Design of a Feature-object-based Mechanical Assembly Library

Y. -S. Ma[1], G. A. Britton[2], S. B. Tor[3], L.-Y. Jin[4], G. Chen[5] and S. -H. Tang[6]

[1]Nanyang Technological University, mysma@ntu.edu.sg
[2]Nanyang Technological University, mgabritton@ntu.edu.sg
[3]Nanyang Technological University, msbtor@ntu.edu.sg
[4]iSi-Dentsu Singapore Pte. Ltd, liangyong@isidsgp.com.sg
[5]Nanyang Technological University, pg02198079@ntu.edu.sg
[6]Nanyang Technological University, pg0214852@ntu.edu.sg

ABSTRACT

In this paper, a new feature-based assembly library for injection mould bases is presented. Assembly design feature-object modeling and configuration management are the key technologies developed. This library is an important functional module of QuickMould, which is a productivity software tool for plastic injection mould design. Interactions between the library and mould design processes are realized. This paper also covers the details about the design of the system and the realization of some major functions, with which, the process of traditional mould design is simplified and standardized. Consequently, the lead-time for plastic injection mould can be significantly shortened.

Keywords: Assembly feature; Feature-based design; Injection mould design

1. INTRODUCTION

The procedures for injection mold design are complex and tedious. A lot of research has been done to improve the efficiency throughout the spectrum of injection mould design activities. In 1997, Kruth and Lecluse developed a system to support mould assembly design [1], which can only accommodate wire-frame and simple solid models. Due to the development of 3D parametric modeling, features have been defined and feature-based modeling accepted for product development. In the design stage, geometric features can be instanced, modified, and located quickly by specifying a minimum set of parameters, while the feature modeler works out the details [2, 3, 4]. Object-oriented modeling is a new way for assembly modeling [5, 6, 7, 8]. Ye et al [9] proposed a feature-based and object-oriented hierarchical representation for injection mould assembly. In recent years, with the development of computer technologies, more research works have contributed to the computer tools for injection mould design through different aspects [10, 11, 12, 13, 14, 15]. Knowledge-based systems represent another trend for future development [16, 17]. Currently, some commercial packages have been available in the market, such as iMould [18], IKMOULD [16], and MoldWizard [19]. However, there still exist some limitations of traditional generative modeling methods where geometry of a design is automatically created according to a set of "specific" rules implemented. Such systems are usually too rigid to be adopted in different enterprises or industries because customers' individual design semantics cannot be incorporated.

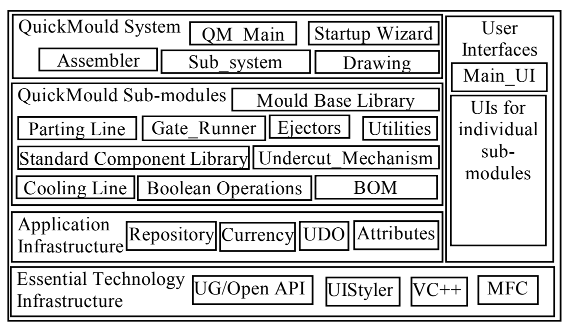

Fig. 1. QuickMould Architecture [20]

In this work, a new assembly feature-object approach is developed. It represents assembly design features and their topology in a more flexible and more convenient way in an assembly pattern library, via the association of a template file, a data file and a configuration file. With the feature-object method, pre-defined modular assembly models can be customized, added, modified, retrieved, and deleted more conveniently by the end-user without any programming effort. The library has been implemented as a functional module of

QuickMould, which was designed as a productivity software tool for plastic injection mould design. It was targeted to improve the effectiveness of design processes, and to significantly shorten the design time. QuickMould adopts a systematic approach for 3D plastic injection mould design with a set of functional modules to assist the designer in designing different subsystems of a mould assembly. These functional modules include, namely, Interactive Mould-base Library (IML), Interactive Gate and Runner, Undercut Mechanism, Standard Component Library [14], Interactive Cooling System [15], etc. Even though these modules perform separate functions, they are well integrated and can be switched from one to another by just the click of a button. Figure 1 shows the overview of QuickMOuld system architecture. This publication is regarding to the mould base library module, i.e. IML.

