulding
2. Differential shrinkage causing component warping

Recommendation – Use single tube configuration

Enter choice:2

STANDARD FLOW WAY:Ronan_CSstd_fl_w2

Position: –89.3296 –26 53

Length: 178.659 mm

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_COR

Configuration: single

STANDARD FLOW WAY:Ronan_CSstd_fl_w3

⋮

Finally the core design strategist is invoked in order to instantiate the core plate (Not shown) using the strategy in Figure 9.16. The mould cavity and core design that is generated is shown in Figures 10.13 and 10.14. After the first mould design cycle the designer opts to change the dimensions of the flower pot design to those in Figure 10.12, re–analysing functional and mouldability constraints before looking at the mould design again. This part of the experiment can be seen in Appendix 10. Design support for the instantiation of the cavity and core impression geometry in the second mould design cycle is exactly the same as above, except obviously for the changed dimensions of the impression. Design support is provided using the cavity design strategies in Figures 9.9 and 9.11, and the core design strategies in Figures 9.13 and 9.15. Having instantiated the geometry of the cavity and core impression the feeding system strategist is now invoked which gives the designer a chance to change the gated wall and type of gating system. In this experiment the designer opts to change from a sprue gate to a rectangular edge gate on a different wall.

Do you wish to change the gated wall? y/ny

Please indicate which wall you wish to gate:
0. Ronan_Mwall1
1. Ronan_Mwall0

Enter choice:1

Enter gate position X Y Z30 0 43

Feeding distance ok

This product is thin walled (thickness < 4mm) and rotational:

Possible choices of gate type:
1. Rectangular edge gate
2. Pin gate
3. Sprue gate

Enter choice:1

Gate type is rectangular edge gate

Afterwards as in the previous mould design cycle, the cooling system design strategist is invoked to support the designer in considering an aspect of the cooling system that is a prerequisite to building a feeding system design. The cooling system design strategist supports the designer in identifying the number of cooling tubes alongside the cavity as described

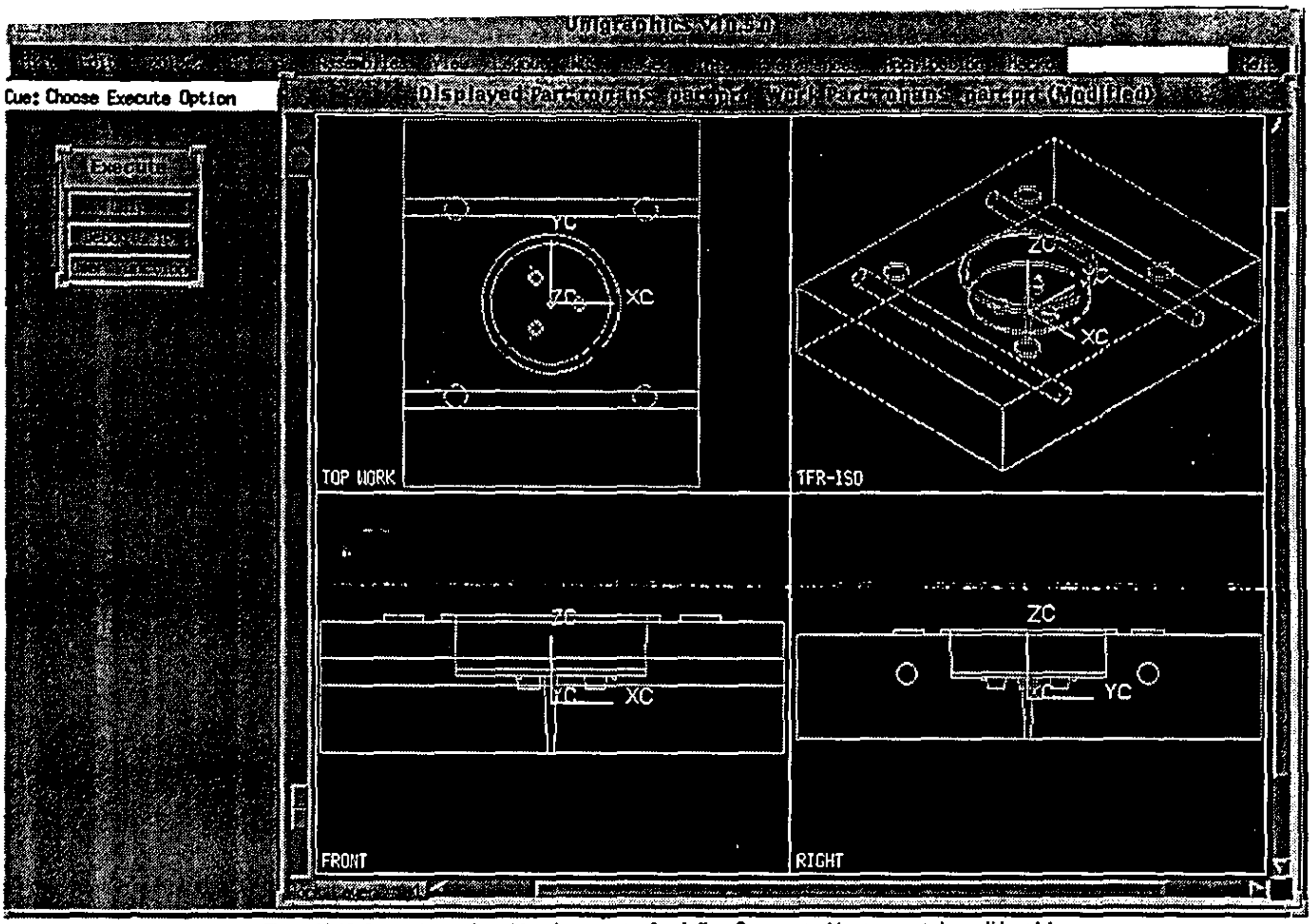


Figure 10.13. Screen dump of mould cavity geometry for shallow flower pot with a sprue gate in a solid model.

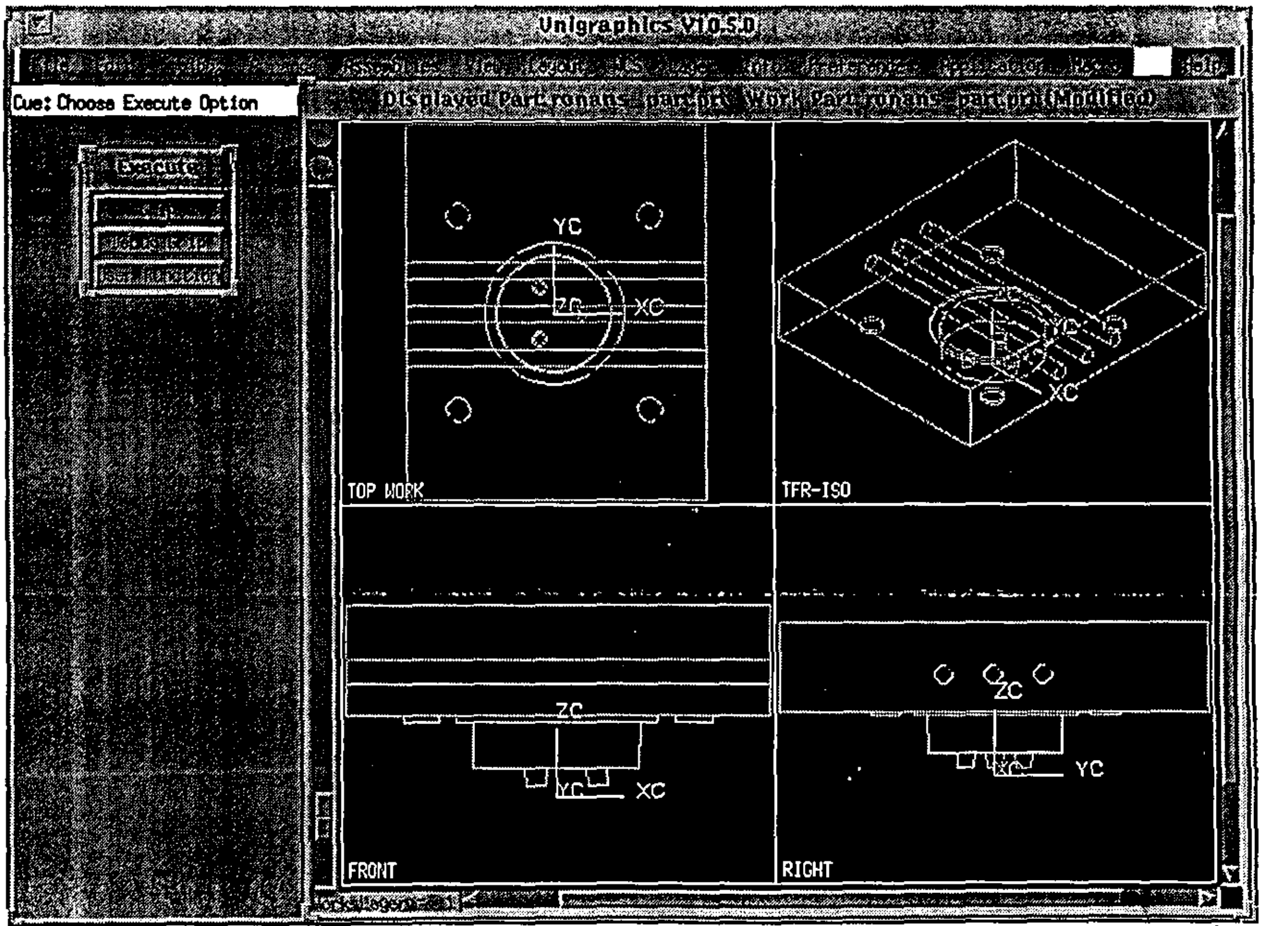


Figure 10.14. Screen dump of mould core geometry for shallow flower pot with a sprue gate in a solid model.

in section 7.8, using the strategy in Figure 9.18. The feeding system design strategist then supports the designer in consideration of the process constraints of the feeding system. In this experiment the designer has chosen a rectangular edge gate and therefore a circular runner system and a main feeding sprue is required. Note how the designer is supported in adjusting the gate position to be on the edge of the cavity. The feeding system application strategy is shown in Figure 9.17.

CREATING FEEDING SYSTEM:

In order to design the feeding system it is necessary to first identify some parameters of the cooling system:

CAVITY COOLING SYSTEM: Ronan_CS

Number of cooling layers using 7 mm flow ways: 1

⋮

Maximum cooling effect for Ronan_CS cavity dimensions: 10 mm diameter

Enter choice (1–4):4

CREATING FEEDING SYSTEM:

FEEDING SYSTEM – RECTANGULAR EDGE GATE:Ronan_FSrcted_g0

WARNING: Application of tapers on Ronan_Mwall0 for manufacturing reasons has increased the width of the cavity opening at the parting line by 0.865742 mm

Consequences:
1. Edge gate position no longer on edge of cavity:
– Reduced land length– weakness in mould construction can lead to wear or failure

Remedial options:
1. Adjust gate position

– No further options
Do you wish to adjust the gate position? y/ny

New gate postion: 30.4329 0 43

Gate position ok

Gate land length:

Land length should be as small as possible and in any case between 0.5 and 0.75 mm
Enter land length.5

Land length ok

Gate depth:

Gate depth has been calculated as 1.4 mm
Do you wish to change the depth?y/nn

Gate width:

Gate width has been calculated as 2.21946 mm
Do you wish to change the width?y/nn

FEEDING SYSTEM – CIRCULAR RUNNER:Ronan_FScirc_r0

Runner length:

Runner length has been calculated as 20.0671 mm

This calculation is based on minimum distance between cavity and main sprue
Do you wish to change the length?y/nn

Runner diameter:

Runner diameter has been calculated as 2 mm

This calculation is based on minimum diameter to ensure cavity is filled before plastic in runner solidifies
Do you wish to change the diameter?y/nn

FEEDING SYSTEM – MAIN FEEDING SPRUE:Ronan_FSmain_s0

Main sprue position: 50.9999 0 48
Enter main sprue taper angle:
Recommended minimum angle = 4.0 degrees (minimum recommended)4

Taper angle ok

Lower diameter of sprue = 3.1 mm to match machine nozzle diameter of 3 mm

No nozzle recess required

Sprue length has been calculated as 60 mm

This allows the minimum cavity block depth (below cavity) to avoid mould distorsion, and allows space for the cavity cooling system.
Note: 5 mm of the sprue length is to create a sprue puller in the core block
Do you wish to change the sprue length?y/nn

Having instantiated the feeding system, the cooling system design strategist supports the designer in considering the appropriate configuration of cavity cooling system, as described in section 7.8. Having identified the appropriate cooling system configuration the standard flow ways are instantiated to build up the required cooling system geometry in the cavity. In the second mould design cycle a U tube configuration is chosen requiring two parallel standard flow ways and a third perpendicular. The cooling system application strategy is shown in Figure 9.19.

CAVITY COOLING SYSTEM:Ronan_CS

Choice of cooling system configurations for mould cavity:
1. paired tube configuration
2. U tube configuration

U tube configuration recommended when using single rectangular edge gate:

Bottom of the U cooling the gated side provides more even cooling of the moulding, can reduce cycle time.

Enter choice:2

STANDARD FLOW WAY:Ronan_CSstd_fl_w0

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_CAV

Configuration: U_tube

Vertical coordinate: 22 mm
Making standard flow way

STANDARD FLOW WAY:Ronan_CSstd_fl_w1

⋮

As in the first mould design cycle, not all the dimensions of the cavity cooling system can be identified and the integer cavity plate features must be instantiated to enable completion of the cavity cooling system geometry. The designer is therefore supported in instantiating the cavity plate features by the cavity design strategist using the strategy in Figure 9.12. In the second cycle of mould design the impression is side gated using a rectangular edge gate. Therefore as described in section 7.5 the mould plate includes an integer cavity mould block, and a wide (rectangular) inner land. As before, although the guide system is not instantiated in the present work the size of the guide system is identified to determine the dimensions of the integer cavity block.

CREATING CAVITY BLOCK FOR: Ronan_CAV

INTEGER CAVITY RECTANGULAR MOULD BLOCK :Ronan_CAVitcv_rbl0

Cavity block position: 0 0 -12

Depth of cavity block: 52.6 mm

Choice of standard guide pin diameters:
1. 10 mm
2. 13 mm
3. 16 mm
4. 19 mm
5. 22 mm
6. 25 mm
7. 32 mm
8. 38 mm

Single rectangular edge gate causes unbalanced forces in the mould, tending to open the mould on one side

Possible consequences:
1. Larger wall section thickness one one side of the mould than on the other

Remedial options
Use guide pin size one larger than that recommended to ensure alignment of mould halves

–No further options

Recommended size for current mould parameters: 3. 16 mm

This product has a single rectangular edge gate – USE NEXT SIZE UP

Enter choice:(1–8):3

Guide pin diameter is 16 mm

Cavity block length: 186 mm

Cavity block width: 208 mm

INTEGER CAVITY RECTANGULAR LAND: Ronan_CAVitcv_rcl0

Rectangular land position: 0 0 40.6

Rectangular land depth:2.4 mm

Rectangular land length: 122.866 mm

Rectangular land width: 70.8656 mm

The remaining dimensions of the cavity cooling system are then identified as shown in Figure 9.20. Subsequently the cooling system design strategist looks at the cooling system in the core using the strategy in Figure 9.21. This time the core is identified as deep and therefore the designer is supported in considering the appropriate deep core cooling system. In the present work only a baffle tube configuration of deep core cooling system is considered. The number of cooling tubes inside the core is established using the maximum cooling capability calculation (section 7.8). Having identified the appropriate cooling system configuration and the maximum cooling capacity the standard flow ways, baffle flow ways and baffle blades are instantiated to build up the required cooling system geometry in the core.

CORE COOLING SYSTEM:Ronan_CS

Choice of cooling system configurations for deep mould core:

1. Baffled straight hole system –
Large cooling capacity, easy to manufacture.

2. Angled hole system –
Does not work for the deepest cores, hard to manufacture due to angled holes, small cooling capacity compared to baffle system.

3. Stepped circuit system –
Holes drilled through core into cavity, requiring plugging and finishing, small cooling capacity compared to baffle system.

