The Fourth International Conference on Control and Automation (ICCA'03), 10-12 June 2003, Montreal, Canada

Standard Component Library Design and Implementation for Plastic Injection Mold Design with a CAD Tool

Y. –S. Ma* G. A. Britton, S. B. Tor,
E. Gunawan C. H. Lee
School of Mechanical and Production Engineering,
Nanyang Technological University
Nanyang Avenue, Singapore 639798
*E-mail: mysma@ntu.edu.sg

Abstract

This paper presents the design of a 3D parametric Standard Component Library (SCL) based on a CAD tool. This library can effectively generate standard components automatically from different suppliers based on Constructive Solid Geometry (CSG) method. Group Technology (GT) method is applied to control the geometry configurations of component models. A prototype has been developed to explore the feasibility of the proposed design with modular feature primitives internally without using pre-defined templates. The new design has demonstrated promising advantages compared to an existing library. Future research and development are discussed.

1 Introduction

In plastic injection molding industry, the time-to-market is more and more compressed. This has created a great challenge in building molds with CAD tools. An injection mold consists of many standard components in different sub-assemblies, such as moldbase plates, locating rings, sprue bushings, ejector pins, hose nipples, and plugs. The similarity of the structural build-up of molds makes certain standardization desirable. This permits standard components to be produced in quantity and thereby reduce manufacturing costs. Many companies, such as DME, HASCO, FUTABA, MISUMI, provide services in standard mold units for the injection mold industry. Some CAD systems for injection mold design have incorporated the standard mold units into their libraries, which are considered as one of the most important features of such software. However, if the library is not designed in a condensed and efficient manner, great amount of data storage and development time are required implement such libraries for CAD systems. An effective and efficient design of standard component library (SCL) is therefore needed.

2 Literature Review

Traditional research works in plastic mold design are concentrated on conceptual design with early engineering decisions [1, 2]. More recently, mold structure design tools are emphasized. Jin et al [3] proposed an automatic pin ejector design system, which consists of four modules, the calculations of the ejection forces, the minimum dimension of the ejector pin, and the number of ejector pins, and the design of the ejector grid. Lee et al [4] and Fu et al [5] worked on a systematic approach to generate mold systems based on certain practical design procedures. Their effort covers several function subsystems, such as feeding, cooling, ejection, local tools, and standard subsystems. Mok et al [6] presented a prototype knowledge-based system, called IKMOLD, for mold assembly generation. In this system, the computational, knowledge-based and CAD-based modules were integrated for generating mold features. They also implemented a Group Technology (GT) method for coding mold configurations with reference to Opitz coding system. However, it seems that little was done on the design of the standard component library. Peklenik and Grum [7] reported on a design database used in CAD/CAM/CAPP applications with particular reference to group technology (GT) principles, but parametric modeling was not considered.

QuickMould is a set of 3D CAD software tools for mold design [8]. QuickMould has a Standard Component Library (SCL) module [9]. With object technology, each type of standard components is categorized as an element in the library. Geometrical configurations, such as feature alterations are embedded into each element [9]. An element of SCL consists of three related files: a predefined CAD template, a configuration file, and a data file (see Figure 1).

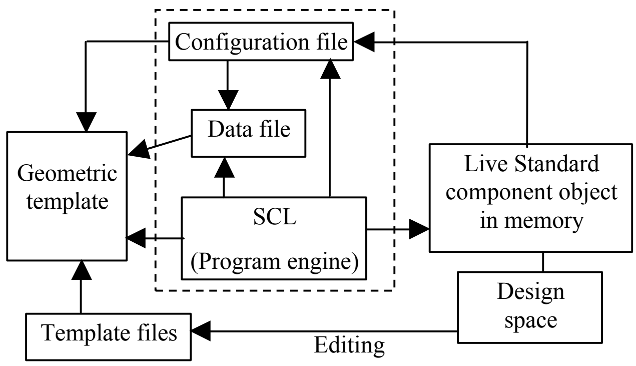


Figure 1 QuickMould SCL components

If user selects a type of components from a supplier whose catalogue has been incorporated in the SCL, a predefined standard component template will be loaded. The program engine will retrieve the configuration file and the data file for the respective component. Then the object representing this component is established, which includes feature configuration options, methods for the geometry updating, different sizes, and the values for the dimensions required. The component loaded can be modified and updated through object methods by updating low-level CAD expressions automatically with the expression reference mechanism [10].