IML provides designer an efficient interface for the selection of mould bases from the 3D feature-based library by specifying the supplier, system, standard, type, size, and plate thickness. It also allows the designer to customize a standard mould base by changing the parameters of key components and features. Furthermore, customized mould bases can be added to the library for future retrieval. IML has implemented several catalogues from different commercial suppliers, such as FUTABA and DME. The parameters and features of mould base assemblies as well as components can be easily managed through software dialogue boxes. The feature-object modeling method enables systematic interactions between the library and the design environment.

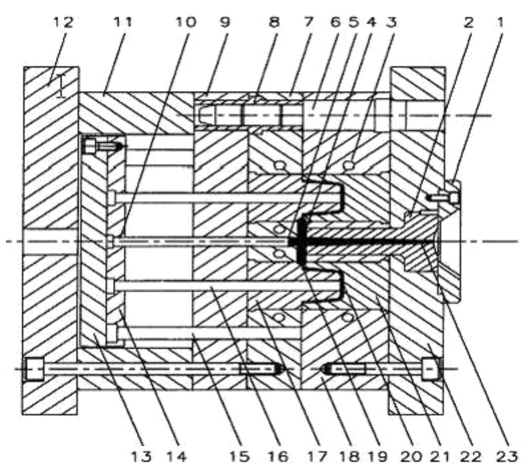

Fig. 2. An example mould assembly [24] (1: register ring; 2: sprue bush; 3: cooling channel; 4: runner; 5:cold slug; 6: guide pin; 7: core plate; 8: guide bush; 9: backing plate; 10: sprue ejector; 11: riser block; 12: bottom clamping plate; 13: ejector plate; 14: ejector retaining plate; 15: return pin; 16: ejector pin; 17: core insert; 18: cavity insert; 19: gate; 20: molded part; 21: cavity insert; 22: top clamping plate; 23: sprue)

2. MOULD BASE ASSEMBLY

A typical mould assembly is shown in Figure 2. The selection of mould base is a crucial process for the development of injection mould. Normally, many suppliers provide varieties of standard mould bases. A mould base assembly is traditionally classified into several subassemblies (see Figure 3), such as, namely fixed, movable and ejection subassemblies. Each subassembly is made up of other subassembly parts, or components, such as plates, pins, guiding bushes, and screws. Dimensional and fitting relationships exist among these parts and subassemblies. However, according to different requirements, it is necessary for the designer to customize certain configurations and dimensions. To check the design validity, analysis and evaluation during the different processes of mould design is also very important. For most mould companies, simplifying the design of mould bases offers the improvement of manufacturing productivity and product quality.