Use baffle system for deep core:

Number of baffle flow ways using 12 mm flow ways: 1

Number of baffle flow ways using 13 mm flow ways: 1

Number of baffle flow ways using 14 mm flow ways: 1

Number of baffle flow ways using 15 mm flow ways: 1

Number of baffle flow ways using 16 mm flow ways: 1

Choice of cooling tube diameter for deep core cooling system:
1. 12mm
2. 13mm
3. 14mm
4. 15mm
5. 16mm

Maximum cooling effect for Ronan_CS core dimensions: 16 mm diameter

Enter choice (1–5):5

Number of baffle flow ways through the centre of the core : 1

Diameter of standard flow way to connect baffle flow ways has been calculated as 10 mm

STANDARD FLOW WAY:Ronan_CSstd_fl_w3

Position: -92.9999 0 69

Length: 186 mm

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_COR

Configuration: deep

BAFFLE FLOW WAY:Ronan_CSbff_fl_w0

Baffle flow way position: -2.49993 0 92.4

Baffle flow way diameter: 16 mm

Baffle flow way length: 64.4 mm

Configuration: deep

Cavity/core name: Ronan_COR

BAFFLE BLADE:Ronan_CSbff_bl0

Baffle blade: -2.49993 0 92.4

Baffle blade length: 48.4 mm

Baffle blade width: 15.5 mm

Baffle blade thickness: 2 mm

Configuration: deep

Cavity/core name: Ronan_COR

Finally the core design strategist is invoked in order to instantiate the core plate (Not shown) using the strategy in Figure 9.16. The mould cavity and core design that is generated from the second mould design cycle is significantly different than that from the first, as shown in Figures 10.15 and 10.16. The results of the above experiment in the Product model are shown in Figures 10.17 and 10.18. Figure 10.17 shows the two integer cavity volumes and the two integer core volumes that are the cavity and core equivalent of the mouldability walls in the flower pot. Figure 10.18 shows the elements of thefeeding system and the cooling system after the second design cycle, showing a rectangular edge gate, a circular runner and a main feeding sprue in the feeding system, three standard flow ways for a U tube in the cavity cooling system, and a fourth standard flow way, a baffle flow way and a baffle blade for a deep cooling system in the core. The above experiment can be seen in its entirety in Appendix 10.

The above experiment has shown:

1. That an injection moulding strategist of the structure described in Chapter 9 can support the designer in the concurrent design of an injection moulded product and its mould.

2. That a link is established via a set of translation processes between the geometry of the product and that of the mould, such that changes in product geometry result in changes in the geometry of the mould.

3. A translation process allows the interaction of design viewpoints within the product and the mould.

4. The cavity design strategist, the core design strategist, the feeding system design stra–

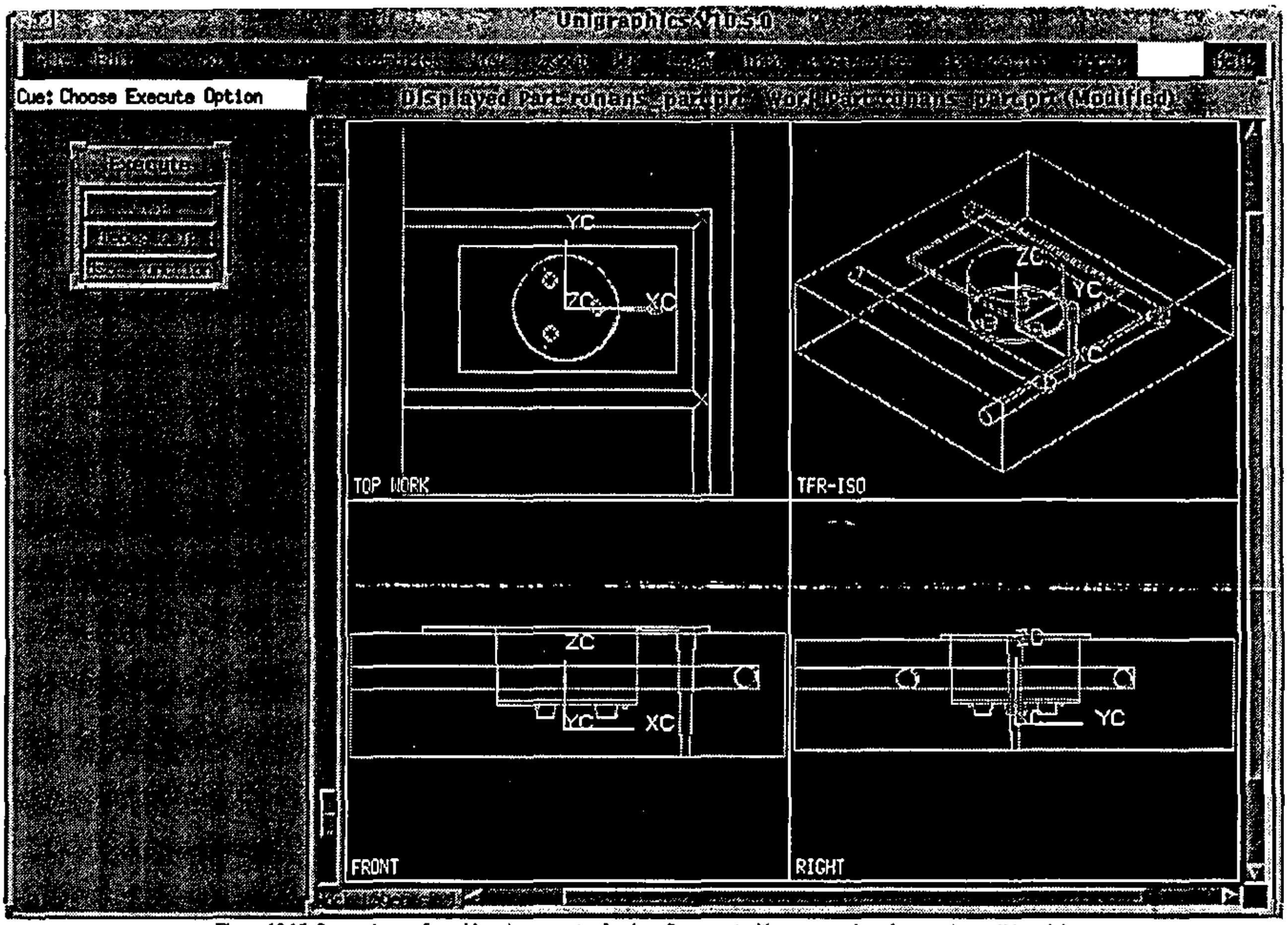


Figure 10.15. Screen dump of mould cavity geometry for deep flower pot with a rectangular edge gate in a solid model.

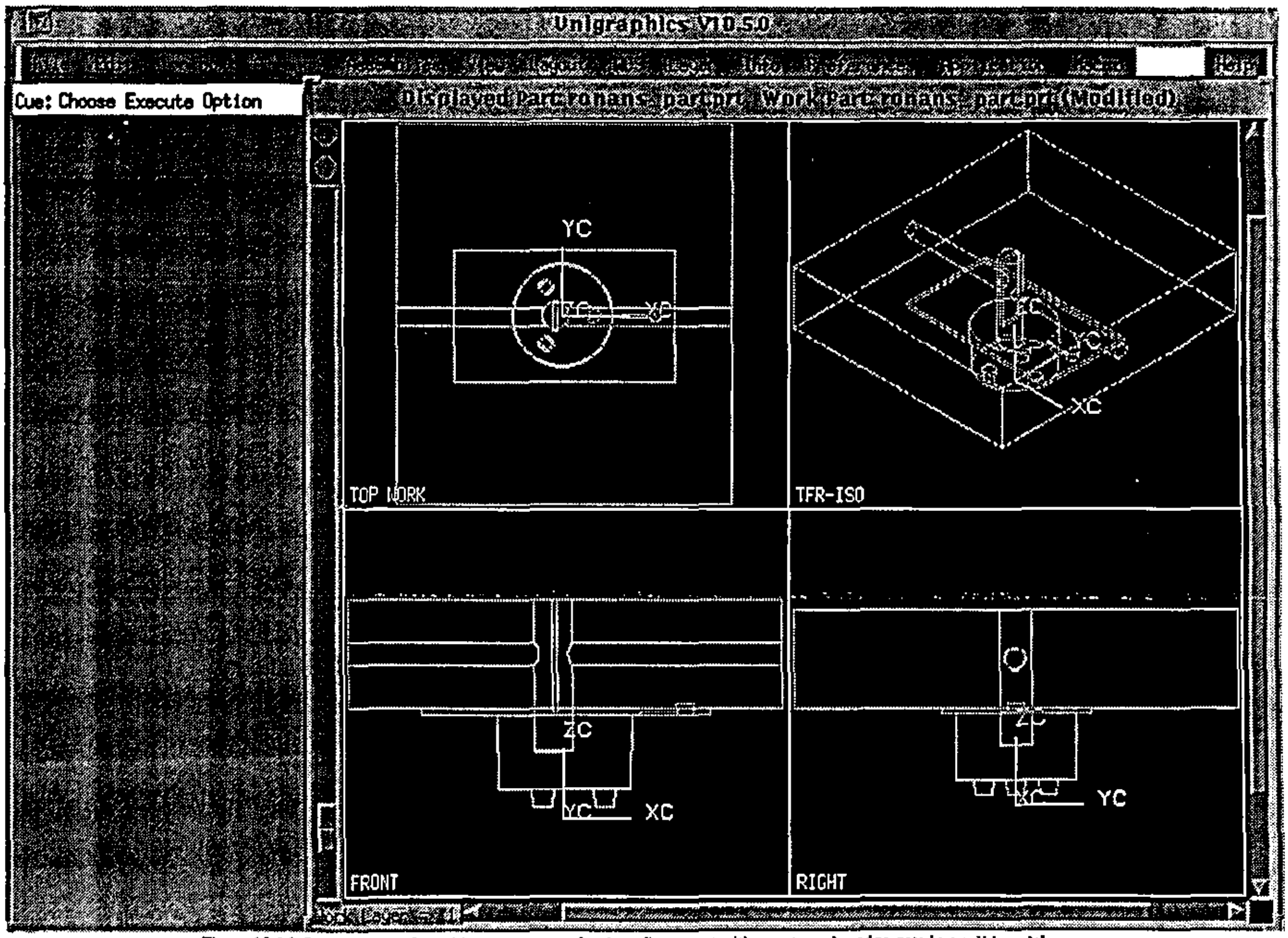


Figure 10.16. Screen dump of mould core geometry for deep flower pot with a rectangular edge gate in a solid model.

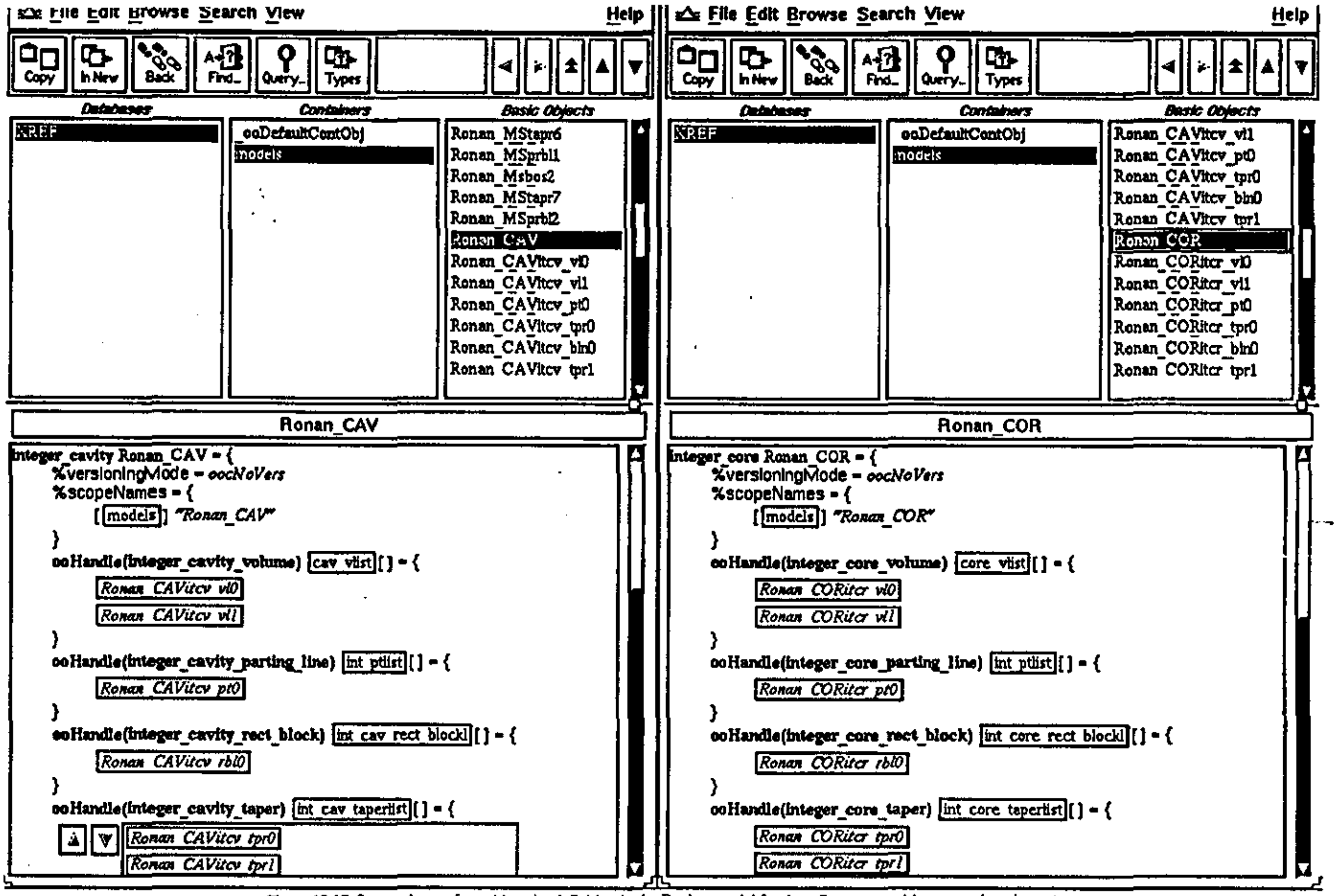


Figure 10.17. Screen dump of mould cavity definition in the Product model for deep flower pot with rectangular edge gate.