The limitation of the QuickMould SCL design is that even though the number of CAD templates has been reduced significantly, considerable modeling effort and a strict configuration convention is still required for modeling the pre-defined template interactively for every type of standard components.

3 New SCL Design

In order to overcome the above limitation of QuickMould SCL, in this paper, a new generative approach for standard component library is proposed. The idea is to generate common standard component geometry with parametric feature primitives in an object-oriented approach. This design has to be generic because knowledge-based design procedures must be applied to solve problems that require significant human expertise for the solutions. Self-contained and consistent object methods are defined for different library components. The library functions are designed to allow users to create, edit, query, and destruct the standard components. On the other hand, library elements need to be easily developed and extended since the catalogues of suppliers will be updated regularly while in-house-made components are to be incorporated too. Considering the potential application of this library for concurrent engineering, library elements are required to be information rich such that all the information of each standard component can be easily extracted and made available to different applications at different levels.

Figure 2 shows the new design of the SCL, which is a modification based on the QuickMould SCL [9]. A pre-defined template is not required to create a library element. The new design explores the capability of CAD application programming interface (API) functions that can be called under an object-based environment to generate standard component models. The CAD API functions have been designed to enable an easy integration between CAD functions and external user-defined application programs [10]. They can be called in either C or C++ programming languages.

In this work, the application program creates the geometry of a component model internally with the CAD API functions. One advantage to this is that the executables are much smaller and they can be compiled much faster. Once an internal application program is loaded into memory, it stays for the remainder of the CAD session. Then it will run interactively, giving the user opportunities to see the results that the application has on displayable objects in the standard component model. The geometry creations of the model are composed directly by the program via a GT coding scheme. By generating the standard component using API functions internally, pre-defined part templates are eliminated.

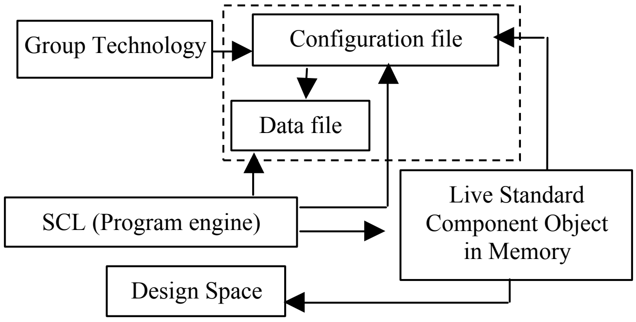


Figure 2 Proposed SCL design

The architecture proposed for the new SCL design is based on Constructive Solid Geometry (CSG) method which is simply a combination of feature solid primitives, such as cuboids, cylinders, spheres, cones, and so on. The set of operations in CSG method are **union** (encompassing all points in space that are contained in the primitives A OR B), **intersection** (encompassing all points contained in A AND B) and **difference** (encompassing all point in shape contained in A – B). Most standard component geometries are generally a combination of simple primitives.

Ejector pins have been chosen as the component type for prototyping. They have the basic shape of cylinder and, are one of the most common types of standard components and available in all the supplier's catalogs. Furthermore, the proposed solution can be extended for other types of components such as core pins, center pins, and sprue bushings. The hierarchal construction to achieve the final model of an ejector pin is shown in Figure 3. The construction of an ejector pin is initialized from the very basic primitive, which is the cylindrical flange. This is the first level primitive of the ejector pin. Then it is connected to second level primitives, followed by the third level. Finally, other geometry deviations and alterations will be assigned to the model according to the type of ejector pins. In a component, all feature primitives have topological relations to one another. Such topological relations are modeled in the form of a graph. Other relations like dimensions and tolerances are also modeled in modular manner in which they can be configured accordingly for every supplier.