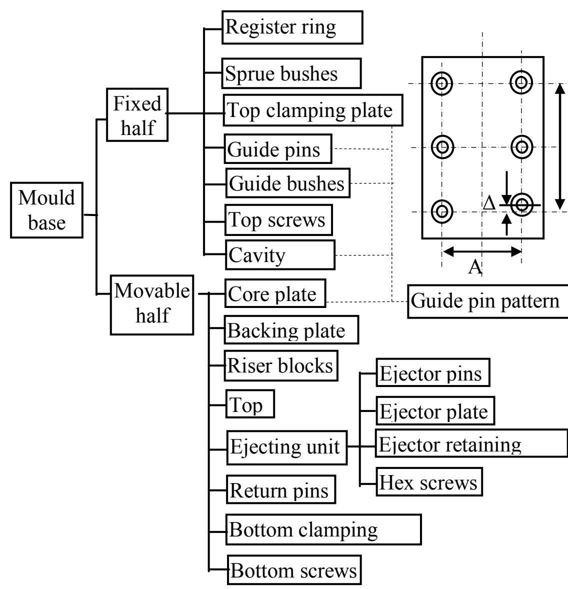

Fig. 3. An assembly design feature related to a mould base assembly structure

3. ASSEMBLY DESIGN FEATURE MODELING

Traditionally, most feature-based design works were limited in a part, or a component. In the context of assembly modeling, only a few research works were reported [21, 22, 23] and they focused on production assembly planning. As to assembly design features, such as assembly design patterns, no literature is found; neither for any feature-based assembly library. In this work, assembly design feature modeling is focused and a feature-object approach is developed for the design of a mould base library, because, in practice, it is quite common that a mould base assembly is used as a standard item and is analyzed or evaluated as a whole. The complex assembly relations that involve several subassemblies and components are embedded in the form of design patterns. Comparatively, it is easy to define features in a single component, but for assembly associative relations, there exists a lot of concise modeling work to be done.

For example, the size of the mould base affects the sizes of all the plates and the support/guide pins. The thickness values of plates affect other elements, such as pin lengths and their corresponding bushes or fitting holes on those plates. Figure 3 gives an example, where the components across subassemblies associated with a design pattern, i.e. guide pin pattern, are indicated with dashed lines.

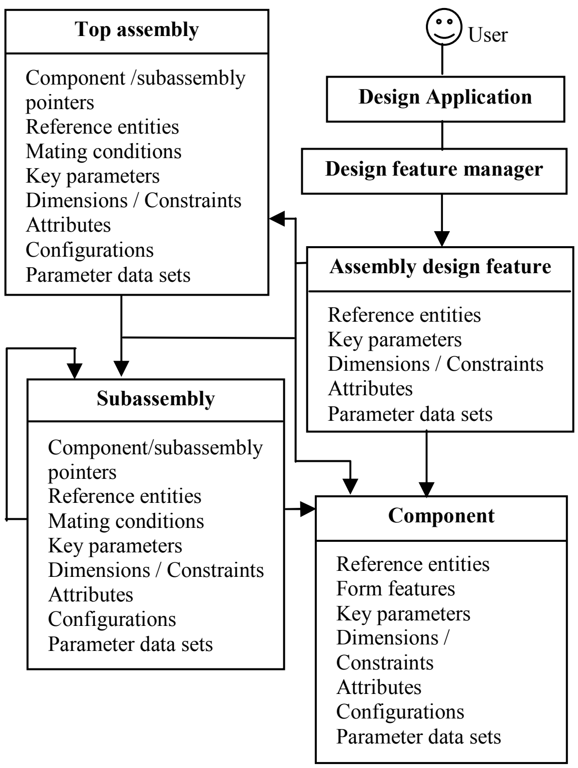

Fig. 4. The definition of assembly design features and relations created with other entities

Obviously, the pattern dimensions directly affect the positions and sizes of those linked components and their relevant features. Within the mould base assembly, there exist several such patterns, e.g. top screw pattern, plate thickness pattern, and so on. The associative nature cannot be easily generalized in a template form; a more explicit and flexible approach is needed.

In this work, such associative design relations, like design patterns, or dimensional relations, or assembly geometrical constraints are collectively defined as assembly design features. They are qualified with the following characteristics:

- These relations are for assembly design purpose with a clear design intent (any type of features is for a specific purpose);
- Such relations occur across different components or subassemblies;
- They are associated with certain assembly design semantics for the application.

The commonality of properties and behaviors can be defined with a common object class; hence, they should be well defined and self-contained. These assembly design features are another type of associative features as defined in [15]. It is difficult and unnecessary to apply hard constraints over all those dimensions or parameters; but it is rather appropriate to model the patterns in the object form. Figure 4 shows our proposed object definition for the assembly design feature and its relations with other geometrical elements, such as form features, components, subassemblies and the top assembly.

