International Journal of CAD/CAM Vol. 3, No. 2, pp. 67~75 (2003)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Design of a Feature-based Multi-viewpoint Design Automation System

KwangHoon Lee, Chris A. McMahon[1] and Kwan H. Lee

Department of Mechatronics, Kwangju Institute of Science and Technology (K-JIST), 1 Oryong-dong, Puk-gu, Kwangju, 500-712, Republic of Korea

[1]Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK

Abstract – Viewpoint-dependent feature-based modelling in computer-aided design is developed for the purposes of supporting engineering design representation and automation. The approach of this paper uses a combination of a multi-level modelling approach. This has two stages of mapping between models, and the multi-level model approach is implemented in three-level architecture. Top of this level is a feature-based description for each viewpoint, comprising a combination of form features and other features such as loads and constraints for analysis. The middle level is an executable representation of the feature model. The bottom of this multi-level modelling is an evaluation of a feature-based CAD model obtained by executable feature representations defined in the middle level. The mappings involved in the system comprise firstly, mapping between the top level feature representations associated with different viewpoints, for example for the geometric simplification and addition of boundary conditions associated with moving from a design model to an analysis model, and secondly mapping between the top level and the middle level representations in which the feature model is transformed into the executable representation. Because an executable representation is used as the intermediate layer, the low level evaluation can be active. The example will be implemented with an analysis model which is evaluated and for which results are output. This multi-level modelling approach will be investigated within the framework aimed for the design automation with a feature-based model.

***Keywords*: Computer-aided design, Feature-based design, Viewpoint dependency, Design automation, Collaborative design**

1. Introduction

A number of computer-aided design research activities have focused particularly on the problems involved when integrating various previously separate systems and representations into a framework that covers all major aspects of the automation of the product development phases. During the design process, different engineers take information about the design, and manipulate it in order to generate the new information required for the development of the new product. Various engineering representations of the designed artefact are created and used for different engineering tasks. These represent, for example, a manufacturing engineering viewpoint, a structural analysis viewpoint, a process engineering viewpoint, and so on. These representations for the various engineering requirements are obtained by translation between engineering models, augmenting them as necessary with information specific to the specialist viewpoint [1].

A significant issue in design is the amount of time and effort that is spent first of all by design engineers in creating CAD models of design parameters and then by specialist engineers in creating their models, based on the design parameter models, so that they may make their specialist judgments. There are various reasons why the work is time-consuming: creating a new design model from geometric primitives can be quite laborious; the modification that is often needed to the design parameter model by a specialist – for example to simplify or approximate the model – can also involve a lot of effort. The manipulation of the model also has to be done at the level of low-level geometric entities such as faces and edge loops, and this contributes to the time taken. However, if model representations can be used that facilitate the automatic or semi-automatic translation between representations, then a good deal of the required effort may be eliminated. Furthermore, if a high-level representation is used, then it may be possible to create design models very rapidly, thus allowing thorough exploration of the design space. An approach to such a high-level representation is multi-viewpoint feature-based design – computer-aided design using features that are compatible with multiple engineering viewpoints.

In this paper a multi-viewpoint feature-based design approach for an automated design environment will be described. The system approach is based on building a feature-based model of a design and mapping to executable representations of secondary viewpoint models. The approach to construction of models using the design-by-features approach involves controlling execution of a commercial CADCAM system through

---

*Corresponding author:
Tel: +82-(0)62-970-2407
Fax: +82-(0)62-970-2384
Homepage: http://kyebek9.kjist.ac.kr/
E-mail: mekl@kyebek.kjist.ac.kr

commands from the system's macro language with dynamic modification of the macro language by an external control program to allow different artefact parameters to be varied. Variable parameters used to describe a design feature model are defined and a probabilistic assessment of a design evaluated simply by repeatedly varying the values of the describing parameters. This system is also designed to be collaborated over the network for the purpose of handling macro language files in distributed workstations. Furthermore, the entire architecture is introduced as a Tool Management System (TMS). The objectives of the work presented here are, firstly, to explore whether, by using features to construct design models and as a basis for specialist models, the models could be constructed much more simply so as to avoid lengthy initial construction, and to avoid time-consuming conversion between models. Secondly, it was wished to examine whether the construction and manipulation of features could be carried out programmatically so that complete design and analysis procedures might be incorporated into optimisation or probabilistic analysis processes. Each of these aspects will now be reviewed, and a brief overview of each of the elements of the system will be provided.

