COLLABORATION IN DISTRIBUTED INJECTION MOLD DESIGN: PROCESS ANALYSIS AND SYSTEM IMPLEMENTATION

DU XIAOJUN

(B.Eng)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude and appreciation to my supervisors, A/P Lee Kim Seng, and A/P Wong Yoke San, for the opportunities they have given to me and for their invaluable guidance and sustained support throughout the duration of my project. Their guidance, advices and supports have brought me great inspirations in my research work and kept me oriented in my journey. Their encouragement and enthusiasm gave me great motivation and confidence, which got me through difficulties that I encountered. From them, I have learned a lot about the attitudes towards both work and life, and skills of conducting research as a researcher. I will always remember their invaluable guidance wherever I go.

Many thanks also go to my colleagues at Imold lab of the National University of Singapore. They are Mr. Sun Yifeng, Mr. Woon Yong Khai, Mr. Atiqur Rahman, Mr. Saravanakumar Mohanraj, Mrs. Cao Jian, Miss. Low Leng Hwa, Maria, and Miss Zhu Yada. Thanks to their kindness and friendship for an enjoyable and unforgettable time working together.

Upon my research project, I want to give my utmost appreciation to researchers and students at Laboratory For Concurrent Engineering and Logistics (LCEL) of the National University of Singapore, in particular the former staff Dr. Wang Jianguo, and students Khoo Boon Bing, Thoo Lee Ming, and Koh Guan Lee. I am indebted to them for helping me in the implementation of project. Without their assistance, my research project would not be able to be carried out smoothly. Thanks are extended to the

National University of Singapore for providing the research scholarship for the two-year research work.

I would also like to thank my ex-boss, Mr. Edward H.Depew, for his full support and understanding when I made the decision of studying aboard, and my friends Yao Xiaofeng, Zhang Wei, Chen Yu who have made my life in Singapore exciting and memorable.

Lastly, my sincere gratitude goes to my parents for their love, trust and support.

TABLE OF CONTENTS

SUMMARY

Plastic product development is typically a highly iterative process that involves collaboration groups of designers, manufacturers, their subcontractors, and suppliers. With the evolution of computer-aided design tools and the widespread availability of the Internet application as the medium for information sharing and distribution, the plastic product development is increasingly collaborated globally. The injection mold manufacturing is an important part of plastic industry, which is widely adopted because of its advantages in mass production of parts of complex geometry. With the abovementioned emerging trend, injection mold design and manufacturing also become involved in the collaboration of dispersed discipline groups.

In the globalized plastic injection mold product development, the bottlenecks are often the process control, information transfer and resource relocation. This thesis examines and attempts to provide methods to some aspects: to develop a mold design process that facilitates concurrent engineering-based practice and to implement this proposed process model in a computer-supported and web-enabled system for the collaboration of geographically dispersed users. The research objectives are summarized as follows:

1. **Process modeling of injection mold design process with the design structure matrix.**

   The mold development process model discussed in this thesis is based on the system engineering methodologies - the design structure matrix (DSM) that facilitates the application of concurrent engineering concept. DSM analyzes the system and the sub-activities from the perspective of the relationship of information dependencies: the parallel, the series, and the iteration. By identifying and interpreting the different types of information relationships among activities involved in the injection mold design, DSM helps to evolve the process model. This process model proposes the concept of collaboration among different discipline groups.

2. **The Computer-supported and web-enabled system to implement the process model.**

The computer-supported and web-enabled system developed in this system is based on the principle of Computer-Supported Collaborative Work (CSCW). The CSCW is a mechanism that supports the work activity of a group of people working on the same product development or technical area. It basically comprises of Computer Technical Features, the Decision Modeling Tools, and the Group Communication Support. This thesis focuses on the application of Computer Technical Features and the Group Communication Support with the SmarTeam system. The areas of data management, process control, and Internet collaboration of this system are extensively developed to improve the applicability of the proposed process model and set up a collaborative product development environment for the injection mold design and manufacturing.

NOMENCLATURE

CE	Concurrent engineering
CPD	Collaborative product development
CSCW	Computer-supported collaborative work
DSM	Design structure matrix
F_i	The information input or output upon which a task/activity operates
IDEF0	Integrated definition for function modeling
IMM	Injection molding machine
IT	Internet technology
OODB	Object-oriented database
Partitioning	Process of manipulating or reordering the matrix rows and columns
PD	Product development
PDM	Product data management
PERT/CPM	The project evaluation and review technique associated critical path method
PM	Process modeling
T_i	The element task/activity of a system process
WBS	Work breakdown structure

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In today's competitive market, the requirements for cost saving and reduced product lead-time have created a challenging environment in product development. Product development company has also become increasingly globalized and decentralized. The trend is towards a team effort involving various groups of designers, manufacturers, suppliers, customers, and other outsourced parties across the world.

In plastic product development, injection molding is widely used in the manufacturing of the plastic parts [Potsch 1995]. Traditionally, the injection mold design and fabrication are carried out at the same geographical location. However, with increasing globalization, die and mold companies have become more involved in the plastic product development in a globally distributed manner, and the injection mold design and fabrication are the co-efforts of globally dispersed specialized groups.

The spatial and geographical discontinuities due to this change have raised the concerns of both researchers and practitioners of the injection mold product developers. Since the injection mold product development is highly dependent on the close cooperation among different groups in the product design, mold design, mold making, and standard components suppliers, the lack of basic interpretation of the dependencies and relationships among the various groups can result in a lack of effective information communication as well as delay in the mold fabrication. For this reason, it is important to study the system framework to facilitate the collaboration among distributed discipline groups and identify suitable configuration that can streamline the collaborative injection mold product development.

1.1 The plastic injection molding process

Injection molding is the primary manufacturing process in plastic industry. It consists of heating the thermoplastic material until it melts, forcing this melt material into a steel mold, and converting this melt material to the finalized plastic parts.

Figure 1.1 shows an injection molding process. It starts with putting the pre-heated raw plastic material in the form of pellets or powder into the hopper. From there the material flow enters the injection unit, where a screw rotates in a cylinder (barrel) and transports the melt in front of the screw to the screw chamber. Because of the increasing melt volume in front of it, the screw moves axially forward. The melt is then ejected into the mold and held under pressure. In the mean time, the clamping unit on the injection mold machine moves forward until the mold halves are in close contact. After the melt plastic inside cools and solidifies, the mold is open and the finished part is ejected.

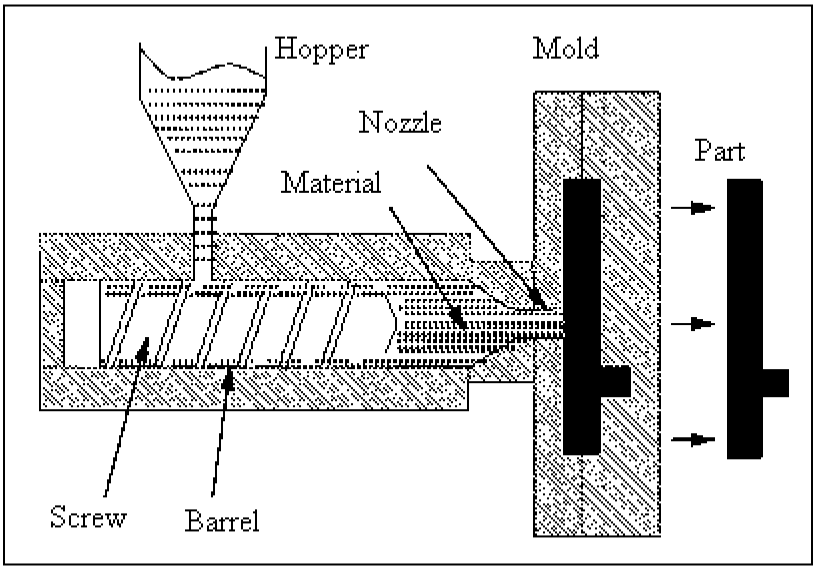


Figure 1.1 An injection molding process

The main components in the injection molding process are the ***injection molding machine (IMM)*** and the ***mold***. The major tasks of the *injection molding machine* are to melt and pressurize the plastic material, inject the molten material into the cavity of the mold, cools the mold, and eject the molding part. Figure 1.2 shows the basic configuration of an ***injection molding machine***. Among the components, the clamping unit and the plasticating unit perform the main functions. The clamping unit exerts a clamping force to keep the mold closed tightly against the injection pressure so that the pressure in the mold cavities could be retained. The plasticating unit is used to melt the plastic material and inject the molten material into the cavity of the mold. The task of melting the plastic material is achieved by a screw inside. It takes in material from the hopper while it rotates. The action of rotating causes the material to proceed towards the nozzle, shearing the material, producing friction, and heating the material.

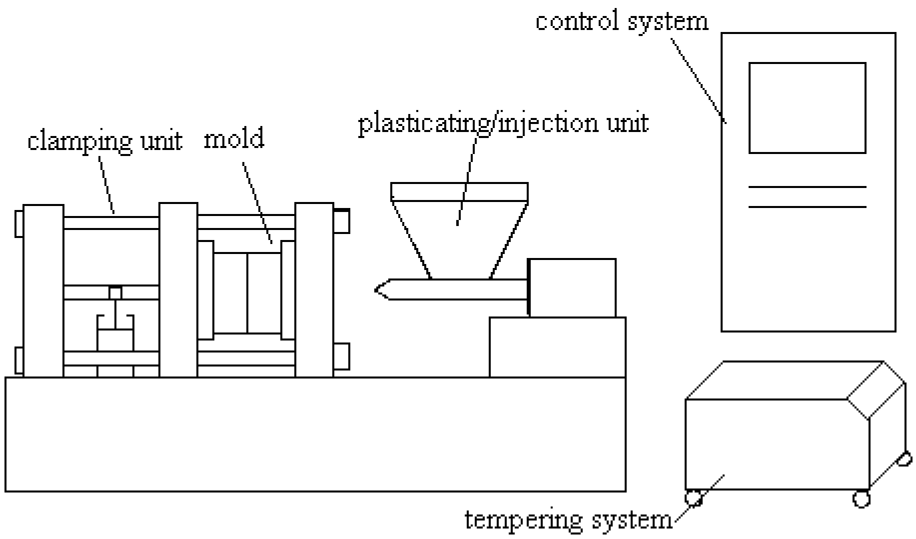


Figure 1.2 Elements of the injection molding machine

The mold is the key component in the injection molding process. It is normally made of metals, primarily steel. The mold is an assembly tool with multiple components. Its function is twofold: imparting the desired shape to the plasticized melt and solidifying the molded product. The injection mold has two basic sets of components: (1) the cavities and cores, and (2) the base in which the cavities and cores are mounted. Figure 1.3 and Figure 1.4 demonstrate the typical layout of core, cavity, and other components in an injection mold. The space between matched molds is known as the mold cavity, which forms the outer surface of the final product. There can be single or multiple cavities in one mold. The mold core functions to form the interior surface of the molded part. The separation between the female and male mold parts is called the ***parting line***. The male part and female part together form the mold base. The mold base is also the place where the mold components are held. Such components include sprue bush, register ring, ejection parts, mechanisms tools like sliders or lifters and the alignment screws etc.

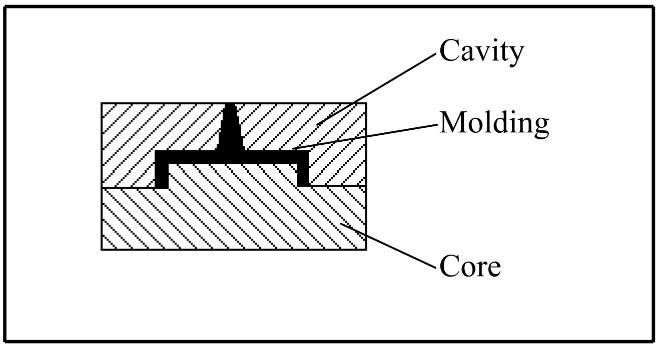
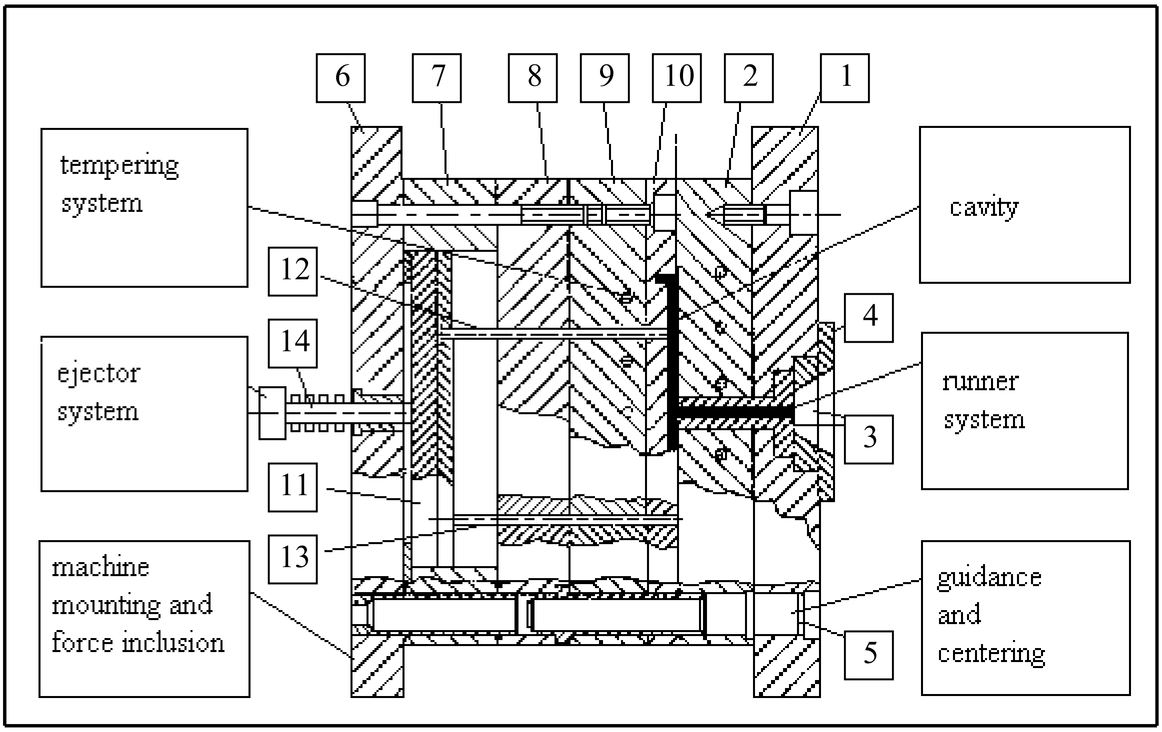


Figure 1.3 The mold core and cavity



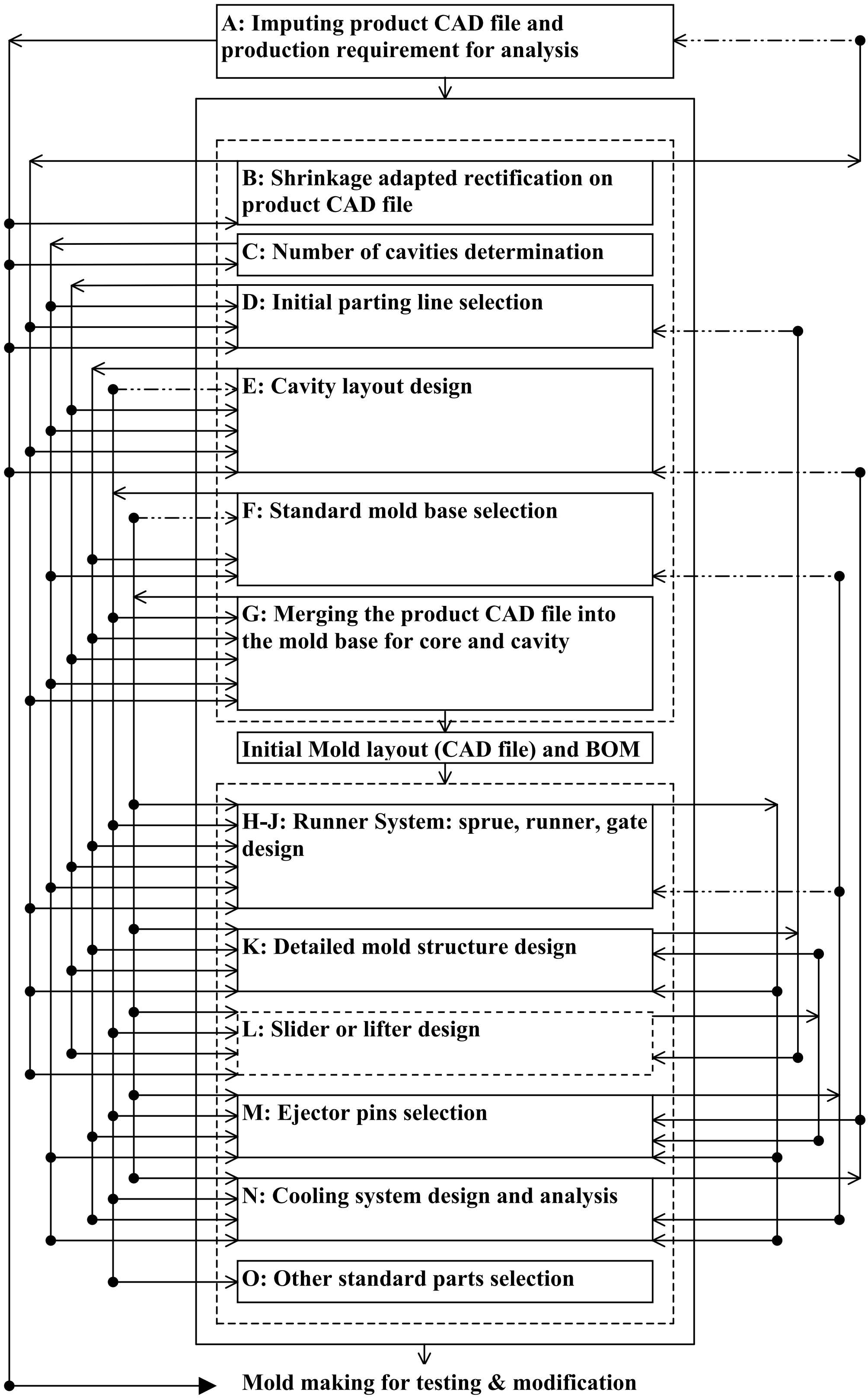
1 base plate	5 guiding pin	9 center plate	13 push back pin
2 molding plate	6 mounting plate	10 molding plate	14 ejector bolt
3 sprue bushing	7 adapter plate	11 ejector plate	
4 locating ring	8 center plate	12 ejector pin	

Figure 1.4 A mold base layout

1.2 The injection mold design

The injection mold development includes molded part evaluation, mold design, mold manufacturing process planning, and mold making. Increasingly, it is concurrently involved in plastic product development at the early stage, interacting with product design and down stream processes. In injection mold development, mold design plays a key role. It contributes important information like part features, detailed mold drawing, and bill of material to mold manufacturers and part suppliers. The information of the design outcome affects the performance of the final product significantly.

The injection mold design is composed of multiple activities, each of which is dedicated to the design of different mold components, and most of which depend on each other for information. The general interdependence relationships between the injection mold design activities are illustrated in Figure 1.5 [Menges and Mohren 1993].



Forward dependency on information; Feedback dependency on information;

Figure 1.5 The interaction relationship among injection mold design activities

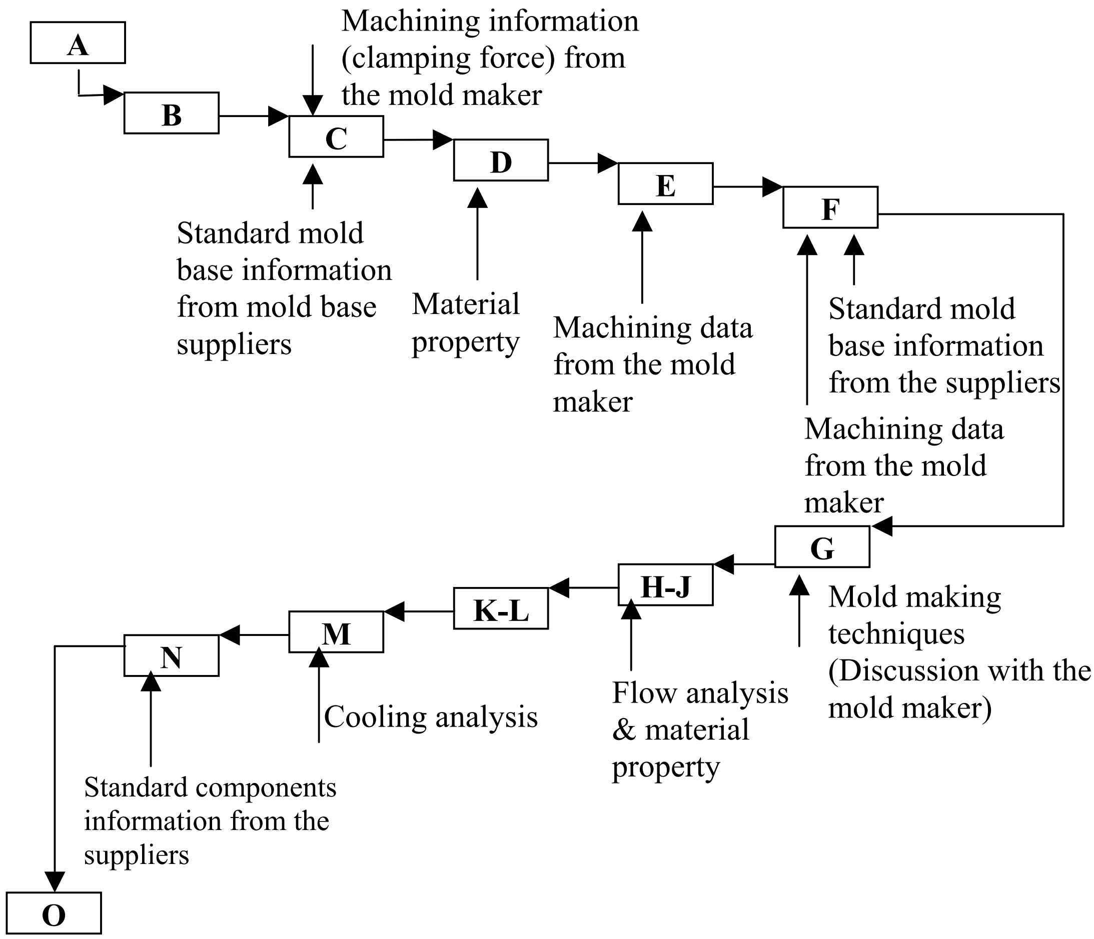


Figure 1.6 The conventional injection mold design sequence

The conventional injection mold design scenario based on the interrelationships interpreted in Figure 1.5 is shown in Figure 1.6. According to the design sequence, the injection mold design activities involve several major specialized groups, namely the product designer, the mold designer, the mold maker, the raw material suppliers including the mold base suppliers and the standard mold components suppliers. The whole process involves extensive communication between these specialized groups, and is based on the coordination of individual design tasks. Most of the decisions made at each design task are based on the mold designer's comprehensive understanding and consideration of the information provided by different collaborating groups involved in the mold design process and the knowledge related to the injection mold development. However, the uncertainty and misunderstanding of the information can lead to incorrect decisions being made and thus increase the complexity of the injection mold development. For efficient management of the information, it is advisable that a process model, with information flows and iteration among design activities comprehensively described, be set up in the mold design stage.

1.3 Research objectives

The objective of this research is to identify and develop a system framework to facilitate the effective collaboration in distributed injection mold product development, especially the injection mold design stage. Three steps are proposed. The first step is to configure the injection mold design process by structuring the information flows among of activities with the Design Structure Matrix methodology. The second step is to model the mold design process for multi-parties' concurrent and collaborative involvement. This part of the work is based on the DSM interpretation in the previous step. The third step is to implement the process model on a computer network-based system, based on commercially available the SmarTeam system.

The proposed three steps are based on the theories of system modeling and simulation, and adapted to the injection mold design context. The network-based system-the SmarTeam system is basically a part of the Product Development Profile from the Dassaut Systems company, and currently used for distributed product development in the Lab for Concurrent Engineering and Logistics (LCEL) at the National University of Singapore. In order to test the utility of this proposed framework, two case studies are demonstrated to illustrate the interaction and communication among distributed discipline groups involving in an injection mold design project. The result and validity of this proposed framework for collaborative injection mold design is discussed.

1.4 Thesis outline

The rest of the thesis is organized as follows.

Chapter 2 is the literature review on the related works in the area of product development in a distributed environment.

Chapter 3 presents the process-modeling method using the design structure matrix. This chapter also discusses the procedure of setting up a matrix model for general product development and injection mold design, and the interpretation about the information relationship among injection mold design activities.

Chapter 4 presents the architecture of the computer-supported and web-enabled system that is used to support the application of the proposed injection mold design process model. Two cases are also presented and discussed.

Chapter 5 concludes the research work in this thesis and discusses the present limitations and future works.

CHAPTER 2

LITERATURE REVIEW

The literature on collaborative product development was quite limited until 1990s when the ***Internet technology*** became widely applied. In the beginning of this development, most of the research works were conducted on large-scale or complex projects [Park and Cutkosky 1999]. Small or medium-size product development, such as the injection mold product development, was not focused until the end of last century. Despite this, these previous research works pave the way for further development and provide inspiration for this research. In this section, a review of the related research work is presented.