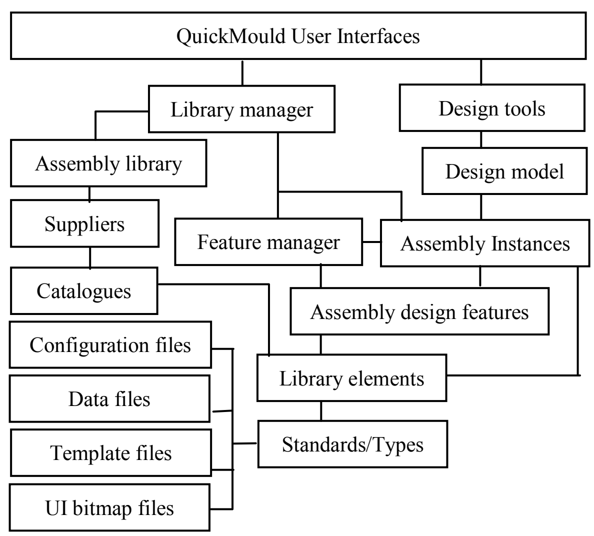

Fig. 5. Assembly library system design

The novelty here is that the assembly features are defined out of the assembly-subassembly-component framework but as separate entities associated with them. This definition provides the flexibility to model relationships across different members of the design. An assembly design feature uses components or other (sub-) assemblies in different layers, such as parameters, attributes, dimensions / constraints, etc. The pre-requisition is that they are also defined in object manner and are accessible via multiple layers. The object definition for components can be found in [14].

4. SYSTEM DESIGN

The system design for this library is shown in Figure 5. In this work, standard mould base catalogues are classified according to different suppliers, because they have different feature definitions and dimensional conventions. Each catalogue is subdivided according to different standards or types. For each standard, a CAD assembly template is built, together with a configuration file, a data file and the relevant UI bitmap files as shown in Figure 5.

The configuration file specifies the assembly features existed within the template and their corresponding parameters, dimensions as well as other attributes. Once the user confirms the selection of a standard, an instance of the assembly is inserted into the design model and its corresponding assembly features are registered by the feature manager, which in turn associate it with the corresponding library source via the library manager. This feature manager is also responsible to maintain the validity of each assembly features through a set of general methods, such as "*update_parameters()*" and "*evaluate_constraints()*", etc. UIs are also dynamically configured according to the user's selection and the bitmap files specified in the configuration file. Due to the object-oriented implementation, any modification within the given feature-editing environment will be verified against all the constraints related to the targeting entities, hence the feature consistency is assured. All feature attributes are associated to the geometrical entities that can be persistently maintained in the design model. Hence, their objects can be reestablished once the QuickMould program and the design models are loaded, and in turn they can be managed via the library and the feature managers.

5. SYSTEM IMPLEMENTATION

The system was developed based on and seamlessly integrated with UG V18.0, via UG/OPEN API, using C++ language. Figure 6 shows some of the application steps with connected UIs. When the library is invoked in the QuickMould environment, the program automatically initializes the session and checks for available catalogues, standards, types, as well as configurations, data and template files. Then the library resource tree is established which in turn supports the mould base library main UI as shown in the upper left corner of Figure 6. For mould design, if there is no existing mould base, designers can choose the mould base by specifying the suppliers, standards, and types (see Figure 6). Once specified, the corresponding assembly template model is retrieved from the library to the buffer together with the configuration file and data file specifying the available sizes, configurations, dimensional parameters and their default values. This buffer model can be further modified or customized interactively via assembly design features. Modified mould base can be eventually saved into design model repository as a member subassembly with the necessary attributes. It can always be retrieved, edited or deleted as an object within QuickMould environment. Choosing different options and loading different parts achieve configuration changes for components. Form-features are configured with controlling expressions to turn on or off the relevant feature candidates and to change the dimensions.

Within QuickMould, a Generic Assembly (GA) class has been defined. It enables the system to identify existing modular assemblies instanced from a given assembly library. To do that, the system cycles the existing QuickMould objects at the moment of initialization. It checks the properties of the assembly tree recursively to find out any subassembly that can match the attributes of an assembly library element. The iteration continues until the end of the list. If any has been found, the corresponding instance object with the embedded attributes is constructed including the associated geometric entity pointers. Matching the earmarking attributes identifies instances from different library source elements, then the original library catalogues and standards are re-connected.