2. Research Background

2.1. Approaches to multiple viewpoint feature-based design

In feature-based design, models of the designed artefact are represented as collections of features - model elements that have engineering significance [2] Feature based approaches encompass both design using features (design-by-features) and automatic identification of features on a conventional geometric model. In design by features, features are entities that can be modified by changing feature parameters, typically related to the geometric form of the feature. Feature-based design is regarded as a promising approach to design representation for various product development phases. Nevertheless, improvements are sought through increased capability for design (e.g. especially geometry specification and modification of complex shapes such as castings or pressed panels) and a better ability to act as the integration for manufacturing applications such as process planning, assembly planning and analysis. The design by features approach comprises all operations that are part of the process of creating a feature-based model and transformation of the model to different applications. The approach provides two principal advantages in the design process in that (1) the designer can store in the feature model non-geometric information which is available at the design stage and can later be applied to various engineering domains, and (2) features can be used to access information associated with particular feature types during the design process. This makes it possible to implement advanced applications such as concurrent design, real time geometry modification, and integration with other applications. The general purpose of the modelling methodology is the support of methods for automatically generating models for carrying out specialist activities, especially engineering analysis tasks. In general, however, the geometric and other attributes of a component may be combined into features in a variety of ways that reflect the needs of different design and manufacturing applications at different product development phases. This is currently a problem in feature representation: different specialists may choose to use different feature sets for the same part because they assign different engineering meaning to the elements of the part. This is known as viewpoint dependency. In this regard there is a need for feature definitions, or for a method of feature mapping or conversion features, that allows wide coverage of different viewpoints in the design process.

In recent years there have been a number of attempts to represent multiple viewpoint modelling based on feature technology. In order to propagate feature modification automatically across different domains, an algorithm has been developed by Jha [3]. In his algorithm, the called "feature-tree" as output by a multiple feature extraction was proposed to propagate changes made in one feature model to other feature models of the same part. The feature-tree makes available to the propagation algorithm a history of the extraction process that results in the multiple feature models. The algorithm of modifications is however limited on only the geometry (e.g., not for changes of topology) of volumetric features. A mechanism for maintaining consistent product views in a distributed product information database was proposed by Hoffmann [4]. In his work, a single repository called a "master model", in which all-relevant product data resides, was proposed for the integration of different product information domains while the other views must be updated to maintain consistency [4]. This architecture builds on different applications distributed over different CAD or CAM systems. Bronsvoort [5] proposed a multiple viewpoints feature modelling approach to overcome the shortcomings from which current multiple viewpoint modelling is done only for form features. His system supports conceptual design, assembly design, part detail design and manufacturing planning by providing an own interpretation of the product for each of these applications. Recently, some researchers [6, 7, 8] are focusing on the collaborative systems in corporation with feature technology. Collaborative feature-based modelling systems are distributed multi-user systems. These systems provide concurrent working environment, synchronisation of modelling data for distributed users, and user interaction facilities among users. More details on collaborative design will be discussed in the next section.

2.2. Collaborative design in automation

Competitive markets require rapid product development, and product development over computing networks is a new development made in modern engineering design. In this circumstance, the distributed system is thought of as a design environment that provides support for a collaboration work in the product development stages. In order to coordinate such activity, a mechanism for coordination of processes in a distributed working environment is required. Design today is therefore undertaken in a collaborative work by distributed design teams that may be on multiple sites. The computing environments used also needs a collaboration implemented by distribution, with computers connected in networks that allow them, in principle, to communicate with other machines all over the world. Software approaches have been developed to exploit these collaborations. In particular, processes may be executed on one computer under control of another computer, allowing different modelling, analysis and simulation components to be collaborated over a network with heterogeneity of computer platforms and languages, and to be executed as desired to achieve design automation by automatically invoking processes in an appropriate order. Hence, distributed computationally expensive computing activities such as optimisation and probabilistic design may be collaborated in the most appropriate way. For this reason, large-scale design work related to the product development life cycle is moving to collaborative design work.