The literature on collaborative injection mold product development can be categorized into two areas: applying the ***Concurrent Engineering*** concept to systematically integrate the interdisciplinary co-operation in the injection mold product development; and implementing web-based computer system for virtual mold product development, real time communication, and data sharing in a distributed engineering environment.

The research work of system approaches of ***Concurrent Engineering*** application is based on the perspective of process management: from the view point of process management, the product development or a system is "a collection of hardware, software, people, facilities, and procedures organized to accomplish some common objectives." ***Concurrent Engineering*** is the integration of all company resources needed for product development, and it aims to decrease the product lead-time and increase the product competitiveness by enabling the parallel cooperation of multi-parties and multi-sources [Forsberg and Mooz 1992; Lake 1992]. In the injection mold product development, the concurrency implies the concurrent interaction among mold development activities and the concurrent interactions between groups involved. The research work in this area is therefore focused on how to enable the concurrent interactions.

Implementing the web-based collaborative framework is inspired by the increasing use of ***Internet technology*** (***IT***). In contrast to the conventional stand-alone systems in product development, web-facilitated systems are set up to streamline the information sharing and exchange by allowing users to simultaneously access to distributed data and providing a virtual design environment for dispersed users. For this reason, the interest in this area is centralized on how the application of ***Internet technology*** can support the collaborative injection mold product development.

2.1 System approaches of applying the concurrent engineering philosophy in the injection mold design process

The ***Concurrent Engineering*** philosophy has been embraced in a wide range of engineering fields. The system approaches of ***Concurrent Engineering*** application in the injection mold product development are mostly adapted from these applications, and the focus is put on developing the appropriate information translation mechanism with process modeling tools to adapt to the process management requirement in the injection mold design and fabrication.

2.1.1 IDEF0 model application

IDEF0 (Integrated Definition for Function Modeling) model is a structural analysis and modeling technique specially designed to define the information translation in the decision-making and activity execution of a product development process. Generally, IDEF0 model denotes the product development process as a group of functional activities, each of which consists of four basic elements: input, output, control, and mechanism. Inputs to a functional activity are the sources an activity needs to process. Outputs are the results the activity generates. Controls to an activity are the constraints or conditions governing the performance of an activity function. Mechanisms are the means used to perform or the resources used to support the function requirement. Unlike the traditional sequential approaches like ***DIGRAPH*** or ***PERT,*** IDEF0 modeling hierarchically decomposes a general and abstract process into more specific and detailed activities by defining and interpreting the relationship among activities in the level of input and output, in turn, the interpreted relationships within a process help different parties involved in the product development to understand the potential areas where the concurrent and parallel works can be performed.

The representative works have been presented by Rong-Shean Lee et al. [Lee et al., 1997] and R. J. V. Lee et al. [Lee et al., 1998]. Rong-Shean Lee proposed a concurrent mold design process by applying the IDEF0 modeling. In their work, the injection mold design process is denoted as a group of process functional activities, such as cavity layout design, parting line determination, feed system design etc. The inputs, outputs, and controls among these activities are displayed in a matrix. To interpret this matrix, Chen used the mechanism that is based on calculating the value of the influence factor an activity possesses due to the influence of information input and output. The activity with a large influence factor indicates that it has the most influence on other activities, it is therefore more decisive in the process and able to be performed earlier. By this analysis, Chen identified the interaction between injection mold design activities, and rearranged the sequence of activities for concurrent injection mold design. The kernel in their work is defined as the information dependencies among injection mold activities, and the approach of IDEF0 model is centralized on interpreting the sophisticated relationships. However, Chen does not provide techniques to define the relationship between activities with equal influence factors. In the mean time, mold design has been considered retrospectively, the negotiation and feedback in mold design as the product geometry evolves are inevitable. However, this issue is not discussed in their work.

R. J. V. Lee et al. [Lee et al., 1998] also used the IDEF0 modeling methodologies to address the activities and information flow interactions in the injection mold design process and product function requirements. In their work, the injection mold design activities are categorized into three areas, each of which is composed of design activities that interact closely. These areas are the moldability analysis, the core, cavity and mold plate design, and the mold elements design that includes design jobs for the necessary mold assemble components. Based on this grouping, the injection mold design process in their work is considered being only the result of the interaction among these three areas. Upon this simplified activity structure model, Lee applied the clustering methodology on this model to integrate the product design function requirement with the mold design, and enable the concurrent interaction between product design and mold design (See Figure 2.1).

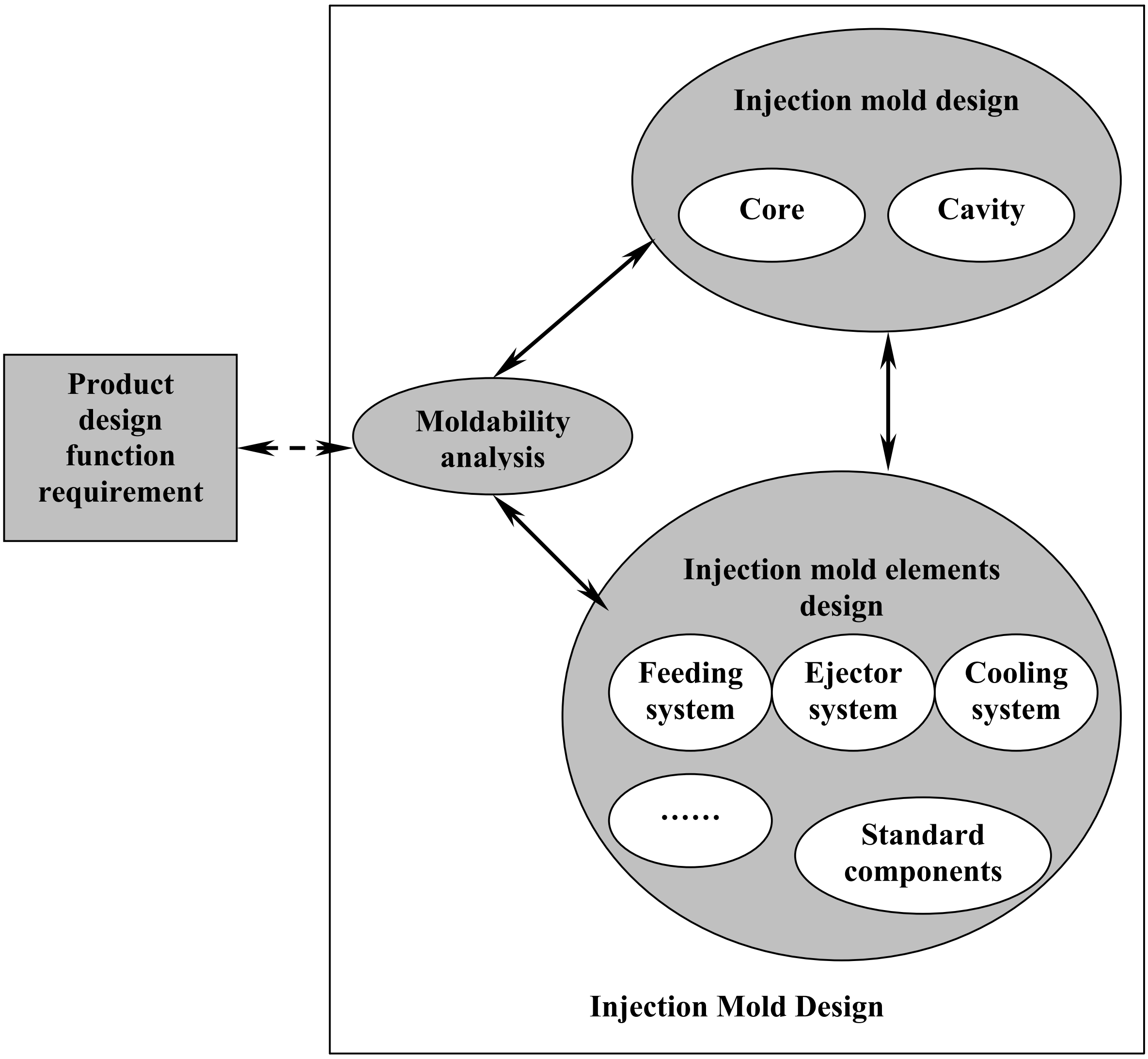


Figure 2.1 The simplified interaction between injection mold design activities

The work by Lee et al simplified the system analysis of injection mold design process by integrating the closely related activities into the same group, thus eliminated unnecessary information processing and reduced the intricacy of mold design process.

However, to some activities which are not integrated in the same group but depend on each other for necessary information, their work did not provide a solution to effectively interpret the relationship among these activities. Also, how to get rid of the influence of such activities on the entire process model are not discussed.

2.2 Setting up the computer-supported and web-facilitated collaborative framework for the injection mold product design process

Whilst the ***Concurrent Engineering*** philosophy is applied by the researchers and practitioners to keep competitiveness, research works have also been done in the area of web-facilitated collaborative framework to offer the concurrent support. Most of these research works are based on the concept of ***Computer-Supported Collaborative Work*** (***CSCE***). The ***CSCE*** is the mechanism that supports the work activities of networked groups working on the same product development or technical area [Monplaisir and Singh 2002]. Figure 2.2 presents the architecture of such an integrated computer system.

2.2.1 The computer-supported concurrent mold design system

Rong-Shen Lee et al. [Lee et al., 1997] proposed a knowledge-based computer framework for concurrent injection mold design. This framework consists the user interface, the object-oriented databases, and the knowledge-based mold development facilities. The knowledge-based mold development facilities are developed according to the characteristics of injection mold design and manufacturing. In their work, the knowledge is first abstracted from the practice, then the abstracted knowledge is transferred into design and manufacturing rules with the semantic description method. In the mean time, the knowledge is classified into a knowledge hierarchy that links the knowledge with the corresponding design or manufacturing activities. By doing this, the rational decision-making in mold design and manufacturing is supported. The data processed in their framework is managed in an object-oriented model. Data entities and their attributes involved in mold development are firstly identified from the information "resources" and the design and manufacturing rules. Then these data are analyzed and constructed in an object-oriented model that connects the data information with the corresponding design and manufacturing activities. In this way, the information or data that are needed in each design or manufacturing activity becomes available. The framework proposed by Lee theoretically integrates the mold design and fabrication processes with the necessary knowledge support and data management. Despite this, its applicability for distributed and collaborative injection mold product development is yet to be tested.

2.2.2 The web-facilitated collaborative framework for distributed injection mold design process

With regard to the web-facilitated collaborative framework for the injection mold design, the representative work is presented by Chung et al. [Chung et al., 2001]. Chung suggested a network-based framework for collaborative injection mold design evaluation. In their work, the framework is set up on the basis of the ***Common Object Request Broker Architecture*** (***CORBA***) client/server standard with the ***eXtensible Markup Language*** (***XML***). The ***XML*** format is primarily a data modeling language that is used to describe and pass the information among networked applications. Since the ***XML*** documents the information in a human-readable form, Chung utilized the ***XML*** as the information transfer standard to facilitate the data exchange among networked groups and parse the information. In their framework, the mold design evaluation is realized by integrating the ***Application Programming Interface*** (***API***), the design software tool-***Unigraphics***, the standard mold components databases, and the design evaluation criteria that is written in C language. Figure 2.2 shows the proposed collaborative architecture for distributed injection mold design.

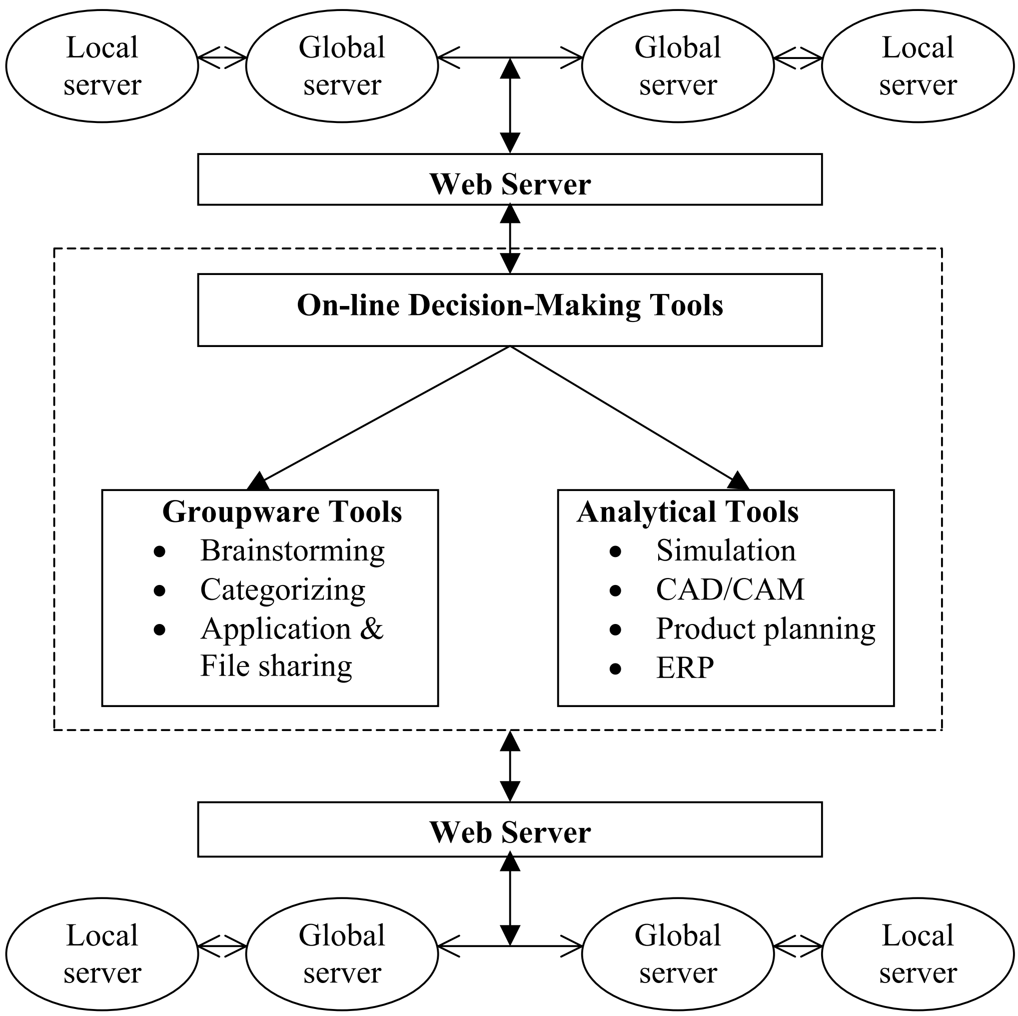


Figure 2.2 The architecture of the integrated CSCW and product development

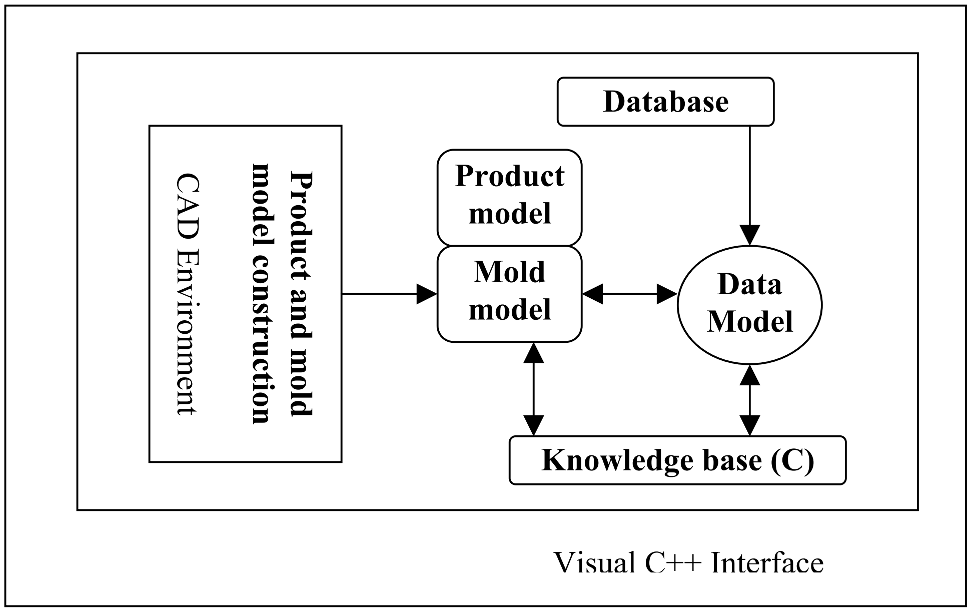


Figure 2.3 The concurrent system architecture

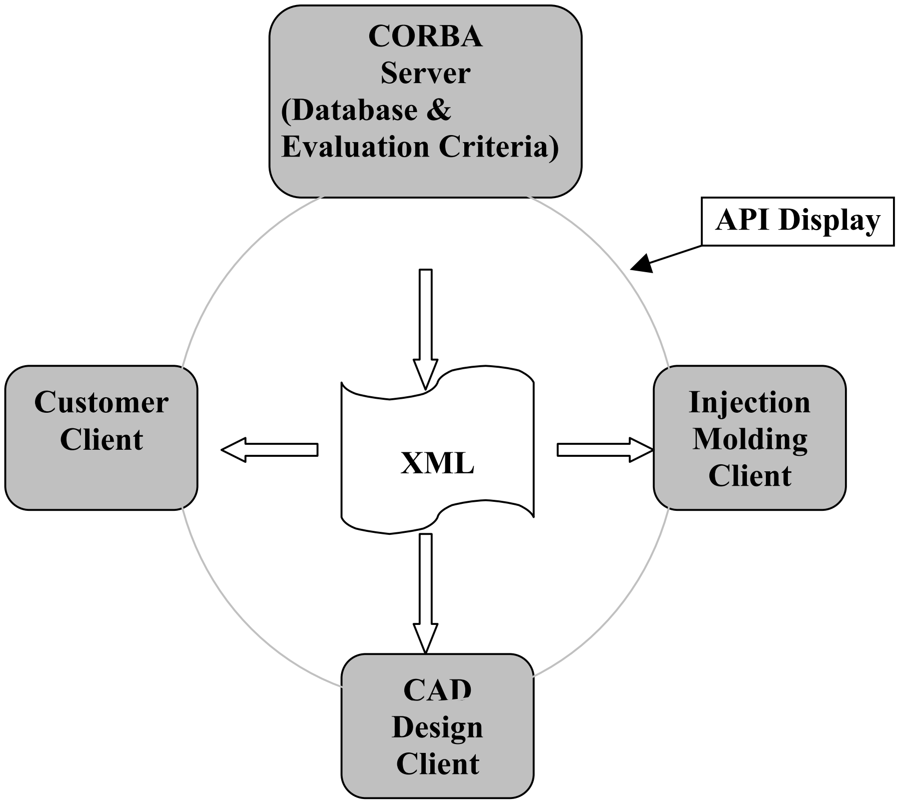


Figure 2.4 The collaborative framework for distributed injection mold design evaluation

The work by Chung et al provides a collaborative environment for distributed injection mold design process by utilizing the CORBA standard and XML language, both of which have been approved to be effective in improving the software interoperability and facilitating the networked data exchange. The concept illustrated in their work describes the feasibility of the design information propagation for real-time maintenance of design validity, and the knowledge-based decision-making in injection mold design process. However, this collaborative injection mold product development approach is basically for design evaluation, the collaboration and concurrent iteration among mold design activities as the product geometry evolves is not discussed. This overlook might result in redesign work in the injection mold design process.

2.3 Chapter summary

This chapter reviews pertinent research works for concurrent and collaborative injection mold design systems. The merits and demerits of these previous works are summarized. The review shows that the concurrent and collaborative injection mold system could be feasible through both the system approach and the implementation of ***Internet technology***. Although there are still limitations, these previous works provide a good references for this thesis.

CHAPTER 3

PROCESS MODELING METHOD USING THE DESIGN STRUCTURE MATRIX IN THE INJECTION MOLD DESIGN PROCESS

3.1 Introduction to the design structure matrix representation

This chapter describes the procedure to construct a matrix model for mapping information flows among process tasks using the method of design structure matrix (DSM). It also discusses the application of the procedure and DSM method in the injection mold design process. The DSM method discussed in this chapter uses the triangulation algorithm in the hierarchical decompositions of process activities or tasks, and identifies various types of information dependencies. Based on this information interpretation, the operation sequence of mold design process is rearranged, and the process road map is illustrated.

Steward's Design Structure Matrix has been developed and put into product design process mapping and modeling since the 1980s [Steward 1981a]. Now it is one of the most popular tools used in process modeling and re-engineering in product development process. Compared with other process modeling tools, DSM views the process from the perspective of components relationship and information flow for their functional requirement. Its philosophy is that a design project is divided into individual activities and the relationship among these activities can be analyzed to identify the underlying structure of the project. Normally, information flow is crucial and decisive in project activity analysis, decision-making, testing, process review, etc. DSM therefore provides a good means of displaying the process in the levels of details and revealing everyone involved in the decision-making the structure and semantic of the process.

The DSM deployed in this thesis is the activity-based DSM model. An activity-based DSM is a square matrix with identical rows and columns. Activities composing a design project are assigned to the matrix rows and their corresponding columns in a roughly chronological order (See Figure 3.1). The relationship or information dependency between activities is marked with either an “X” or a numerical value. The numerical value could be varied to represent the degree of dependencies. In reading the matrix, the marked matrix cell before the diagonal line indicates the forward information; the marked matrix cell after the diagonal line is referred as the feedback information. Take activity D in the matrix in Figure 3.1 for example, the forward information input that activity D receives is from activity A, and the feedback information input that it receives is from activity E and F.

Activity		1	2	3	4	5	6	7
A	1	A						
B	2		B		X			
C	3		X	C	X			
D	4	X			D	X	X	
E	5	X			X	E		
F	6						F	
G	7	X	X	X	X	X	X	G

Figure 3.1 A sample activity-based DSM

3.2 The procedure of constructing a process model for engineering projects with DSM

From the viewpoint of system engineering, it is necessary to identify and define the component activities and the information exchanged among these activities before the process structure can be represented to guide us understand the system semantic. In this section, the procedure of constructing a process model with the DSM method is presented in the following four steps:

(1) Define the process scope and decompose the process activities into the matrix representation.

(2) Use the triangulation algorithm of process partitioning and process tearing to classify the information dependency among activities and minimize information iteration.

(3) Map the information flow and rearrange the process to improve the process road map.

3.2.1 Decompose engineering process into manageable tasks/activities

Decomposing is the procedure that is used to divide a system process or engineering project into multiple component tasks and sub-activities for process modeling. The concept of decomposing stands out in Galbraith's definition of technology [Galbraith 1967]:

"Technology means the systematic application of scientific of other organized knowledge to practical tasks. Its most important consequence, at least for purposes of economics, is in forcing the division and subdivision of any such task into its component parts. Thus, and only thus, can organized knowledge be brought to bear on performance."

Simon [Simon 1973] stated the necessity of decomposing a project or process into sub-tasks:

"From the information-processing point of view, division of labor means factoring the total system of decisions that need to be made into relatively independent subsystems, each one of which can be designed with only minimal concern for its interactions with the others. The division is necessary because the processors that are available to organizations, whether humans or computers, are very limited in their processing capacity in comparison with the magnitude of decision problems that organizations face. The number of alternatives that can be considered, the intricacy of the chains of consequences that can be traced – all these are severely restricted by the limited capacities of the available processors."

It is understandable that there might be some adverse effects on the process modeling and the process operation if the process is ill defined and the information dependencies among tasks are not fully explored. These adverse effects may lead to extended product life cycle, low efficiency, inaccurate decisions, and decreased flexibility of teams or groups etc. For these reasons, it is necessary and important to divide a process or project into a set of appropriate information levels and an understandable scope.

From above perspectives, a successful decomposition should be able to achieve the following results in process modeling:

- The process is decomposed in a clear way for everyone involved to understand the process scope.
- The relationships between process activities or sub-activities are clearly represented.
- The process can be displayed and represented for further process modeling operation.
- The process activities are measurable and tractable.

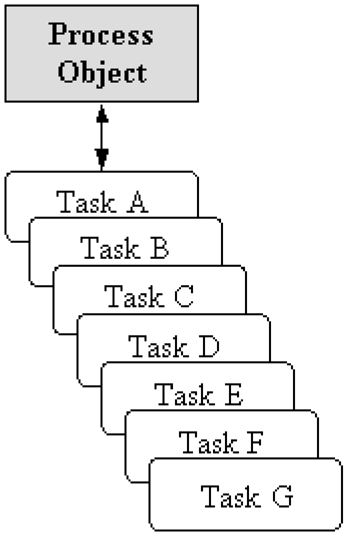
Take the design project displayed in Figure 3.1 for example, the process decomposition is performed in the following steps (See Figure 3.2):

1. Identify the scope of a process and establish an overview about the work content by defining the activities or tasks and their deliverables.
2. Represent the initial structure of the process with a ***GRAPH***.