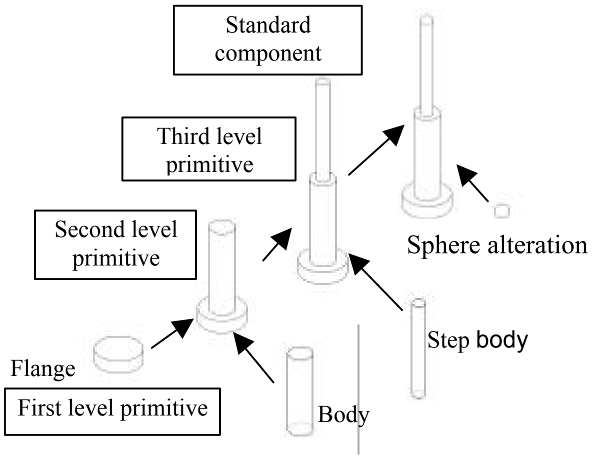
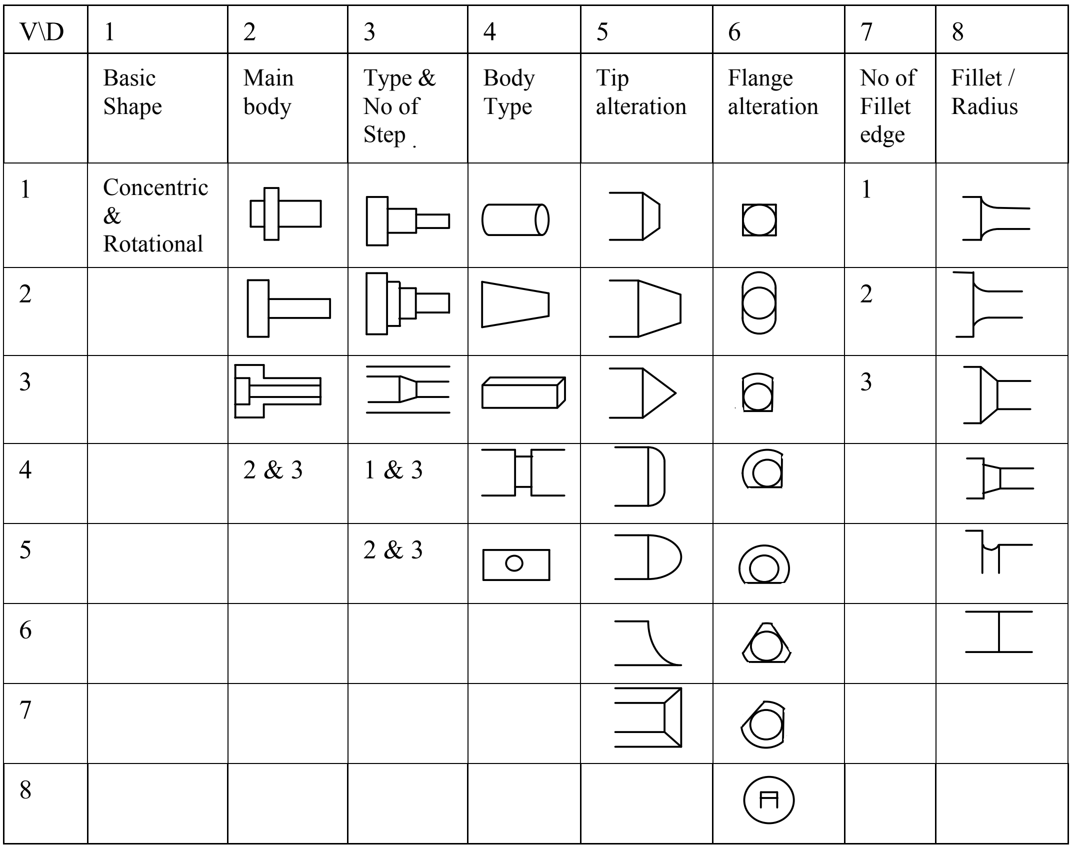