Achieving design automation involves the integration of the information processing required by the various disciplines involved at the different stages of the design process [9]. For well-established designs, e.g. in the automotive industry, it may be possible to model the flow of information between different disciplines and stages, and to use these models to assist in the automation of aspects of the design process. For example, the optimisation or probabilistic design assessment of an engineering part may involve generation of a geometric model of the part, use of the model to generate an analysis model, and use of the analysis model to produce information to guide the modification of the geometric model. However, using current technology, considerable human input is needed to set up geometric models or parametric models for use in such activities, and then to use these models in the preparation of analysis models for the activities.

A major issue in reducing the human effort required in information processing in design is the use of model representations that are sufficiently semantically rich. Feature based modelling is regarded as a key technology in this respect. Computer-aided design with feature representation as the mechanism may be used to define and maintain product design information for analysis and simulation of products from the early stage of the product development cycle. By using a feature-based description of engineering parts, the automation of such processes as optimisation and probabilistic design may be assisted – the human input required in the modelling of the geometry and then in the preparation of the analysis model may be very much reduced.

In the use of features to develop a product model, it is necessary for feature-based design systems to work with other systems and their application programs. From this requirement in which the ability to share information is enabled, feature data may be exchanged by product data exchange standards between the various systems and applications. One of the efforts of developing international standards in data exchange is STEP, and ISO 10303 covers a wide variety of different product types (e.g., electronic, electro-mechanical, sheetmetal, fibre composites, ships, architectural, process plant, etc.) and life cycle stages (design, analysis, planning, manufacture, etc.) [10]. The parametric group of ISO TC184/SC4 has developed the parametric data exchange scheme, and in an example system called "macro-parametric approach" in which it is capable that parametric information of models can be transferred in the form of macro files [11]. In addition, CORBA's growing number of Application Programming Interfaces (APIs) and extensions have made it possible to build distributed, collaborative (in which integral parts of various engineering design environment are built), multi-tier applications (in which one part runs on one computer and another part runs on another, and the individual disciplines are used as objects) [12]. On the basis of this emerging technology, significant efforts in various research fields have been taken to automate and to distribute a complex computing task over a network. Among the various researches on distributed systems, the software environment called NetBuilder was introduced in [13]. NetBuilder provides a mechanism for coordinating collaborative activities in an automated and distributed work environment, which is being used to support integration of designers and design tools involved in multi-disciplinary engineering analysis in a number of domains.

Tool Management System (TMS) has been recently developed by Lee, McMahon, and Kaymaz [9, 14] and [15] for the purpose of handling the flow of data and the control of multi-process execution for distributed design activities. A TMS is a system which can maintain consistent versions of data, and invoke user-specified tools as necessary in some problem-solving application. Generally, these employ an object-oriented approach and use such mechanisms as CORBA or other distributed object systems. In a TMS, a standard interface to the tool should be provided to invoke tools and their sub-applications. In the procedures having a standard interface, the terms "tool integration" and "tool encapsulation" may be defined for tool control. Tool integration describes the process in which tools are connected to a TMS, and tool encapsulation defines the process by which a tool management system is working to execute and control tools. In tool integration, the source code of the tools or APIs should be available to implement communication with the TMS. Then, the tool can directly obtain data to operate. In tool encapsulation, a software wrapper between a tool management system and tools normally defines the communication that works through only the TMS's interface. In example systems, Kaymaz developed ADAPRES_NET in which a structural reliability problem is resolved by applying structural reliability methods. In the work by Lee [14], the basic structure is proposed to be based on an automated computer aided design process for creating and modifying a model, and invoking ADAPRES for the evaluation of multiple performance functions of a structural reliability problem using response surface functions or direct calls to the performance function. This process can be distributed over the workstations on a network by the advantage of using CORBA [16]. The multi-viewpoint feature-based design automation system described here will be based on this approach.

3. Research Methodology

3.1. Multi-level feature mapping approach to design automation

In order to use features to construct design models and as a basis of specialist models, it is necessary to be able to specify feature models and their manipulation into different viewpoints in such a way that they may be constructed and manipulated easily. Ideally, some sort of language would be provided for the specification of features, and of the way in which they are assembled together into a model and then manipulated for different applications. This language would provide mechanisms for the definition of features, for the combination of features into a model, for the manipulation of features (simplification, approximation etc.) and for the association of such aspects as loads and boundary constraints with a feature model. These mechanisms eventually have to be translated into operations for a CAD system. For example, features in a feature model need to be translated into the system commands needed to construct these features. When the design feature model is manipulated into a FE model, then there may need to be simplified or approximated feature models and then loads and boundary conditions need to be applied to the features of the design model. In practice this means firstly that a different feature model may need to be created with alternative geometric representation of the features (to account for the simplification/approximation) and secondly that loads and boundary conditions need to be applied to the features, or more precisely to the faces of the features. The viewpoint-dependent manipulation should involve execution of CAD system commands in some way such that the appropriate geometric entities and FE