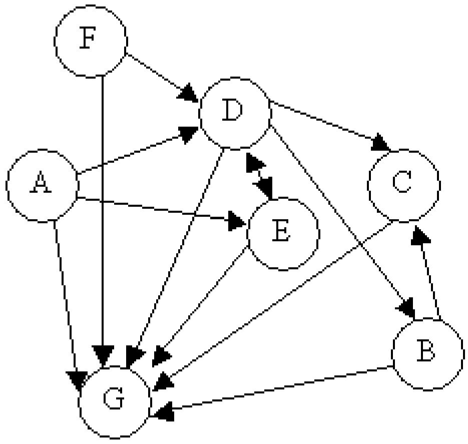
3. Determine a list of information or parameters that are needed to generate each activity deliverable.
4. Create the corresponding precedence matrix.

The ***GRAPH*** theory is the fundamental and well-established technique used for representing the structure of a system or process [Crirca 1736]. A graph is composed of set of vertices and arcs. The vertices represent the process activity; the arcs are drawn from one vertex to another vertex that it affects. The direct information effect between two vertices is shown by one arc. The indirect information effect among two or more vertices is shown by a path, which is a sequence of one or more arcs from vertex to vertex along the direction of arrows. Normally, the number of arcs within a path is defined by this length, which explains how complicated the indirect information effect could be.

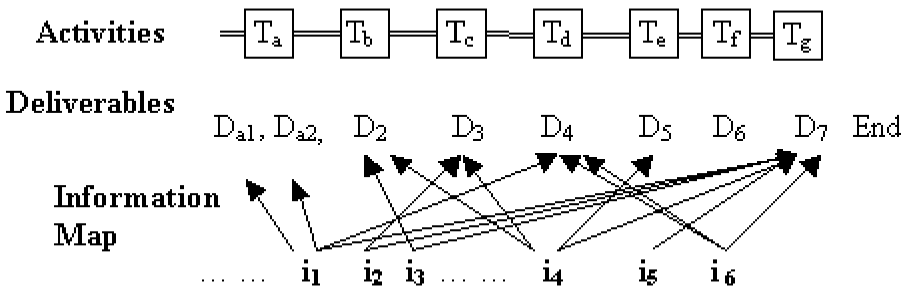
A graph provides an intuitive display of a process. However, it has limitation in representing and parsing the process into the level of information details. DSM is utilized for further process representation and interpretation.



(a). A Process Object



(b). A Graph Representation



(c). The Information Map

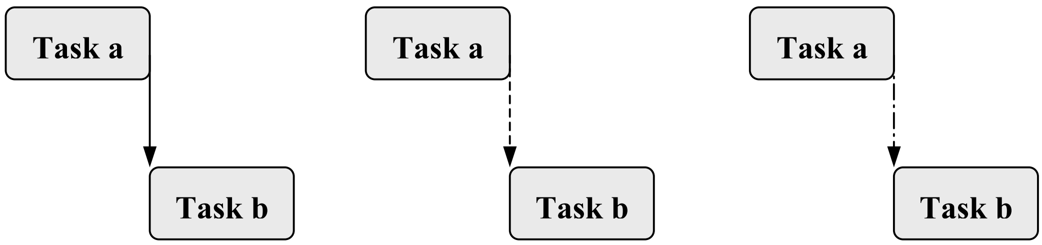
Activity		1	2	3	4	5	6	7
A	1	A						
B	2		B		X			
C	3		X	C	X			
D	4	X			D	X	X	
E	5	X			X	E		
F	6						F	
G	7	X	X	X	X	X	X	G

(d). The Corresponding Matrix Representation

Figure 3.2 An example of the process decomposition

According to the concept of information processing, the information a design activity receives has various effects on the result an activity could pass to subsequent tasks. Based on this understanding, the information types that are identified in this research are defined by the weight of dependency of the information being transferred.

In the DSM representation adopted in this thesis, three levels of numerical marks are defined to represent the various information dependencies. The three levels are: #1 for the high information dependency, #2 for medium dependency, and #3 for low or weak information dependency. The advantage of identifying the different information dependency and using numerical value to describe lies in the necessity of identifying the information input that is either predicable or has little impact on the subsequent tasks at the stage of DSM process tearing. The information input that has predicable effect on the subsequent tasks does not increase the intricacy or uncertainty of a process since the results are certain and predictable. Such information dependency could be eliminated, as it does not contribute to the efficiency of the process. Figure 3.3 illustrates the three levels of information dependencies among individual design activities. The different types of arcs, which indicate the different strength of information dependency a task as defined, describes the information processed in a task execution with level of details. This more detailed description of information relationship provides a better understanding of the relationship between tasks and the entire process. Based on this understanding, the process represented in Figure 3.2 can be displayed as a numerical matrix in Figure 3.4.



(a) High dependency (b) Medium dependency (c) Low dependency

Figure 3.3 Types of information flow between tasks/activities

Activity		1	2	3	4	5	6	7
A	1	A						
B	2		B		1			
C	3		2	C	2			
D	4	2			D	1	1	
E	5	3			2	E		
F	6						F	
G	7	2	1	2	1	2	3	G

Figure 3.4 The numerical matrix representation of the process decomposition

After the information dependency is defined, the process modeling is performed. We break the process modeling into two phases. Phase one is called process partitioning. Phase two is process tearing. Process partitioning is a process of reordering process tasks after a process is decomposed, so that information-flow marks in the matrix are placed in the lower diagonal or grouped together within square blocks on the diagonal (Steward 1981). The main aim of process partitioning is that the new DSM arrangement has fewer system elements involved in the iteration cycle that could result in a faster development process. The process tearing is used to remove the feedback information from the iteration cycle or circuit, and determine how the tasks should be relatively ordered with it. The main purpose of process tearing is to break the iteration and reduce its influence on the entire process. For the numerical DSM representation, process tearing aims to get only the highest number of marks above the diagonal line. Since the highest marks define information that has predictable effect on design activities, it is reasonable to conclude that these marks will not increase the intricacy of the entire design project if the semantic represented by these marks could be well understood. It is still possible that there are still some low-level feedback marks left in the matrix even after the process tearing. In this situation, the assumption or estimate that is based on the semantic of interactive activities and the requirements for information become necessary to loosen the information loop.

For the numerical DSM representation, Steward (1981) has suggested the following procedure to execute the process partitioning and the process tearing.

1. Partition the whole matrix with all the marks being treated equally to get the operation sequence of the entire design project.

2. For the information blocks or iterations, the mark with the highest-level number will be torn first and the matrix is reordered again. This process is repeated until all feedback marks disappear so that the sum of the dependency strengths above the diagonal line can be minimized.

3. Reorganize the process sequence and draft the process map.

Both of the process partitioning and process tearing discussed in this thesis are based on the triangulation algorithm developed by Kusiak et al. [Kusiak et al., 1994, Belhe et al., 1995]. We also define three numerical levels to display the numerical matrix. Improvements are made to get this modeling technique fit into the injection mold product design process.

3.2.2 Use the triangulation algorithm of process partitioning to classify the activity relationships

Kusiak et al. have discovered that most systems or projects are too information-intensive. This fact has resulted in the long duration of performing tasks. Kusiak's triangulation algorithm therefore attempts to solve this problem by eliminating the controls. Typically, a control is a piece of information that is required to perform the activity. By doing this, the volume of information processing can be reduced and the duration of tasks can be reduced.

Generally speaking, Kusiak's triangulation algorithm is carried out along the following procedure:

- Decompose tasks

  In many cases, a single task can be partitioned into multiple concurrent sub-tasks. These parallel tasks involve less information requirement and iteration, and thus shorten the overall duration of process. Kusiak's triangulation algorithm aims to identify such parallel tasks and define the information requirements of these tasks.

- Combine the serial tasks

The serial tasks refer to tasks that are independent of each other for information, or have no information requirement from other tasks in a process. To shorten the overall duration of a process, the rearrangement of the serial tasks therefore becomes necessary. Kusiak's triangulation algorithm identifies these serial tasks and combines the sequence of the tasks in process operation. By doing this, the work performed in two different areas may be performed at the same time, and thus shorten the duration of tasks.

- Identify the information cycles and reduce the volume of information involved

  In information processing of a process or project, tasks can get entangled or performed back and forth because of information cycles or loops between them. The breaking of information loops and streamlining of tasks are often the main purpose in process modeling and structuring. Kusiak's triangulation algorithm solves this problem by focusing on the elimination of many smaller and shorter information cycles.

To present this algorithm, the terminology defined in it is first introduced. As shown in Table 3.1, an activity is called the origin activity (*OA*) if there is no other activity preceding it. Simply put, an *OA* activity does not rely on other activities for information. Such activities can be identified by observing the empty row (except the diagonal cell) in the matrix. For example, if t_i has only one non-empty element (a diagonal element) on its row, then t_i is an *OA*. Normally, an OA activity can be performed at the beginning level of a design project. An activity is called the succeeding activity if it delivers no information to other activities. Such activities can be identified by observing the empty column (except the diagonal cell) on the matrix, and assigned to the end level of a design project. Besides these two types of activities, this algorithm also defines the circuit or cycle for activities that are interrelated, and arranges the sequence of these activities according to the interpretation of information iteration.

Figure 3.5 shows the logic of the process partitioning based on Kusiak's triangulation algorithm. To better illustrate this process partitioning, the sample matrix in Figure 3.4 is used as the example.

Table 3.1 The terminology of the triangulation algorithm

Terminology and Symbols	Interpretation
A	The Corresponding Matrix of a design project $A=[a_{ij}]_{n \times n}$
t	A component activity or task involved in a process
OA	Set of $OA = \{t \mid t$ is an activity without other preceding activities$\}$
SA	Set of $SA = \{t \mid t$ is an activity without other succeeding activities$\}$
E	Set $E = \{t \mid t$ is an activity involved in an information cycle$\}$
L	Set $L = \{L(i) \mid$set of activities in level number $i\}$
C	Set $C = \{C(j) \mid$set of coupled activities in group number $j\}$
$O(k)$	Activity Level Number $O(k)$, for $k = 1, \ldots, n$

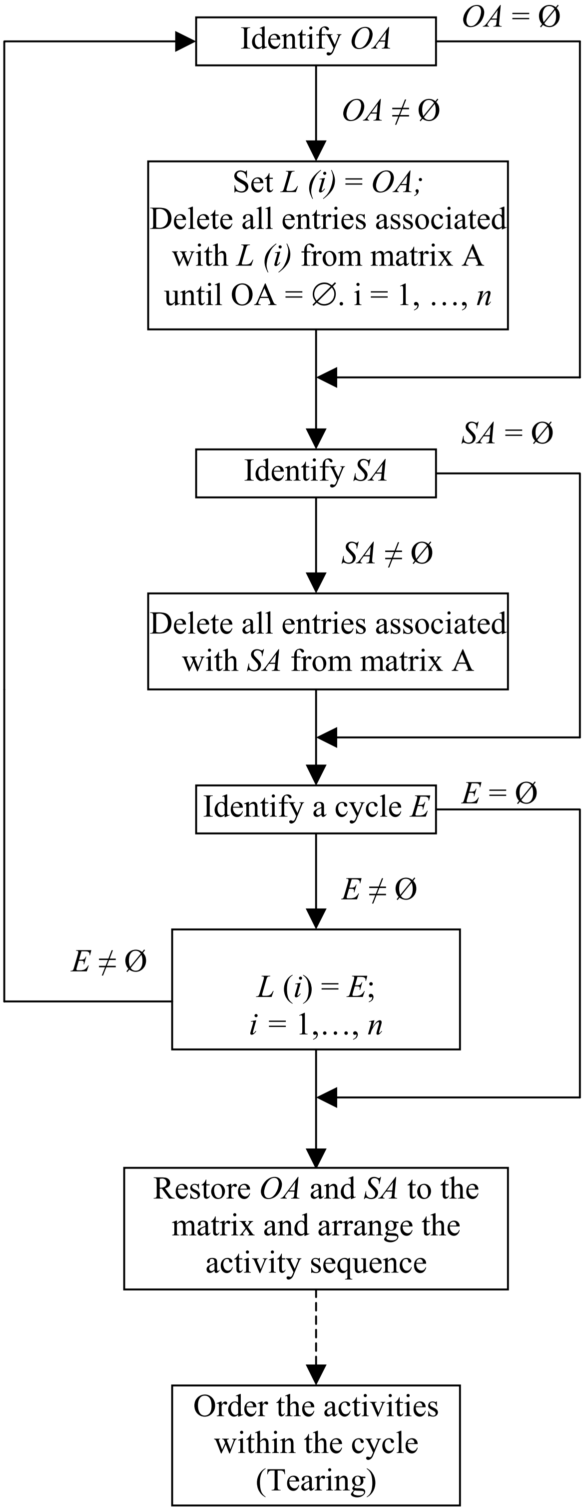


Figure 3.5 The logic of the triangulation algorithm for process partitioning

Based on the logic shown in Figure 3.5, the process partitioning in Figure 3.3 can be performed as follows [Kusiak et al., 1994].

Step 1. Set $i \leftarrow 1$, $j \leftarrow 1$, $L(1) \leftarrow \varnothing$, $C(1) \leftarrow \varnothing$, $OA \leftarrow \varnothing$, $E \leftarrow \varnothing$, and $O(k) \leftarrow 1$ for k = 1, .., 7. (See Figure 3.6 a)

Step 2. Since $A \neq \varnothing$, go to Step 3.

Step 3. $OA \leftarrow \{1, 6\}$; since $OA \neq \varnothing$ and $\{1, 6\} \notin C$, set $L(1) \leftarrow \{1, 6\}$and go to Step 4.

Step 4. Delete all entries associated with activities 1 and 6 from matrix A; set $O(4) = O(5) = O(7) = 2$; since $L(1) \neq \varnothing$, set $i \leftarrow 2$ and $L(2) \leftarrow \varnothing$; $OA \leftarrow \varnothing$; go to Step 2. (See Figure 3.6 b)

Step 2. Since $A \neq \varnothing$, go to Step 3.

Step 3. Since $OA = \varnothing$, go to Step 5.

Step 5. Since $SA \leftarrow \{7\}$; Delete all entries associated with activity 7 from matrix A; go to Step 6. (See Figure 3.6 c).

Step 6. Find a cycle {4, 5, 4}. Set E = {4, 5}, and go to Step 7.

Step 7. Since $C \cap E = \varnothing$, set $C(1) \leftarrow \{4, 5\}$, set $j \leftarrow 2$, $C(j) \leftarrow \varnothing$; $O(C(1)) \leftarrow \text{Max}\{O(4), O(5)\} = 2$; set $E \leftarrow \varnothing$ and go to Step 2.(See Figure 3.6 c)

Step 2. Since $A \neq \varnothing$, go to Step 3.

Step 3. $OA \leftarrow \{2\}$; since $OA \neq \varnothing$ and $\{2\} \notin C$, set $L(3) \leftarrow \{2\}$and go to Step 4.

Step 4. Delete all entries associated with activity 2 from matrix A; set $O(3)$ =4; since $L(3) \neq \varnothing$, set $i \leftarrow 4$ and $L(4) \leftarrow \varnothing$; $OA \leftarrow \varnothing$; go to Step 2.(See Figure 3.6 d)

Step 2. Since $A \neq \varnothing$, go to Step 3.

Step 3. $OA \leftarrow \{3\}$; since $OA \neq \varnothing$ and $\{3\} \notin C$, set $L(4) \leftarrow \{3\}$and go to Step 4.

Step 4. Delete all entries associated with activities 3 from matrix A; since $L(4) \neq \varnothing$, set $i \leftarrow 5$ and $L(5) \leftarrow \varnothing$; $OA \leftarrow \varnothing$; go to Step 2.(See Figure 3.6 e)

Step 2. Since $A = \varnothing$, go to Step 8.

Step 8. Restore the OA and SA and E to the matrix A, arrange the sequence according to the level order.

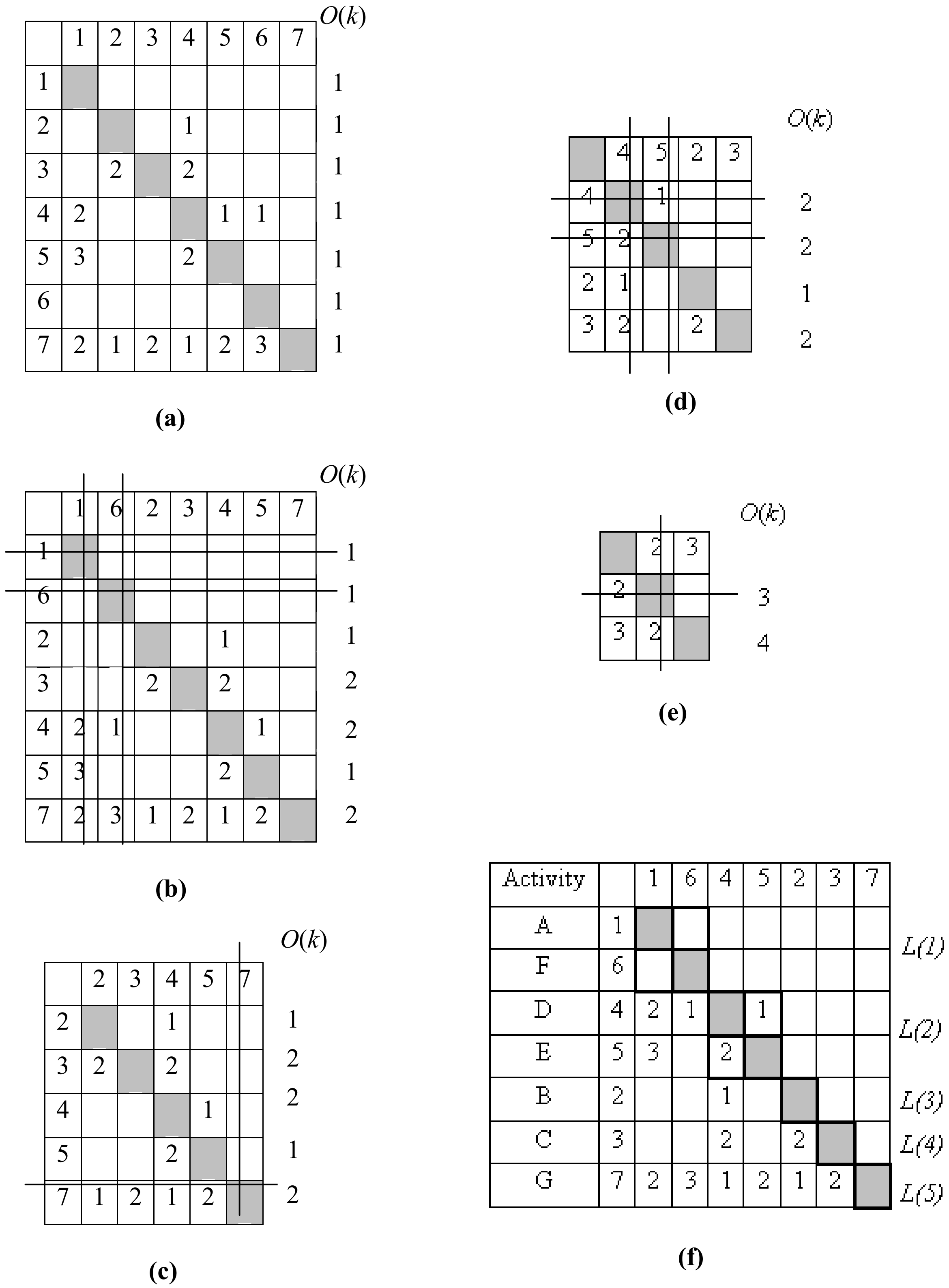


Figure 3.6 The result of process partitioning based on the triangulation algorithm

Basically, three types of information dependencies between activities can be obtained after the process partitioning. With reference to Figure 3.6 d, there are the relationship of parallel or independency between activities, such as activities A and F; the relationship of coupled or interacted activities, such as activities D and E; and the relationship of sequential or dependent activities, such as activities B, C, and G. In process operation, activities that are in parallel relationship could be executed at the same time. Activities that are coupled should be performed simultaneously since they rely on each other for information. Assumptions might be necessary if there are too many uncertainties involved in their information requirements. Regarding activities that are in the relationship of sequence, the leading activity should be performed in the process as early as possible to enable a quick start of the following activities.

Activity		1	6	4	5	2	3	7
A	1							
F	6							
D	4	2	1		1			
E	5	3		2				
B	2			1				
C	3			2		2		
G	7	2	3	1	2	1	2	

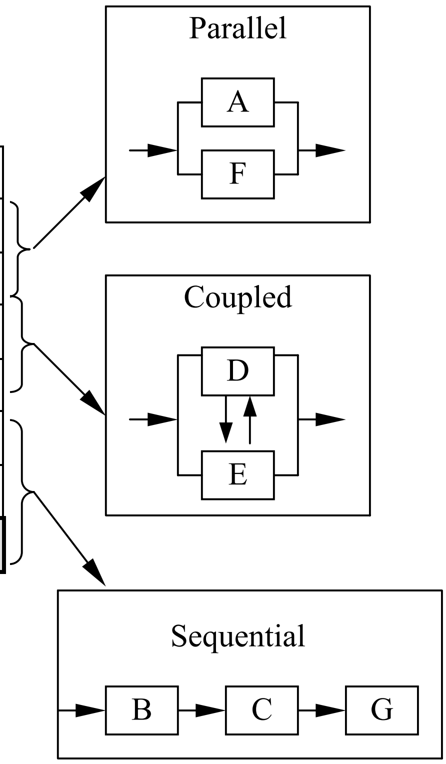


Figure 3.7 Types of information dependencies between activities in matrix interpretation

Although the process partitioning arranges the design project activities into a more manageable sequence, it does not say anything about the relative ordering of activities that are coupled or in information circuit. More importantly, it is essential to break the iteration among coupled activities before a complete process map could be drafted. The process tearing is therefore deployed for further process interpretation.

3.2.3 Use the triangulation algorithm of process tearing to eliminate the information circuits and iteration

Tearing is the process used to remove the tear arcs-the set of feedback marks, and reorder the activities within the information circuits. The main purpose of process tearing is to break the iteration and streamline the information flow in process operation. Generally, the process tearing is performed according to the following principle [Steward 1981b]:

"We could remove (tear) arcs where our knowledge of the semantics indicates that the arcs represent acceptable feedbacks, and where our knowledge of the structure indicates that the removal of the arc would be effective in breaking circuits".

From the above perspective, we can find the difference between the process partitioning and the process tearing. To partition a design project, we need to consider only the graph and the representation matrix. But to tear the circuits and order the activities within the circuits, we must use our understanding of the semantics of the design activities as well as the design project structure.

According to Steward (1981), the logic of tearing the information circuit in numerical matrix is shown in Figure 3.8.

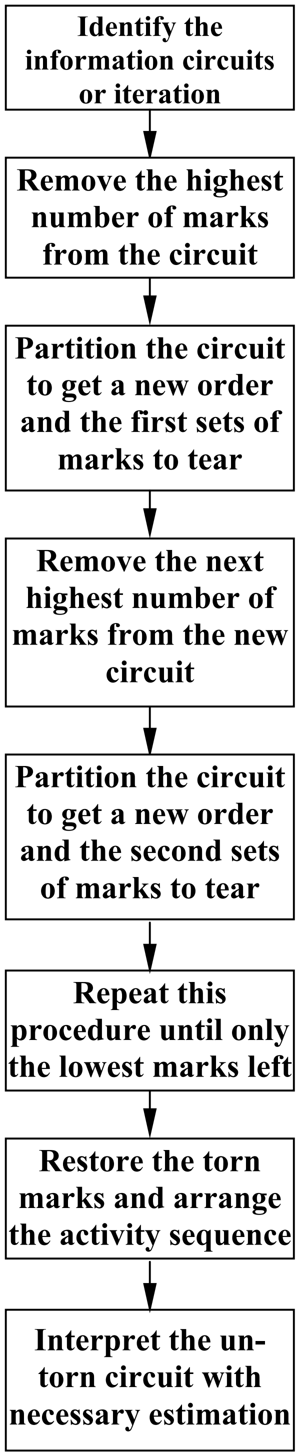


Figure 3.8 The logic of the process tearing

Based on this logic, given an activity iteration, the process tearing can be performed in the following steps (See Figure 3. 8)

	1	4	5	6	10	12
1						1
4				2		
5				1		
6	1		1			3
10		2	3	2		
12					3	

(a)

	12	1	4	5	6	10
12						
1	1					
4						
5					1	
6		1		1		
10						

(b)

	12	4	10	1	5	6	*O*(*k*)
12							
4							*L*(1)
10							
1	1						*L*(2)
5						1	
6				1	1		*L*(3)

(c)

	12	4	10	1	5	6	*O*(*k*)
12			3				
4						2	*L*(1)
10		2			3	2	
1	1						*L*(2)
5						1	
6	3			1	1		*L*(3)

(d)

Figure 3.9 The process tearing and the results

The process tearing shown in Figure 3.9 set up an ordering for activities within the circuit. Compared with the activity sequence before tearing, the new activity order reduces the feedback marks on the iteration. By doing this, the influence of high feedback dependency is eliminated. What's more, tearing gives suggestions on the tear set –the most decisive feedback that can be worked to simplify the information circuit.

As shown in Figure 3.9 d, activities 5 and 6 are the tear set for this circuit. These two activities affect each other greatly, also sent tolerable feedback information to activities like 10 and 4. To break such tears, we also need to be knowledgeable of the semantics and consider what it would mean to the design process to make the estimates these tears represent. Besides this, considering the fact that the execution of highly coupled activities within the circuit is often the team-efforts in information processing and information translation, the demands for hardware facilities that can facilitate the efficient information transfer in project management is therefore increasing.