Figure 3 Constructive Solid Geometry of a stepped pin

The group technology method implemented is based on Wilson et al [11] and Hyde et al's [12] approach. It is designed that the primitives of standard components from different supplier catalogs are modeled in a matrix form with particular reference to a classification coding system. The GT method used in coding the standard components selected by the user is crucial since the code represents the necessary information about the shape of standard components and the construction methods to be called. The classification method used in grouping standard components is based on the configurations of features, including shapes, sizes and other physical characteristics. Figure 4 illustrates the matrix for ejector pins. Each code consists of a certain number of digits (shown as the columns in Figure 4). In general, each digit represents a specific feature of the standard component. If the corresponding feature of a digit does not exist in a component, zero is assigned to the digit. For example, the first digit of the matrix represents the base body shape, such as a concentric cylinder in ejector pins. As usual, digits in the front represent major features while the rest represent minor alterations.

In the case of ejector pins, the first digit indicates the cylindrical base feature type, but the sixth digit indicates flange alterations. Since code numbers are assigned on the basis of its configurations, two standard components with similar shapes will have similar code numbers and vice versa. For a free flange stepped ejector pin as shown in figure 5, the code is "11110012". The last two digits in the code represent the type of fillet in the ejector body. Note that the second last digit indicates one fillet and the last digit represents the type of fillet respectively.



V\D	1	2	3	4	5	6	7	8
	Basic Shape	Main body	Type & No of Step	Body Type	Tip alteration	Flange alteration	No of Fillet edge	Fillet / Radius
1	Concentric & Rotational						1	
2							2	
3							3	
4		2 & 3	1 & 3					
5			2 & 3					
6								
7								
8								

Figure 4 Group Technology matrix for ejector pins (D: Digit, V: Value)

This coding system can be expanded automatically by applying a "polycode" group for each nominal "digit". For example, digit 7 (number of fillets) will determine the selection digit 8. If there are 3 fillets as indicated by digit 7, then the fillet type code must consists of 3 digits, which is treated as "digit 8 group". In fact, the scheme is also organized in the form of a graph tree. For the rectangular pin shown in figure 6, the code assigned is "12030000".