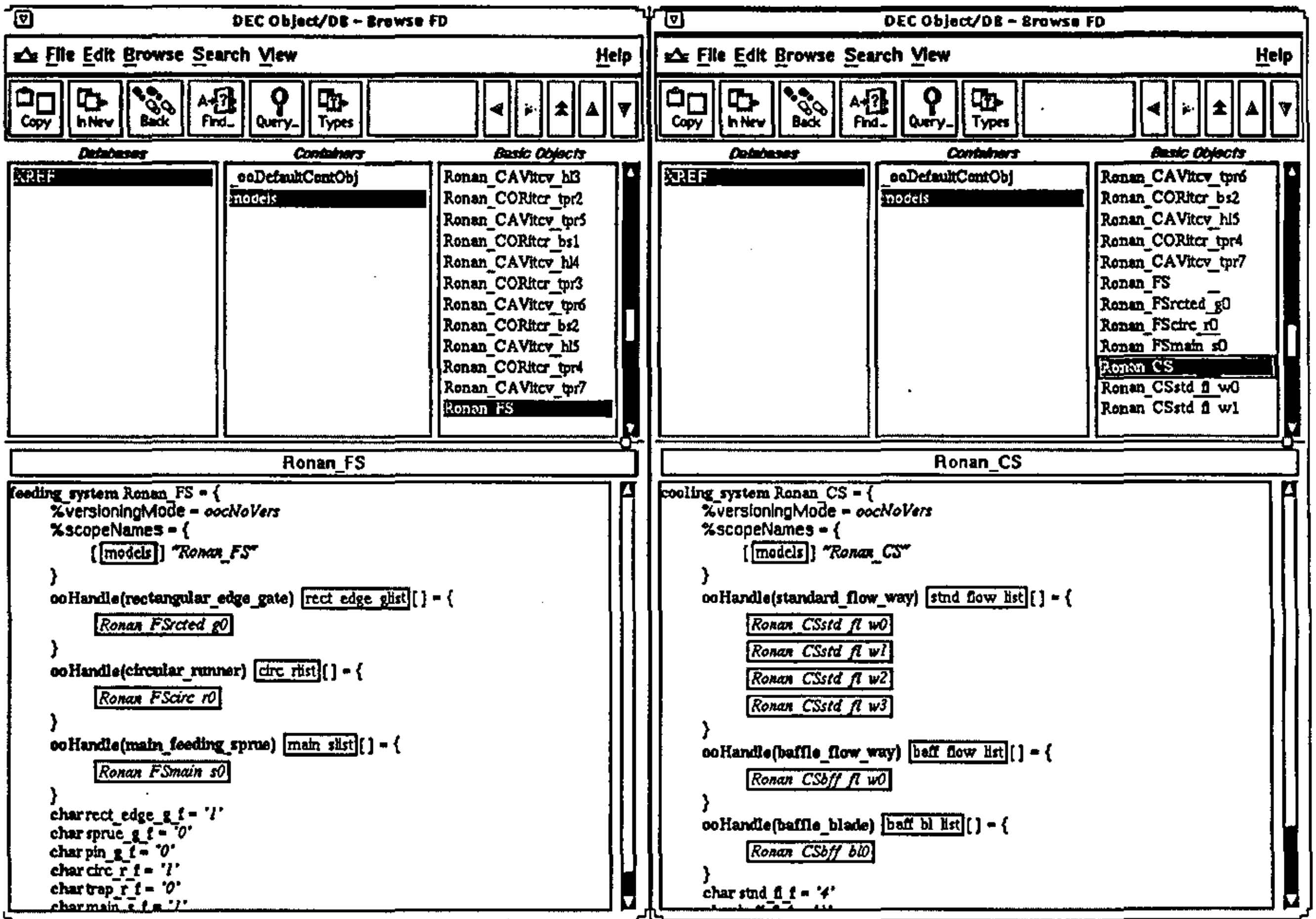


Figure 10.18. Screen dump of mould core definition in the Product model for deep flower pot with a rectangular edge gate.

tegist and the cooling system design strategist invoke data from the Manufacturing model and relate this to individual feature instantiations in the Product model.

5. The Manufacturing model provides feedback advice to the designer about mould design constraints for 3D geometry.

10.2.7. Supporting concurrent product and mould design for a PTPlus.

The experiment described in this section has been performed to demonstrate how an injection moulding strategist supports the designer in considering the interaction of constraints for the design viewpoints of the product and mould when building up a PTPlus design. The objectives are identical to those listed for the previous experiment, in order to show that a variety of different products and their moulds can be considered. An additional objective is to show how an injection moulding strategist deals with gaps in the geometry of the cavity and core.

In this experiment the designer considers the options for design of the mould during build up of a PTPlus product design. Therefore the cavity and core design strategists are invoked as well as the feeding system design strategist and the cooling system design strategist. The starting point for the product is shown in Figure 10.19. The experiment shows how the designer is supported in considering the options for mould design with respect to the PTPlus design in Figure 10.19, when the designer has opted for a rectangular edge gate. Subsequently the designer addresses another product function on the PTPlus, creating the product design shown in Figure 10.20. The designer then analyses the options for mould design with respect to a pin gated component. The entire experiment can be seen in Appendix 11.

Having built up the product geometry to be as in Figure 10.19, the designer selects the menu option for mould design, is warned that changes to mould geometry also change the product geometry, and opts to continue with the mould design.

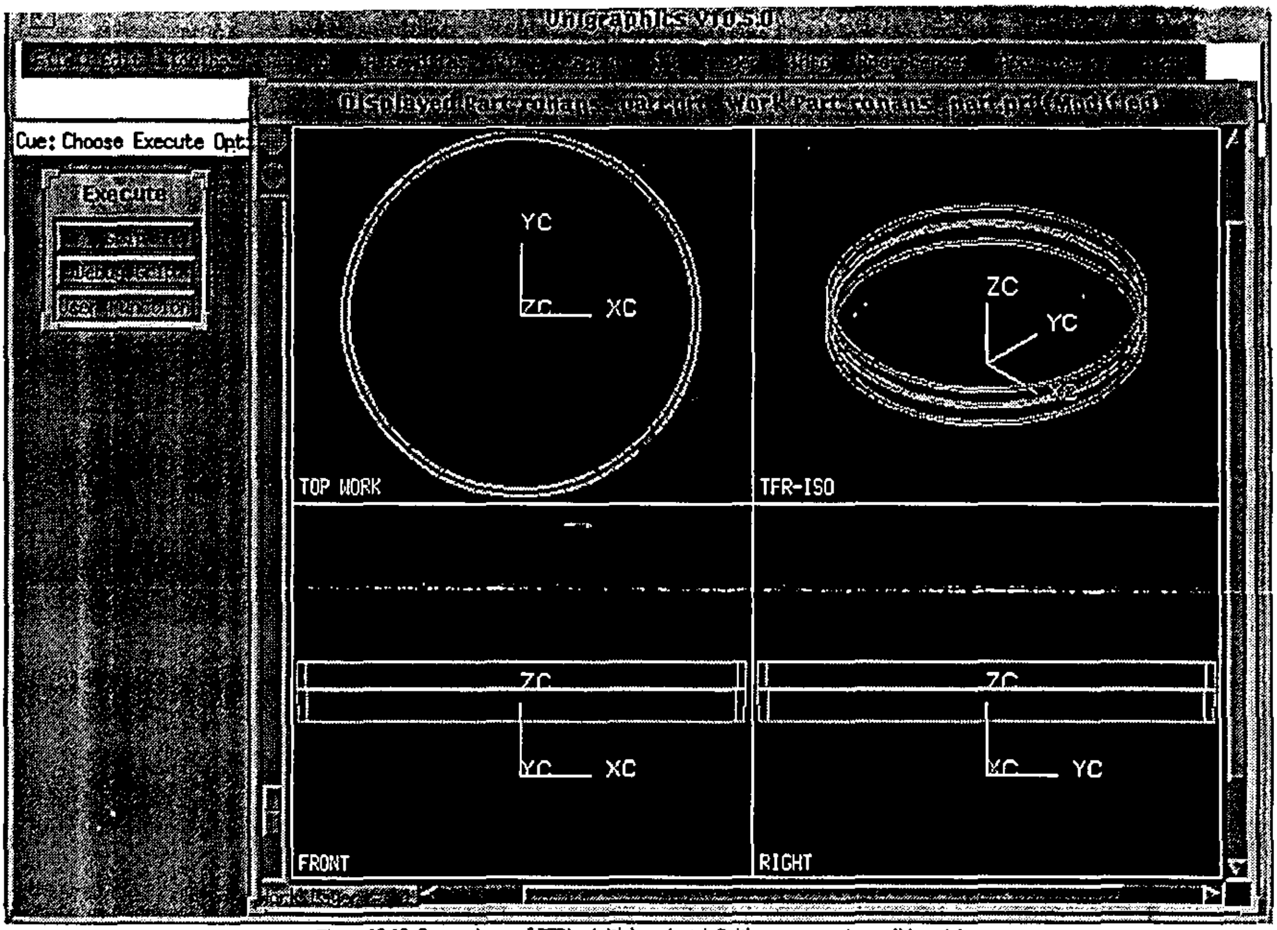


Figure 10.19. Screen dump of PTPlus initial product definition geometry in a solid model.

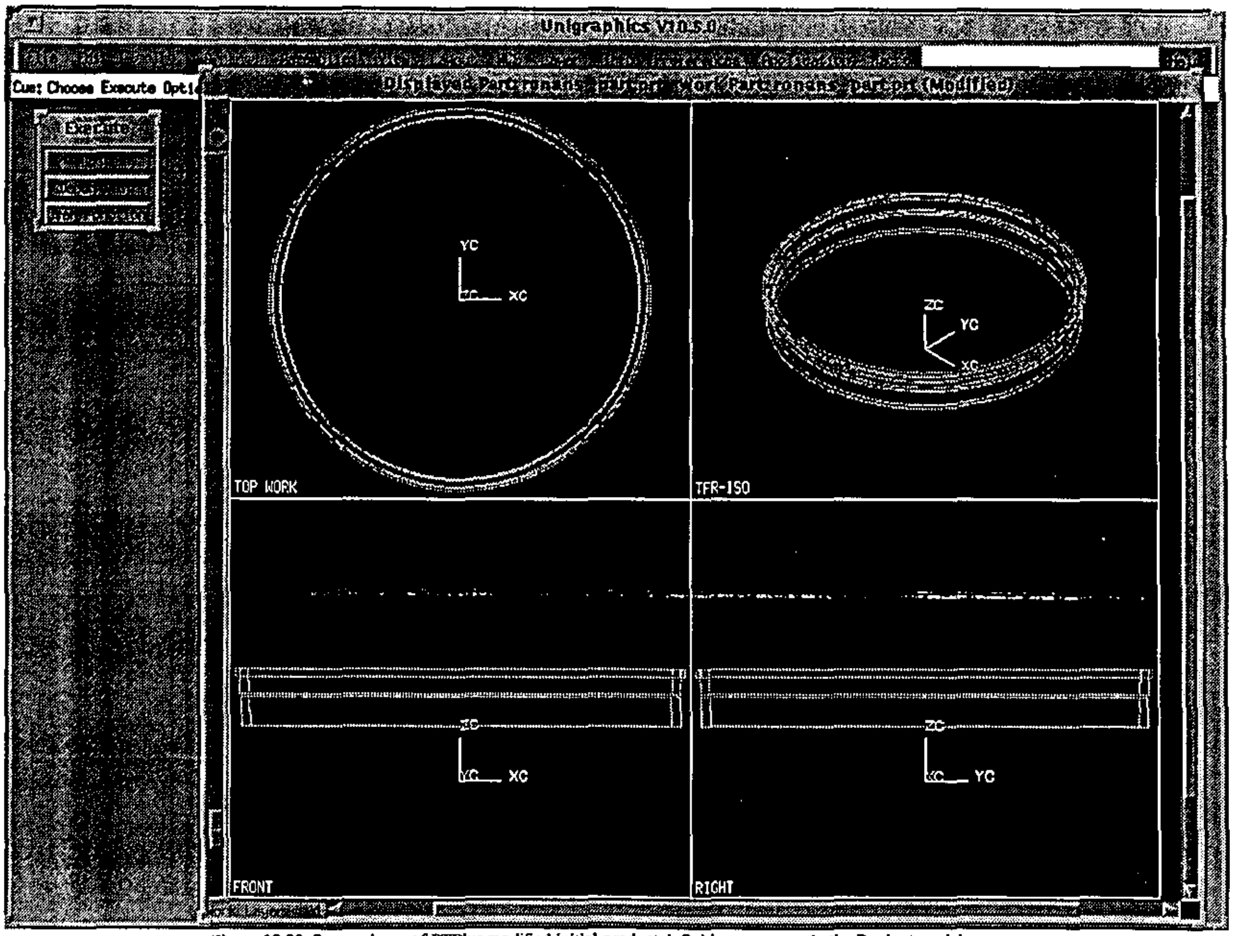


Figure 10.20. Screen dump of PTPlus modified initial product definition geometry in the Product model.

1. Go on to Interactive product Modification design phase
2. Modification/re–analysis of existing forms
3. Display options
4. Go on to mould design
5. End session4

WARNING: Dimensional changes to cavity/core elements and application of tapers during mould design will result in dimensional changes and creation of tapers on corresponding product features

As a consequence of the above, functional relationships within the product may be invalidated

Re–analysis of all functional features in the product must be undertaken in original order of creation

Are you sure you want to go on to mould design?y/ny

The first strategist triggered in the mould design cycle is the cavity design strategist, as shown in Figure 9.8. As in the previous experiment, this strategist identifies the parting line and translates from the mouldability viewpoint to the cavity, starting with the walls. The walls are translated to the cavity design viewpoint as described in section 7.5. This application strategy is shown in Figure 9.9. Having instantiated the parting line and cavity equivalent of the mouldability wall geometry, the core design strategist is invoked to do the same in the core. The walls are translated to the core design viewpoint as described in section 7.6. This application strategy is shown in Figure 9.13. The above part of the experiment can be seen in Appendix 11.

Having instantiated the main cavity and core impression geometry, it can be seen from Figure 10.19 that a gap exists between the product walls that is bridged by protrusions. In order to deal with this geometry the cavity and core design strategists instantiate an integer cavity group volume and an integer core group volume respectively, using the strategies in Figures 9.10 and 9.14. It is identified that the diameter of the group of protrusions causes an overhang in the cavity, and the designer is asked to increase the diameter of the grouping to avoid a split mould or non–mouldable component. However, when looking at the core the designer is asked for a reduction in the diameter in the core to avoid a requirement for component stripping or a collapsible core. Thus a conflict exists between the process constraints of the cavity and of the core. The designer opts for satisfying the constraints in the cavity since the consequences in the core are less severe than those in the cavity. This decision is assisted by the core feedback advice because it can be identified that the current dimensions of the product mean the component is capable of being stripped from the core.

INTEGER CAVITY GROUP VOLUME FEATURE: Ronan_CAVitcr_grv0

Cavity volume diameter is too small, an overhang exists

Possible consequences:
1. Split mould required
2. If there is a rim the component cannot be removed from the mould even
 if it is split – COMPONENT NON MOULDABLE

Remedial options:
1. Increase volume diameter to a minimum of 52.2776 mm

–No further options

Change volume diameter? y/ny

Enter new volume diameter (mm):52.28

New diameter ok

INTEGER CORE GROUP VOLUME FEATURE: Ronan_CORitcr_grv0

Core volume diameter is too large, an overhang exists

Possible consequences:
1. For an overhang of up to 1.5 mm, stripping of the component
 from the core is required for removal
2. If the overhang is larger than 1.5mm the component cannot be removed
 from the mould unless a collapsible core can be designed

Remedial options:
1. Reduce volume diameter to a minimum of 50.4399 mm

–No further options

WARNING: changing the core group volume diameter also changes the CAVITY
 group volume diameter, which can result in the need for a split cavity
Maximum diameter on core to prevent split cavity requirement :52.4399
Minimum diameter on core to prevent split cavity requirement :52.2776

Change volume diameter? y/nn

Present diameter recorded – Component stripping required

INTEGER CAVITY GROUP VOLUME FEATURE: Ronan_CAVitcr_grv0

Volume diameter ok

The geometry of the cavity and core impression is now complete. The feeding system strategist is now invoked which gives the designer a chance to change the gated wall and type of gating system. Subsequently the cooling system design strategist is invoked to support the designer in considering an aspect of the cooling system that is a prerequisite to building a feeding system design. The cooling system design strategist supports the designer in identifying the number of cooling tubes alongside the cavity as described in section 7.8., using the strategy in Figure 9.18.

Do you wish to change the gated wall? y/nn

CREATING FEEDING SYSTEM:

In order to design the feeding system it is necessary to first identify some parameters of the cooling system:

CAVITY COOLING SYSTEM: Ronan_CS

Number of cooling layers using 7 mm flow ways: 1

Number of cooling layers using 8 mm flow ways: 1

Number of cooling layers using 9 mm flow ways: 1

Number of cooling layers using 10 mm flow ways: 1

Choice of cooling tube diameter for cavity cooling system:
1. 7mm
2. 8mm
3. 9mm
4. 10mm

Lower than 7 mm – insufficient cooling effect – difficulty drilling deep holes

Higher than 10mm – high volume of water to be pumped around the mould for cooling

Maximum cooling effect for Ronan_CS cavity dimensions: 10 mm diameter

Enter choice (1–4):4

The feeding system design strategist now supports the designer in consideration of the process constraints of the feeding system. In this experiment the designer has chosen to use a rectangular edge gate and therefore a circular runner and a main feeding sprue are required. The feeding system application strategy is shown in Figure 9.17.