3.3 DSM application in the injection mold design process

This section describes the application of DSM process partitioning and tearing in the injection mold design process [Du, XJ et al., 2002]. Upon the understanding of the semantic of injection mold design and the DSM representation methodology, the injection mold design process is first decomposed into multiple sub-activities with the information dependency being graphed. Then the matrix is used to represent the structure of the injection mold design process according to the information dependency. After that, the process partitioning and tearing algorithms are deployed to rearrange the injection mold design and structure the process model.

Figure 3.10 demonstrates the graph of the injection mold design activities and their interrelated conditions. The numerical matrix representation for the injection mold design process can be obtained in Figure 3.11. Based on the DSM process partitioning and process tearing (see Appendix B), the rearranged process sequence of injection mold design could be obtained in Figure 3. 12.

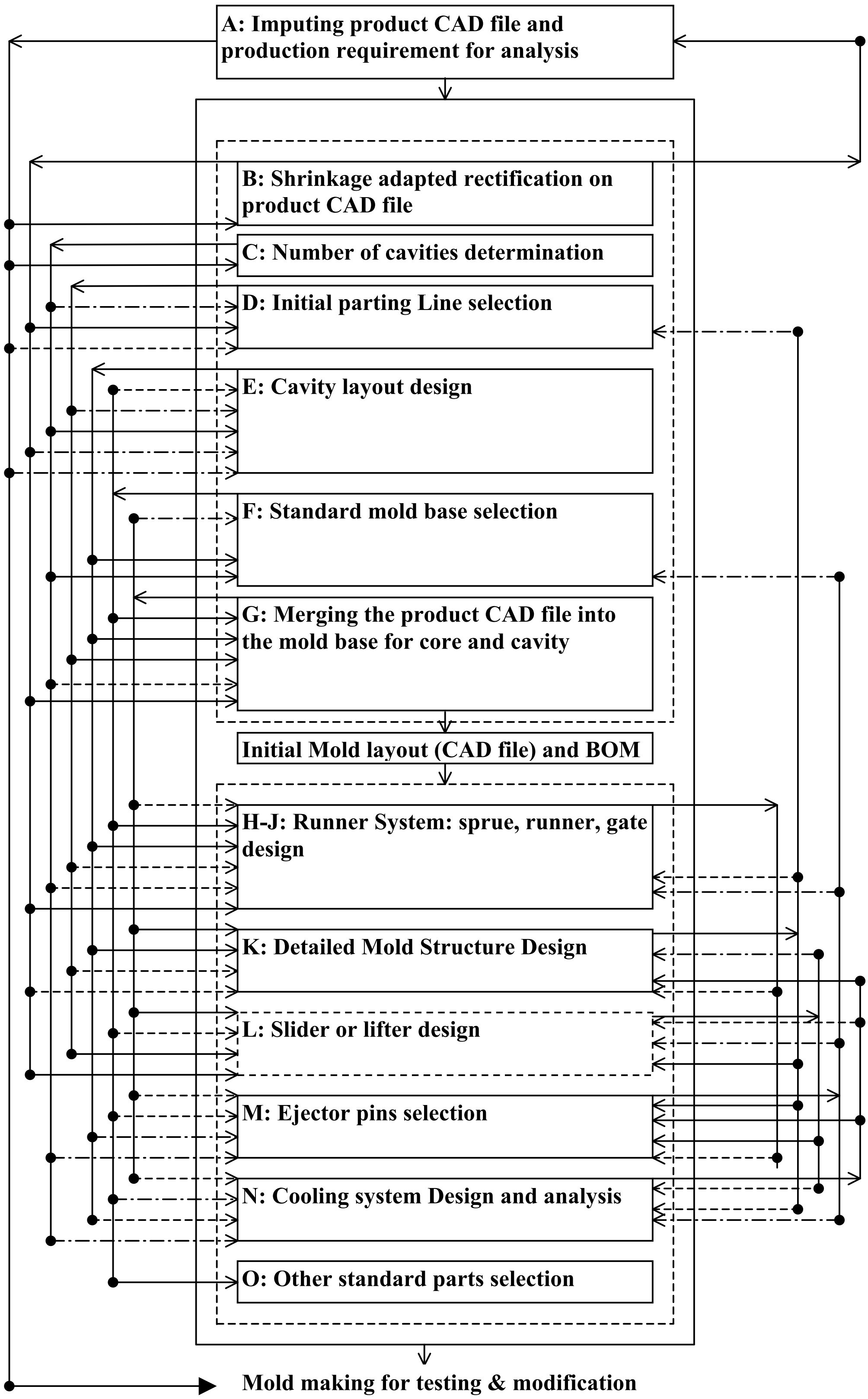


Figure 3. 10 The graph representation of an injection mold design project

Injection Mold Design Activities		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	1		1													
B	2	1														
C	3	1														
D	4	2	1	1												
E	5	3	3	1	3		2									
F	6		1	1		1								3		
G	7		1	2	1	1	1									
H	8		1				1			3	2			3		
I	9	2	1	3	2	1	1	2	1		1	2		3		
J	10		2	2	2	1	3		2	1		2		3		
K	11		2		2	1		1	2	1	3		3		1	
L	12		1		3		1					1		3	2	
M	13			3		3	1	2	2	2	2		1		1	
N	14			3		1	1		3			2	2	3		
O	15						1									

Figure 3.11 The numerical matrix representation of an injection mold design project

Injection Mold Design Activities		1	2	3	4	5	6	7	8	14	9	10	11	12	13	15	Activity level	
A	1		1														$L(1)$	
B	2	1																
C	3	1															$L(2)$	
D	4	2	1	1													$L(3)$	
E	5	3	3	1			2										$L(4_1)$	
F	6		1	1		1									3		$L(4_2)$	
G	7		1	2	1	1	1										$L(4_3)$	
H	8		1				1				3	2			3		$L(4_3)$	
N	14		1	3		1	1		3				2	2	3		$L(4_3)$	$L(4)$
I	9	2	2	2		1	1	2	1			1	2		3		$L(4_4)$	
J	10		2			1	3		2		1		2		3			
K	11		1	3		1		1	2	1	1	3		3			$L(4_5)$	
L	12			3			1			2			1		3		$L(4_6)$	
M	13					3	1	2	2	1	2	2	1	1			$L(4_7)$	
O	15						1										$L(5)$	

Figure 3.12 The process sequence of the injection mold design after the DSM operation

Necessary estimation and assumption based on information available are needed to simplify the information circuit. It is noticed that a new ordering of activities involved in the information circuit is suggested after the process tearing. The big and single circuit is divided into multiple information subsets with the smaller level of numbers eliminated and feedback marks confined to the smaller block. However, except activities which are closely interacted and could not be torn because of the significant information dependency, some feedback restrictions due to weak or minor information dependencies still exist above the diagonal line in the matrix. As defined, a mark above the diagonal indicates where an assumption or estimate must be used for the predecessor activity in DSM interpretation, we would generally consider that it is reasonable to think less of the low-level marks if the likely error in the assumption or estimate of the predecessor represented by the mark would not have a significant effect on the task it interacts. That is, the task that interacts with minor or weak information influence is insensitive to the error in the estimate made. With the DSM operated injection mold design process, we apply some necessary assumptions and estimates to minimize the impact of such weak or minor information dependencies among activities, in the mean time, we look into the semantics of activities that interact with significant information exchange, and understand the request of each discipline group involved, so that the information processing within these activities could be streamlined.

As suggested in Figure 3.12, the injection mold design process can be performed along a five-stage sequence. The first stage contains two activities, importing the product CAD file and shrinkage adapted rectification on the product CAD file, involving two key collaborating parties: the product designer and the mold designer. These two parties iterate to obtain the appropriate product moldability. The information loop at this stage is normally the result of less experienced mold designers who misunderstand the customers' requirements. To avoid this situation and to loosen the information loop, mold designers need to take external information into account beside the information from customers, such information include injection molding machine availability, operation requirements, and material handling specification. This means that relevant information from mold makers and suppliers is necessary. Their active participations could help mold designers to make the decision more rapidly and accurately.

Stage 2 contains the activity of cavity number determination, and stage 3 contains the activity of initial parting-line selection. These two activities are serial, and decisions can be made after their required preceding activities are finished. The parties that need to collaborate at these stages include the mold designer, the mold maker and the standard mold base suppliers. The mold designer needs to know the dimension and cost of standard mold bases before he or she can determine the most appropriate number of cavity. That is why the early participation of the standard mold base suppliers is necessary at this stage. The involvement of the mold maker at the stage 3 is the need to increase the accuracy in parting line selection. For unskilled mold designers, experience and suggestion of mold makers are helpful to streamline this task.

The forth stage which includes the mold layout design, mold base selection, and some mold components design which contains activities that form a big and single information block, involving customers, mold designers, mold makers, and standard mold components suppliers. The DSM tearing suggests an ordering for these entangled activities. The first two phases within this block include a small iteration between activities of cavity layout design and standard mold base selection. Based on our understanding of the semantics of these two activities, the mold base is usually determined after the cavity number and cavity layout are defined. However, since the standardization of mold base and components has given rise to shorter lead-time and saved cost saving in mold fabrication, standard mold base and components are now widely adopted in injection mold design. Sometimes, mold designers would have to accommodate their designs to existing standard mold bases because of cost reasons. That is why standard mold base information could be a decisive factor in cavity layout design and mold base selection, and there is information iteration between these two activities. The possible solution to streamline the decision-making at this stage is to ensure the timely updating of standard mold base and components information and increasing the accuracy of mold cavity number and layout design. The first could be achieved by better managing the information from the mold base suppliers. The latter requires the mold designer's thorough understanding of customer's requirements and purchase order.

The third phase within the information block includes 3 activities that could be performed at the same time. These are the activity of merging the product CAD file into the mold base, the activity of sprue design, and the activity of cooling system analysis and design. According to the semantics of injection mold design, these three activities are independent of one another for information. Therefore these three activities could be performed simultaneously. In the mean time, we need to bear in mind that the sprue design is sometimes affected by the size of runner and gate design, and the ejection pin selection. This is because that the sprue functions as the channel to guide the plasticated material to runner and gates. Therefore the sprue designed must ensure sufficient transmission of holding pressure in runner. Practitioners usually choose semi finished or finished sprues that are manufactured according to mold standards to simplify the runner system design. In our research, we assume that the sprue needed in an injection mold project is based on the information provided by standard components suppliers; the interaction between sprue design and ejection pin selection is often due to the estate conflict when the sprue and ejection pins need to be positioned. This conflict can be smoothed out as long as the mold base and sprue bushing are defined on the basis of the information provided by suppliers.

The task of cooling system design at phase 3 is relatively one of the most important tasks in the injection mold design. It has a definite effect on the economics of the process and the quality of the final articles. In cooling system design, mold designers need to consider many factors like material density, cooling time, wall thickness, thermal conductivity, distance, cooling time, etc. Most of these factors can affect the efficiency of cooling design to different extends. From the perspective of information iteration, downstream activities like detailed mold structure design, slider or lifter design, and ejection pin selection could cause the redesign in cooling channel layout, dimension, and positioning. To avoid the redesign work and reduce the information interaction, mold designers have to make necessary compromises without sacrificing the efficiency of heat transfer rate and cooling time. For this reason, we make necessary assumptions for the cooling system design.

1. The material used for injection molding is confirmed before the cooling system design starts. Since most of factors used in cooling design such as thermal diffusivity are related to the material itself, the assumption is therefore made

with the purpose to eliminate unnecessary uncertainties due to material characteristics.

2. Necessary cooling analysis should be conducted before the cooling system design starts. The development of computer-aided engineering has made it possible to simulate the cooling stage during mold design. This simulation helps mold designer to do a more accurate thermal design, and thus increases the accuracy of cooling design.
3. The cooling channel layout is greatly affected by the detailed mold structure, wall thickness and etc. Therefore, it is necessary to assume that the modification on the product structure could be delivered to the downstream activities without unnecessary delay due to the information transferring. This assumption also defines that the information iteration from the perspective of system engineering is restricted to the semantics of the design process.
4. The main iteration between the cooling system design and the ejection pin selection is the estate conflict in positioning the cooling channel and ejection pins. Practitioners usually solve this problem by giving priorities to the efficiency of cooling system. That is why sometimes the ejection pins are positioned to adjust to the cooling channel layout. On the other hand, ejection pins have to be positioned to ensure the releasing of molded part without difficulty. For this reason, compromises are often necessary when mold design comes to this stage.

The fourth phase within the information block contains activities of runner design and gates design. The runner system to be used can be either a hot runner system or a cold runner system. The appropriate hot runner system design can be determined with close collaboration among the mold designer, the mold maker and the standard hot runner supplier. The cold runner system design is an inter-related process. The designs of the gate and the runner closely depend on each other for information. In the mean time, the runner and the gate design can be affected by the detailed mold structure design and the ejection pin position. These feedback information influences can be minimized with the accuracy of product structure design being increased.

Phase 5, 6 and 7 within the information block contain activities of detailed mold structure design, slider or lifter design and ejection pins selection respectively. A minor information loop is still seen between these activities. The Slider and the lifter are mechanical components used for releasing molded products with undercuts. Since the slider or lifter design complicates the molding procedure and the mold structure, it results in higher cost. However, if sliders or lifters cannot be avoided, they are designed together with detailed core and cavity insert because the detailed mold structure determines whether the slider or lifter should be used in releasing the molded part, and the position and size of appropriate slider or lifter if it has to be used. The ejection pin positioning at phase 7 might change the position of slider or lifter that had been defined in the previous phase. However, this feedback influence is not decisive in slider or lifter design, and its effect can often be eliminated if the detailed mold structure design is accurate and confirmed. The final stage in the injection mold design is the standard mold components selection, such as screws and pins etc. This decision is usually made after the mold base is selected and the mold component designs are finished, so that the most appropriate standard mold components can be selected.

Based on the DSM analysis, it can be seen that injection mold design process is a complicated process that is highly relied on the active participation of different collaborating groups. For this reason, it is necessary to set up a concurrent and collaborative process map where the participation of different groups is clearly defined and illustrated (see Figure 3.13).

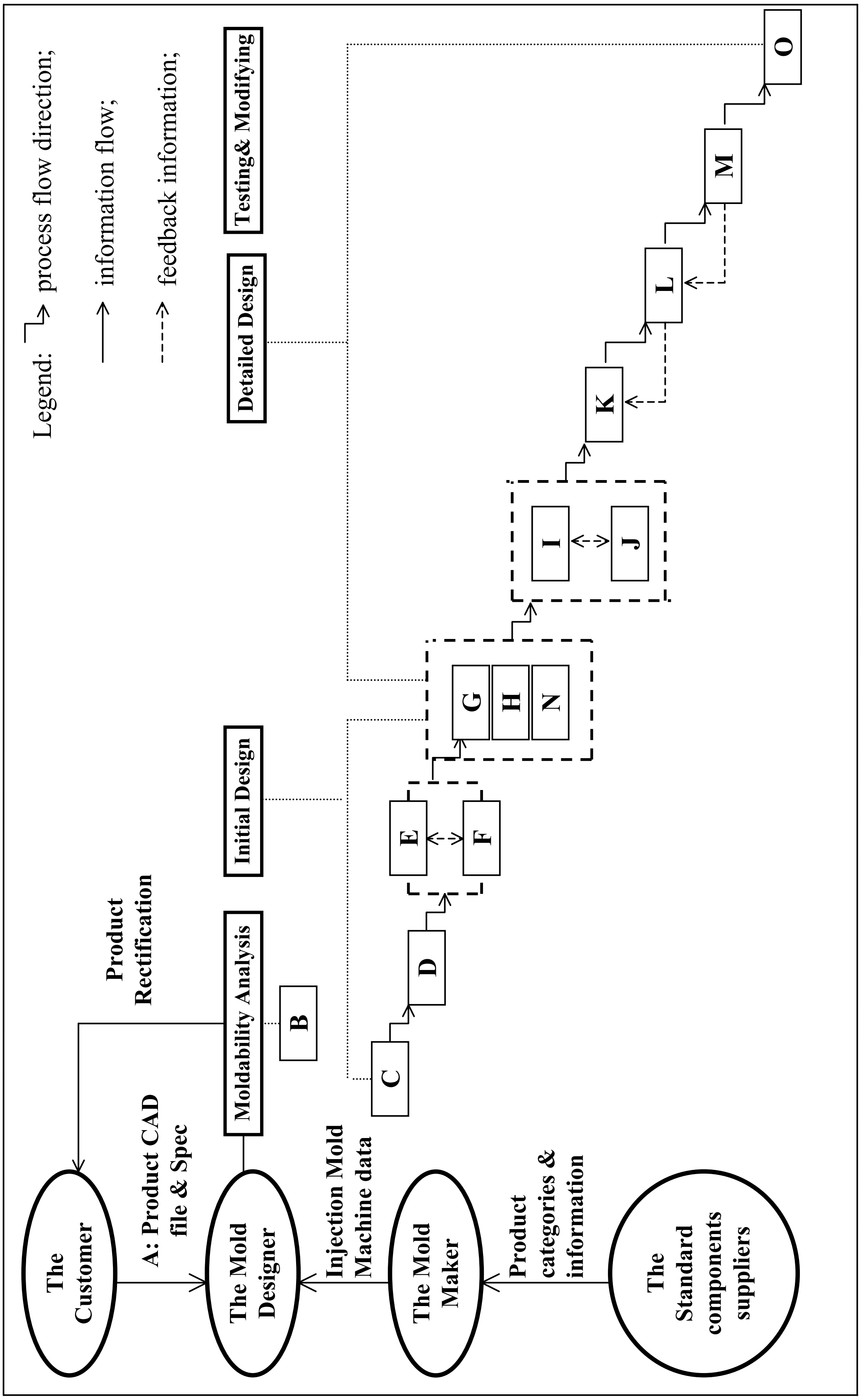


Figure 3. 13 The injection mold design process flow map

3.4 Chapter summary

This chapter presents the process-modeling method using the design structure matrix to draft the process flow map for the injection mold design process. The chapter also presents the procedure for setting up a matrix model for the injection mold design, and the interpretation about the information relationship among injection mold design activities. The DSM process partitioning and tearing developed from Kusiak's triangulation algorithm are discussed and applied to decompose the injection mold design process into manageable hierarchies around levels, and to eliminate the redundant information loops. Necessary assumptions are also made for effective process modeling.

CHAPTER 4

WEB-FACILITATED COMPUTER SYSTEM IMPLEMENTATION

Due to considerable interaction among the mold design activities and the necessary collaboration between dispersed specialized groups in today's injection mold product development, there is a need to develop a computer-supported system to facilitate the coordination of tasks among the specialized groups involved in the injection mold design process, and to streamline the decision making activities at each design stage. In this section, a framework for a computer-aided injection mold development based on the proposed concurrent process modeling is presented and discussed.

4.1 System overview

The proposed concurrent and collaborative mold product concept is implemented in JAVA™ language-enabled SmarTeam system. The technical framework of SmarTeam system is depicted in Figure 4.1.

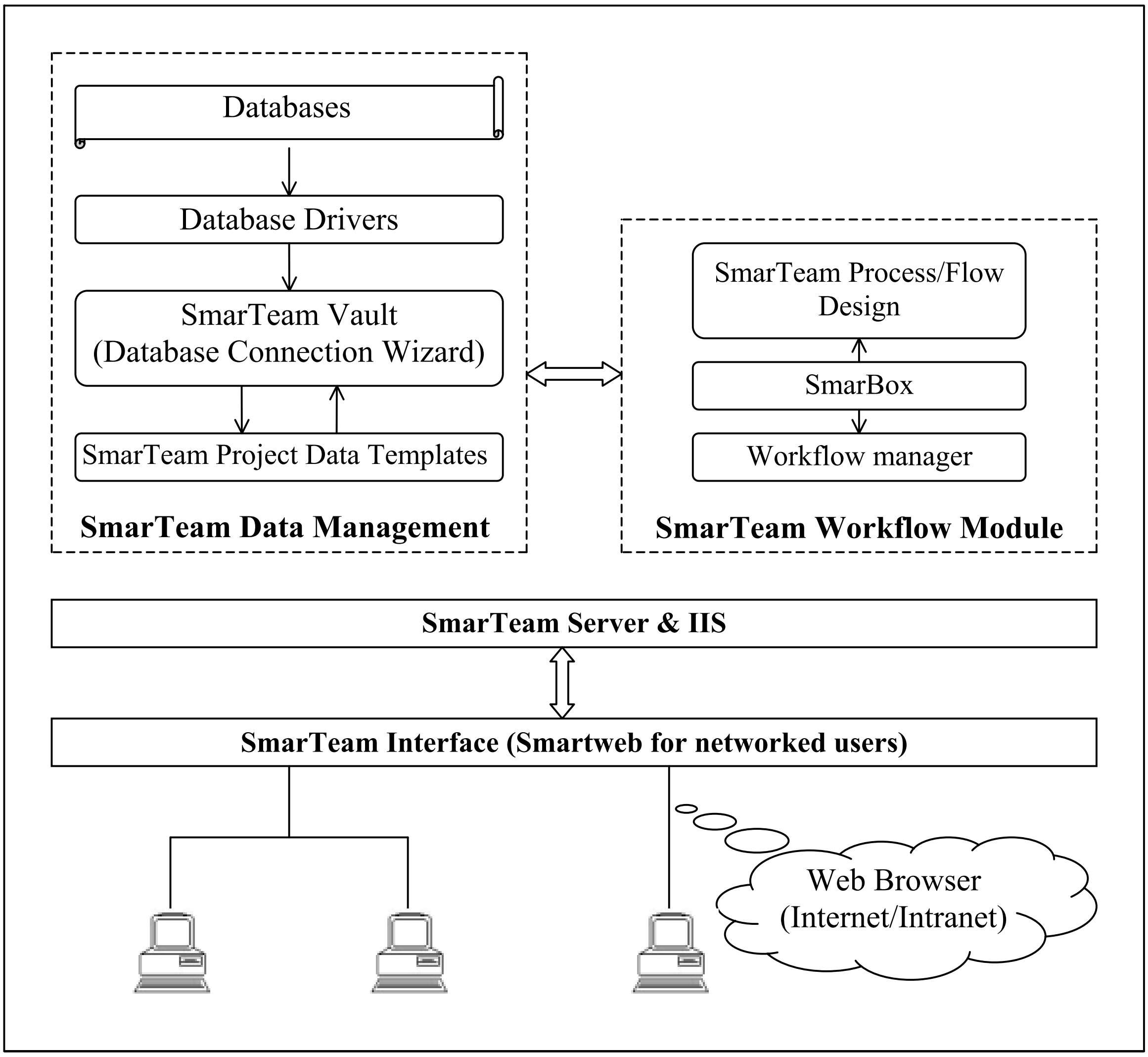


Figure 4.1 The SmarTeam system architecture

There are three main components in the SmarTeam system. These are the collaboration module, the data management module, and the process control (workflow) module. The collaboration module functions to eliminate the geographic restriction by enabling the web communication among clients and their access to the SmarTeam database server. The data or information that is obtained from remote clients via web communication is managed in the data management module, which organizes information in object-oriented data structure for easy and quick sourcing. The process control (workflow) module defines the flow path of data movement according to the functional requirements of each activity. Participants in the course of product developments then follow the flow path to perform the tasks.

4.1.1 Collaboration module

Currently, there are two types of client/server applications that are used in web-based computer systems [Shyamsundar et al., 2002; Berg 1999; Nidamarthi et al., 2001]. One is the thick-client and thin-server type (thick-client); the other is the thin-client and thick-server type (thin-client). In the thick-client mode, the client-end has a high hardware configuration where most of the data operations like product geometry modeling, file documentation are conducted; the server-end functions as the data storage place for users' sourcing in their work. The greatest advantage of utilizing this type of client/server structure is that the workload on the server end is small and the operation problems associated with network translation speed can be overcome. However, the limitation is also obvious: high processing demands on the client side which requires considerable maintenance and technical support, and if the clients work on different types of platforms, the maintenance can be tremendous. In the thin-client and thick-server mode, clients have a low hardware configuration because the bulk of data processing operations occur on the server end, the client essentially performs the visualization tasks or communication tasks, and therefore the workload on client ends is reduced.

The SmarTeam system utilizes the hub server and thin-client structure to facilitate the collaboration environment. To eliminate the restriction brought by the increasing number of users (clients) in the application of the thin-client structure, SmarTeam system develops the 3-tier client authorization for web-based users (See Figure 4.2). The SmarTeam Server is a Windows (NT/2000/XP) service, and utilizes the API that requires the license for users operation. Since only the system administrator or the fully authorized company has the complete access to the product related data, the SmarTeam client/server mode ensures a secure collaborative environment for enterprises and their divisions or subcontractors.