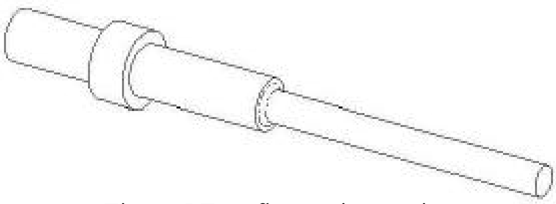


Figure 5 Free flange ejector pin

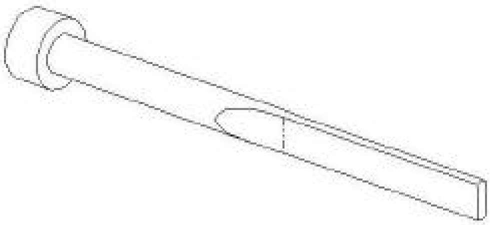


Figure 6 Rectangular ejector pin

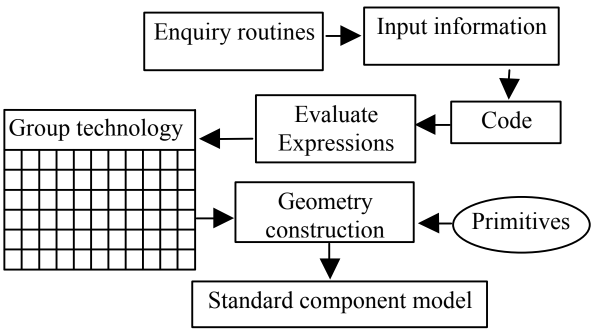


Figure 7. Implementation of GT in feature configuration

This matrix can also be expanded to represent other types of pins such as core pins, and center pins. In addition, the matrix can also be used to compose the particular attributes, such as dimension, tolerance, and combination of primitives. Figure 7 shows the implementation of GT into the SCL design proposed in this project. After user has selected necessary options about a component, such as supplier, major type, sub-type, and all the alterations, the component configuration is then represented with a classification code. The corresponding attributes and values are retrieved from the configuration and data files. The corresponding object is then established and the geometry generated. One of the roles played by GT codes is to control the geometry construction methods for the standard component library, because they are organized according to the code input from the user interfaces.

4 Implementation of "SCL"

"SCL" (Standard Component Library) program is written using Visual C++ with UG/Open API internally. A constructive solid geometry method is implemented in the program. Figure 8 shows the flow chart. In creating the element geometry models of standard components, the implementation has showed that the implementation of the GT method into object-oriented feature generation methods is promising to increase program reusability.

However, due to the time constraints, the UIs of this prototype were not organized according to GT codes with object technology. As experienced by QuickMould development [8, 9], generic UIs with consistent object definitions would be able to display multiple selections (supplier, category, major type, subtype, and alterations) in a dynamic interface configuration.

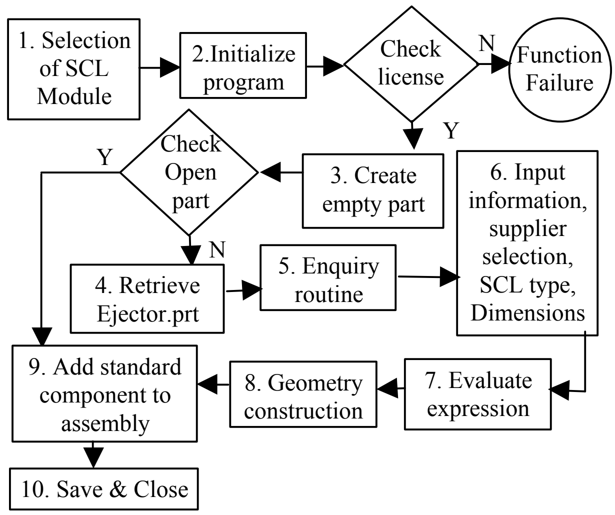


Figure 8 The flow chart for creating ejector pins

5 Discussion

The combination of the primitives in the program allows the user to create many variations of standard components from different suppliers. The design based on object-oriented geometry methods is generic since all the primitives are coded only once and they can be reused again and again to create features with different configurations. This design has eliminated the need of pre-defined templates. The program is expandable by including more primitive routines. Therefore other standard family components can be incorporated into this SCL.

The pre-defined templates approach [9] requires considerable amount of effort and training to develop CAD templates. The new generative SCL design offers some advantages. Firstly, the GT method is a generic method in controlling the geometry creation of the standard component, which can cater for many types of standard component. Secondly, the classification code for a component can be used for indexing, searching and organizing modular methods. However, matching supplier specific dimensions to the generic generation parameters can be a tricky problem. In this sense, automatic generation approach is not as flexible and universal as template-based approach. Ideally, a hybrid approach can be developed.

6 Future Work and Conclusion

Future research and development include 3 areas (Figure 9). **(1)** User-defined primitives should be enabled such that the existing user designed features can be converted into model generation subroutines automatically. **(2)** Data files should be converted into a database; hence the large amount of the data records can be better managed. **(3)** Lastly, the design of this SCL should be extended into a web-based environment. Extensible Markup Language (XML) can be used to communicate and share design information [13].