CREATING FEEDING SYSTEM:

FEEDING SYSTEM – RECTANGULAR EDGE GATE:Ronan_FSrcted_g0

WARNING: Application of tapers on Ronan_Mwall0 for manufacturing reasons has increased the width of the cavity opening at the parting line by 0.0837814 mm

Consequences:
1. Edge gate position no longer on edge of cavity:
– Reduced land length– weakness in mould construction can lead to wear or failure

Remedial options:
1. Adjust gate position

– No further options
Do you wish to adjust the gate position? y/ny

New gate postion: 26.2619 0 12

Gate position 26.2619 0 12 is not on the parting line

Consequences:
1. Component and feed system cannot be ejected- COMPONENT NON MOULDABLE
2. Gate and runner system cannot be machined into cavity block- MOULD NON MANUFACTURABLE

Remedial options:
1. Move gate to parting line

-No further options

Gate position has been recalculated to 26.2619 0 13
Do you wish to change the new position?y/ny

Enter new gate position (X Y Z):26.2619 0 13

New position recorded

Gate land length:

Land length should be as small as possible and in any case between 0.5 and 0.75 mm
Enter land length.5

Land length ok

Gate depth:

Gate depth has been calculated as 0.7 mm
Do you wish to change the depth?y/nn

Gate width:

Gate width has been calculated as 1.33192 mm
Do you wish to change the width?y/nn

FEEDING SYSTEM – CIRCULAR RUNNER:Ronan_FScirc_r0

Runner length:

Runner length has been calculated as 20.458 mm

This calculation is based on minimum distance between cavity and main sprue
Do you wish to change the length?y/nn

Runner diameter:

Runner diameter has been calculated as 2 mm

This calculation is based on minimum diameter to ensure cavity is filled before plastic in runner solidifies
Do you wish to change the diameter?y/nn

FEEDING SYSTEM – MAIN FEEDING SPRUE:Ronan_FSmain_s0

Main sprue position: 47.2199 0 18
Enter main sprue taper angle:
Recommended minimum angle = 4.0 degrees (minimum recommended)4.0

Taper angle ok

Lower diameter of sprue = 3.1 mm to match machine nozzle diameter of 3 mm

No nozzle recess required

Sprue length has been calculated as 47 mm

This allows the minimum cavity block depth (below cavity) to avoid mould distorsion, and allows space for the cavity cooling system.
Note: 5 mm of the sprue length is to create a sprue puller in the core block
Do you wish to change the sprue length?y/nn

Having instantiated the feeding system the cooling system design strategist supports the designer in considering the appropriate configuration of cavity cooling system, as described in section 7.8. Having identified the appropriate cooling system configuration the standard flow ways are instantiated to build up the required cooling system geometry in the cavity, in this experiment a U tube configuration is chosen requiring two parallel standard flow ways and a third perpendicular. This cooling system application strategy is shown in Figure 9.19.

CREATING CAVITY COOLING SYSTEM:

CAVITY COOLING SYSTEM:Ronan_CS

Choice of cooling system configurations for mould cavity:
1. paired tube configuration
2. U tube configuration

U tube configuration recommended when using single rectangular edge gate:

Bottom of the U cooling the gated side provides more even cooling of the moulding, can reduce cycle time.

Enter choice:2

STANDARD FLOW WAY:Ronan_CSstd_fl_w0

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_CAV

Configuration: U_tube

Vertical coordinate: -8 mm

STANDARD FLOW WAY:Ronan_CSstd_fl_w1

⋮

As in the previous experiment, not all the dimensions of the cavity cooling system can be identified and the integer cavity plate must be instantiated to enable completion of the cavity cooling system geometry. The designer is therefore supported in instantiating the cavity plate features by the cavity design strategist using the strategy in Figure 9.12. In this experiment the impression is sidegated using a rectangular edge gate. Therefore as described in section 7.5 the mould plate includes an integer cavity mould block and a wide (rectangular) inner land. It can be seen that although the guide system is not instantiated in the present work the size of the guide system is identified to determine the dimensions of the integer cavity block. As described in section 7.5 the recommended size of the guide system pins is based on the area of the cavity.

CREATING CAVITY BLOCK FOR: Ronan_CAV

INTEGER CAVITY RECTANGULAR MOULD BLOCK :Ronan_CAVitcv_rbl0

Cavity block position: 0 0 -29

Depth of cavity block: 39.6 mm

Choice of standard guide pin diameters:
1. 10 mm
2. 13 mm
3. 16 mm
4. 19 mm
5. 22 mm
6. 25 mm
7. 32 mm
8. 38 mm

Single rectangular edge gate causes unbalanced forces in the mould, tending to open the mould on one side

Possible consequences:
1. Larger wall section thickness one one side of the mould than on the other

Remedial options
Use guide pin size one larger than that recommended to ensure alignment of mould halves

-No further options

Recommended size for current mould parameters: 3. 16 mm

This product has a single rectangular edge gate – USE NEXT SIZE UP

Enter choice:(1-8):3

Guide pin diameter is 16 mm

Cavity block length: 178.44 mm

Cavity block width: 200.44 mm
Making rblck

INTEGER CAVITY RECTANGULAR LAND: Ronan_CAVitcv_rcl0

Rectangular land position: 0 0 10.6

Rectangular land depth:2.4 mm

Rectangular land length: 114.524 mm

Rectangular land width: 62.5237 mm
Making rcld

The remaining dimensions of the cavity cooling system are then identified using the strategy shown in Figure 9.20. The cooling system design strategist then looks at the cooling system in the core using the strategy in Figure 9.21. The core is identified as shallow and therefore the designer is supported in considering the appropriate configuration of shallow core cooling system. The number of cooling tubes across the core is established using the maximum cooling capability calculation (section 7.8). Subsequently, the possible configurations of shallow core cooling system are dependant upon the number of tubes that can fit across the core. In this experiment two tubes can fit across the core and the choice is either a paired tube or U tube configuration. Having identified the appropriate cooling system configuration the standard flow ways are instantiated to build up the required cooling system geometry in the core. In this experiment a U tube configuration is chosen requiring two parallel standard flow ways and a third perpendicular.

CREATING CORE COOLING SYSTEM:

CORE COOLING SYSTEM:Ronan_CS

Number of standard flow ways using 7 mm flow ways: 2

Number of standard flow ways using 8 mm flow ways: 2

Number of standard flow ways using 9 mm flow ways: 2

Number of standard flow ways using 10 mm flow ways: 2

Choice of cooling tube diameter for shallow core cooling system:
1. 7mm
2. 8mm
3. 9mm
4. 10mm

Maximum cooling effect for Ronan_CS core dimensions: 10 mm diameter

Enter choice (1–4):4

Choice of cooling system configurations for shallow mould core at diameter 10 mm:

1. paired tube configuration

2. U_tube configuration

U tube configuration recommended when using single edge gate or single pin gate:

Bottom of the U cooling the gated side provides more even cooling of the moulding, can reduce cycle time.

Enter choice:2

STANDARD FLOW WAY:Ronan_CSstd_fl_w3

Position: -89.2199 -13 34

Length: 141.44 mm

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_COR

Configuration: U_tube
Making standard flow way

STANDARD FLOW WAY:Ronan_CSstd_fl_w4

⋮

Finally the core design strategist is invoked in order to instantiate the core plate (Not shown) using the strategy in Figure 9.16. The mould cavity and core design that is generated is shown in Figures 10.21 and 10.22. After the first mould design cycle the designer opts to change the dimensions of the PTPlus design to those in Figure 10.20 by addressing the hold in lid product function. The support for addressing of this product function has been shown in the experiment in section 10.3.2, and is shown as part of the current experiment in Appendix 11. The designer then opts to look at the mould design again. Design support for the instantiation of the cavity and core impression geometry in the second mould design cycle is exactly the same as above, except obviously for the dimensions of the impression and the extra wall on the product. Design support is provided using the cavity design strategies in Figures 9.9 and 9.10, and the core design strategies in Figures 9.13 and 9.14. Having instantiated the geometry of the cavity and core impression the feeding system strategist is now invoked which gives the designer a chance to change the gated wall and type of gating system. This time the designer opts to change from a rectangular edge gate to a pin gate on a different wall. The designer is warned that a pin gate requires a three plate mould but still chooses this gate type.

Do you wish to change the gated wall? y/ny

Please indicate which wall you wish to gate:
0. Ronan_Mwall1
1. Ronan_Mwall0
2. Ronan_Mwall2

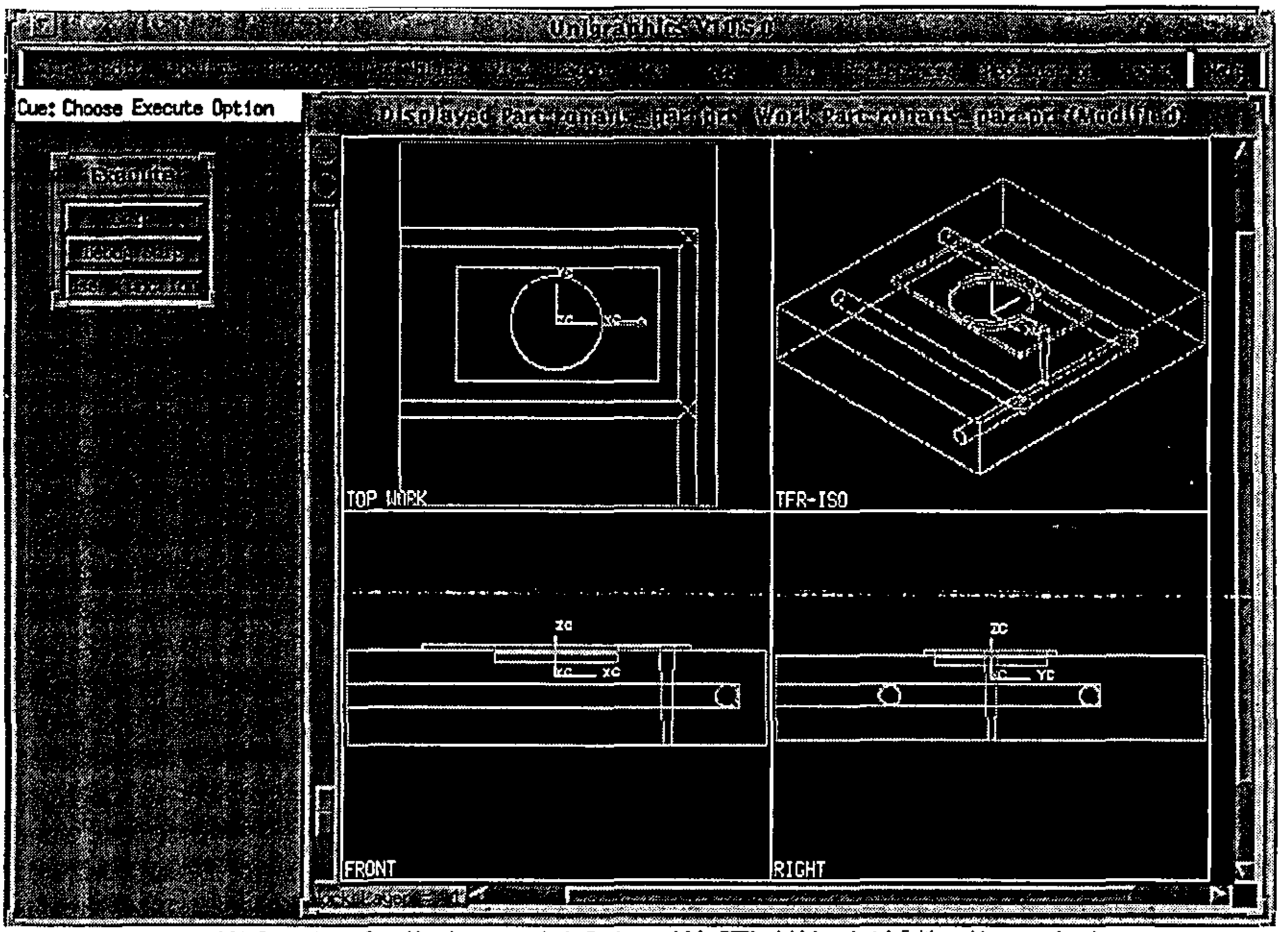


Figure 10.21. Screen dump of mould cavity geometry in the Product model for PTPlus initial product definition with rectangular edge gate.

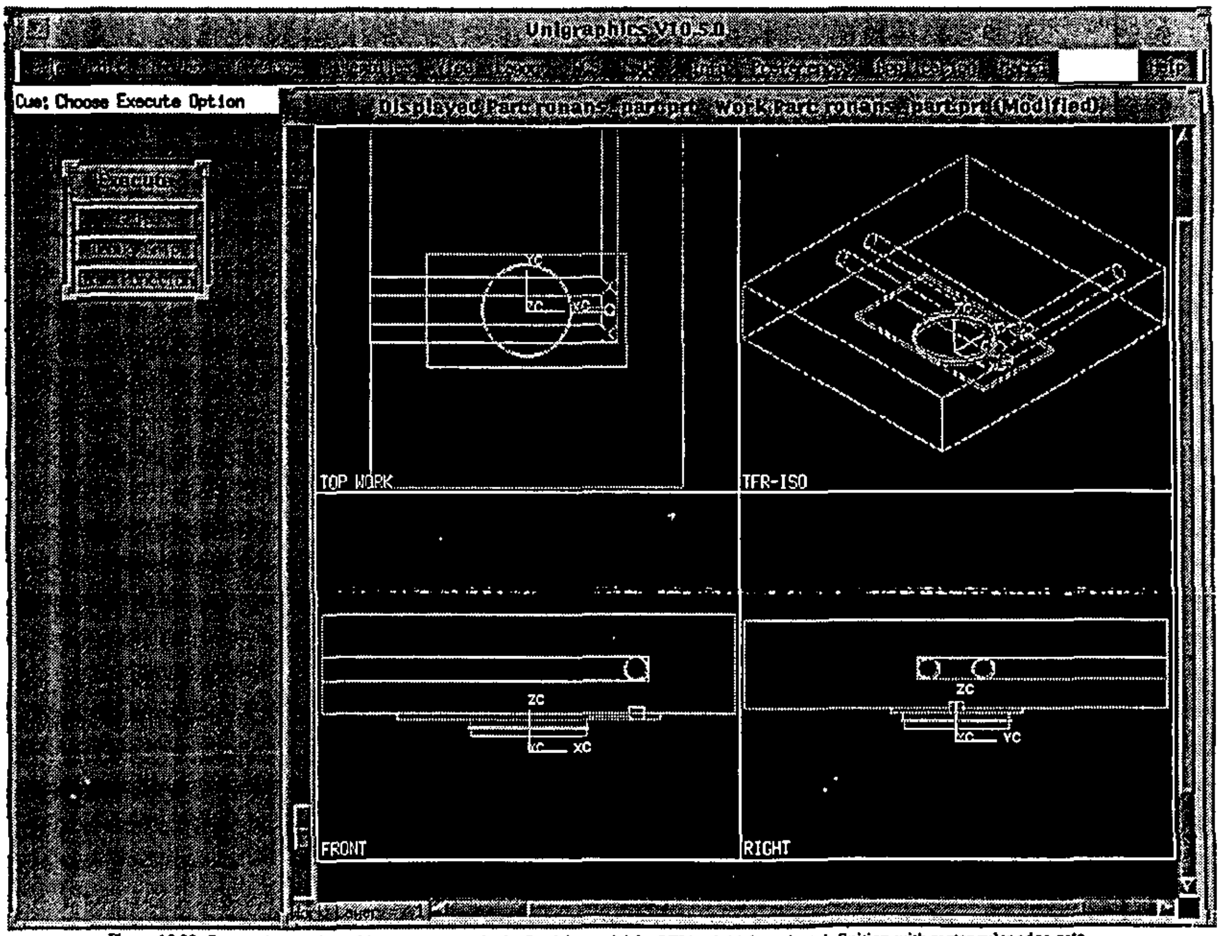


Figure 10.22. Screen dump of mould core geometry in the Product model for PTPlus initial product definition with rectangular edge gate.