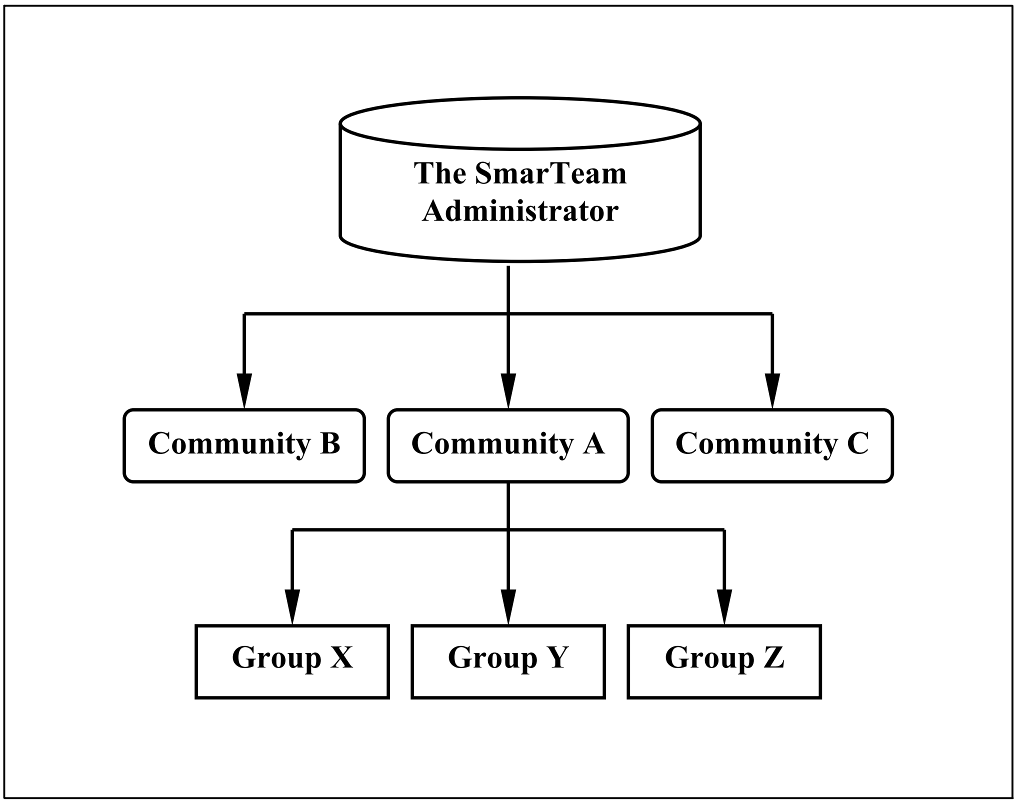


Figure 4.2 The 3-tier user authorization in the SmarTeam system

The user interface between the clients and the SmarTeam server is the Smartweb. The Smartweb interface is based on the technology of standard web browser: the HTTP protocol is used to enable clients' communication with SmarTeam server. The requests from the clients and the responses from the web server are processed by the technologies of ASP with JavaScript, XML, and COM+ in SmartWeb interface. As shown in Figure 4.3, when authorized clients want to access SmarTeam database, they load the URL address of SmarTeam web server into their web browsers. Then they will enter the index page of SmarTeam system. The SmartWeb supports the manipulation of both engineering-type and office-type documents. The manipulation of engineering file like product solid models is visualized by a JAVA-invoked viewer that is embedded in SmartWeb. This viewer processes the control of viewing, rotating, adding, deleting, and modifying on the product CAD models, and the CAD models can be from SolidWorks, AutoCAD, ProEngineer, or CATIA, etc. Besides this, SmarTeam users can use SmartBox, the internal massaging tool and an essential component of SmarTeam process control module, to send and receive massage for further communication.

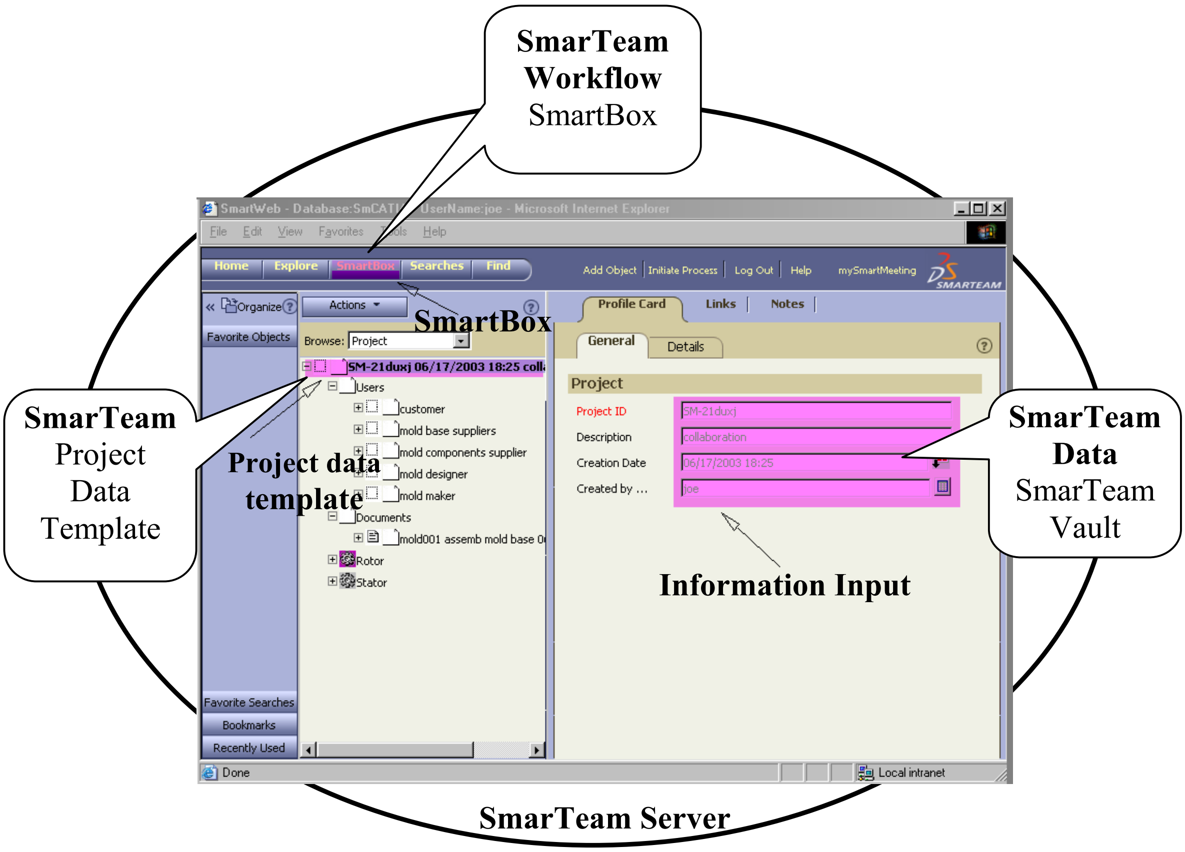


Figure 4.3 The SmarTeam user interface-SmartWeb

4.1.2 Data management module

In the SmarTeam system, the product information stored in the database is managed along the levels of objects, classes, subclasses and attributes. At the object level, the information of different perspectives of a product related project such as business items, project documents, partner groups are established. At the level of classes, subclasses and attributes, the detailed information pertinent to each object is listed, such as the drawings, specifications, dimensions, and feature types. The information associated with one defined project data model is linked together through the methods, which is pre-defined in the system to interpret the logic relationship of a project related data. The information in SmarTeam data model can be modified or interactively defined by users. Various types of databases where users store their data can be integrated with SmarTeam system through SmarTeam Data Connection. Users classify the information of the database into different levels of object-oriented data templates with SmarTeam Wizard. Since the SmarTeam data templates are integrated with World Wide Web. The distributed users can access the data or information through the JAVA enabled query that is predefined in the SmartWeb (See Figure 4.4). The predefined queries direct users to the specified data entities that are managed in the object-oriented data templates, so that users can define, add, or update data.

Figure 4.5 presents the data processing in SmarTeam system. To access the information or data stored in SmarTeam system, authorized users first need to get the information checked into the system vault. When this requirement is approved, the information can then be released and accessed by other users. In this research work, the data or information associated with the case study is first managed with Mircosoft

Access. Then SmarTeam Data Connection links this information with SmarTeam system for project management and remote user access.

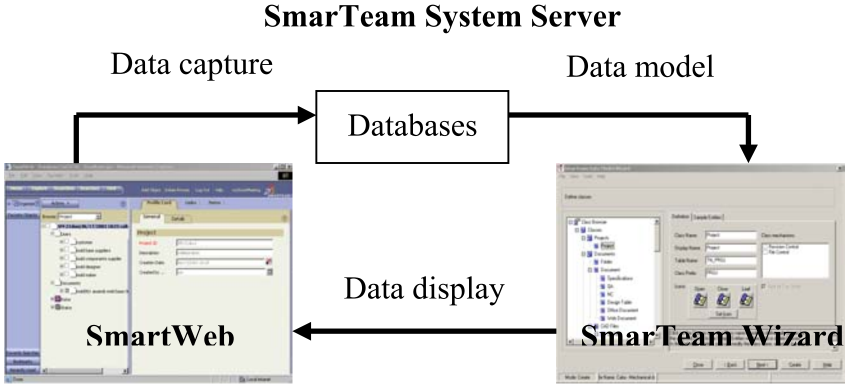


Figure 4.4 Data management module in the SmarTeam system

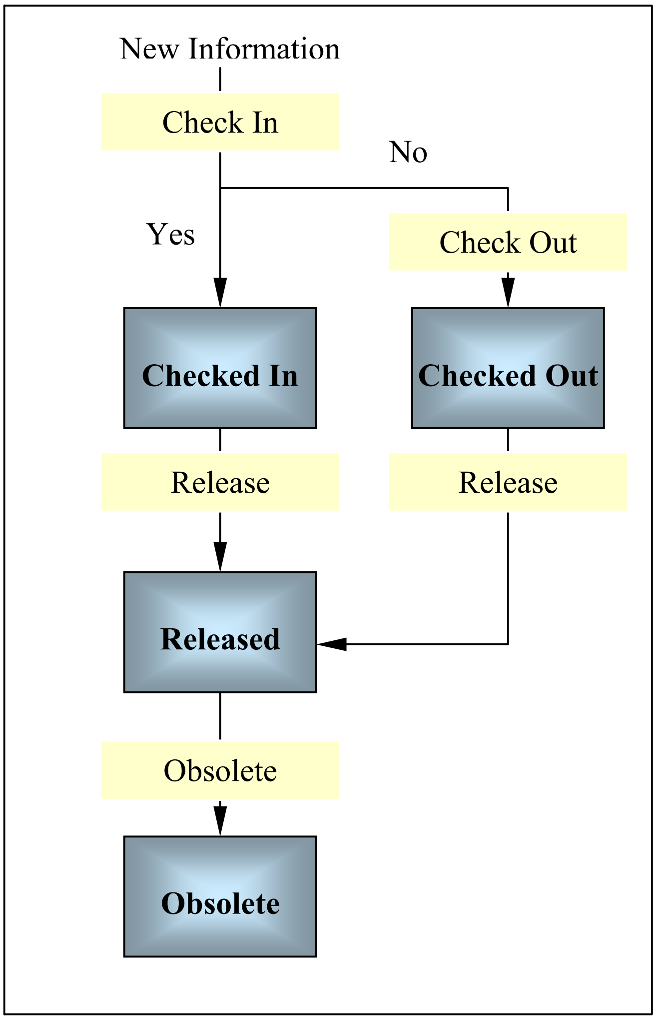


Figure 4.5 The SmarTeam dataflow process

4.1.3 Process control (workflow) module

The process control module in SmarTeam system is used to manage the flow of information in the product design and manufacturing processes. In this module, system administrators can employ the flow chart designer to define a variety of process flows for the needs of different users (See Figure 4.6), in the mean time, they can use the workflow manager to monitor processes already initiated. The process flow managed in this module can be either routine business procedures or technical procedures that require specific information support. Users can initiate the process, view the process waiting for them, work on the tasks designated to them, and pass the process to the next step with the aid of SmartBox. These tasks could be SmarTeam operation tasks such as Check in/out, Approve, user-defined tasks and tailor-made activities. Through this, people, data, and information that are involved in a project are linked together, and the traceability and control of the project is improved. Further explanation of this module is given in the next section.

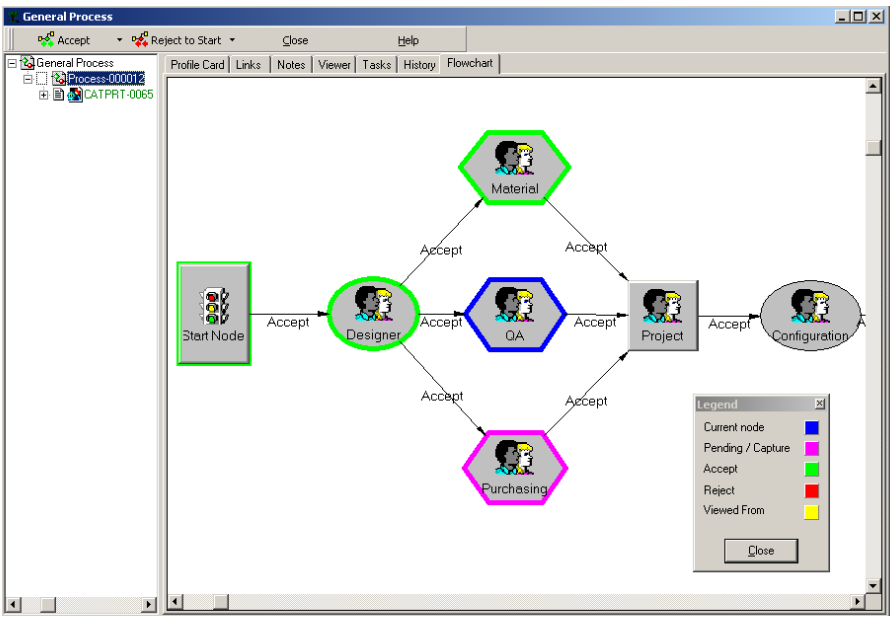


Figure 4.6 The flow chart designer in the SmarTeam system

4.2 Case studies

In this section, the authors present the implementation of the proposed process model and the SmarTeam network system with two case studies to support efficient collaboration in distributed injection mold product development. The presentation of the results is focused on the essential injection mold design activities.

4.2.1 Case study of a hand phone cover

According to the approach and proposed mold design process model, the SmarTeam system is implemented to visualize the concurrent and collaborative injection mold design environment. One example of a hand phone cover (See Figure 4.7) is given to illustrate part of the functions of the SmarTeam integrated collaborative and concurrent injection mold design process, and the synchronous and dynamic aspect of this new design environment. The injection mold design of this case example is performed in accordance with the procedure displayed in Figure 3. 14.

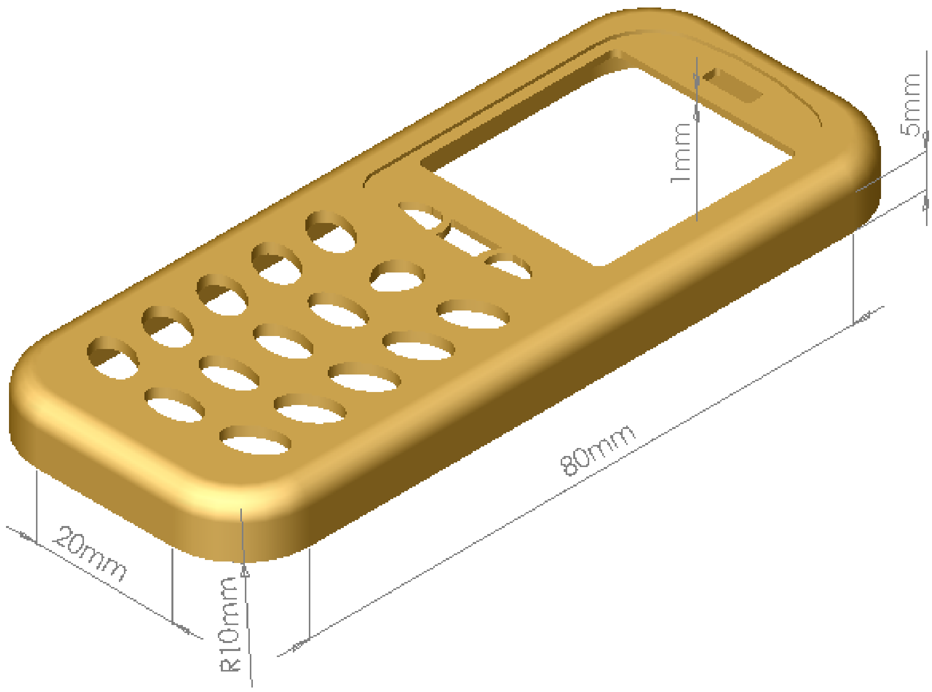


Figure 4.7 The hand phone cover CAD model

- **Stage I: Initial Product Moldability Analysis and Technical Discussion between customer (the product designer) and the mold designer**. The interaction among different discipline groups at this stage can be captured in Figure 4.8. A mold design project starts with the SmarTeam system administrator setting up a project folder named SM-HPcover within the SmarTeam system and granting users involved in the project with the controlled authority to the SmarTeam server (See Figure 4.9). After that, the information of user name and password are past to respective users via email or fax. The SmarTeam administrator then defines the process flows and tasks pending their attention, and send these processes to users via SmartBox.

  In this case study, the process flow at the first stage is depicted in Figure 4.10. Upon receiving the information of authority, users log onto the SmartWeb and check their SmartBox to find out what job is pending for them. For example, the customer uploads the hand phone cover CAD file, specification, and other requirements to the designated project folder via the SmartWeb (See Figure 4.11); the mold designer retrieves this CAD model for the moldability assessment through SmartWeb; the mold makers provides necessary machining information as required for the mold design, etc.

  After the mold designer gets the CAD file, he or she first has a technical discussion with the customer. The technical discussion usually covers the basic information for the initial mold design, such as the resin material to be used and its shrinkage value, the mold cavity number required by the customer, the molding machine details, the desired mold layout, the type of mold base wanted, the gating system, and other basic information. The mold designer or the customer could put their opinion under the project folder at the SmartWeb for other users access, or they can exchange their opinion on these fundamental technical requirements via SmartBox. After the mold designer and the customer reach the agreement on these fundamental requirements, the mold designer proceeds to inspect the mold CAD model for moldability analysis and rectifies the mold structure for material shrinkage requirement in operation. If there is modification, the mold designer then uploads the modified CAD file back to the project folder, and uses the SmartBox to send messages to the customer for their upgrading and approval. When the customer finally accepts the mold designer's suggestions, the mold designer starts the mold layout design. In this case study, the moldability evaluation is basically the assessment on part weight, wall thickness, and the structure (undercut and sharp corner). One design error occurs, there are sharp corners found around buttons. Rounding is therefore required to modify this structure and prevent failure in production. The mold designer gives the suggestion under the directory of SmartWeb note pad and leaves the message at SmartBox for customer' s tracking.

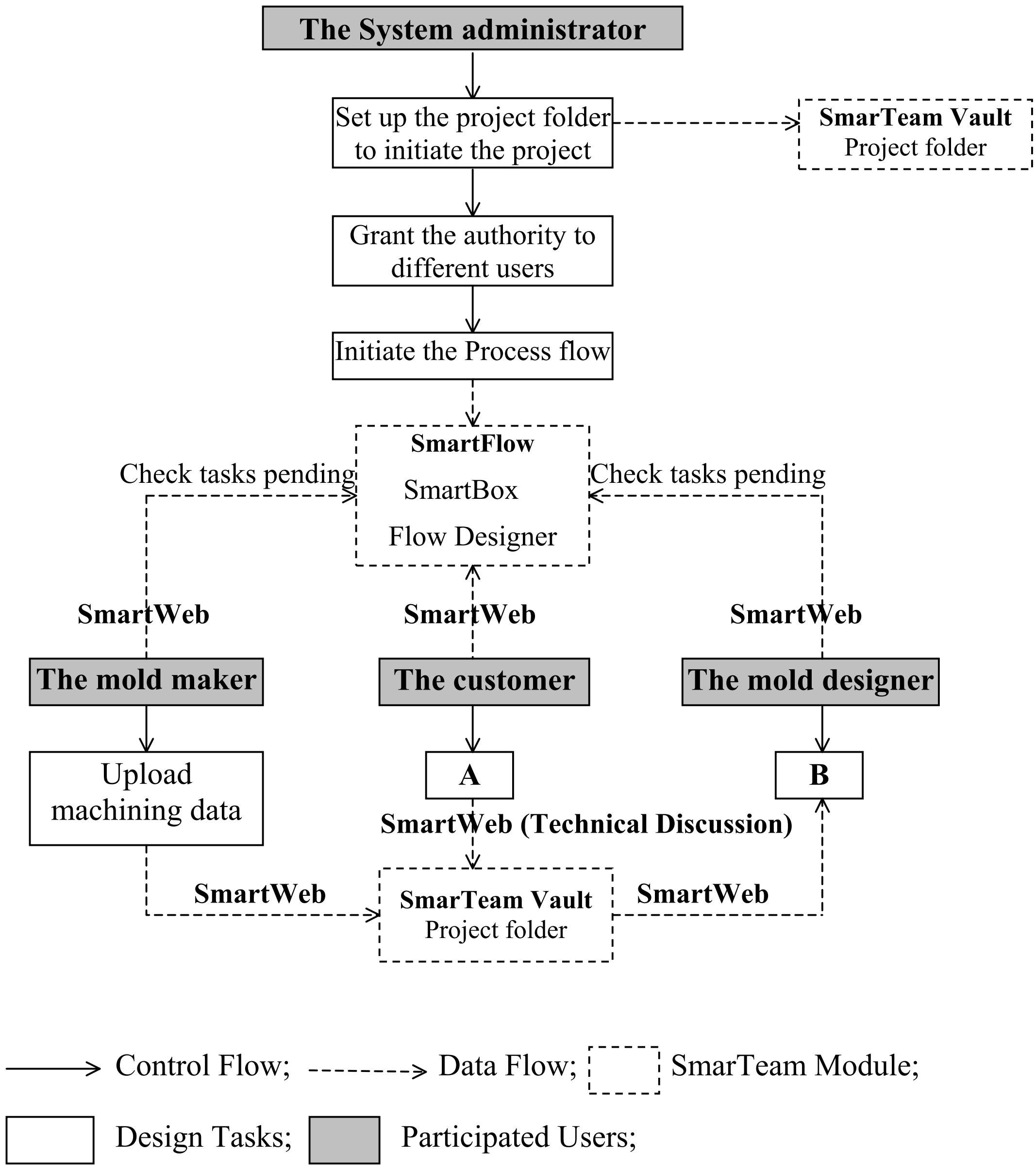


Figure 4. 8 The collaboration and interaction among users at the 1st Stage of the injection mold design process