In this system, assembly features such as the thickness of plates, cap screw pattern, guide pin pattern, etc. are identified and managed. As shown at the center of Figure 6, when clicking "Edit Parameters" button from the main UI of the mould base library, a list of defined assembly features as well as the member components is compiled and made selectable in another UI menu. Double click on a feature or a component, its corresponding editing UI appears. In background, the feature manager instances the feature object by reading in its definition from the configuration file and extracting the related parameters as well as the expression values from the assembly instance model. The editing UI allows the designer to input values on specified parameters or to select the configuration from available feature options. This editing function can update assembly features or components of the assembly instance automatically. The configuration of these UIs is carried out on the fly based on the configuration file where all the choices of types, sizes, features, parameters and constraints are specified.

Once the user confirms the inputs, the editing function will check the input selections or values against the availability of the configuration as well as the legal parameter value ranges. Relevant constraints are checked as well. Any change in the key parameters will be brought into an updating expression file. If no violation occurs, the corresponding instance of the assembly model is updated via the links to the assembly template. Key variables are changed with the newly generated expressions; hence the involved features in the assembly model are updated. Correspondingly, other related features or components will be automatically modified because of constraints existed. Within a library assembly template, associative parametric links are automatically created via relations among expressions across the assembly model. Such links are retrieved, managed and saved via a set of feature object modification methods.