Enter choice:0

Enter gate position X Y Z25.5 0 6.25

Feeding distance ok

This product is tubular:

Possible choices of gate type:
1. Rectangular edge gate
2. Pin gate
3. Diaphram gate
4. Ring gate

Enter choice:2

Gate type is pin gate

WARNING: Use of a pin gate requires a three plate mould.
Other possible gate types only require two plate moulds
Do you still want to specify a pin gate type?y/ny

Gate type is pin gate

–Three plate mould required

Afterwards as in the previous mould design cycle, the cooling system design strategist is invoked to support the designer in considering an aspect of the cooling system that is a pre-requisite to building a feeding system design. The cooling system design strategist supports the designer in identifying the number of cooling tubes alongside the cavity as described in section 7.8, using the strategy in Figure 9.18. The feeding system design strategist then supports the designer in consideration of the process constraints of the feeding system. In this experiment the designer has chosen a pin gate and therefore a trapezoidal runner system and a main feeding sprue is required. The feeding system application strategy is shown in Figure 9.17.

CREATING FEEDING SYSTEM:

In order to design the feeding system it is necessary to first identify some parameters of the cooling system:

CAVITY COOLING SYSTEM: Ronan_CS

Number of cooling layers using 7 mm flow ways: 1

:
:
:

Maximum cooling effect for Ronan_CS cavity dimensions: 10 mm diameter

Enter choice (1–4):4

CREATING FEEDING SYSTEM:

FEEDING SYSTEM – PIN GATE:Ronan_FSpin_g0

WARNING: Application of tapers on Ronan_Mwall1 for manufacturing reasons has decreased the base diameter of the cavity by 0.0279271 mm

Consequences:
1. Pin gate position no longer same distance from edge of section:

Remedial options:
1. Adjust gate position

– No further options
Do you wish to adjust the gate position? y/ny

New gate postion: 25.486 0 6.25

Gate position ok

Gate land length:

Land length should be as small as possible and in any case between 0.5 and 0.75 mm
Enter land length.5

Land length ok

Gate diameter:

Gate diameter has been calculated as 1.22121 mm
Do you wish to change the diameter?y/nn

Enter gate taper angle:
Recommended minimum angle = 4.0 degrees (minimum recommended)4.0

Taper angle ok
Min under:18

Secondary sprue length has been calculated as 34.75 mm

This allows the minimum cavity block depth (below cavity) to avoid mould distorsion, and allows space for the cavity cooling system
Do you wish to change the secondary sprue length?y/nn

FEEDING SYSTEM – TRAPESOIDAL RUNNER:Ronan_FStrap_r0

Runner length:

Runner length has been calculated as 45.486 mm

This calculation is based on minimum distance between pin gate secondary sprue and a central main sprue
Do you wish to change the length?y/nn

Secondary sprue diameter at junction with runner: 3.64821

Runner width:

Runner width has been calculated as 3.64821 mm

This calculation is based on the diameter of the secondary sprue where it joins the runner. Runner width should be at least as large as the sprue diameter up to a maximum of 10mm
Do you wish to change the width?y/nn

FEEDING SYSTEM – MAIN FEEDING SPRUE:Ronan_FSmain_s0

Main sprue position: 0 0 –32.6482
Enter main sprue taper angle:
Recommended minimum angle = 4.0 degrees (minimum recommended)4.0

Taper angle ok

Lower diameter of sprue = 3.1 mm to match machine nozzle diameter of 3 mm

Sprue length has been calculated as 18 mm

This allows the minimum backing plate depth to avoid mould distorsion
Do you wish to change the sprue length?y/nn

Having instantiated the feeding system, the cooling system design strategist supports the designer in considering the appropriate configuration of cavity cooling system, as described in section 7.8. Having identified the appropriate cooling system configuration the standard flow ways are instantiated to build up the required cooling system geometry in the cavity. In the second mould design cycle a U tube configuration is chosen requiring two parallel standard flow ways and a third perpendicular. The cooling system application strategy is shown in Figure 9.19.

CAVITY COOLING SYSTEM:Ronan_CS

Choice of cooling system configurations for mould cavity:
1. paired tube configuration
2. U tube configuration

U tube configuration recommended when using single pin gate:

Bottom of the U cooling the gated side provides more even cooling of the moulding, can reduce cycle time.

Enter choice:2

STANDARD FLOW WAY:Ronan_CSstd_fl_w0

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_CAV

Configuration: U_tube

Vertical coordinate: –8 mm
Making standard flow way

STANDARD FLOW WAY:Ronan_CSstd_fl_w1

⋮

As in the first mould design cycle, not all the dimensions of the cavity cooling system can be identified and the integer cavity plate must be instantiated to enable completion of the cavity cooling system geometry. The designer is therefore supported in instantiating the cavity plate features by the cavity design strategist using the strategy in Figure 9.12. In this experiment the impression is under gated using a pin gating system. Therefore as described in section 7.5 the mould plate includes an integer cavity mould block, and a narrow (circular) inner land, four peripheral lands and a backing plate to make a three plate mould. As before, although the guide system is not instantiated in the present work the size of the guide system is identified to determine the dimensions of the integer cavity block.

CREATING CAVITY BLOCK FOR: Ronan_CAV

INTEGER CAVITY RECTANGULAR MOULD BLOCK :Ronan_CAVitcv_rbl0

Cavity block position: 0 0 -29

Depth of cavity block: 39.6 mm

Choice of standard guide pin diameters:
1. 10 mm
2. 13 mm
3. 16 mm
4. 19 mm
5. 22 mm
6. 25 mm
7. 32 mm
8. 38 mm

Recommendation: Use smallest suitable guide pin diameter to minimise size and weight of mould assembly

Recommended size for current mould parameters: 2. 13 mm

Enter choice:(1-8):2

Guide pin diameter is 13 mm

Cavity block length: 142.54 mm

Cavity block width: 189.44 mm

INTEGER CAVITY CIRCULAR LAND:Ronan_CAVitcv_crl0

Circular land position: 0 0 10.6

Circular land depth:2.4 mm

Circular land diameter:63.4678 mm

INTEGER CAVITY PERIPHERAL LAND:Ronan_CAVitcv_pf0

Peripheral land position: 45.2698 45.2698 10.6

Peripheral land depth:2.4 mm

Peripheral land diameter:13.5499 mm

⋮

INTEGER CAVITY PERIPHERAL LAND:Ronan_CAVitcv_pf3

Peripheral land position: -45.2698 -45.2698 10.6

Peripheral land depth:2.4 mm

Peripheral land diameter:13.5499 mm

INTEGER CAVITY BACKING PLATE:Ronan_CAVitcv_bk0

Backing plate position: 0 0 -50.6482

Backing plate width: 189.44 mm

Backing plate length: 142.54 mm

Backing plate depth: 21.6482 mm

The remaining dimensions of the cavity cooling system are then identified as shown in Figure 9.20. The cooling system design strategist then looks at the cooling system in the core using the strategy in Figure 9.21. The core is again identified as shallow and therefore the designer is supported in considering the appropriate shallow core cooling system. The number of cooling tubes across the core is established using the maximum cooling capability calculation (section 7.8). Subsequently, the possible configurations of shallow core cooling system are dependant upon the number of tubes that can fit across the core. In this experiment two tubes can fit across the core and the choice is either a paired tube or a U tube configuration. Having identified the appropriate cooling system configuration and the maximum cooling capacity the standard flow ways are instantiated to build up the required cooling system geometry in the core. In this experiment a U tube configuration is chosen requiring two parallel standard flow ways and a third perpendicular.

CREATING CORE COOLING SYSTEM:

CORE COOLING SYSTEM:Ronan_CS

Number of standard flow ways using 7 mm flow ways: 2

Number of standard flow ways using 8 mm flow ways: 2

Number of standard flow ways using 9 mm flow ways: 2

Number of standard flow ways using 10 mm flow ways: 2

Choice of cooling tube diameter for shallow core cooling system:
1. 7mm

2. 8mm
3. 9mm
4. 10mm

Maximum cooling effect for Ronan_CS core dimensions: 10 mm diameter

Enter choice (1–4):4

Choice of cooling system configurations for shallow mould core at diameter 10 mm:

1. paired tube configuration

2. U_tube configuration

U tube configuration recommended when using single edge gate or single pin gate:

Bottom of the U cooling the gated side provides more even cooling of the moulding, can reduce cycle time.

Enter choice:2

STANDARD FLOW WAY:Ronan_CSstd_fl_w3

Position: -71.2698 -13 34

Length: 123.99 mm

Diameter: 10 mm

Orientation: 0

Cavity/core_name: Ronan_COR

Configuration: U_tube
Making standard flow way

STANDARD FLOW WAY:Ronan_CSstd_fl_w4

⋮

Finally the core design strategist is invoked in order to instantiate the core plate (Not shown) using the strategy in Figure 9.16. The mould cavity and core design that is generated from the second mould design cycle is significantly different than that from the first, as shown in Figures 10.23 and 10.24. The results of the above experiment in the Product model are shown in Figures 10.25 and 10.26. Figure 10.25 shows the three integer cavity volumes and three integer core volumes which are the cavity and core equivalent of the mouldability walls in the PTPlus. Also the circular land can be seen. Figure 10.26 shows the feeding system and the cooling system after the second design cycle, showing a pin gate, a trapezoidal runner and a main feeding sprue in the feeding system, three standard flows ways for a U tube in the cavity cooling system, and three standard flow ways for a U tube shallow cooling system in the core. The above experiment can be seen in its entirety in Appendix 11.

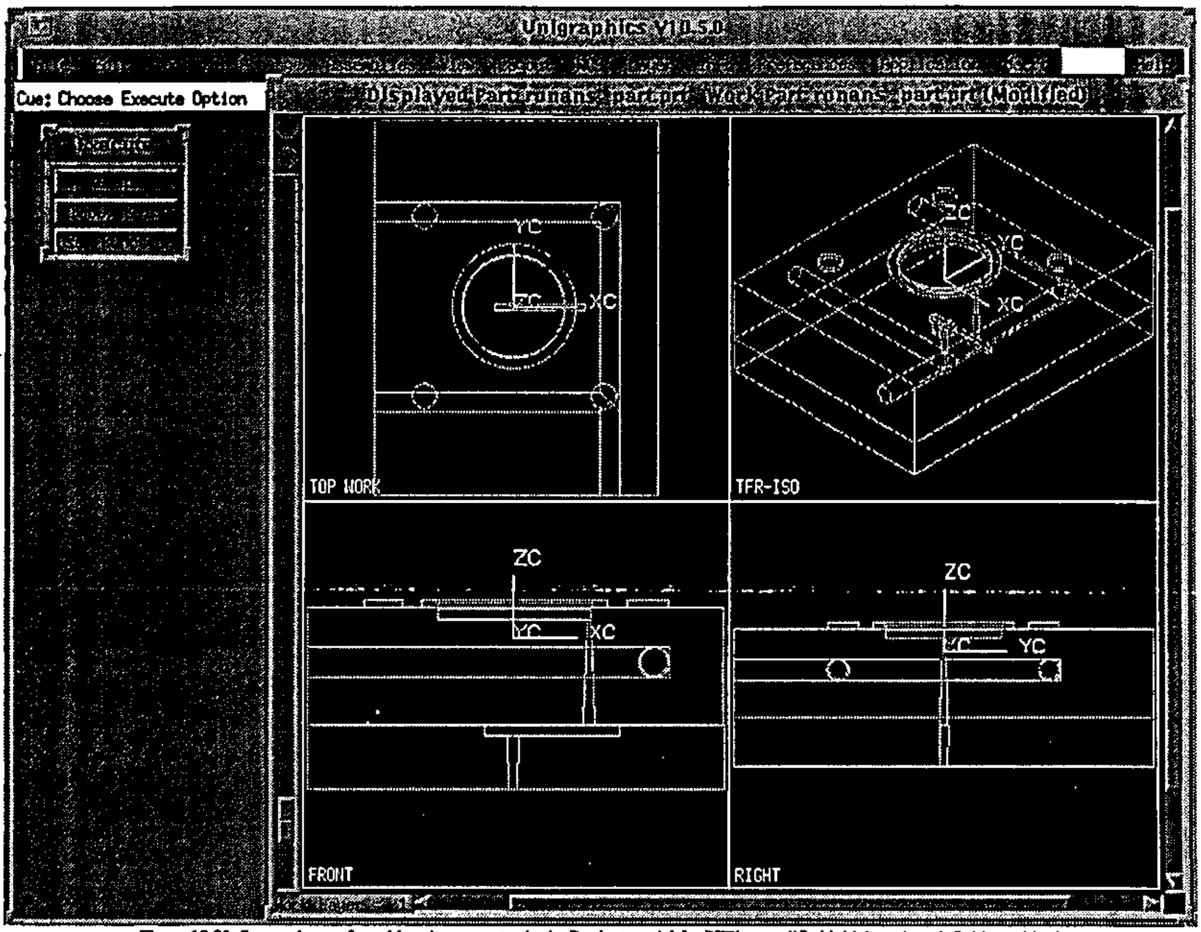


Figure 10.23. Screen dump of mould cavity geometry in the Product model for PTPlus modified initial product definition with pin gate.

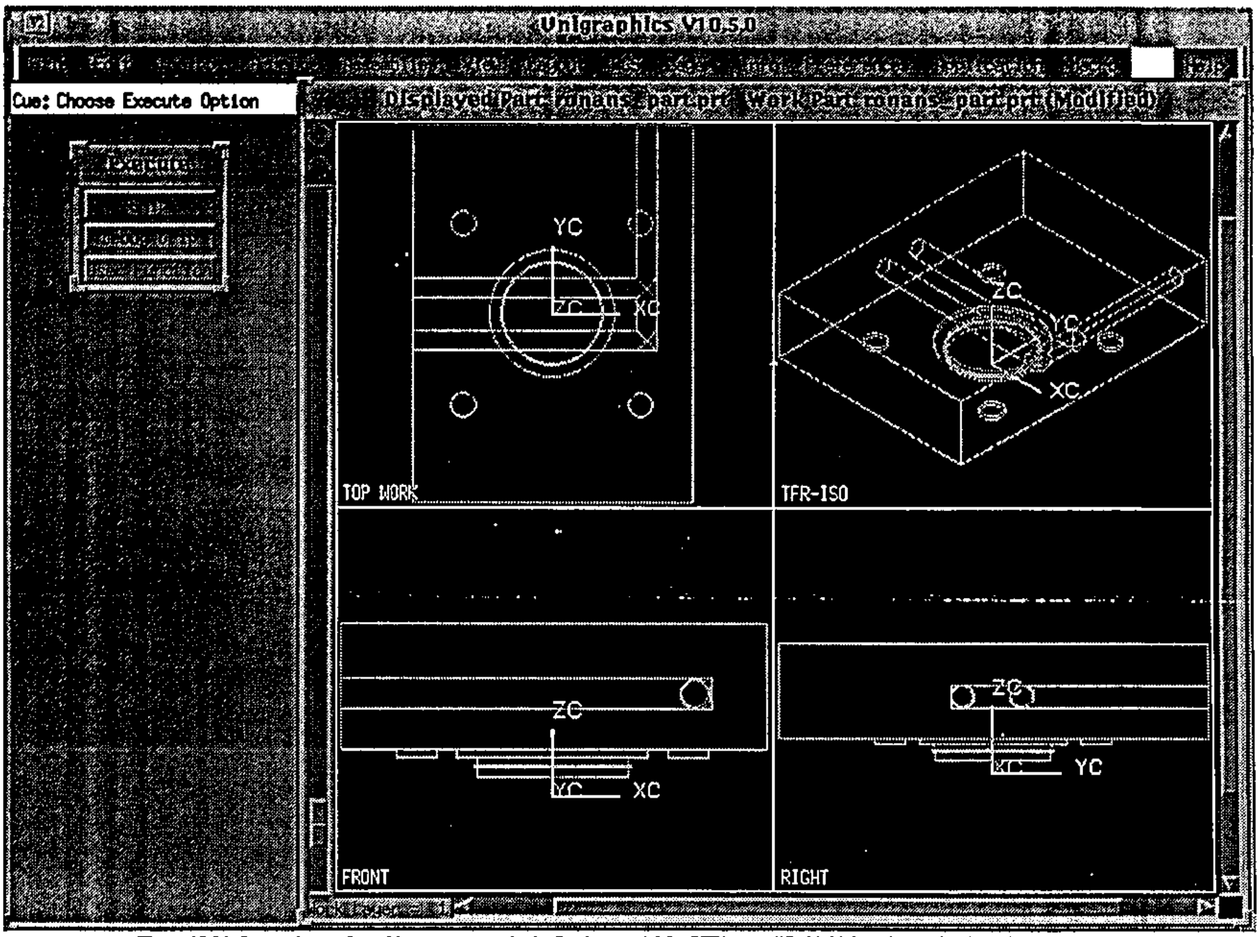


Figure 10.24. Screen dump of mould core geometry in the Product model for PTPlus modified initial product definition with pin gate.