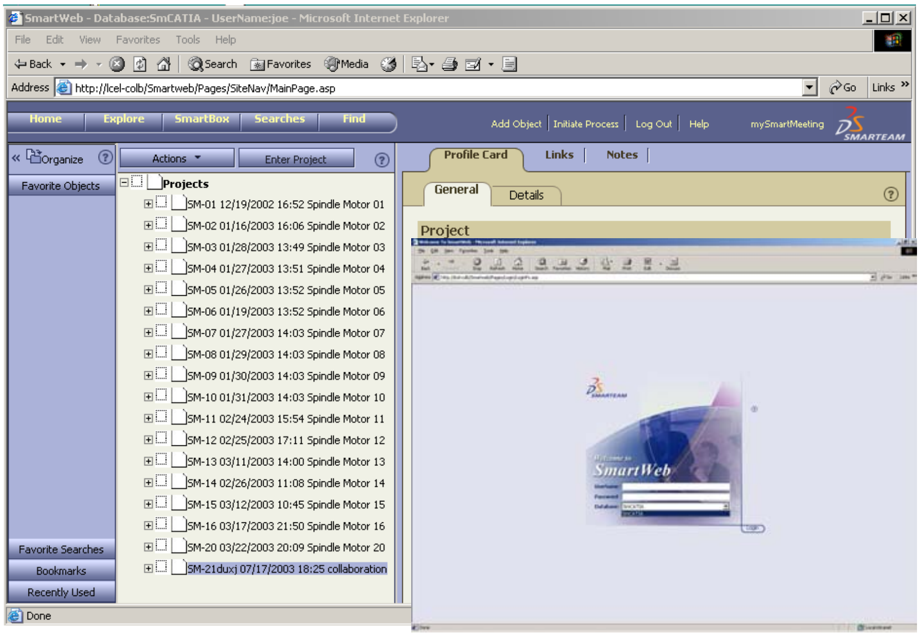


Figure 4.9 Initiate an injection mold design project with the SmarTeam system

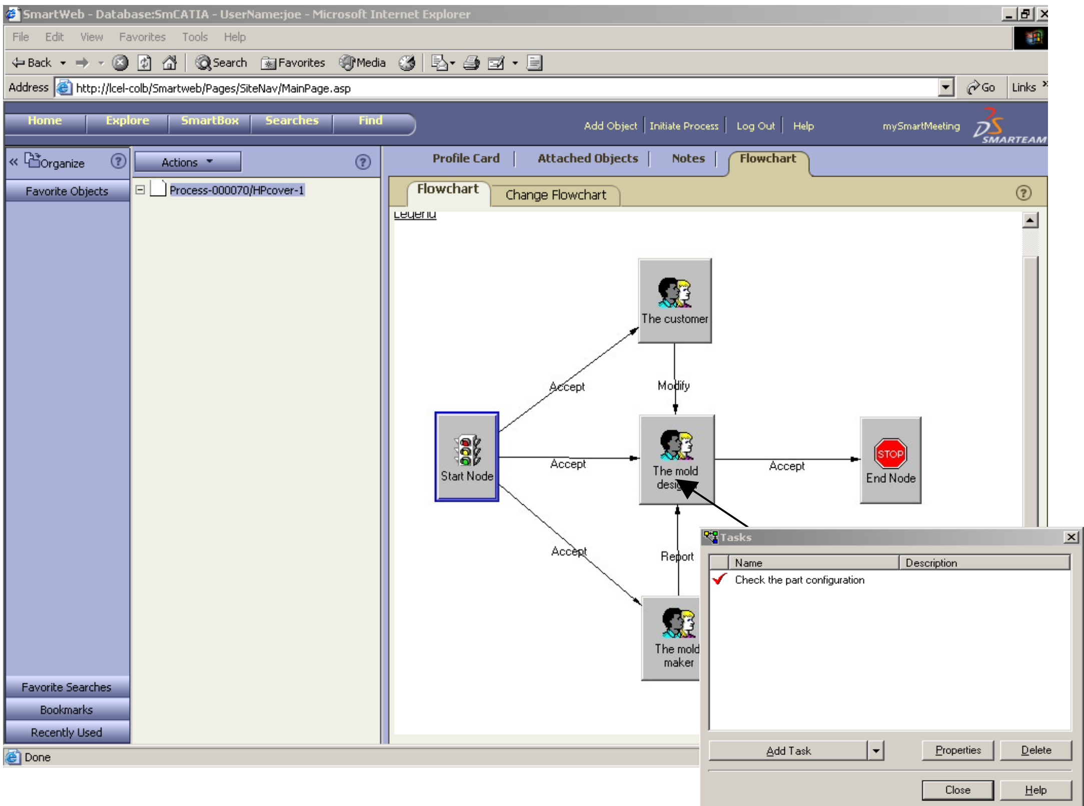


Figure 4.10 The process flow at the 1st stage of the injection mold design

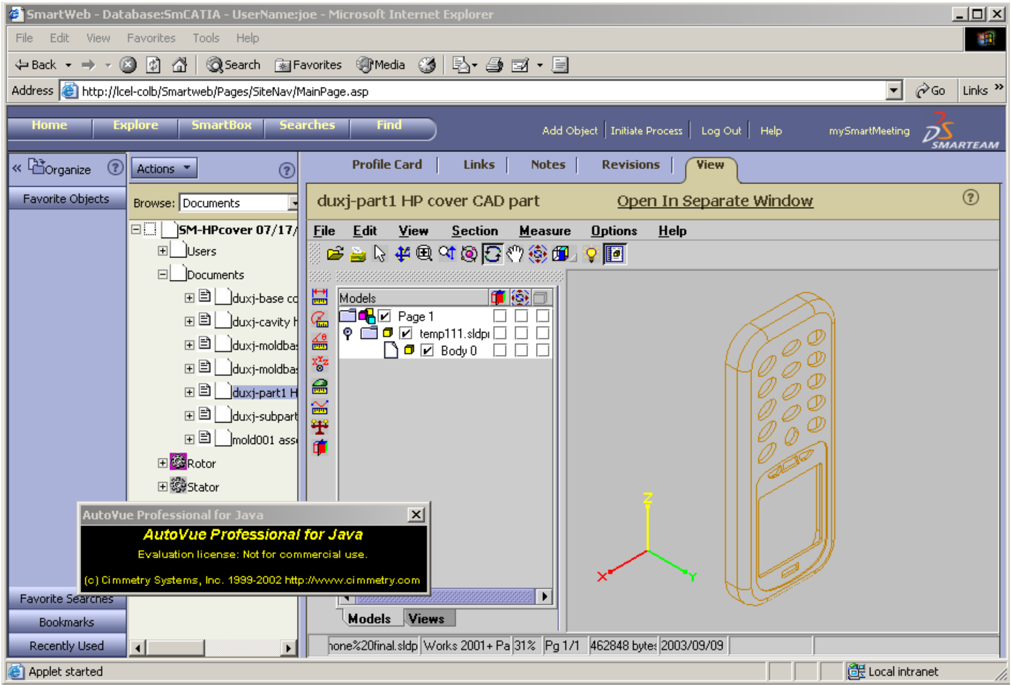


Figure 4. 11 The hand phone cover CAD model displayed at the SmartWeb

- **Stage II to Stage III: The cavity number and initial parting line determination.** As the process model suggests, these two stages are performed collaboratively, and the interaction among different parties can be captured in Figure 4.12. Upon being granted the authority to the SmarTeam system server, the standard mold base supplier logs onto the SmartWeb and checks what task is pending for his or her at the SmartBox. After that, the standard mold base supplier uploads the soft copy of stand mold base categories to the project folder. The administrator then sends messages to the mold designer for his or her attention. Based on the information provided by the standard mold base suppliers, the mold designer then needs to determine the number of cavities. In this case study, the cavity number finally selected is four. When the cavity number is determined, the mold designer proceeds to select the parting line to separate the hand phone cover into the core and the cavity. To improve the accuracy of the parting line selection, the mold designer sends messages to the mold maker via the SmartBox and asks for his or her opinions on the parting line selection. In this case study, the parting line is positioned at the outer surface (See Figure 4. 13). The flowchart at these two stages is illustrated in Figure 4.14.

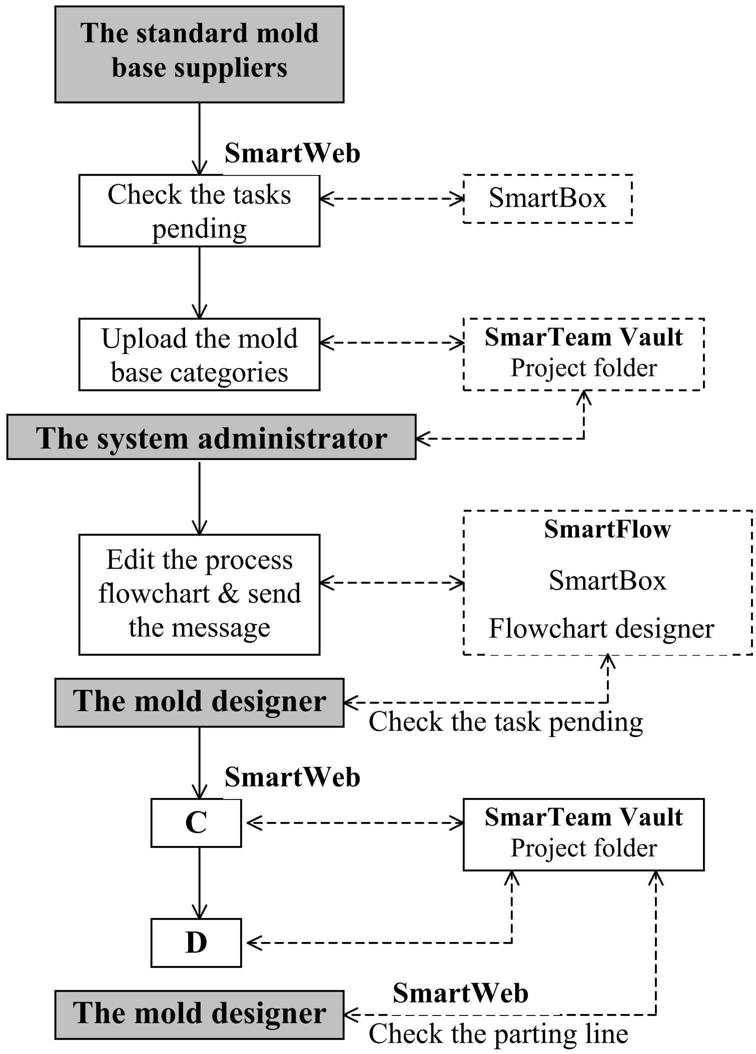


Figure 4. 12 The collaboration and interaction among users at the 2nd and 3rd stages of the integrated injection mold design

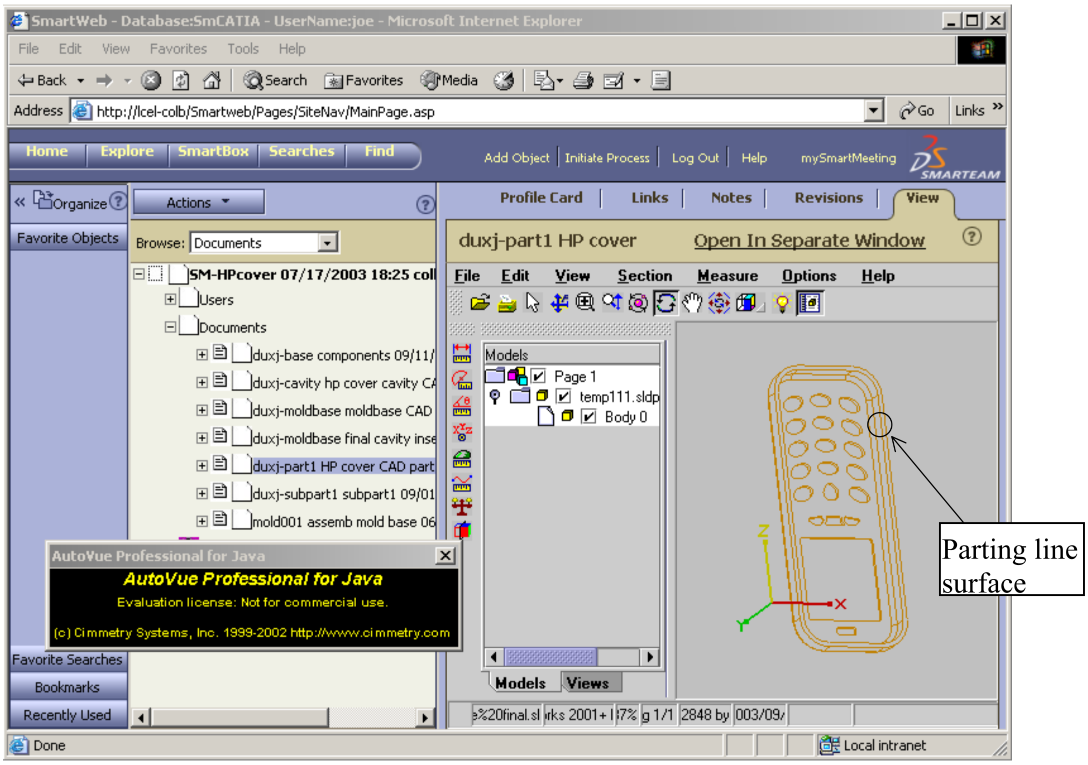


Figure 4. 13 The selected parting line surface

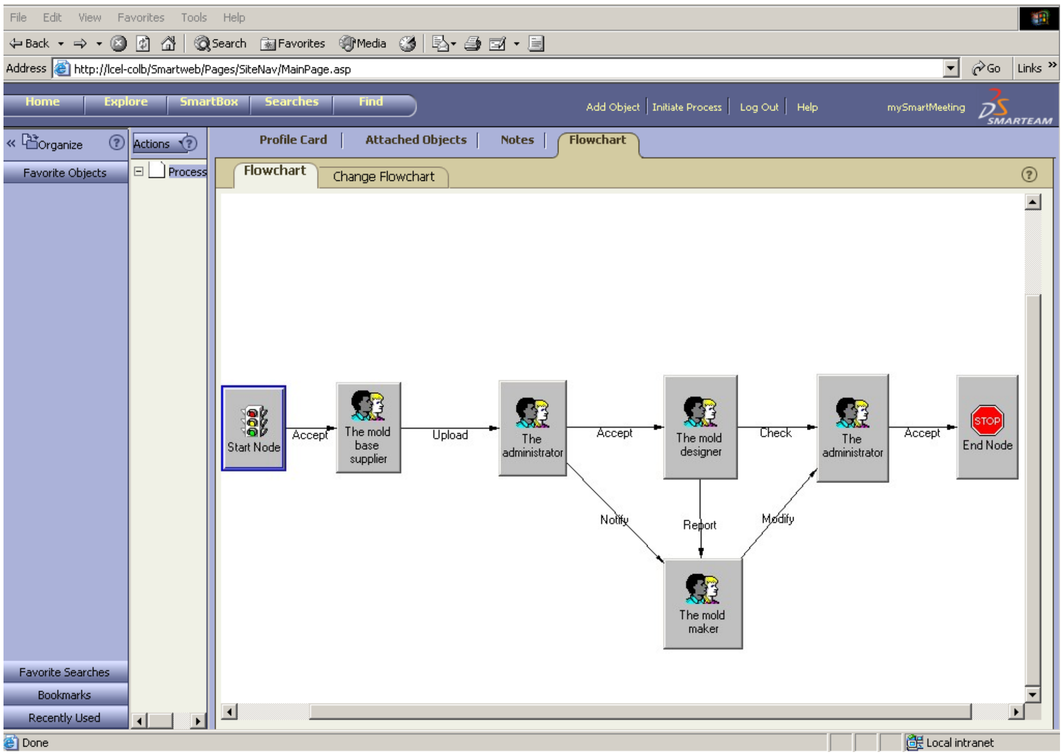


Figure 4.14 The flow chart of process at the 2nd and 3rd stages of the mold design

- **Stage IV: the mold layout design, the mold base selection, and mold components design**. Based on the understanding of the process model at this stage, the interaction among different parties can be captured in Figure 4.15. The mold designer does the cavity layout and selects the most appropriate mold base at the same time since these two activities are interacted. As the cavity number chosen in the previous stage is 4, the mold designer calculates the most appropriate mold base size to hold 4 cavities. After the mold base size is determined, he or she checks the standard mold base information via SmartWeb to design the appropriate mold layout and select the most appropriate mold base for this layout (See Figure 4. 16). After these two activities are finished, the mold designer uploads the CAD file of the mold cavity layout to the project folder for the mold maker's approval. If the layout design and the mold base selected are accepted by the mold maker, the mold designer then proceeds to do the following activities at the same time, which are merging the mold cavity layout with the mold base, selecting the sprue, and doing the cooling system analysis and design.

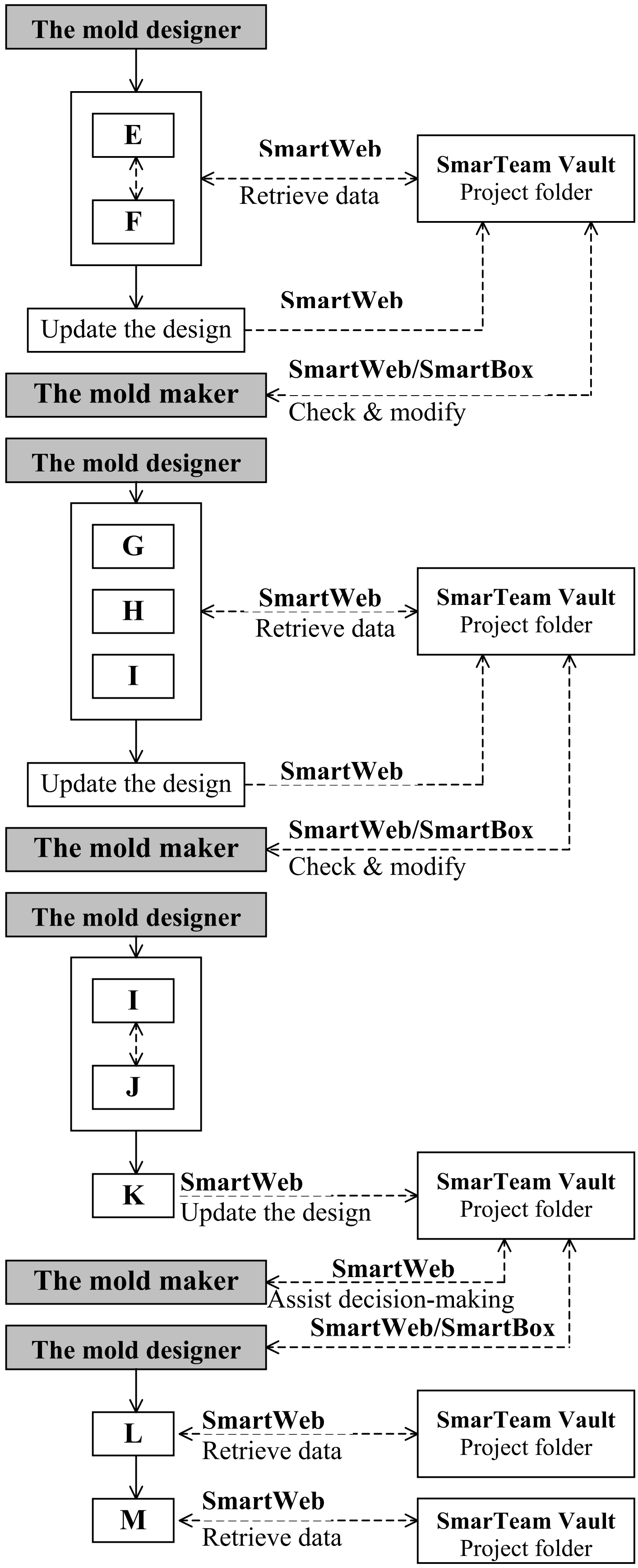


Figure 4.15 The collaboration and interaction among users at the 4th stage of the integrated injection mold design

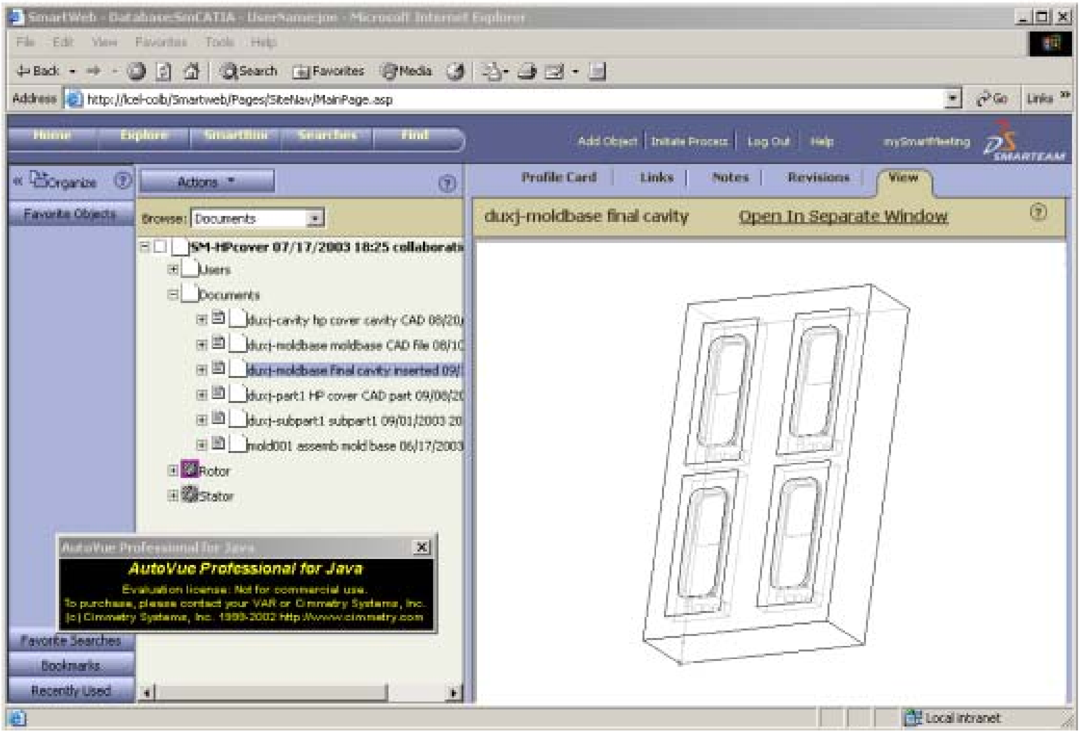


Figure 4. 16 The cavity layout of the hand phone cover mold

As the sprue used in the existing mold fabrication industry is most often semi-finished or fully finished, the mold designer selects the sprue to match with the mold base in accordance with the customer's production requirement and the information provided by the standard mold component suppliers. The cooling system analysis and design are based on the CAE software like COSMOS used in mold flow analysis and cooling analysis. After the mold designer finishes this design, the CAD file is uploaded to the project folder again for the mold maker's inspection. In this case study, the CAD file resulted from the design of the above three activities is shown in Figure 4. 17.

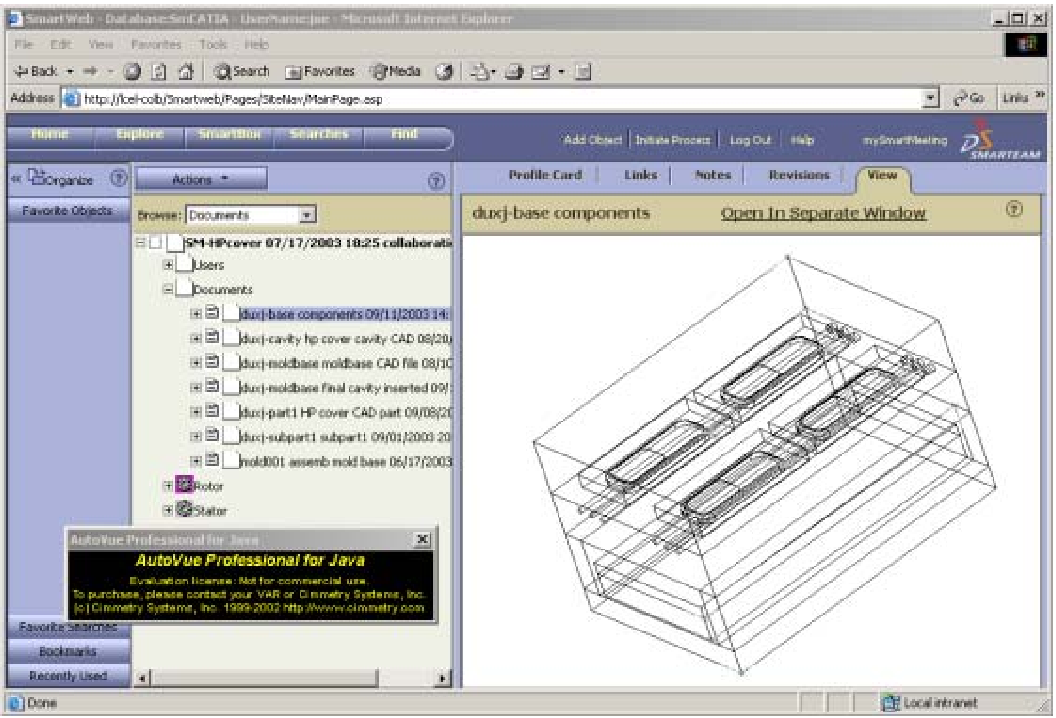


Figure 4. 17 The mold base of the hand phone cover (with the cooling channel included)

The runner and gate designs are also performed collaboratively. In this case study, the cold runner is designed and used as the reference at the gate design. When the runner and design are finished, the activities of the detailed mold design, the slider or lifter selection, and the ejection pin selection can be carried out in a serial manner. In the detailed mold design, the detection of draft faces and round edges are performed first. In this case study, the mold designer adds the fillet to the concave edge and the round to the convex edge. The final design of the hand phone cover is then sent to the project folder so that the mold maker can download and check the hand phone cover structure before the decision on using the slider or lifer is finally made. If there is the need to use the slider or lifter, the mold designer needs to check the information about standard slider or lifter provided by the standard mold component suppliers. In this case study, the mold maker suggests that the slider or lifter is not necessary, the mold designer therefore proceeds to select the appropriate ejection pin. Upon the finalized hand phone cover geometry; the appropriate ejection pins can be selected with the reference of the information from the standard mold components suppliers.

- **Stage V: other standard mold components selection**. With referencing the final product model, the mold components designed in previous stages, and the standard components data stored at the project folder, the mold designer selects the appropriate ejection pins. After that, the mold designer sends messages to the mold maker for the modification and improvement. The collaboration and interaction among different users at this stage can be captured in the Figure 4. 18.

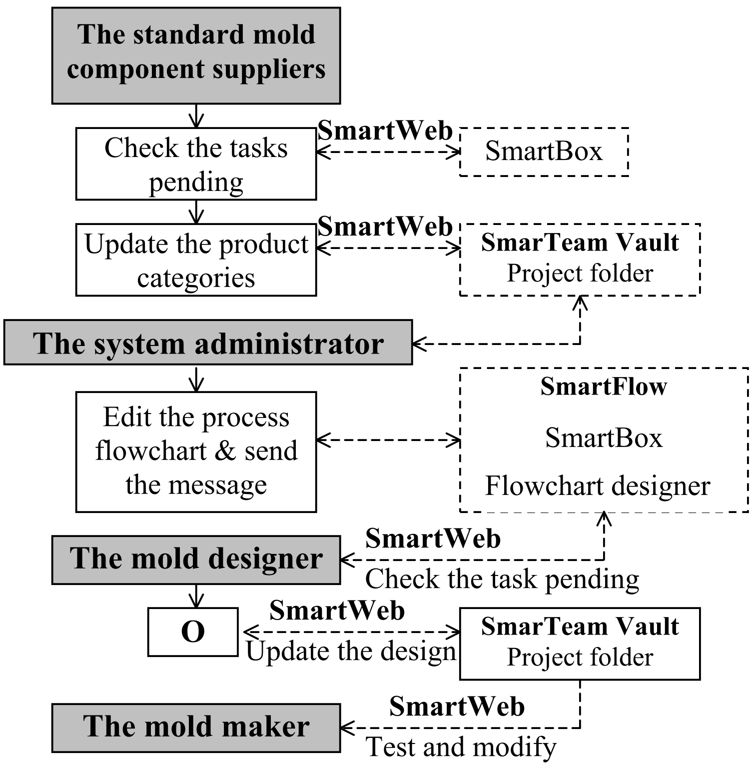


Figure 4. 18 The collaboration and interaction among users at the 4th stage of the integrated injection mold design

4.2.2 Case study of a cored hold boss

Another case study of a part based on that by Lee RS et al. [Lee RS et al., 1997] (See Figure 4. 19) is used in our research to illustrate the dynamic characteristic of the concurrent and collaborative injection mold design development.