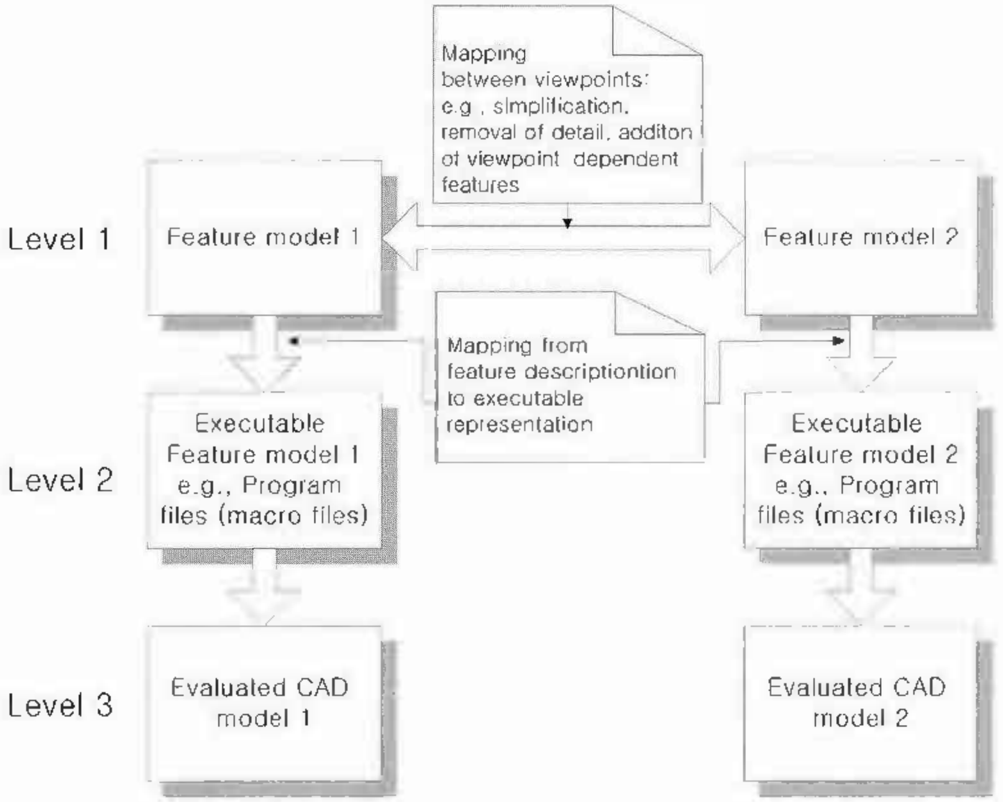


Fig. 1. Architecture and mappings of the multi-level feature-based modelling system.

entities are created.

Viewpoint dependency means that the features needed and the way in which they are defined may vary with the engineering viewpoint. Feature mapping is therefore seen as an important requirement for the flexibility of feature-based design systems including multiple viewpoint modelling [14]. The approach that has been adopted in this work is to develop an approach for viewpoint-dependent feature modelling, called Dynamic Feature Evaluation (DFE) and implemented using a commercial CADCAM system (SDRC I-DEAS) in a distributed automation environment [9]. The approach uses a combination of mapping between representations and a multi-level modelling approach, similar to that employed by de Kraker [17], but here comprising:

1. A feature-based description for each viewpoint, comprising a combination of form features and other features such as loads and constraints for analysis;
2. An executable representation of the feature model, for example, in the form of an executable Macro file or object model;
3. An "evaluation" of the feature model obtained by executing the representation defined in (2).

This three level architecture is shown in Fig. 1 and there are two sets of mappings associated with this architecture: firstly, mapping between the level I feature representations, for example for the geometric simplification and addition of boundary conditions associated with moving from a design model to an analysis model, and secondly mapping between level I and level 2 representations in which the feature model is transformed into the executable representation. Note that because an executable representation is used as the intermediate layer, then the low level evaluation can be active – for example an analysis model which is evaluated and for which results are output.