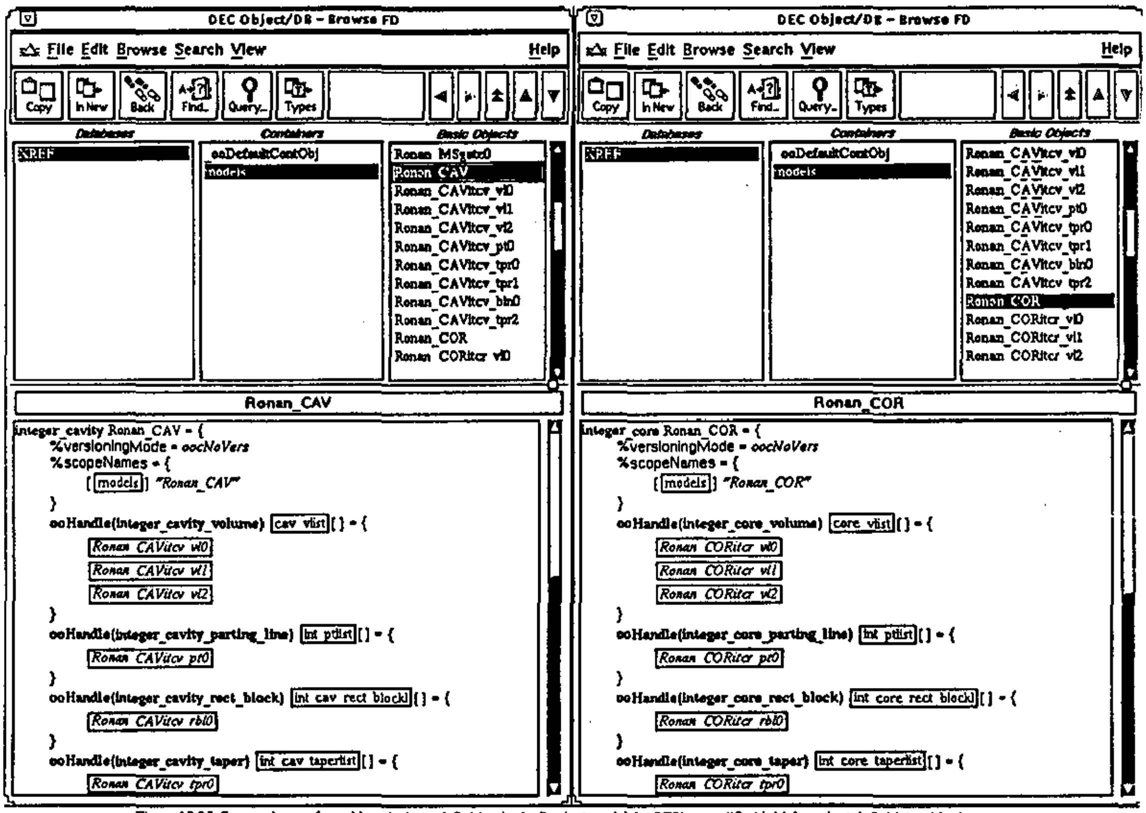


Figure 10.25. Screen dump of mould cavity/core definition in the Product model for PTPlus modified initial product definition with pin gate.

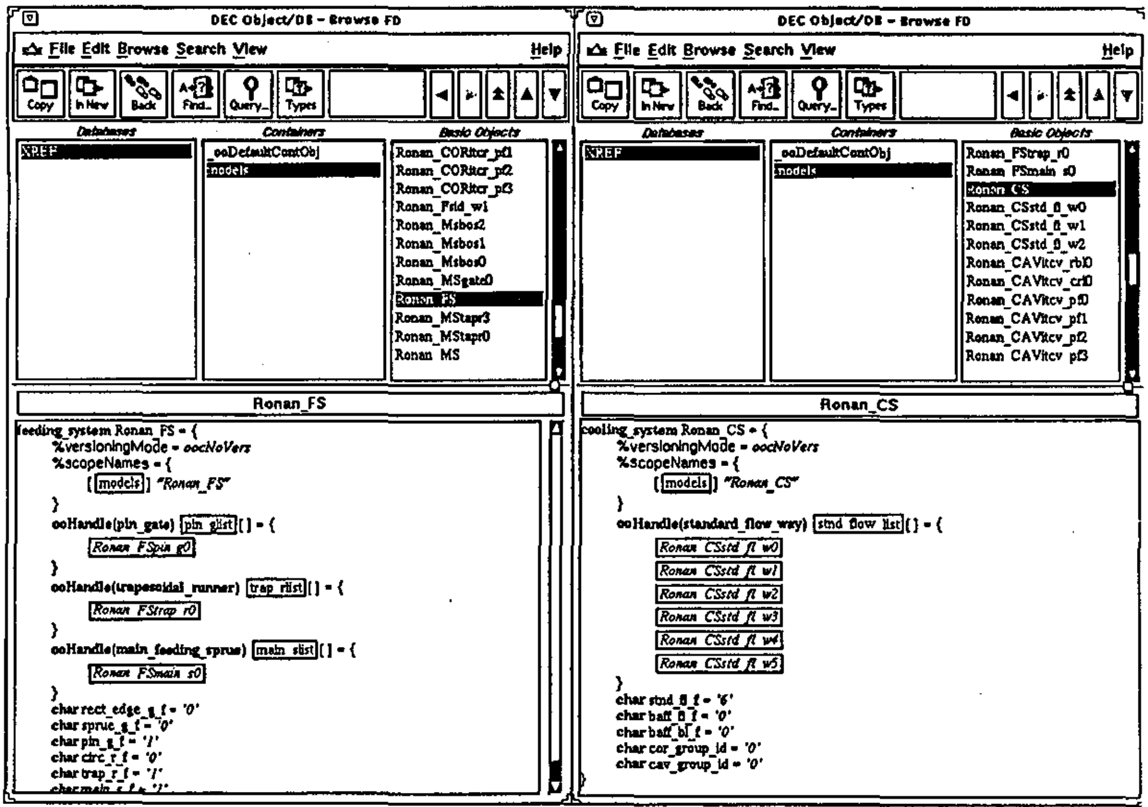


Figure 10.26. Screen dump of mould feeding/cooling definition in the Product model or PTPlus modified initial product definition with pin gate.

The above experiment has shown:

1. That an injection moulding strategist of the structure described in Chapter 9 can support the designer in the concurrent design of an injection moulded product and its mould.

2. That a link is established via a set of translation processes between the geometry of the product and that of the mould, such that changes in product geometry result in changes in the geometry of the mould cavity/core and mould system elements.

3. An injection moulding strategist can support the designer in considering the process constraints with respect to gaps in the geometry of the cavity/core.

4. A translation process allows the interaction of design viewpoints within the product and the mould.

5. The cavity design strategist, the core design strategist, the feeding system design strategist and the cooling system design strategist invoke data from the Manufacturing model and relate this to individual feature instantiations in the Product model.

6. The Manufacturing model provides feedback advice to the designer about mould design constraints for 3D geometry.

10.3. Discussion of experimental results.

The above experiments have shown that an injection moulding strategist can be structured within an information modelling environment to provide concurrent support for the design of injection moulded products and their moulds. The experimental software system can support the design of a range of rotational products and their moulds. It has been shown that a menu drives the experimental software system which provides interactive sessions to support the designer in the instantiation of feature types to define the product and mould from multiple viewpoints in a Product model.

The above experiments show that an injection moulding strategist application is capable of supporting the designer in considering the interaction of functional and manufacturing constraints during the progression of a product design. An injection moulding strategist is also shown to be capable of supporting the designer in the consideration and re–consideration of possible options for mould design at any juncture in the development of the product geometry. The translation processes defined in Chapter 7 between viewpoints in injection moulding design have been shown to enable interactions between applications within the injection moulding strategist so that functional constraints data in the Product Range Model and injection moulding process constraints data in the Manufacturing model can be used to provide feedback advice to the designer from multiple viewpoints in product and mould design.

The experiments have shown that the Manufacturing model is capable of capturing the process constraints of a set of mouldability features in a form that can be related to individual feature instantiations in a Product model. The Manufacturing model has also been shown to be capable of capturing the process constraints of a set of cavity and core features in a form that can be related to individual feature instantiations in a Product model, enabling a link between the geometry of a product and that of a mould. Finally the Manufacturing model has been shown to be capable of capturing the process constraints of sets of features representing the mould system elements. This data is also captured in a form that can be related to individual feature instantiations in a Product model.

The experiments one to three showed that the Product Range Model is capable of capturing data to support the designer in the association of function and form in the Product model and can capture functional constraints of product ranges in a form that can be related to individual feature instantiations in the Product model.

The first three experiments showed that the injection moulding strategist supports the instantiation of functional and manufacturing features in the Product model as the product design progresses. In these experiments the designer is supported in the association of functional specification data with a form feature in the Product model as a product range function is addressed. After each new instantiation or a modification to the product geometry, functional feedback advice is provided to the designer using data in the Product Range Model. A translation process to the mouldability viewpoint instantiates the equivalent mouldability feature(s) or updates the equivalent mouldability feature(s) in response to a geometry modification, providing mouldability feedback advice using data from the Manufacturing model.

Experiments four and five showed that the injection moulding strategist supports the designer in considering the design of a mould at any juncture in the development of the product geometry. In experiments four and five the designer is supported in consideration of process constraints during the build up of the cavity/core impression geometry, of the feeding system and cooling system and of the cavity and core plate. A translation process enables the designer to drive the mould design process and receive feedback advice on the process constraints of the viewpoints within mould design using data from the Manufacturing model. After mould design the designer is able to continue developing the product or re–analyse the mould design, possibly progressing to a completely different mould configuration in the Product model.

The use of geometry analysis techniques such as mold–flow has not been explored in this work, as the software cannot be linked to the Object DB software. The role of such analysis techniques within the functionality of the above experimental system, for example with respect to the analysis of feeding or cooling in a mould requires investigation.

The above experimentation has shown that an injection moulding strategist application can support the designer in consideration of three dimensional product and mould geometry from multiple viewpoints in injection moulding design. However the use of complex geometry, ie non–rotational, non–symmetrical etc requires further investigation. This can have implications for Manufacturing model support for the consideration of injection moulding process constraints, for Product Range Model support for the association of function and form, and for the enabling of interactions between strategist applications. However the author believes that the general structure and functionality of an injection moulding strategist application is still applicable to support the designer in concurrent design of an injection moulded product and its mould using complex geometry, that the process constraints captured in the Manufacturing model are still applicable to the complex geometry of a product and a mould, and that the Product Range Model structure is still applicable for the support of functional data and complex forms.

Software support for the design of multi–cavity moulds has not been considered in this work. The extensibility of the approach in this work for consideration of complex production moulds and any additional requirements for design support needs to be investigated. Also the potential effect of different injection machine tooling has not been considered.

The above experiments have demonstrated that an injection moulding strategist application can be structured in an information modelling environment to provide concurrent support for the design of injection moulded products and their moulds, a Manufacturing model can be built in software form to provide a common source of information for a range of interacting strategist applications, supporting a link between the geometry of the product and that of the mould, and functional data can be captured in a Product Range Model to support the association of function and form. The software has provided an adequate platform to prove the ideas put forward in this research. However for a practical system to be developed the issues identified above must be explored.

Chapter 11

Conclusions and recommendations for further work.

11.1. INTRODUCTION.

The research described in this thesis has explored the functionality and structure of an injection moulding strategist to support concurrent product and mould design in an information modelling environment. The information support requirements of such an application have also been explored. An experimental design support system has been implemented to test the research thesis. The experiments which have been performed to explore the ideas developed in the thesis have led to the conclusions and recommendations for further work which are made in this chapter.

11.2. Conclusions.

1 – The functionality of an injection moulding strategist application has been captured in an IDEF0 activity model, based on a detailed analysis of injection moulding design methods to ascertain the scope for concurrency and the key activities and their relationships. The information on design methods was obtained from discussions with people in industry about their design methods for injection moulding and by the use of literature about the methodology and considerations in injection moulded product and mould design (Appendix 1 and 2).

2 –Multiple features sets have been defined within information models to provide appropriate information input to multiple design for manufacture applications. An independent set of 'enhanced' form features has been defined which captures inter–form relationships, allowing form to be the central attribute through which all viewpoints interact.

3 – It has been shown possible to achieve the necessary data translations between feature types to provide input to the range of applications involved in design for injection moulding support.

4 – It has been shown that the structure of a design support system for injection moulded products, identified in this research as necessary for concurrent support of design for injection moulding, can be captured in an experimental software system, termed an injection moulding strategist, which supports concurrent design of injection moulded products and their moulds. The applications within the injection moulding strategist are:

a – A design for mouldability application which supports the designer in building a product definition with sound mouldability.

b – A cavity design application which supports the designer in building a cavity definition which is capable of producing the outside shape of a product with sound mouldability, and allowing its ejection. The designer is supported in building a cavity definition that has sufficient strength to withstand the forces of the injection process, is capable of accommodating the mould system elements and does not have non–producable geometry.

c – A core design application which supports the designer in building a core definition which is capable of producing the inside shape of a product with sound mouldability, and allowing its ejection. The designer is supported in building a core definition that is capable of accommodating the mould system elements and does not have non–producable geometry.

d – A feeding system design application which supports the designer by providing the available choices of feeding system configuration at each juncture in product design and in building a feeding system definition that provides adequate feeding to the impression and does not have non manufacturable geometry.

e – A cooling system design application which supports the designer by providing the available choices of cooling system configuration at each juncture in product design and in building a cooling system definition that provides maximum cooling capability and optimum cooling characteristics.

5 – The specification and structure of the injection moulding process capabilities has been defined in Booch and EXPRESS language in a form suitable to support manipulation of spatial relationships in design for injection moulding.

6 – It has been shown that the capabilities of the injection moulding process can be represented in a software model in a form suitable to support manipulation of spatial relationships in design for injection moulding. This Manufacturing model has captured the representation of mouldability features and mould system elements.

7 – It has been shown that the Manufacturing model can support the injection moulding strategist in the manipulation of spatial relationships in design for injection moulding.

8 – The specification and structure of functional features relating to product ranges has been defined in Booch and the EXPRESS language. The representation, termed the Product Range Model, has been shown to be suitable to support the designer in the linking of function and form.

9 – It has been shown that the Product Range Model can provide the source of information to enable a design support application to support the designer in the association of function and form.

10 – The appropriate structures in the Product model to facilitate analysis and re–analysis of the current state of a product and a mould for multiple design viewpoints have been defined.

11 – It has been shown that the Product model can facilitate analysis and re–analysis of the current state of a product and a mould for multiple design viewpoints by an injection moulding strategist application.

12 – The Object DB software has been shown to be an adequate tool for the implementation of a data driven concurrent design support system. There is a minor limitation that the software does not support multiple inheritance, so that exact replication of Booch is not always possible. The Object DB software is based on C++ which enabled the interfacing of the Product model to the Unigraphics solid model package. This has provided an adequate platform for modelling and displaying the contents of the Product model.

11.3. Recommendations for further work.

1 – There is a requirement to investigate software support for the consideration of more complex geometry than investigated in this thesis, eg non–rotational, complex surfaces.

2 – Investigating the above area also requires investigation into the changes that may be required to the translation mechanisms identified in chapter 7.

3 – Investigation is required into the extensibility of the Product Range Model concept for products which have more complex geometry than investigated in this thesis.

4 – There is a requirement to investigate software support for consideration of insert moulds and of the machine elements.