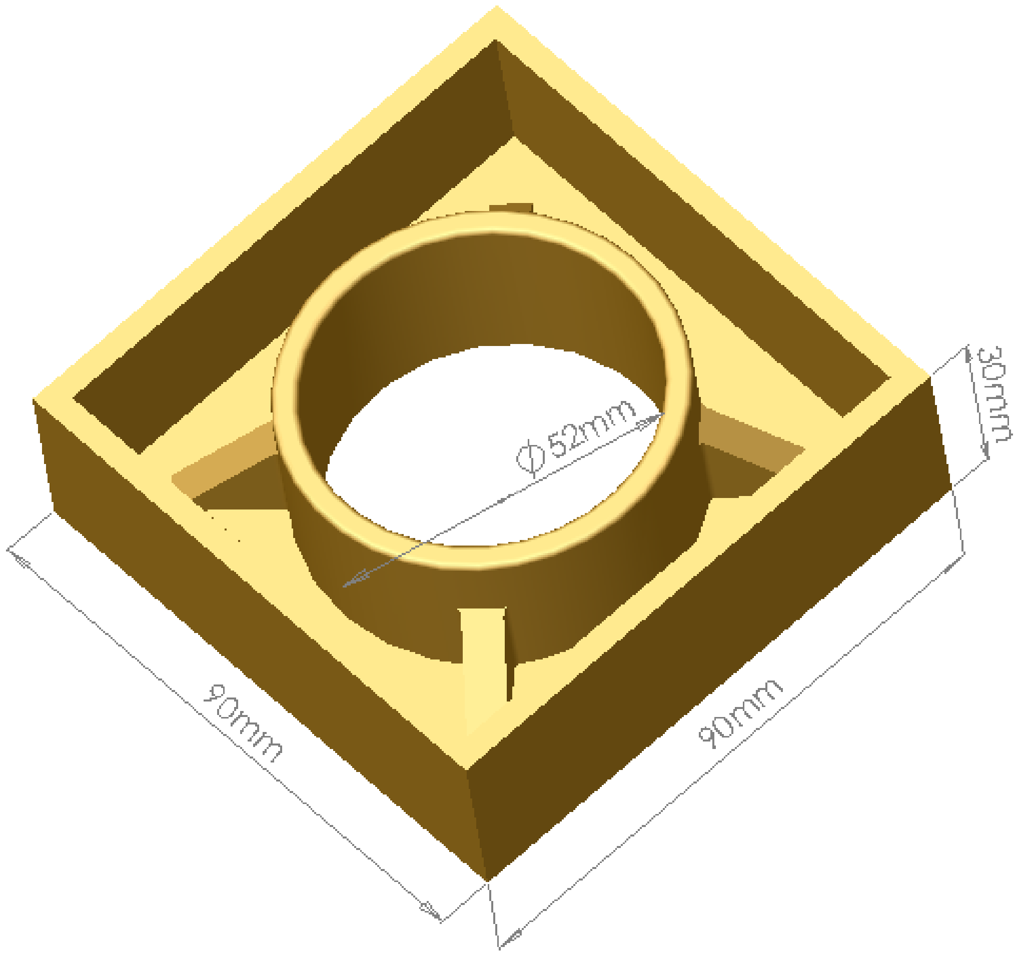


Figure 4.19 The boss CAD model

The development of this casing can be carried out along the following procedures.

- **Stage I:** This mold design project starts with the SmarTeam system administrator setting up a project folder named SM-New Case within the SmarTeam system and granting users involved in the project the controlled authority to the SmarTeam server. After that, the information of user names and password are passed to respective users via email or fax.

Upon receiving the information of authority, the customer logs on the SmartWeb and uploads the part CAD file, specifications, production requirements which include molding machine information, cavity numbers, cost requirements, and lead-time, etc to the designated project folder via the SmartWeb. The mold designer retrieves this CAD model and customer's requirements at the project folder through SmartWeb. Meanwhile, the injection mold maker and the standard mold base supplier upload the necessary files for the mold designer to retrieve. After that, the injection mold designer conducts the initial technical discussion with the customer via SmartWeb. They finally come to the agreement about the cavity number, the resin to be used, and other basic information needed to be determined in the initial discussion (See Figure 4. 20). In this case, two-cavity mold is suggested from the discussion. When the initial technical discussion is over, the mold designer proceeds to inspect the part model for moldability analysis and rectifies the mold structure for material shrinkage requirement in operation if necessary. In this case, the mold designer suggests that ribs are added with fillets to ensure the easy release of the finished part (See Figure 4.21). In the mean time, the mold designer defines the specific shrinkage coefficient according to the polymer selected and the dimension of this part. After all these activities are accomplished, the mold designer starts the mold layout design.

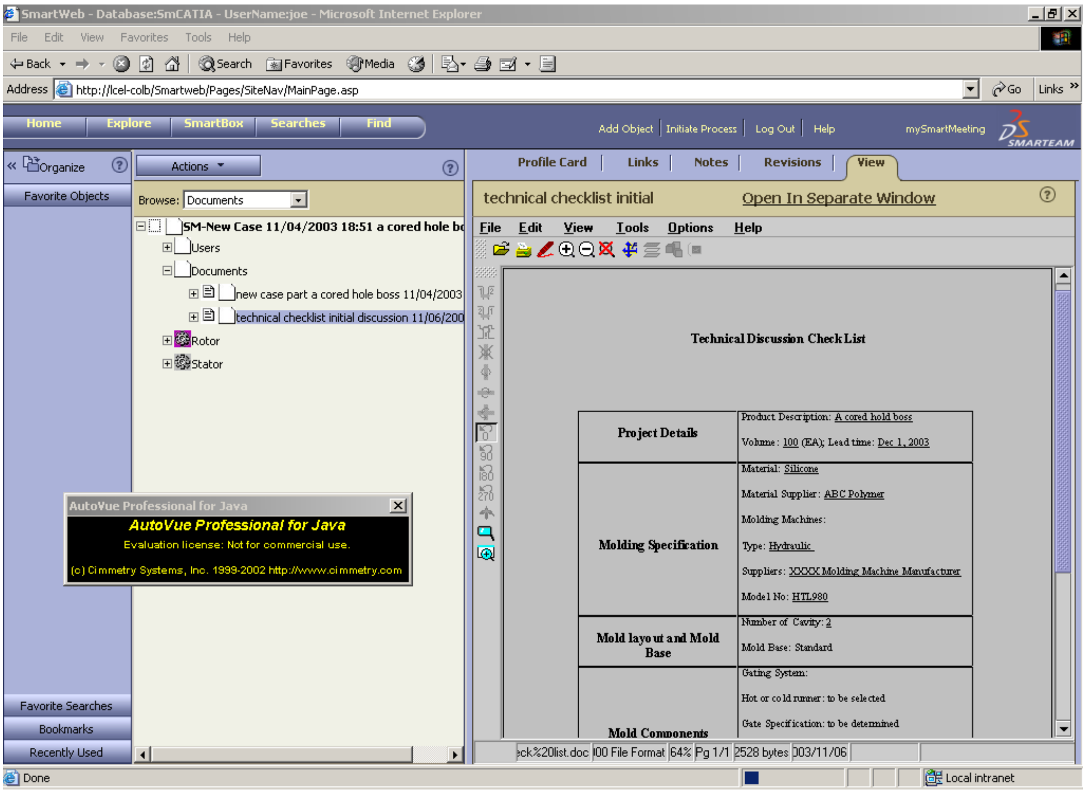


Figure 4.20 The initial technical checklist

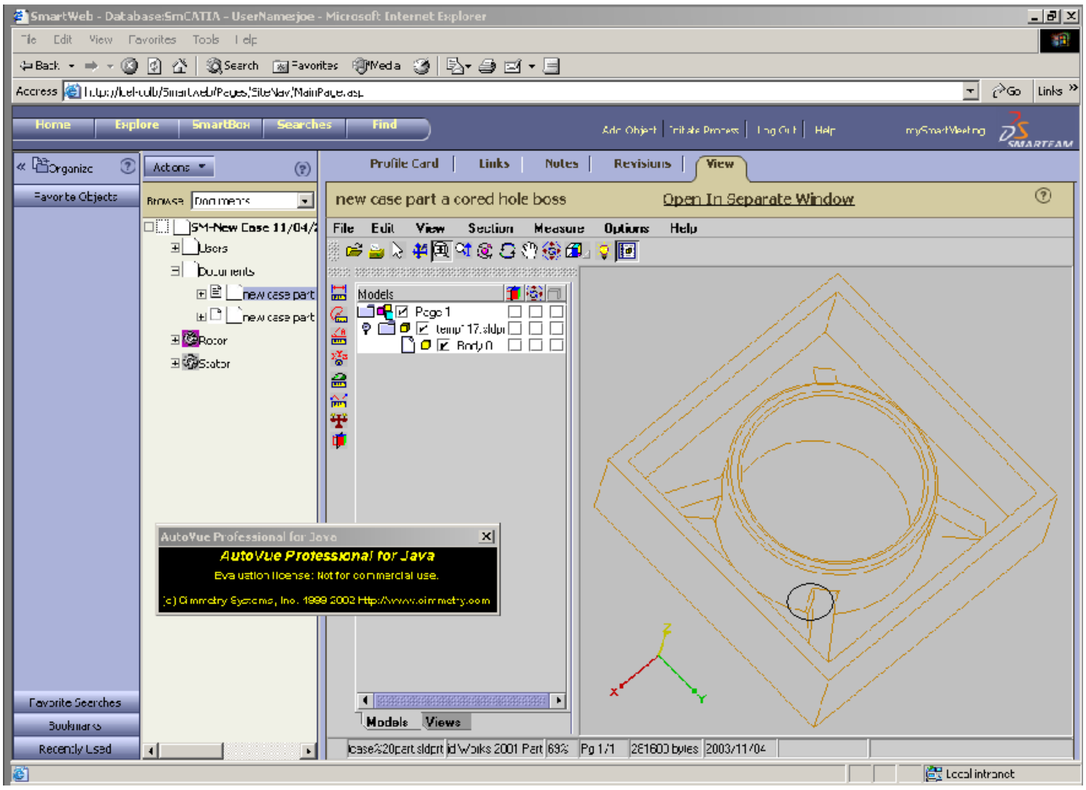


Figure 4. 21 The cored hole boss with fillets added

- **Stage II to Stage V:** At these stages, the mold designer designs the cavity layout, inserts casing cavities into the standard mold base, and assembles the mold components into the mold base. As the cavity number determined in the initial technical discussion is two, the mold designer first retrieves the standard mold base information from the project folder. If information on appropriate mold bases is not available, the mold designer sends messages to request from the mold base supplier for additional information on appropriate mold bases. After the mold base is finally determined, the mold designer proceeds to select the parting line to separate the part into the core and the cavity. To improve the accuracy of the parting line selection, the mold designer sends messages to the mold maker via the SmartBox and asks for his or her opinions on the parting line selection. After the parting line is determined, the mold designer separates the part into the core and the cavity, and inserts them into the mold base (see Figure 4. 22).

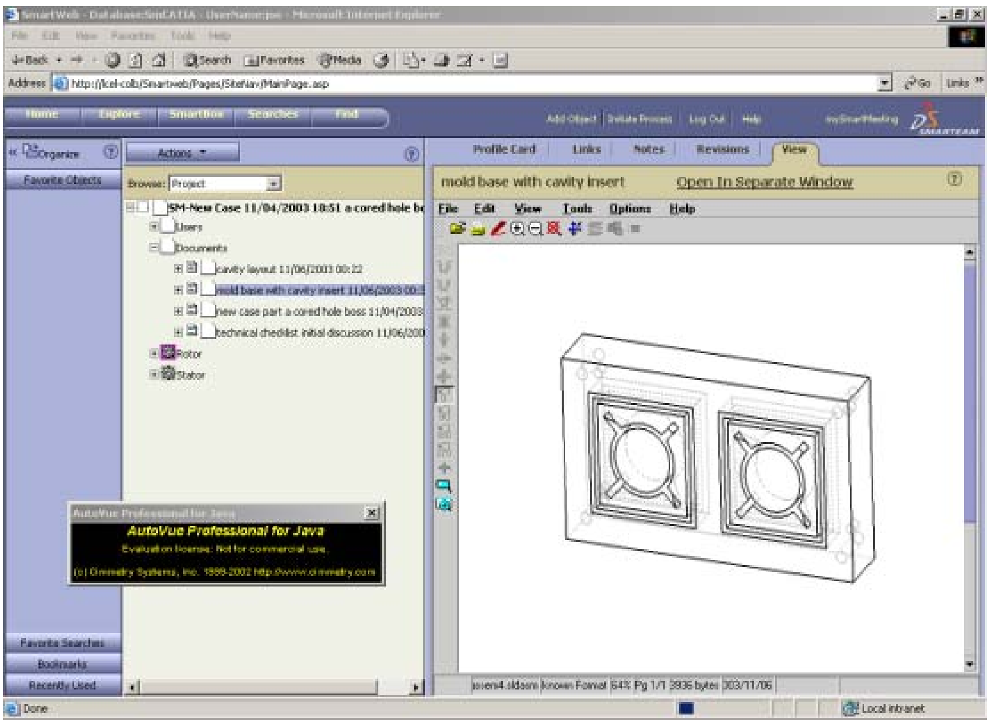


Figure 4.22 The mold base with the cavities inserted

After the cavity and the core are inserted into the mold base, the mold designer proceeds to assemble the mold base with the necessary components like runner components, cooling channels, and etc. The finalized CAD file is then uploaded to the project folder for the customer and the mold maker's inspection.

4.3 Discussion

The proposed injection mold design process is managed and planned on the basis of the integrated SmarTeam system and the concurrent product development concept. Compared with the conventional injection mold design process, the proposed process framework stands out with the following advantages.

1. The resources involved in the integrated injection mold design process are well analyzed and attended, and the complexity of the injection mold design process is reduced. Compared with the conventional injection mold design, the injection mold design process performed under the new environment is derived from the concurrent process model, which integrates both the necessary internal and external resources to meet the end. Under the new environment, the mold design can be carried out along a five-stage procedure. Within each of these stages, the types of activities' relationship are clearly illustrated and the responsibilities of each party or user involved are well defined. Through this, the integrated process model is more manageable and controllable. For example, the rearranged injection mold design process suggests the concurrent design sequences for activities of cavity layout and mold base selection, detailed mold design and slider/lifter design, and subsequent runner system design and ejector pins selection. These rearrangements optimize the injection mold design process and reduce the time wasted on unnecessary information

iteration. The SmarTeam system supports the new design sequence by activating the real-time participation of related collaborators and the instant messaging system.

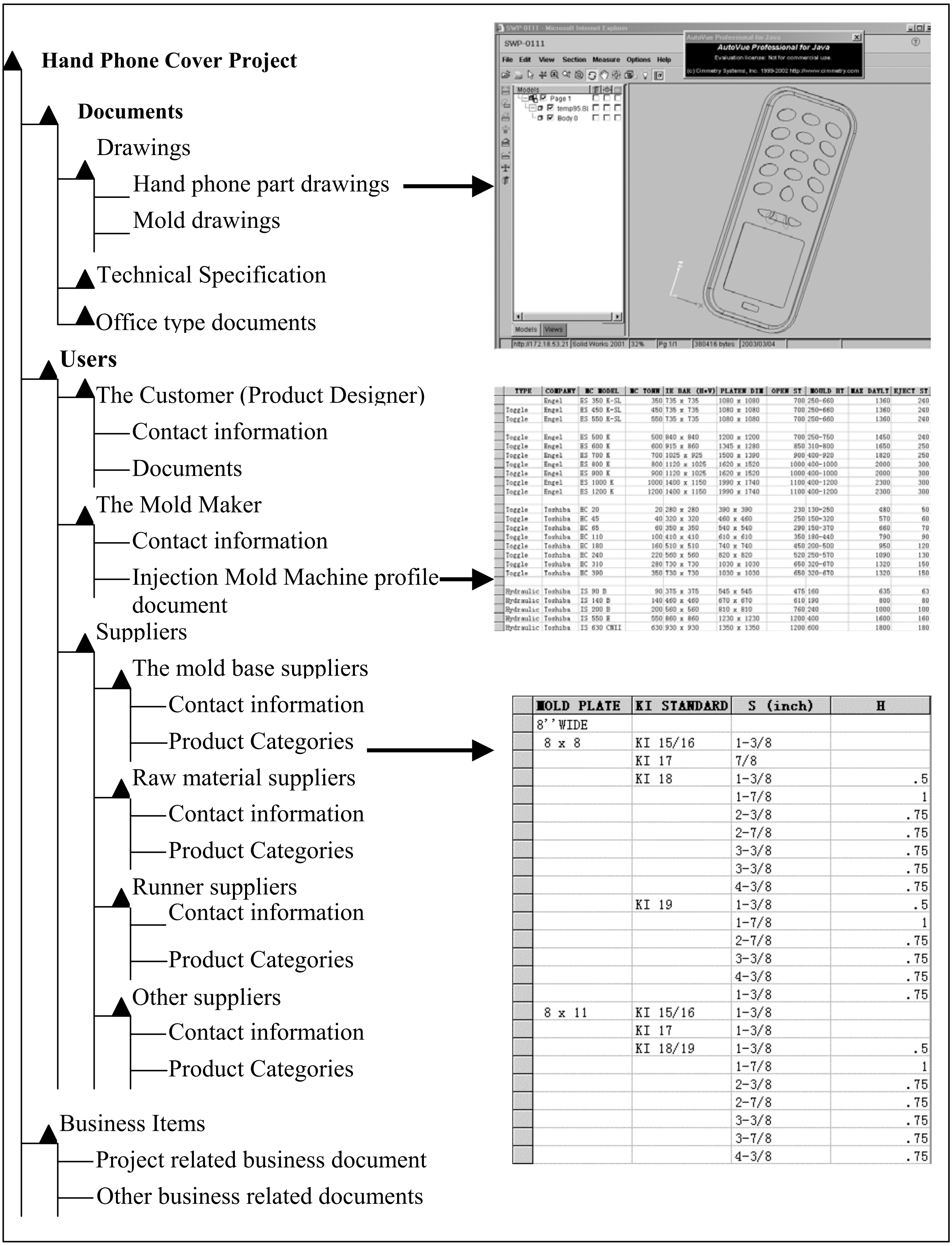
2. The project related data and information in the new injection mold product design environment are systematically managed for the collaboration of geographically distributed parties. The product and project related data in the integrated injection mold development environment is classified into an object-oriented structure. The data hierarchy links the information with the corresponding objects, design tasks, and parties. For example, the information or data related to the standard mold base supplier can be stored in the same sub-directory of the project folder. Such an arrangement provides a clear display of project related information, and facilitates the knowledge-based injection mold design.

Figure 4.23 illustrates the data structure and part of the detailed information used in the hand phone cover project discussed in the case study. It is noted that both the technical and non-technical data or information that are related to the same object are grouped under the same directory, and all the project related data can be represented with this data structure. In this way, Users can r manage and retrieve the data or information for their own needs. In the mean time, the availability of non-technical but essentially related information simplifies the design process and saves users' time in data outsourcing. The information sourcing and transferring are also simplified and accelerated in this integrated mold design environment. The web function of the SmarTeam system enables real-time information transfer and users communication. The

SmarTeam interface, the SmartWeb, provides pre-defined forms for user to enter data, thus simplifies the procedure of data capture. In the mean time, this interface bridges users and the SmarTeam system server where the project-related data are stored. This makes it convenient for distributed users to track information. The most advantage that is brought by this integrated design environment is the process control function (workflow management). This function helps users to define the flow chart of a process in advance. This process can be either the general process of a mold design project or a particular process involved in a design task. The pre-defined flow chart depicts the information flow along this process and pinpoints the responsibility of different parties involved. Through this function, the problems commonly seen in conventional injection mold design process due to users' unawareness or misunderstanding of the process or tasks can be avoided. Also, users can use this function of SmarTeam system to keep track of the development of the entire project.

4.4 Chapter summary

This section presents the implementation of the proposed concurrent injection mold design model in web-based computer system-SmarTeam system. The SmarTeam system facilitates the collaborative injection mold design with three modules: the collaborative module which is based the standard web browser technology and the thin client/server mode, the data management module which manages the product related information, and the process control module which helps users to define the work contents and monitor the process workflow. Two cases are later presented and discussed to test the feasibility of the web-based SmarTeam system implementation.



TYPE	COMPANY	MC MODEL	MC TONN	IE BAR (H*V)	PLATEN DIM	OPEN ST	MOULD HT	MAX DAYLT	EJECT ST
	Engel	ES 350 K-SL	350	735 x 735	1080 x 1080	700	250-660	1360	240
Toggle	Engel	ES 450 K-SL	450	735 x 735	1080 x 1080	700	250-660	1360	240
Toggle	Engel	ES 550 K-SL	550	735 x 735	1080 x 1080	700	250-660	1360	240
Toggle	Engel	ES 500 K	500	840 x 840	1200 x 1200	700	250-750	1450	240
Toggle	Engel	ES 600 K	600	915 x 860	1345 x 1280	850	310-800	1650	250
Toggle	Engel	ES 700 K	700	1025 x 925	1500 x 1390	900	400-920	1820	250
Toggle	Engel	ES 800 K	800	1120 x 1025	1620 x 1520	1000	400-1000	2000	300
Toggle	Engel	ES 900 K	900	1120 x 1025	1620 x 1520	1000	400-1000	2000	300
Toggle	Engel	ES 1000 K	1000	1400 x 1150	1990 x 1740	1100	400-1200	2300	300
Toggle	Engel	ES 1200 K	1200	1400 x 1150	1990 x 1740	1100	400-1200	2300	300
Toggle	Toshiba	EC 20	20	280 x 280	390 x 390	230	130-250	480	50
Toggle	Toshiba	EC 45	40	320 x 320	460 x 460	250	150-320	570	60
Toggle	Toshiba	EC 65	60	350 x 350	540 x 540	290	150-370	660	70
Toggle	Toshiba	EC 110	100	410 x 410	610 x 610	350	180-440	790	90
Toggle	Toshiba	EC 180	160	510 x 510	740 x 740	450	200-500	950	120
Toggle	Toshiba	EC 240	220	560 x 560	820 x 820	520	250-570	1090	130
Toggle	Toshiba	EC 310	280	730 x 730	1030 x 1030	650	320-670	1320	150
Toggle	Toshiba	EC 390	350	730 x 730	1030 x 1030	650	320-670	1320	150
Hydraulic	Toshiba	IS 90 B	90	375 x 375	545 x 545	475	160	635	63
Hydraulic	Toshiba	IS 140 B	140	460 x 460	670 x 670	610	190	800	80
Hydraulic	Toshiba	IS 200 B	200	560 x 560	810 x 810	760	240	1000	100
Hydraulic	Toshiba	IS 550 E	550	860 x 860	1230 x 1230	1200	400	1600	160
Hydraulic	Toshiba	IS 630 CNII	630	930 x 930	1350 x 1350	1200	600	1800	180

MOLD PLATE	KI STANDARD	S (inch)	H
8''WIDE			
8 x 8	KI 15/16	1-3/8	
	KI 17	7/8	
	KI 18	1-3/8	.5
		1-7/8	1
		2-3/8	.75
		2-7/8	.75
		3-3/8	.75
		3-3/8	.75
		4-3/8	.75
	KI 19	1-3/8	.5
		1-7/8	1
		2-7/8	.75
		3-3/8	.75
		4-3/8	.75
		1-3/8	.75
8 x 11	KI 15/16	1-3/8	
	KI 17	1-3/8	
	KI 18/19	1-3/8	.5
		1-7/8	1
		2-3/8	.75
		2-7/8	.75
		3-3/8	.75
		3-7/8	.75
		4-3/8	.75

Figure 4. 23 The object-oriented project data structure model

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The injection mold manufacturing is a major part in plastic industry. With the trend towards globalization in industry environment, injection mold manufacturing increasingly involves participations by different parties from geographically dispersed locations. Hence, there are increasing requirements on efficient information processing and active involvement in distributed injection mold product development.