In conclusion, a generative approach for the development of a Standard Component Library (SCL) has been presented. The new approach uses a Group Technology method. This approach offers improvements over the template-based approach. A prototype has been developed with a simple GT coding matrix, and it has demonstrated the promising ability to generate standard component models with modular generic methods. The GT method makes the object developed more reusable across different component types.

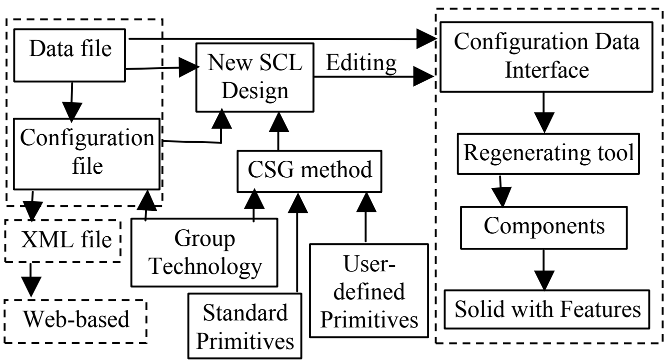


Figure 9. Proposed future research and development

Acknowledgment

QuickMould research project was sponsored by Singapore Infocom Development Authority, and jointly developed by NTU and SIMT. Currently, QuickMould is being used as a research package in NTU.

References

[1] J. Borg, and K.J. MacCallum, "A Hypercad Expert System For Plastic Product Design", Conference on Computer Integrated Manufacturing, vol. 1, pp.295-302, 1995.

[2] K.S. Chin, and T.N. Wong, "A Framework To Develop An Expert Injection Mold Planning System For Early Product Design Decision", Conference on Computer Integrated Manufacturing, vol. 1, pp. 303-310, 1995.

[3] S. Jin, S.B. Tor, and G.A. Britton, "A Proposed Automatic Pin Ejector System Design For Plastic Injection Molding", Conference on Computer Integrated Manufacturing, vol. 2, pp. 1125-1131, 1995.

[4] R.S. Lee, Y.M. Chen, and C.Z. Lee, "Development of a Concurrent Mold Design System: A Knowledge-Based Approach", Computer Integrated Manufacturing Systems, vol.10 no.4, pp. 287-307, 1997.

[5] M.W. Fu, J.Y.H. Fuh, and A.Y.C. Nee, "Core and Cavity Generation Method in Injection Molding". International Journal of Production Research 39, pp. 121-138, 2001.

[6] C.K. Mok, K.S. Chin, and K.L. Ho, "An Interactive Knowledge-Based CAD System for Mold Design in Injection Molding Processes". International Journal of Advanced Manufacturing Technology 17, pp. 27-38, 2001.

[7] J. Peklenik, J. Grum, B. Logar, and G. Hlebanja, "Design of the Database for CAD Based on Group Technology", Robotics & Computer-Integrated Manufacturing, vol. 4, no.1/2, pp. 49-62, 1988.

[8] QuickMould User's Guide, Singapore Institute of Manufacturing Technology, Singapore, 1999.

[9] Y.-S. Ma, S.B. Tor, and G.A. Britton, "The development of a Standard Component Library for Plastic Injection Mould Design Using an Object-oriented Approach", accepted by International Journal of Advanced Manufacturing Technology.

[10] EDS Inc., UG Documentation Help, Maryland Heights, MO, USA, 2000.

[11] R.C. Wilson, and R.A. Henry, "Introduction to Group Technology in Manufacturing and Engineering", Institute of Science and Technology, The University of Michigan, Ann Arbor, 1977.

[12] W.F. Hyde, "Improving Productivity by Classification, Coding, and Data Base Standardization: The Key to maximizing CAD/ CAM and Group Technology".

[13] J. Chung, K. Lee, "Computers in Industry: A Framework of Collaborative Design Environment for Injection Molding", 2002.