The work reported here involves both mappings, although it has concentrated on the second of the two mappings. It is assumed that techniques such as that described by Kugathasan [18] would be used for complex mapping of form features between viewpoint representations. Simple mapping, limited to addition of analysis features, will be described. The question that has been primarily addressed concerns how feature models should be mapped to executable intermediate representations, and then how these representations may be used in design automation applications. This has been done through experiments that explore the automated construction of feature models from different viewpoints.

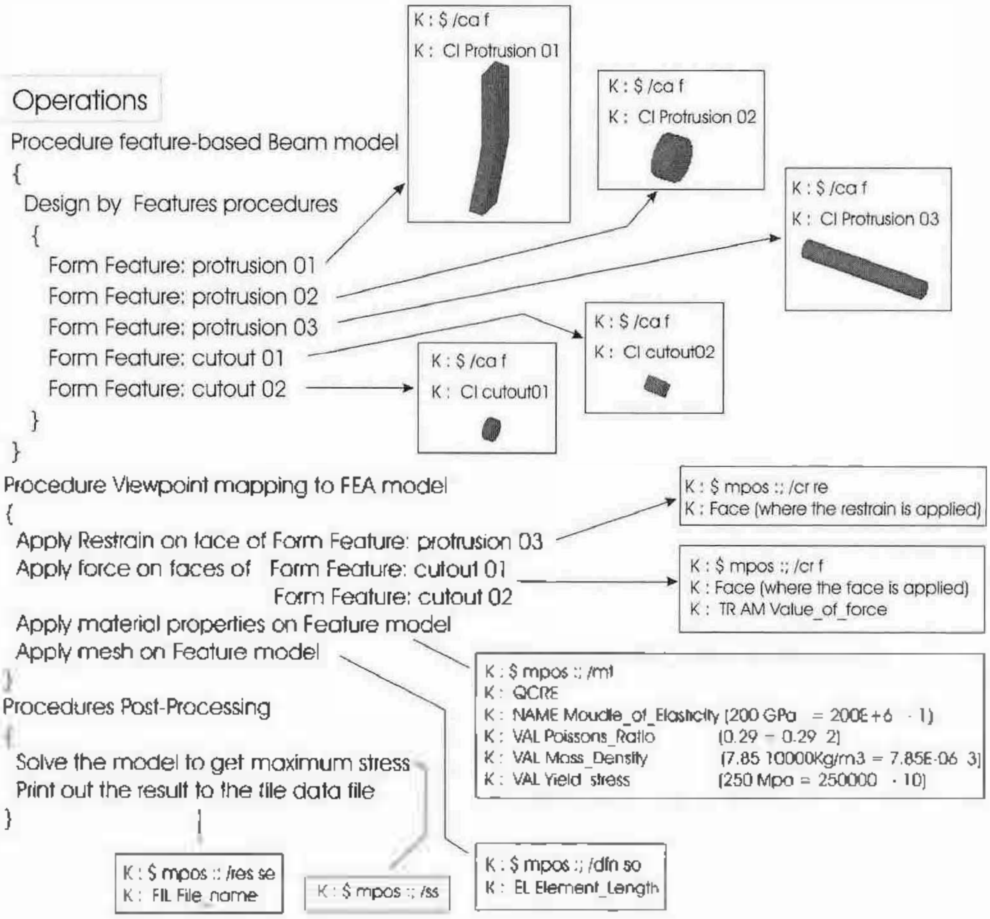

```
Operations
Procedure feature-based Beam model
{
  Design by  Features procedures
   {
     Form Feature: protrusion 01
     Form Feature: protrusion 02
     Form Feature: protrusion 03
     Form Feature: cutout 01
     Form Feature: cutout 02
   }
}
Procedure Viewpoint mapping to FEA model
{
 Apply Restrain on face of Form Feature: protrusion 03
 Apply force on faces of   Form Feature: cutout 01
                           Form Feature: cutout 02
 Apply material properties on Feature model
 Apply mesh on Feature model
}
Procedures Post-Processing
{
 Solve the model to get maximum stress
 Print out the result to the file data file
}
```

Fig. 2. Macro operations for the FE model of a bicycle crank.