This thesis presents an integrated system framework for the collaboration and coordination in distributed injection mold product development, especially in the mold design stage. This framework is based on the view of system engineering and the view of the computer-supported information processing. The advantages of deploying this framework in the injection mold design process are as follows:

1. As a key contribution, the injection mold design process is decomposed into five more manageable stages in accordance with the concurrent process model. This means a more clarified information flow and process control in the mold design process, and enables a great saving in time needed in the injection mold design. The system engineering analysis is performed to structure the injection mold design process from the perspective of information dependencies and set

up the matrix-based process model. This model, being interpreted with the Design Structure Matrix triangulation algorithm, accounts for the important characteristics of product development processes, including the basic information transfer patterns, the characteristics of information relationships of each task, and the information flow through the process. What's more, the concurrent process model also gives suggestions about the participations of different parties involved in an injection mold design process. Through this, the structure of the entire process is well represented, the resources needed, both the external and the internal, are understood, and the semantics of the entire process is interpreted. It is therefore understandable that the confusion due to users' misunderstanding and less awareness of the injection mold design process itself can be eliminated.

2. Others contributions of this research lie in that the SmarTeam system deployed supports the concurrent process model by providing the databases to record the entire project and product-related data such as standard mold bases, standard mold components, molding machines, and business documents, etc, and manage the information in an object-oriented manner. This means that all the information or data associated with one particular object is grouped under the same directory for users' easy access. This enables a systematical and scientific data management, and improves the efficiency of project management and the reusability of recorded data. In the mean time, the process control (workflow management) of the SmarTeam system can be utilized to define the workflow of a process on the basis of the information transfer, no matter the particular process needed to accomplish a certain task or the general process of the entire

project. With this function, users can keep track of the project development, and the resources needed at each stage of the mold design model can be allocated rapidly and efficiently.

3. The SmarTeam system also facilitates the collaboration among different users involved in the concurrent mold design process by offering the user-friendly interface and the web service. The user-friendly interface and the web service which are supported by three fundamental modules of the SmarTeam system enable real-time data transfer and processing in the area of remote data capturing, data retrieving, data storage and management for web users and distributed parties involved in the injection mold design process.

4. The resultant system framework presented in this research suggests a technical structure of the injection mold design that takes into account both the system perspective and the infrastructure view. This technical structure provides an approach, and can be referred in future research.

5.2 Future works

The integrated framework has been developed and applied in two case studies. While the flexibility and applicability of this framework has been proven, there are further works needed to improve this framework and enable its wide application.

1. The representation of the framework is presently confined to the essential and relevant injection mold design activities. Extended application is necessary for the injection mold process planning and mold making. Since these processes

have their own characteristics in information processing, the exploration of other process-modeling tools becomes indispensable.

2. The DSM process modeling simply assumes that a fixed resource is input into the information flow. However, the resource could be variable and more sophisticated when the geometric complexity and intricacy of molded parts are increased. Extended application of other DSM methodologies that target at complex product development should be tested and proved.

3. The decision-making that is based on the injection mold knowledge and ration in SmarTeam system needs to be further developed. The SmarTeam system is originally developed for the general collaborative product development. The injection mold product process has its own characteristics that need to be specially attended to when the system is applied. For example, there is usually a discussion list used in the initial mold design stage to collect the customers' requirements. Based on the information collected, the mold designers make the decisions such as the cavity layout design, and the mold base size. If this decision-making procedure can be well integrated into the system architecture, the efficiency at this point will be further improved.

4. The proposed framework can be further developed to incorporate the business issues including development cost, budgeting in the distributed injection mold product development. This part of work might be based on a better understanding of the business operation procedure in individual mold companies and their outsourced partners.

5. The proposed framework in this research is applied in the operation system of Windows 2000. For users who rely on other operation systems such as Macintosh, Unix, the applicability of this framework needs to be tested.

REFERENCES

Berg E.V.D. (1999). COLLABORATIVE MODELING WITH FEATURES. In Proc. DETC'01 2001 ASME Design Engineering Technical Conferences, September 2001, Pittsburgh, Pennsylvania, USA, pp. 1-5.

Belhe U. and Kusiak A. Resource Constrained Scheduling of Hierarchically Structured Design Activity Networks. IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, Vol. 42, No. 2, pp. 150-156. 1995.

Chang HC and Lu WF, Liu XQ F. WWW-Based Collaborative System for Integrated Design and Manufacturing', CONCURRENT ENGINEERING: Research and Applications, Vol 7, No 4, pp. 319-334. 1999.

Chung JH, Lee KW. A framework of collaborative design environment for injection molding, COMPUTERS IN INDUSTRY, 47, pp.319-337. 2002.

Du, XJ, Lee, KS, Wong, YS, and Wang JG. Process Modeling and Analysis with DSM for Collaborative Product Design and Injection Mold Design Framework. In Proc. 4th International DSM Conference, October 2002, MIT, Boston, USA.

Eppinger S.D, Whitney D.E, Smith R.P, Gebala D.A. Computer-Aided Cooperative Product Development. pp. 229-252, New York: Spring-Verlag. 1991.

Forsberg, K., and Mooz, H. The relationship of systems engineering to the project cycle, *Engineering Management Journal*, 4(3), pp. 36-43. 1992.

Galbraith. J.K. The New Industrial State. pp.12, Boston: Houghton Mifflin. 1967.

Huang G.Q, Huang J and Mak K.L. Early Supplier Involvement in New Product Development on the Internet: Implementation Perspectives, CONCURRENT ENGINEERING: Research and Applications, Vol 8, No. 1, pp. 40-50. 2000.

The Design Structure Matrix, HTTP://WEB.MIT.EDU/DSM/2003.

CATIA Team PDM 4.0, HTTP://WWW.IBM.COM/2003.

SmarTeam, HTTP://WWW.SMARTEAM.COM/2003.

Lake, J. Systems engineering re-energized: Impacts of the revised DoD acquisition process, Engineering Management Journal, 4(3), pp. 8-14. 1992.

Lee, R. J. V and Young, R. I. M. Information supported design for manufacturing of injection-moulded rotational products, International Journal of Production Research, Vol. 36, No. 12, pp. 3347-3366. 1998.

Lee Rong-Shean, Chen Yuh-Min and Lee Chang-Zou. Development of a concurrent mold design system: a knowledge-based approach, Computer Integrated Manufacturing Systems, Vol.10, No.4, pp. 287-307. 1997.

Kusiak A, Larson N.T. and Wang J. REENGINEERING OF DESIGN AND MANUFACUTING PROCESS, *Computers ind. Engng*, Vol. 26, No. 3, pp 521-536, 1994.

Martin, J. Principles of Object-Oriented Analysis and Design. pp. 241-261, NJ: Prentice-Hall. 1991.

Medhat, S. (ed). Concurrent Engineering, The Agenda for Success. New York: Wiley. 1997.

Menges, G. and Mohren, P. How to Make Injection Molds, 2nd Ed. pp. 1-100, New York: Hanser Publishers. 1993.

Meredith D.D, Wong K.W, Woodhead R.W, Wortman R.H. Design Planning of Engineering Systems. NJ: Prentice-Hall. 1985.

Monplaisir L, and Singh N. (ed). Collaborative Engineering for Product Design and Development. CA: American Scientific Publishers, 2002.

Nidamarthi S, Alllen R.H and Sriram R.D. Observations from supplementing the traditional design process via Internet-based collaboration tools, Computer Integrated Manufacturing, Vol. 14, No. 1, pp. 95-107. 2001.

Pallot M, and Sandoval V. Concurrent Enterprising: Towards the Concurrent Enterprise in the Era of the Internet and Electronic Commerce. pp. 65-100, Boston: Kluwer Academic Publishers. 1998.

Park H and Cutkosky M.R. Framework for Modeling Dependencies in Collaborative Engineering Processes, Research in Engineering Design, 11, pp. 84-102. 1999.

Potsch, G. and Michaeli, W. Injection Molding: An Introduction. pp. 80-103, New York: Hanser Publishers. 1995.

Simon, H.A. The structure of ill-structured problems, Artificial Intelligence, 4, pp. 145-180. 1973.

Steward D.V. The design structure system: A method for managing the design of complex systems, IEEE Trans Eng Manage, 28, pp. 71-74. 1981.

Steward D.V. Systems Analysis and Management: Structure, Strategy and Design. New York: Petrocelli Books. 1981.

Syan, C. S, and Menon, U. (ed). Concurrent Engineering, Concepts, Implementation, and Practice. pp. 3-46, London: Chapman & Hall. 1994.

Tkach, D, and Puttick, R. Object Technology in Application Development. pp. 51-70, 125-130, Redwood City, Calif: Benjamin/Cummings Pub. Co. 1994.

Truss, J.K. Discrete Mathematics for Computer Scientists Reading. pp. 369-380, Mass: Addison-Wesley. 1999.

Vorwerk, R. Towards a True OBBMS. Object Magazine, 3, 5, pp. 38-39. 1994.

Wiest, J.D and Levy, F.K. A Management Guide to PERT/CPM. N.J.: Prentice-Hall. 1969.

APPENDIX A

THE INTERPRETATION OF THE INTERACTION AMONG INJECTION MOLD DESIGN ACTIVITIES

Generally speaking, the mold design can be divided into several phases: the initial design phase, the detailed design phase, and the testing and modification phase. The initial design phase starts with the mold design when the mold designer receives the specification of plastic parts and basic information from the plastic product designers or customers. The injection molding machine information such as machine plasticating rate, clamping force, the maximum injection pressure are sometimes included. The mold designer compensates the product geometry to account for material shrinkage during the molding operation. This compensation used to be accomplished manually, but it can now be done with the aid of computer analysis software, such as mold flow analysis. With the analysis software, the mold designer can predict flow pattern and obtain useful information for the design of cavity layout, runner system (runner, gating, sprue), cooling layout, etc.

After the CAD model of the plastic part has been approved and analyzed for its moldability, the mold cavity layout design starts. The mold layout design usually includes the determination of cavity numbers and the cavity layout design. The Cavity number selection determines whether a single- or multiple-cavity mold should be used, and the layout design determines the positioning and arranging of cavities in a mold. Decisions at this stage are mainly based on customers' requirements on production, cost, and lead-time, etc. With the wide application of standard components in injection molding manufacturing, mold cavity layout is also affected by standard mold base information provided by suppliers. While determining the cavity layout, the mold designer locates the initial parting line to divide the molded part into cavity and core parts. This decision mainly depends on the geometry of the molded part. The detailed parting location is drafted in the detailed mold design phase when the design of detailed cavity and core is carried out. This may result in the modification of the number and location of parting lines that are initially defined.

Once the mold cavity layout is determined, the mold designer proceeds to select the appropriate mold base for mold assembly. Selecting the type of mold base is an important task in mold design. Beside the decision information such as cavity number and cavity layout that are determined in the previous steps, the mold designer also needs to take into consideration the information such as the availability of mold making and molding equipments, molding pressure, production requirements, molded part dimensions and mold structure complexity etc. The standard mold base is now widely used in mold making industry. Its data library can be found in commercial CAD/CAM programs like Unigraphics, ProEngineer, and SolidWorks. After the mold designer selects the appropriate mold base from the data library, he/she merges the product CAD file into it. If the standard mold base is not used, the mold designer will then construct a customized mold base to appropriately accommodate the dimensions and geometry of the molded part.

The initial mold design requires the assistance of the mold maker and the standard mold base supplier. The moldmaker checks on the product CAD file after it is merged with the standard mold base. By doing this, experienced mold makers can give advice on factors such as parting line location, slider or lifter location, runner location, gate location and size, etc. Based on these suggestions, the necessary modifications on the initial mold design can be addressed, and the correctness in detailed mold design can be improved. The mold base suppliers provide the standard mold base information for the mold design. The comprehensive mold base libraries could prompt the mold designer to make the appropriate selection.

The detailed mold design is focused in the design of mold assembly components. These components are detailed core and cavity insert, slider, lifter, runner system-sprue, runner, gate, ejector system and cooling channel etc. The detailed design of core and cavity insert normally includes detailed core and cavity structure design and detailed parting line location determination. The detailed parting line location can be determined when information about flow pattern, runner system layout, the location of sliders and lifters etc are available. The finalized parting line location may require changes to the initially selected location of parting line. The runner system is the channel system in a mold that directs the flow of molten plastic into the cavities. Its configuration, dimensions, and connection with the molded part affect the filling process considerably and thus influence the quality of the molded part greatly. There are some factors that a mold designer needs to consider in the runner system design. First of all, the choice of hot runner or cold runner to be used in the mold needs to be determined before the runner layout is drafted. This decision is mainly based on the customer requirements, cost and final product appearance. In turn, the use of hot or cold runner will affect the sprue selection, runner layout and design, and the subsequent mold making process planning etc. This is because hot runner systems are mainly made with standard components; however, cold runner systems are mainly customized by the mold maker. The cold runner system design is a highly interrelated design process in determining the main components of the sprue, runner and gate. Necessary compromise has to be made so that the whole runner system can function normally and adapt to the production requirements. The principle for runner system components design can be summarized as:(1), The size of the sprue should not speed up the pressure drop and impair its ability in distributing material to the extreme point; (2), The runner size should be kept small in comparison to the size of plastic part and also kept balanced with respect to the pressure loss, heat exchange efficiency, and material saving; (3), Gate design should meet the principle that the molding or molded part should not exhibit blemishes or be distorted due to the connection with gate. Other factors concerned with cold runner system design include wall thickness of part, filling pressure and temperature, cooling time, cost of manufacturing, material shrinkage, etc.

The ejector system design and the cooling system design are two other important tasks in the detailed mold design. The ejector system is composed of pins and plates that are used to remove the part from the mold after it has solidified and cooled down. The ejector system design is mainly based on the mold designer’s understanding about the ejector mechanisms. In commercial standard mold base data library, the mold base usually includes the ejector plate, upon which the mold designer incorporates the ejector pins. The cooling system in a mold is used to cool the molten plastic in the mold cavities. Its effectiveness in heat exchanges between the injected plastic and the mold is a decisive factor in the quality and appearance of the final product. Major considerations in the cooling system design therefore include the size and the location of the cooling channel, filling velocity, cooling time, the thermo properties of polymer, etc. Meanwhile, in mold components design and ejector design, dimension allowances should be made for the proper sizing and positioning of the cooling channel. Information exchange is therefore required between cooling system design and these activities.

The outcome of the detailed mold design is a mold assembly with all the necessary components fitted together. The last phase in the injection mold design is therefore the testing and modification on the first article of the mold assembly. The testing and modification check whether the first article achieves the customer's requirements on appearance and performance etc. Modifications are made directly on the mold assembly. After the modifications are approved and the customer accepts the first article, the related information will be sent to the mold designer for updating of the mold CAD file.

APPENDIX B

APPLICATION OF DSM IN INJECTION MOLD DESIGN PROCESS

Based on the logic of Kusiak's algorithm, the process partitioning can be performed on the injection mold design process as follows.

Step 1. Set $i \leftarrow 1$, $j \leftarrow 1$, $L(1) \leftarrow \varnothing$, $C(1) \leftarrow \varnothing$, $OA \leftarrow \varnothing$, $E \leftarrow \varnothing$, and $O(k) \leftarrow 1$ for k = 1, .., 15.

Step 2. Since $A \neq \varnothing$, go to Step 3.

Step 3. Since $OA = \varnothing$, go to Step 4.

Step 5. $SA \leftarrow \{15\}$; Delete all entries associated with activity 15 from matrix A; go to Step 6.

Step 6. Find a cycle {1, 2, 1}. Set $E = \{1, 2\}$, and go to step 7.

Step 7. Since $C \cap E = \emptyset$, set $C(1) \leftarrow \{1, 2\}$, set $j \leftarrow 2$, $C(j) \leftarrow \emptyset$; $O(C(1)) \leftarrow \mathrm{Max}\{O(1), O(2)\} = 1$; set $E \leftarrow \emptyset$ and go to Step 2.

Step 2. Since $A \neq \varnothing$, go to Step 3.

Step 3. $OA \leftarrow \{C(1)\}$; since $OA \neq \varnothing$ and $C(1) \in C$, set $L(\mathrm{O(C(1))}) \leftarrow L(\mathrm{O(C(1))}) \cup C(1) = \{1, 2, 1\}$, where $\mathrm{O(C(1))} = 1$.Go to Step 4.

Step 4. Delete all entries associated with $C(1)$ from matrix A and go to Step 2.

Step 2. Since $A \neq \emptyset$, go to Step 3.

Step 3. $OA \leftarrow \{3\}$; since $OA \neq \emptyset$ and $\{3\} \notin C$, set $L(2) \leftarrow \{3\}$ and go to Step 4.

Step 4. Delete all entries associated with activities 3 from matrix A; set $O(4) = O(5) = O(6) = O(7) = O(9) = O(10) = O(13) = O(14) = 3$; since $L(2) \neq \emptyset$, set $i \leftarrow 3$ and $L(3) \leftarrow \emptyset$; $OA \leftarrow \emptyset$; go to Step 2. (See Figure 3.6 b)

Step 2. Since $A \neq \emptyset$, go to Step 3.

Step 3. Since $OA = \emptyset$, go to Step 5.

Step 5. Since $SA = \emptyset$, go to Step 6.

Step 6. Find a cycle {4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 4}. Set E = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 4}, and go to Step 7.

Step 7. Since $C \cap E = \emptyset$, set $C(2) \leftarrow$ {4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 4}, set $j \leftarrow 3$, $C(j) \leftarrow \emptyset$; $O(C(2)) \leftarrow \text{Max}\{O(4), O(5), O(6), O(7), O(8), O(9), O(10), O(11), O(12), O(13), O(14)\} = 3$; set $E \leftarrow \emptyset$ and go to Step 2.(See Figure 3.6 c)

Step 2. Since $A \neq \emptyset$, go to Step 3.

Step 3. $OA \leftarrow \{C(2)\}$; since $OA \neq \emptyset$ and $C(2) \in C$, set $L(\mathrm{O(C(2))}) \leftarrow L(\mathrm{O(C(2))}) \cup C(2)$ = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 4}, where O(C(2)) = 3. Go to Step 4.

Step 4. Delete all entries associated with $C(2)$ from matrix A and go to Step 2.

Step 2. Since $A = \emptyset$, stop.

Injection Mold Design Activities		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$O(k)$
A	1		1														1
B	2	1															1
C	3	1															1
D	4	2	1	1													1
E	5	3	3	1	3		2										1
F	6		1	1		1								3			1
G	7		1	2	1	1	1										1
H	8		1				1			3	2			3			1
I	9	2	1	3	2	1	1	2	1		1	2		3			1
J	10		2	2	2	1	3		2	1		2		3			1
K	11		2		2	1		1	2	1	3		3		1		1
L	12		1		3		1					1		3	2		1
M	13			3		3	1	2	2	2	2		1		1		1
N	14			3		1	1		3			2	2	3			1
O	15						1										1

(a)

Injection Mold Design Activities		1	2	3	4	5	6	7	8	9	10	11	12	13	14	$O(k)$
A	1		1													1
B	2	1														1
C	3	1														2
D	4	2	1	1												2
E	5	3	3	1	3		2									2
F	6		1	1		1								3		2
G	7		1	2	1	1	1									2
H	8		1				1			3	2			3		2
I	9	2	1	3	2	1	1	2	1		1	2		3		2
J	10		2	2	2	1	3		2	1		2		3		2
K	11		2		2	1		1	2	1	3		3		1	2
L	12		1		3		1					1		3	2	2
M	13			3		3	1	2	2	2	2		1		1	1
N	14			3		1	1		3			2	2	3		1

(b)

Injection Mold Design Activities		3	4	5	6	7	8	9	10	11	12	13	14	$O(k)$
C	3													2
D	4	1												3
E	5	1	3		2									3
F	6	1		1								3		3
G	7	2	1	1	1									3
H	8				1			3	2			3		1
I	9	3	2	1	1	2	1		1	2		3		3
J	10	2	2	1	3		2	1		2		3		3
K	11		2	1		1	2	1	3		3		1	1
L	12		3		1					1		3	2	1
M	13	3		3	1	2	2	2	2		1		1	3
N	14	3		1	1		3			2	2	3		3

(c)

Injection Mold Design Activities		4	5	6	7	8	9	10	11	12	13	14	$O(k)$
D	4												3
E	5	3		2									4
F	6		1								3		3
G	7	1	1	1									4
H	8			1			3	2			3		3
I	9	2	1	1	2	1		1	2		3		4
J	10	2	1	3		2	1		2		3		4
K	11	2	1		1	2	1	3		3		1	4
L	12	3		1					1		3	2	4
M	13		3	1	2	2	2	2		1		1	3
N	14		1	1		3			2	2	3		3

(d)

Injection Mold Design Activities		5	6	7	8	9	10	11	12	13	14	O(k)
E	5		2									4
F	6	1								3		4
G	7	1	1									4
H	8		1			3	2			3		4
I	9	1	1	2	1		1	2		3		4
J	10	1	3		2	1		2		3		4
K	11	1		1	2	1	3		3		1	4
L	12		1					1		3	2	4
M	13	3	1	2	2	2	2		1		1	4
N	14	1	1		3			2	2	3		4

(e)

Figure B.1 Process partitioning on the injection mold design process

After the process partitioning, it is noted that quite a number of mold design activities interact within a large and single information circuit (see Figure 3.12 e). To break this information block and optimize the sequence activities involved in this block, the process tearing algorithm is deployed on this information circuit.

Figure 3.13 demonstrates the process tearing step by step as follows:

Step 1. Remove all the 3's marks from the matrix block shown in Figure 3.12 e, and partition the new matrix to get a new order (see Figure 3.13 a - e).

Injection Mold Design Activities		5	6	7	8	9	10	11	12	13	14	$O(k)$
E	5		2									1
F	6	1										1
G	7	1	1									1
H	8		1				2					1
I	9	1	1	2	1		1	2				1
J	10	1			2	1		2				1
K	11	1		1	2	1					1	1
L	12		1					1			2	1
M	13		1	2	2	2	2		1		1	1
N	14	1	1					2	2			1

(a)

Injection Mold Design Activities		5	6	7	8	9	10	11	12	14	$O(k)$
E	5		2								2
F	6	1									2
G	7	1	1								3
H	8		1				2				3
I	9	1	1	2	1		1	2			3
J	10	1			2	1		2			3
K	11	1		1	2	1				1	3
L	12		1					1		2	3
N	14	1	1					2	2		3

(b)

Injection Mold Design Activities		7	8	9	10	11	12	14	$O(k)$
G	7								3
H	8				2				3
I	9	2	1		1	2			4
J	10		2	1		2			3
K	11	1	2	1				1	4
L	12					1		2	3
N	14					2	2		3

(c)

Injection Mold Design Activities		8	9	10	11	12	14	$O(k)$
H	8			2				4
I	9	1		1	2			4
J	10	2	1		2			4
K	11	2	1				1	4
L	12				1		2	4
N	14				2	2		4

(d)

Injection Mold Design Activities		5	6	7	8	9	10	11	12	14	13
E	5		2								
F	6	1									
G	7	1	1								
H	8		1				2				
I	9	1	1	2	1		1	2			
J	10	1			2	1		2			
K	11	1		1	2	1				1	
L	12		1					1		2	
N	14	1	1					2	2		
M	13		1	2	2	2	2		1	1	

(e)

Figure B.2 The 1st stage of process tearing being performed in the injection mold design process

Step 2. Remove all the 2's marks from the matrix block shown in Figure 3.13 e, and partition the new matrix to get a new order (See Figure 3.14).

Injection Mold Design Activities		5	6	7	8	9	10	11	12	14	13	$O(k)$
E	5											1
F	6	1										2
G	7	1	1									2
H	8		1									1
I	9	1	1		1		1					2
J	10	1				1						2
K	11	1		1		1				1		2
L	12		1					1				1
N	14	1	1									2
M	13		1						1	1		1

(a)

Injection Mold Design Activities		6	7	8	9	10	11	12	14	13	$O(k)$
F	6										2
G	7	1									2
H	8	1									1
I	9	1		1		1					2
J	10				1						2
K	11		1		1				1		2
L	12	1					1				1
N	14	1									2
M	13	1						1	1		1

(b)

Injection Mold Design Activities		6	7	8	9	10	11	12	14	$O(k)$
F	6									2
G	7	1								3
H	8	1								3
I	9	1		1		1				3
J	10				1					2
K	11		1		1				1	2
L	12	1					1			3
N	14	1								3

(c)

Injection Mold Design Activities		7	8	9	10	11	12	14	$O(k)$
G	7								3
H	8								3
I	9		1		1				3
J	10			1					2
K	11	1		1				1	2
L	12					1			3
N	14								3

(d)

Injection Mold Design Activities		7	8	9	10	11	14	$O(k)$
G	7							3
H	8							3
I	9		1		1			4
J	10			1				2
K	11	1		1			1	4
N	14							3

(e)

Injection Mold Design Activities		9	10	$O(k)$
I	9		1	4
J	10	1		4

(f)

Injection Mold Design Activities		5	6	7	8	14	9	10	11	12	13	
E	5		2									$L(1)$
F	6	1									3	$L(2)$
G	7	1	1									$L(3)$
H	8		1				3	2			3	$L(3)$
N	14	1	1		3				2	2	3	$L(3)$
I	9	1	1	2	1			1	2		3	$L(4)$
J	10	1	3		2		1		2		3	
K	11	1		1	2	1	1	3		3		$L(5)$
L	12		1			2			1		3	$L(6)$
M	13	3	1	2	2	1	2	2	1	1		$L(7)$

(h)

Figure B.3 The 2nd stage of process tearing being performed in the injection mold design process

The process tearing rearranges the activities involved in the circuit and provides a possible ordering for the execution of these activities. Together with the analysis of process partitioning, the process sequence of injection mold design could be obtained in Figure 3. 12.