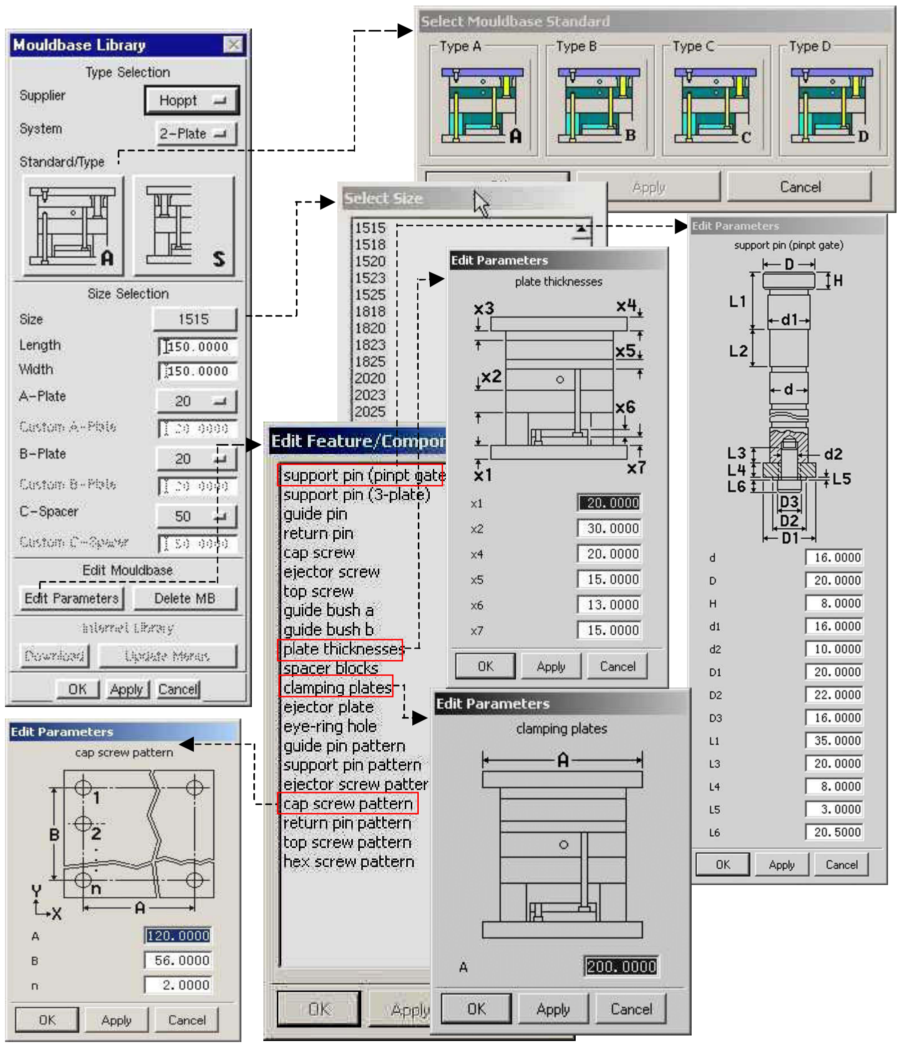

Fig. 6. Some user interfaces for QuickMould mould base library

Figure 7 illustrated a mould base assembly used in a real design. To delete the mould base in the mould assembly, the user needs to just click on the "Delete MB" button on the library main UI as shown in Figure 6.

6. CONCLUSIONS

This paper gives a detailed description about the systematic design of an interactive mould base library. This software tool can help the designer of injection moulds to simplify the mold base design process. This library uses a feature-object method for modeling. The genetic assembly model in this work has the major advantage over other tools that it enables different types and configurations of assemblies being included in a common library framework. Each library element is realized via an innovative data structure, which consists of a configuration file, a CAD assembly template, a data file and some UI bitmap files. It enables the designer to customize the mould base from the standard library elements easily. Customized mould bases can also be added back to the library for future usage and retrieval. Constraints have been established within the mould base assembly via inter-part expressions. By changing one of the key parameters, related features and components are correspondingly changed, and the whole assembly updated. Interactions between the designer and the mould base have been realized. The system have been tested and evaluated with many mould designs. It has demonstrated to be an efficient and effective approach for the development of other modular assembly libraries.

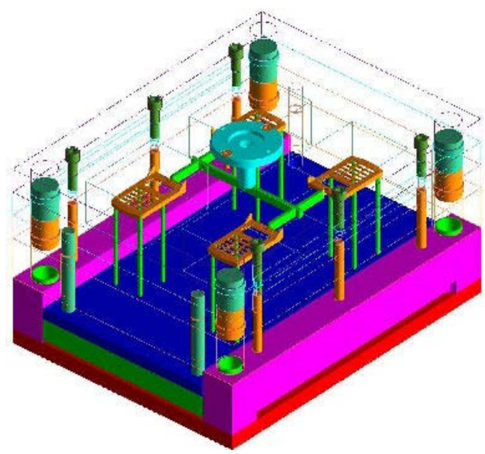

Fig. 7. A mold base assembly used in a design (partially loaded)

7. FUTURE WORK

There are still some limitations with the system, such as complete checking for geometric constraints that occasionally cause runtime errors during the regeneration of the assembly instances. Further research and development are needed in this aspect. The collaborative product development trend has greatly affected injection mould manufacturing cycle. This work provides some basis for web-based collaborative product development. With its modularized design concept, its potential capacity for collaborative product development can be further explored.

8. ACKNOWLEDGEMENT

QuickMould© research project was sponsored by the InfoComm Development Authority (iDA) of Singapore, and jointly developed by Nanyang Technological University (NTU) and Singapore Institute of Manufacturing Technology (SIMTech). After its commercialization, QuickMould© was adopted by EDS Inc. and further developed as part of Moldwizard™.

9. REFERENCES

[1] Kruth J. P. and Lecluse D. W., A design support system using high level mould object, in *Proceedings of the CIRP International Conference and Exhibition on Design and Production of Dies and Moulds*, 1997, Turkey, pp 19-21.

[2] Aldefeld, B., Variation of geometries based on a geometric-reasoning method, *Computer-Aided Design,* Vol. 20, No. 3, 1988, pp 117-126.

[3] Shah, J. J. and Mantyla, M., *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*, John Wiley and Sons, Inc., 1995.

[4] Otto, H. E., From concepts to consistent object specifications: translation of a domain-oriented feature framework into practice, *Journal of computer science & technology,* Vol. 16, No. 3, 2001, pp 208-230.

[5] Gorti, S. R., Gupta, A., Kim, G. J., Sriram, R. D. and Wong, A., An object-oriented representation for product and design process, *Computer-Aided Design,* Vol. 30, No. 7, 1998, pp 489-501.