5 – The production of an injection moulding of the required quality is dependant upon the production of the injection mould. Therefore investigation of support tools for consideration of mould manufacturing processes such as machining is required, and investigation of their role in the support of concurrent product and mould design.

6 – As in the work described in this thesis, investigation of design support tools in the above areas also requires investigation of the information support requirements, ie the data structures in a Manufacturing model.

7 – To provide industrially useful design support tools for the future, there is a requirement to explore the implementation of software support tools, such as the injection moulding strategist described in this thesis, inside a commercial CAD system such as Unigraphics, in order to provide a concurrent design support tool as an integral part of a CAD system.

References.

Allada V., Anand S. 1995. "Feature-based modelling approaches for integrated manufacturing: state-of-the-art survey and future research directions". International Journal of Computer Integrated Manufacturing, v 8, No 6, 1995, pp411–440.

Al-Ashaab A.H., Young R.I.M. 1992. "Information models: An aid to achieve concurrency in injection moulded products design", Winter Annual meeting ASME, California, USA, November 8–13, 1992, pp1–12.

Al-Ashaab A.H., Young R.I.M. 1994. "A Manufacturing Model: Supporting Concurrency in Injection Moulded Products Design. ASME", ASME 2nd Biennial European Joint Conference on Engineering Systems Design and Analysis, July 4–7, 1994, London, UK, pp33–45.

Al-Ashaab, A.H. 1994. "A Manufacturing Model to Capture Injection Moulding Process Capabilities to Support Design for Manufacture", Ph.D. thesis. Loughborough University, Department of Manufacturing Engineering.

Armstrong G.T. Carey G.C. De Pennington A. 1984. "Numerical Code Generation from a Geometric Modelling System", Solid Modelling in Computers, pp139–137. Plenum Press.

Austin C. 1994 "Computer analysis – The optimisation tool of the 90's", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp527–530, San Fransisco, CA, USA.

Bjorke O., and Myklebust O. 1992. "IMPPACT: Integrated Modelling of Products and Process Using Advanced Computer Technologies". TAPIR Publishers, Trondheim, Norway.

Bond A.H., Chang K.J. 1988. "Feature Based Process Planning for Machined Parts", Proceedings ASME International Computers in Engineering Conference, v1, pp571–576, San Diego, CA, USA.

Booch G. 1991. "Object Oriented Design with Applications". The Bengamin/Cummings Publishing Company, Inc. ISBN 0-8053-0091-0.

Booch G. 1993. "Object Oriented Design with Applications". 2nd Edition, The Bengamin/Cummings Publishing Company, Inc.

Bowen J., and Bahler D. 1991. "Supporting Cooperation between Multiple Perspectives in a Constraint-Based Approach to Concurrent Engineering", Journal of Design and Manufacturing, Vol. 1, pp. 89–105.

Bowen J. and Bahler D. 1992. "Frames, Quantification, Perspectives and Negotiation in Constraint Networks for Life–Cycle Engineering". Artificial Intelligence in Engineering, 7 pp. 199–226.

Bowen J. and Bahler D. 1993. "Constraint-Based Software for Concurrent Engineering". IEEE Computer, 26 (1) pp. 66–68.

Brown,D.R., Cutkosky,M.R., Tenenbaum,J.M. 1991. "Next Cut: A second generation framework for concurrent engineering." Computer aided product development. Springer–Verlag 1991.

BSI DD 194. 1990. ENV 40 003:1990 "Computer Integrated Manufacturing (CIM): CIM Systems Architecture Framework for Modelling". British Standards Institute.

Case K. and Gao J. 1993. "Feature Technology: an Overview" International Journal of Computer Integrated manufacturing Vol. 6 No. 1 & 2, pp. 2–12.

Chakrabarti A. 1993. "Towards a Theory for Functional Reasoning in Design", International Conference on Engineering Design ICED '93, August 17–19, 1993, The Hague, Netherlands.

Chandrasekaran B., Goel A.K., Iwasaki Y. 1993. "Functional Representation as Design Rationale", IEEE Computer, v 26, part 1, January, 1993, pp 48–56.

Chen W–J. and Wallace K. M. 1993. "Manufacturing Information for Designers". International Conference on Engineering Design ICED'93. The HAGUE August 17–19 Vol. 2, pp. 1111–1114.

Choi B.K., Barash M.M., Anderson D. 1984. "Automatic Recognition of Machined Surfaces from a 3D Solid Model". Computer Aided Design 16 (2) 81.

Cinquegrana D. 1990(a). "Intelligent CAD Automates Mold Design", Mechanical Engineering, July 1990, pp71–73.

Cinquegrana D. and McCarthy S. 1990(b). "Mold Optimization Using Rule Based Software". ANTEC'90, pp. 1107–1113. May 7–11, Dallas.

Colquhoun G.J., Baines R.W., Crossley R. 1993. "A State of the Art Review of IDEF0". International Journal of Computer Integrated Manufacture, v6, No4, pp252–264.

Corbett J. 1986. "Design for Economic Manufacture", Annals CIRP, v35/1/1986.

Corbett J., Woodward A.J. 1991. "A CAD integrated knowledge base system for the design of die cast components", Annals CIRP,v40/1/1991,pp103–105.

Corrigall M.J., Lee M.K., Young R.I.M. and Bell R. 1992. "Manufacturing Code Generation in a Product Modelling Environment", Proc. Inst. Mech. Engrs., Vol. 206, pp. 165–175.

Cross N. (1989) "Engineering Design Methods", London: John Wiley and Sons. 1989.

Cunningham,J.J.,Dixon,J.R. 1988. "Designing with features: The origin of features", Proceedings ASMS. International Conference Computers in Engineering, 1988, v1, pp237–243.

Cutkosky, M.R. Tenenbaum, J.M. Muller, D. 1988. "Features in Process-based Design". Computers in Engineering. ASME. pages 557–562.

Cutkosky M.R., Brown D.R., and Tenenbaum J.M. 1989. "Extending Concurrent Product and Process Design Toward Earlier Design Stages", Concurrent Product and Process Design, N.H. Chao and S.C.Y Lu (Eds.), The Winter Annual Meeting of the ASME, San Francisco, California, pp. 65–72.

Cutkosky M.R., Tenenbaum J.M. 1990. "A Methodology and Computational Framework for Concurrent Product and Process Design". Mechanism and Machine Theory 25 (3): 365–381, April.

Cutkosky M.R., Brown D.R., and Tenenbaum J.M. 1991. "Next-Cut: A Second Generation Framework for Concurrent Engineering", Computer-Aided Cooperative Product Development, D. Sriram, R. Logcher and S. Fukuda (Eds.), Springer-Verlag, pp. 1–18.

Cutkosky M.R., Engelmore R.S., Fikes R. E., Genesereth M. R., Gruber T.R., Mark W.S., Tenenbaum J.M. and Weber J.C., 1993. "PACT: An Experiment in Integrating Concurrent Engineering Systems". IEEE Computer, January, pp. 28–37.

Delbressine,F.L.M.,Hjink,J.A.W. 1991. "Discrete part design by taking manufacturing restrictions into account", Annals CIRP,v40/1/1991,pp171–174.

De Martino T., Falcidieno B., Giannini F., Hassinger S., Ottcharova J. 1994. "Feature-based modelling by integrating design and recognition approaches", Computer Aided Design, v 26, No 8, August, 1994, pp646–653.

Dewhurst,P.,Boothroyd,D. 1987. "Design for assembly in action", Assembly Engineering, January 1987.

Dixon, J.R. 1988. "Designing with Features: Building Manufacturing Knowledge into More Intelligent CAD System". ASME Manuf. Int. 88. Atlanta, G.A. pp. 51–57.

Dixon J.R. 1995. "Knowledge-Based Systems for Design" Transactions ASME, vol 117, June 1995, pp11-15.

Domazet,D.S. 1992. "Concurrent design and process planning of rotational parts", Annals CIRP, v41/1/1992, pp181-184.

Domazet D.S., Choong F.N., Sng D, Ho N.C., Lu S.C.Y. 1995. "Active Data-Driven Design Using Dynamic Product Models", Annals CIRP, v44/1/1995, pp109-112.

Dong X., De Vries W.R., Wozuy M.J. 1991. "Feature Based Reasoning in Fixture Design", Annals CIRP, v40/1/1991.

Dowlatshahi S. 1992. "Product Design in a Concurrent Engineering Environment: An Optimisation Approach", International Journal of Production Research, v30, No 8, pp1803-1818, 1992.

Dowlatshahi S. 1994. "A Comparison of Apporaches to Concurrent Engineering", International Journal of Advanced Manufacturing Technology, No9, 1994, pp106-113. Springer-Verlag.

Duan W., Zhou J., Lai K. 1993. "FSMT: A Feature Based Solid Modelling Tool for Feature Based Design and Manufacture", Computer Aided Design,v5, No1, January, 1993, pp29-38. Butterworth and Heineman

Dym J. B. 1987. "Injection Molds and Moulding. A Practical Manual", Van Nostrand Reinhold Company.

Ellis T.I.A, Young R.I.M. and Bell R. 1993a. "Modelling Manufacturing Process Information to Support Simultaneous Engineering". ICED'93, August pp. 17-19, The HAGUE, Netherlands.

Ellis T.I.A, Bell R and Young R.I.M. 1993b. "A Design for Manufacture Software Environment to Support Simultaneous Engineering". In Advances in Manufacturing Technology VII. Ed. Bramley A. and Mileham T. The Proceeding of the 9th National Conference on Manufacturing Research. September, pp. 301-305. UK.

Ellis T.I.A., Molina A., Young R.I.M. 1994. "The Development of an Information Sharing Platform for Concurrent Engineering", International Manufacturing Systems Engineering Workshop, December 12-14, 1994, Grenoble, France.

EXPRESS 1992. "EXPRESS Language Reference Manual". ISO DIS 10303-11.

Fagan D.J. 1986. "Extracting and Organising of Manufacturing Features from a Three Dimensional CAD Database", Proc. Computers in Eng. Conf. Chicago, Ill. pp. 265–272.

Farris J., Knight W.A. 1992. "Design for Manufacture: Expert Processing Sequence Selection for Early Product Design", Annals CIRP,v41/1/1992.

Finger S., Fox M.S., Prinz F.B. and Rinderle J.R. 1992. "Concurrent Design". Applied Artificial Intelligence, 6:257–283.

Fridshall R. 1988. Presentation at NSF workshop on Features in Design and manufacturing, UCLA, CA, Febuary, 1988.

Gadh R., Hall M.A., Gursoz E. L., Prinz F.B. 1989. "Knowledge Driven Manufacturability Analysis from Feature–Based Representations", ASME Winter Annual Meeting, Symposium on Concurrent Product and Process Design, pp. 21–34.

Gadh R. and Prinz F. B. 1992. "On the Recognition of Geometric Forms Using the Differential Depth Filter". Computer Aided Design, November, 24 (11) pp. 583–598.

Gadh R. and Prinz F. B. 1995. "Automatic Determination of Feature Interactions in Design–for–Manufacturing Analysis", Transaction ASME, vol 117, March 1995, pp2–9.

Gerdes V.B., Webb D., Chassapis C. 1994. "Concurrent Engineering Approach to Injection Mold Design and Fabrication", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp1111–1117, San Fransisco, CA, USA.

Gopalakrishnan B., Pandiarajan V. 1991. "Materials and manufacturing process selection system for product designs in concurrent engineering", Journal of Material Processing Technology,28(1991), pp93–103. Eslevier.

Hambaba M., Elsherif H.E., O'Brien W. and Bergstorm E. 1992. "Mold–Base Part Design Using Neural Networks", IFAC Workshop, Intelligent Manufacturing System Dearbon, Michigan USA October, pp. 37–41.

Hanada H. and Leifer L.J. 1989. "Inteligent Design System for Injection Molded Parts Based on the Process Function Analysis Method". NSF Engineering Design Research Conference. June 11–14, pp. 597.

Harding J. Popplewell K.P. 1994. "The Rationale for an Engineering Moderator", MOSES Project Report, series 26, November, 1994.

Henson B. Juster N. 1994. "Towards an Integrated Representation of Function, Behavior and Form", Computer Aided Conceptual Design, Proceedings 1994 Lancaster International Workshop on Engineering Design (CAD'94), April, 1994, Lancaster University, UK.

Huh Y.J. and Kim S.G. 1989. "RIBBER: A Knowledge-Based Synthesis System for RIBBED Injection Molded Parts". Concurrent Product and Process Design, (Ed.) N.H. Chao, S.C.Y. Lu., Proceedings of the Winter Annual Meeting of ASME, DE Vol.1, PED Vol. 36, San Francisco, CA, December 10–15, pp. 195–204.

Hui K.C., Tan S.T. 1992. "Mould design with sweep operations – a heuristic search approach", Computer Aided Design, v24, No2, February, 1992, pp81–91. Butterworth Hieneman.

Hui K.C., Tan S.T. 1994. "Automatic Determination of Parting Directions from a Solid Model", Computer Aided Design, 1994, pp359–362. Butterworth Hieneman.

IMPPACT. 1991. Esprit No. 2165, Proceedings of the Workshop, Berlin, 26–27 February 1991.

Irani R.H., Kim B.H., Dixon J.R. 1989a. "Integrating CAE. Features and Iterative Redesign to Automate the Design of Injection Molds", Proc. 1989 ASME Int. Computers in Eng. Conference, Anaheim, California, pp. 27–33.

Irani R.H., Kim B.H., Dixon J.R. 1989b. "Automated Injection Mold Gating Design System". ANTEC'89 . pp. 1265–1269.

Irani R.H., Kim B.H., Dixon, J.R. 1990. "Automated Design of the Feed System of Injection Molds". ASME Manuf. Int. Conf. Atlanta, Ga. pp. 45–51.

Ishii K., Hornberger L., Lion M. 1989. "Compatibility-Based Design for Injection Molding". In Concurrent Product and Process Design. Ed. N.H. Chao and S.C.Y. Lu. The Winter Annual Meeting ASME. San Francisco, Cal.

Ishii K. 1991. "Life-Cycle Engineering Using Design Compatibility Analysis". Proc. of the 17th NSF Design and Manufacturing Systems Conference, SME, January, 1991, Austin, Texas, pp. 1059–1065.

Ishii K., Hornberger L. 1991. "Keys to the Successful Development of an AI-Based Tool for Life-Cycle Design". In Artificial Intelligence in Design '91. J.S.Gere (Ed.).

Ishii K. 1992. "The Fourth Annual Life-Cycle Design Sponsors Meeting". June 9th. Dept. of Mechanical Engineering, The Ohio State University.

Ishii K. 1995. "Life–Cycle Engineering Design", Transactions ASME, v 117, June 1995, pp42–47.

Ishii K., Miller R.A. 1992. "Design Representation for Manufacturability Evaluation in CAD: Beyond Feature Based Design", Computers in Engineering, v1, Proceedings ASME, 1992.

Ishii K. and Mukherjee S. 1992. "Computer-Aided Life–Cycle Design Compatibility Analysis". Proceedings of the NSF Design and Manufacturing Conference, SME, January, 1992, Atlanta, GA.

Iwata, K., Kakino Y., Oba F., Sugimura N. 1980. "Development of non Part Family Type Computer Aided Production Planning System", Advances in Manufacturing Technology, pp171–184, North–Holland.

Iwata K., Sugimur N. 1986. "Development of Product Model for Design and Manufacturing of Machine Products". 14 NAMRC Proc. 1, 528–534.

Iwata K., Matsuo E. and Onosato M. 1992. "Development of New Methodology for Mechanical Design by Symbolized Elements with Movement". Annals of the CIRP Vol. 41, No. 1.

Jared G.E.M. 1986. "Feature Recognition and Expert Systems for Operation Planning in NC Machining", Proceedings International Conference on AI Europa, Wiesbaden, Germany.

Jared G.E.M. 1991. "Geometric Reasoning for Economic Manufacture", Progress report on SERC Grant No GR/F/70785.

Jong W.R., Chen S.C. Wang K.K. 1994. "An Integrated Design System for 3D Runner Balancing of Injection Molding Process", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp749–753, San Fransisco, CA, USA.

Joshi S., Vissa N.N., Chang T.C. 1988, "Expert Process Planning System with Solid Model Interface", International Journal of Production Research, 26, pp863–885.

Joshi S. and Chang T.C. 1990. "Feature Extraction and Feature Based Design Approaches in the Development of Design Interface for Process Planning". Journal of Intelligent Manufacturing Vol. 1 pp. 1–15.

Kahaner D., and Lu S. 1993. "Concurrent Engineering Research in Review". First CIRP International Workshop on Concurrent Engineering for Product Realization, Volume 5, pp. 6–14.

Kang,M.,Kim,S.G.,Chong–won Lee. 1990. "CIM for mould factory automation", Annals CIRP, v39/1/1990, pp467–470.

Kang,M.,Park,S.,Hong,Y. 1992. "Computer integrated mould manufacturing", International Journal of CIM, v5, No4/5, pp229–239.

Kim Y.S. 1992 "Recognition of Form Features Using Convex Decomposition", Computer Aided Design, v24, No9, September, 1992, pp461–476.

Kimura F. 1991. "Software for Product Realization". International Symposium for International Trends in Manufacturing Towards the 21st Century, Oct. 18, Berlin, Germany.

Kimura F. 1993(a). "Product and Process Modelling as a Kernel for Virtual Manufacturing Environment". CIRP Annals Vol. 42. No. 1, pp. 147–150.

Kimura F. 1993(b). "A Computer Supported Framework for Concurrent Engineering – An Approach Based on Virtual Manufacturing", Information Infrastructure Systems for Manufacturing (B–14), 1993, pp345–359. North–Holland.

Kimura F. 1994. "Product and Process Modelling as a Basis for Rapid Product Development", International Conference on Rapid Product Development, January 29 – Febuary 2, 1994, Stuttgart, Germany.

Kimura F. and Suzuki H. 1989. "A CAD System for Efficient Product Design Based on Design Intent", Annals of the CIRP, Vol. 38, No. 1, pp. 149–152.

Kimura F. and Suzuki H. 1995. "Representing Background Information for Product Description to Support Product Development Process", Annals CIRP, vol44/1/1995, pp113–116.

Kinstrey M., Lee V.K., Lewis J.W., Salant E.L., Sarachan B.D. and Wilson P.R. 1990. "PPO Management". Proceedings 2nd National Symposium on Concurrent Engineering, February 7–9, Morgantown, West Virginia, pp. 67–90.

Kjellberg T. and Schmekel H. 1992. " Product Modelling and 'Information–Integrated' Engineering Systems". Annals of CIRP, Vol. 41. No. 1.

Kosanke K. 1990. "CIMOSA – Road to Enterprise Integration". CE and CALS Conference and Exposition, June 1–4, Washington, D.C., USA, pp281–292.

Kusiak A., Larson N. 1995. "Decomposition and Representation Methods in Mechanical Design", Transactions ASME, vol 117, June 1995, pp16–24.

Laakko T., Mantyla M. 1993. "Feature Modelling by Incremental Feature Recognition", Computer Aided Design, v25, No8, August, 1993, pp479–491,

Latif M.N., Boyd R.D., Hannam R.G. 1993 "Integrating CAD and Manufacturing Intelligence through Features and Objects", International Journal of Computer Integrated Manufacture, v6, No1/2, pp87–93, 1993.

Lee,R.J.V. 1992. "Geometric support function in a design for manufacture environment", Discussion document, Department of Manufacturing Engineering,Loughborough University of Technology, December 1992.

Lee R.J.V., Young R.I.M. 1994a. "Design for manufacture: An approach to software support in a concurrent engineering environment", Factory 2000 conference, October 1994, York University,York, UK. IEE publications.

Lee R.J.V.,Young R.I.M. 1994b. "Resolving feature interactions in design for injection moulding",10th National Conference on Manufacturing Research, 5–7 September, 1994, Loughborough University, Loughborough, UK.

Lee Y.C. Fu K.S. 1987. "Machine Understanding of CSG: Extraction and Unification of Manufacturing Features", IEEE Computer Graphics and Applications 7, pp20–32.

Libardi E.C., Dixon, J.R., Simmons M.K. 1986. "Designing with Features: Design and Analysis of Extrusions as an Example". ASME paper No.86–DE4. The Design Eng. Conf. Chicago.

Linberg L.E.A. 1992. "The product structure– The backbone of CIM", Annals CIRP, v41/1/1992, pp165–168.

Linberg L.E.A. 1993. "Notes on Concurrent Engineering", Annals CIRP, v42/1/1993, pp159–162..

Linington P. F. 1992. "Introduction of Open Distributed Processing Basic Reference Model", Proceedings 1st International Workshop ODP, University of Kent, UK. v1, 1992, pp3–13.

Lodenstien M.A., Romps D.M., Tran P. 1994. "Development of Mold Design Software", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp1091–1093, San Fransisco, CA, USA.

Londono F., Cleetus K.J., and Reddy Y.V. 1990. "A Blackboard Scheme for Cooperative Problem–Solving by Human Experts". Computer–Aided Cooperative Product Development. D. Sriram, R. Logcher and S. Fukuda (Eds.), Springer–Verlag, pp. 26–50.

Lu S.C-Y. 1992. "Research, Development and Implementation of Knowledge Processing Tools to Support Concurrent Engineering Tasks". In the Annual Report of Knowledge-Based Engineering Systems Research Laboratory, University of Illinois at Urbana-Champaign, pp. 3–25.

Lu S., Smith K., Herman A., Mattox D., Silliman M., Lucenti M., Jacobs J., Chazin D. Lawley M. and Case M. 1993. "SWIFT: System Workbench for Integrating and Facilitating Teams". Proceedings of the 2nd Workshop on Enabling Technologies Infrastructure for Collabrative Enterprises (WET ICE) Morgantown, West Virgina, April 20–22.

Luby S.C., Dixon J.R., Simmons M.K. 1986. "Designing with Features: Creating and Using a Features Data Base for Evaluation of Manufacturability of Castings". Proc. Computers in Eng. Conf. Chicago, Ill. pp. 285–292.

Manoochehri S. 1994. "An Intelligent Function Based Design System for Injection Molded Plastic Parts", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp680–683, San Fransisco, CA, USA.

Mantyla M. 1993. "Towards Open Architecture Concurrent Engineering Frameworks". Helsinki University of Technology, Laboratory of Information Processing Science, Tekniikantie 12, SF–02150 Espoo, Finland. Otaniemi 1992lTKO–B96.

Mantyla M., Lagus K., Laakko T. 1994. "Application of Constraint Propagation in Part Family Modelling", Annals CIRP, v43/1/1994, pp129–132.

McGinnis B.D., Ullman D.G. 1992. "The Evolution of Commitments in the Design of a Component", Transactions ASME Journal of Mechanical Design, March, 1992, pp1–7.

McKay A., Bloor S. 1991. "The Role of Product Models in Effective CAD/CAM". Proceedings of the Institutions of Mechanical Engineers, Effective CADCAM '91, European Conference, November 6–7, 1991, pp113–119, Coventry, UK.

McKay A. 1993. "The Role of Product Data in Assemble to Order Engineering", MOSES Project Report, series 11, November, 1993.

Medland A.J., Mullineax G. 1993. "A Constraint Approach to Features Based Design", International Journal of Computer Integrated Manufacture, v6, No1/2, pp34–38, 1993.

Michaeli W., Galuschka S. 1994. "New Developments in the Simulation of Injection Moulding and Special Processes", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp565–570, San Fransisco, CA, USA.

Michaeli W., Wisinger S., Galuschka S., Zachert J. 1995. "Avoiding Faults on Mouldings - Weld Lines, Shrinkage and Distortion" Kunststoffe plast europe, November, 1995, pp15–16. Carl Hanser Verlag, Munchen.

Milaric V.R. 1985 "SAPT- Expert System for Manufacturing Planning", Computer Aided/Intelligent Process Planning, ASME Winter Annual Meeting, PED–v19, ASME, 1985.

Mill F.G., Salmon J.C., Pedley A.G. 1993. "Representation Problems in Feature Based Approaches to Design and Process Planning", International Journal of Computer Integrated Manufacture, v6, No 1/2, 1993, pp27–33. Taylor and Francis.

Mochizuki T., Yuhara N. 1992. "Methods of Extracting Potential Undercut and Determining Optimum Withdrawal Direction for Mold", International Journal Japanese Society Precision Engineering, v26, No1, March, 1992.

Molina A., Mezgar I., and Kovacs G. 1992. "Object Knowledge Representation Models for Concurrent Design of FMS". Human Aspects in Computer Integrated Manufacturing. G.J. Olling, F. Kimura (Eds.), Elsevier Science Publishers B.V., (North–Holland), IFIP, pp.. 779–788.

Molina A., Ellis T.I.A., Young R.I.M., Bell R. 1993. "Reference Models for the Computer Aided Support of Simultaneous Engineering". International Journal of Computer Integrated Manufacture, April, 1993. Taylor and Francis.

Molina A., Ellis T.I.A., Young R.I.M. 1994(a) "Methods and Tools for Modelling Manufacturing Information to Support Simultaneous Engineering", Proceedings 2nd IFAC/IFAP/IFORS workshop, IMS'94, June 13–15, 1994, Vienna, Austria.

Molina A., Ellis T.I.A., Young R.I.M. 1994(b). "Modelling Manufacturing Resources, Proceses, and Strategies to Support Concurrent Engineering", First International Conference on Concurrent Engineering Research and Applications, August 29–31, 1994, Pittsburgh, PA, USA.

Molina A., Al–Ashaab A.H., Ellis T., Young R.I.M. and Bell R. 1995. " A Review of Computer Aided Simultaneous Engineering Systems", Research in Engineering Design, No7, 1995, pp 38–62. Springer–Verlag.

Ni S., Guinn R.A. 1994. "Warpage Analysis and Resulting Improved Design for a Lazer Printer Part", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp531–539, San Fransisco, CA, USA.

Nichols D.M. 1992. "Monitoring Design Progress in a Concurrent Engineering Environment". CE & CALS Conference and Exposition, June 1–4, Washington, D.C., pp. 65–76.

Ong S.K., Prombanpong S., Lee K.S. 1995. "An Object Oriented Approach to Computer Aided Design of a Plastic Injection Mould", Journal of Intelligent Manufacturing, v6, No1, Febuary, 1995, pp1–10. Chapman and Hall.

Pande S.S., Prabhu B.S. 1990. "An Expert System for Automatic Extraction of Machining Features and Tooling Selection Automata", Computer Aided Engineering Journal, v7, No4, August, 1990, pp99–103, MCB University Press.

PDES/STEP 1988. "Product Data Exchange Specification". Ist Working Draft, NISTIR 884004 National Institute of Standards and Technology, National Engineering Laboratory, Gaithersburg, MD, USA.

Poli C., Kuo S. and Graves R., 1992. "Design for Injection Molding: A Group Technology-Based Approach". Journal of Engineering Design, Vol.3, No. 4, pp. 307–324.

Pratt, M.J. 1993. "Applications of feature recognition in the product life cycle", International journal of Computer Integrated Manufacture, v6, Nos1&2, 1993, pp13–19.

Pye R.G.W. 1989. "Injection Mould Design – A Design Manual for the Thermoplastics Industry", Longman/George Goodman, 1989.

Rehg J., Elfes A., Talukdar S., Woodbury R., Eisenberger M. and Edahl R. 1988. "CASE: Computer-Aided Simultaneous Engineering". Artificial Intelligence in Engineering: Design. (Ed.) J.S. Gero, Elsevier, pp. 339–360.

Rho,H.M.,Sheen,D.,Lee,C.S. 1990. "A description scheme of mould dies by using specific part features", 22nd CIRP Seminar of Manufacturing Systems,1990,pp1–10.

Sapossnek S., Talukdar S., Elfes A., Sedas S., Elsenberger M. and Hou L. 1989. "Design Critics in the Computer-Aided Simultaneous Engineering (CASE) Project". Concurrent Product and Process Design. N.H. Chao, S.C.Y. Lu. (Eds.) Proceedings of the Winter Annual Meeting of ASME, DE Vol.1, PED Vol. 36, San Francisco, CA, December 10–15, pp. 137–141.

Sebastian D. H. 1994. "Concurrent Design Methodology for Injection Moulded Parts", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp687–689, San Fransisco, CA, USA.

Schmekel H. 1989. "Functional Models and Design Solutions", Annals CIRP, v38/1/1989, pp129–132.

Schute M., Weber C. 1993. "The Relationship Between Function and Shape", Proceedings Inetrnational Conference on Engineering Design ICED '93, August 17–19, 1993, pp9–19, The Hague, Netherlands.

Shah J.J., Rogers M.T. 1988. "Functional Requirements and Conceptual Design of the Feature based Modelling System", Computer Aided Engineering Journal, Febuary, 1988, pp 9 –15.

Sivard G., Linberg L., Agerman E. 1993. " Customer Based Design with Constraint Reasoning", Annals CIRP, v42/1/1993, pp 139 – 142.

Solenius, G. 1992. "Concurrent Engineering", Annals CIRP, 41/2/1992, pp645–655.

Suh N.P. 1990. "The Principles of Design", Oxford Universtity Press. 1990.

Suh N.P. 1995. "Axiomatic Design of Mechanical Systems" Transactions ASME, v117, June, 1995, pp2–10.

Syan, C.S. 1994. "Introduction to Concurrent Engineering, Chapter1– Concurrent Engineering, Concepts, Implementation and Practice", Chapman and Hall, 1994.

Torbenhenau H., Lianchum M. 1993. "Features in Integrated Modelling of Products and Their Production", International Journal of Computer Integrated Manufacture, v6, No 1/2, pp65–73, 1993.

Tseng H.H., Lia F.S., Peng T. 1994. "Integration of CAD/CAM/CAE in Mold Design and Manufacturing", Proceedings Society of Plastics Engineers ANTEC'94, May 1–5, 1994, pp1100–1103, San Fransisco, CA, USA.

Urban S.D., Shah J.J., Rogers M.T. 1993. "Engineering Data Management: Achieving Integration Through Database Technology", Computing and Control Engineering Journal, June 1993, v4, No3, pp119–126. IEE Publications.

Vujosevic R. and Kusiak A. 1991. "Data Base Requirements for Concurrent Design Systems". In Engineering Databases: An Engineering Resource, Proceedings of the ASME International

Computer in Engineering Conference & Exhibition, V. Saxena (Ed.), August 18–22, Santa Clara, California, pp. 109–113.

Weirda,L.S. 1991. "Linking design, process planning and cost information by feature based modelling. Journal of Engineering Design,v2,No1,1991,pp3–9.

WEKA 1991. "New horizons in plastics – A handbook for design engineers", WEKA publishing Ltd, London.

Winjard L., Carlberg P., Kjellberg T. 1992. "Enabling Use of Engineering Terminology in Product Models and User Interfaces of CAD/CAM Systems", Annals CIRP, v41/1/1992, pp205–208.

Winner R.I., Pennell J.P., Bertrand H.E. and Slusarczuk M.M.G. 1988. "The Role of Concurrent Engineering in Weapon System Acquisition". IDA–Report R–338.

Wu J.K., Liu T.H. and Fisher G.W. 1992. "An Integrated PDES/STEP Based Information Model for CAE and CAM Applications". Technical Report R–129, Center for Simulation and Design Optimization of Mechanical Systems, The University of Iowa.

Wu J.K. and Choong F.N. 1992. "A Process Modeling System for Concurrent Engineering of Mechanical Systems". Intelligent Systems in Concurrent Engineering. Proceedings of the AID '92 Workshop, TR–92–15, North Carolina State University.

Young R.I.M., Bell R. 1993. "Design by Features: Advantages and Limitations in Machine Planning Integration", International Journal of Computer Integrated Manufacture,v6, No1/2, pp105–112, 1993.

Young R.E., Greef A. and O'Grady P. (1991). "Spark: An Artificial Intelligence Constraint Network System for Concurrent Engineering". Proceedings First International Conference on Artificial Intelligence in Design, Butterworth–Heinemann Publishers.

Zu F., De Pennington A., Saia A. 1993. "A Graph Grammar Approach to Feature Representation and Transformation", International Journal of Computer Integrated Manufacture, v6, No 1/2, 1993, pp137–151.