[6] Bettig, B. and Shah, J., An object-oriented program shell for integrating CAD software tools, *Advances in Engineering Software*, Vol. 30, No. 8, 1999, pp 529-541.

[7] Britton, G. A., Ma, Y. –S. and Tor, S. B., Object technology development and unigraphics, in *Proceedings, Unigraphics User Group 1999 Spring Conference: Manage design evolution*, 1999, Newport Beach, California, USA.

[8] Jayaram, S., Wang, Y., Jayaram, U., Lyons, K. and Hart, P., A Virtual Assembly Design Environment, in *Proceedings of IEEE Computer Graphics and Applications*, Vol. 19, No. 6, 1999, pp 44-50.

[9] Ye, X. G., Fuh, J. Y. H. and Lee, K. S., Automated Assembly Modeling for Plastic Injection Moulds, *International Journal of Advanced Manufacturing Technology*, Vol. 16, No. 10, 2000, pp 739-747.

[10] Gan, P. Y., Lee, K. S. and Zhang, Y. F., A branch and bound algorithm based process-planning system for plastic injection mould bases, *International Journal of Advanced Manufacturing Technology*, Vol. 18, No. 9, 1998, pp 624-632.

[11] Neo, T. L. and Lee, K. S., Three-dimensional kernel development for injection mould design,

International Journal of Advanced Manufacturing Technology, Vol. 17, No. 6, 2001, pp 453-461.
[12] Fu, M. W., Fuh, J. Y. H. and Nee, A. Y. C., Core and cavity generation method in injection mould design, *International Journal of Product Research*, Vol. 39, No. 1, 2001, pp 121-138.
[13] Chung, J. and Lee, K., A framework of collaborative design environment for injection molding, *Computers in Industry*, Vol. 47, No. 3, 2002, pp 319-337.
[14] Ma, Y. -S., Tor, S. B. and Britton, G. A., The development of a standard component library for plastic injection mould design using an object oriented approach, *International Journal of Advanced Manufacturing Technology*, Vol. 22, No. 9-10, 2003, pp 611-618.
[15] Ma, Y. -S. and Tong, T., Associative feature modeling for concurrent engineering integration, *Computers in Industry,* Vol. 51, No. 1, 2003, pp 51-71.
[16] Mok, C. K., Chin, S., and Ho, J. K. L., An interactive knowledge-based CAD system for mould design in injection moulding processes, *International Journal of Advanced Manufacturing Technology*, Vol. 17, No. 1, 2001, pp 27-38.
[17] Myung, S. and Han, S. K., Knowledge-based parametric design of mechanical products based on configuration design method, *Expert Systems with Applications*, Vol. 21, No. 2, 2001, pp 99-107.
[18] Lee, K. S., Fuh, J. Y. H., Zhang, Y. F., Li, Z. and Nee, A. Y. C., IMOLD: An intelligent plastic injection mold design and assembly system, *IES Journal*, Vol. 36, No. 7-12, 1998.
[19] EDS Inc., *UG help documentation*, Maryland Height, MO, USA, 2000.
[20] Singapore Institute of Manufacturing Technology (SIMTech), *Quickmould user's guide*, Singapore, 1999.
[21] Mäntylä, M., A modeling system for top-down design of assembled products, *IBM Journal of Research and Development*, Vol. 34, No. 5, 1990, pp 636-659.
[22] Sugimura, N., Moriwaki, T. and Kakino, T., A study on assembly model based on STEP and its application to assembly process planning, in *Proceedings of ASME Japan/USA Symposium on Flexible Automation*, 1996, Vol. 2, pp 791-794.
[23] Pham, D. T. and Dimov, S. S., A system for automatic extraction of feature-based assembly information, in *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, Vol. 213, No. B1, 1999, pp 97-101.
[24] Pye, R. G. W., *Injection mould design: a design manual for the thermoplastics industry*, Plastics and Rubber Institute, London, Godwin, 1983.