3.2. Design automation with a feature-based model

Automated construction of feature models requires the feature description to be translated into an executable set of commands for the CAD system to evaluate. It was considered that this could be done in a number of ways: by instructing a CAD system application to execute commands one-by-one, by translating the feature description into code for the system's API to execute, or by translating it into the system's macro language (that in the case of I-DEAS is called Program Files and uses command mnemonics for automated system operation). All three approaches were explored, but the first method is cumbersome, the system used did not support a full set of finite element operations through its API, and therefore the mnemonic approach was concentrated on. It is that approach that will be described here.

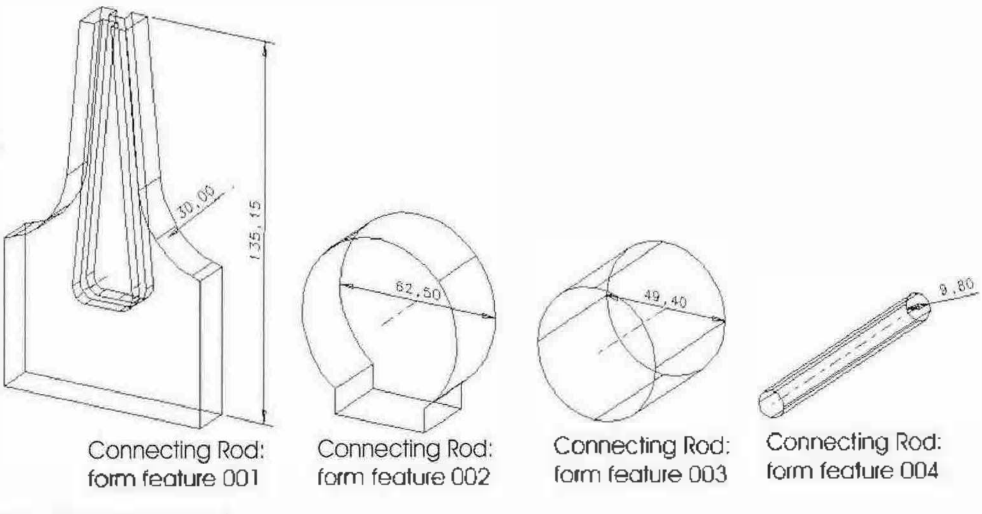
The use of a macro language for the evaluated feature model is achieved by first creating mnemonic sequences for features and for feature operations either by writing them directly or by capture of interactive commands and then editing the captured files as required. Mapping from feature model to macro language is then achieved by creating a file combining code segments corresponding to features and other operations. Fig. 2 shows the code segments corresponding to the development of a finite element model of a bicycle crank.

In this example, a finite element analysis feature model is represented as a collection of executable commands describing feature modifications, feature parameters and locations, and then for each step of the mapping to be described in terms of operations on the whole part (e.g., meshing, materials), or on named elements of features (e.g. loads, constraints applied to specified faces), all described using executable mnemonic commands. In this way, repetitive design automation tasks (e.g. repeatedly constructing models for the purposes of optimisation) may be carried out by writing, modifying and executing mnemonic files.

A further advantage of using a macro file approach is that model variables may be modified easily by editing the program file, and that this may be done programmatically. Distribution of execution may also achieve easily, simply by instructing a remote workstation to execute a program file. This approach to remote execution also reduces the necessary network traffic.

There are negative aspects of using macro files. The most serious difficulty (shared with the other approaches) was that, for the system used, it was not possible to use user-supplied persistent names for geometric entities (e.g. faces of features) to be used in later manipulation of the model. System-supplied names could be used, but these were not persistent and keeping track of them was difficult. Also, the mnemonic files are difficult to read, and macro execution is potentially much slower than programmatic execution using an API. Nevertheless, unless absolute performance is an issue (and it may be in some optimisation applications), macros offer a convenient way of constructing feature models.

Macro and command-by-command approaches also require different ways of arranging for the distributed execution of tasks. In each approach presented here the architecture is that a central computer application works at feature model level, and generates and accumulates the data used for example for optimisation or probabilistic analysis. Construction of the CAD models, execution of finite element analyses and so on is carried out by CAD system software, either on the same or separate workstations. The nature of the interaction is different between the two applications however. In the command-by-command approach there is continuous communication between the central controller and the CAD system command server, and the controller passes a stream of commands to the remote processes. In the macro approach the central controller again deals with



Connecting Rod: form feature 001

Connecting Rod: form feature 002

Connecting Rod: form feature 003

Connecting Rod: form feature 004

Fig. 3. Form features of Connecting Rod.

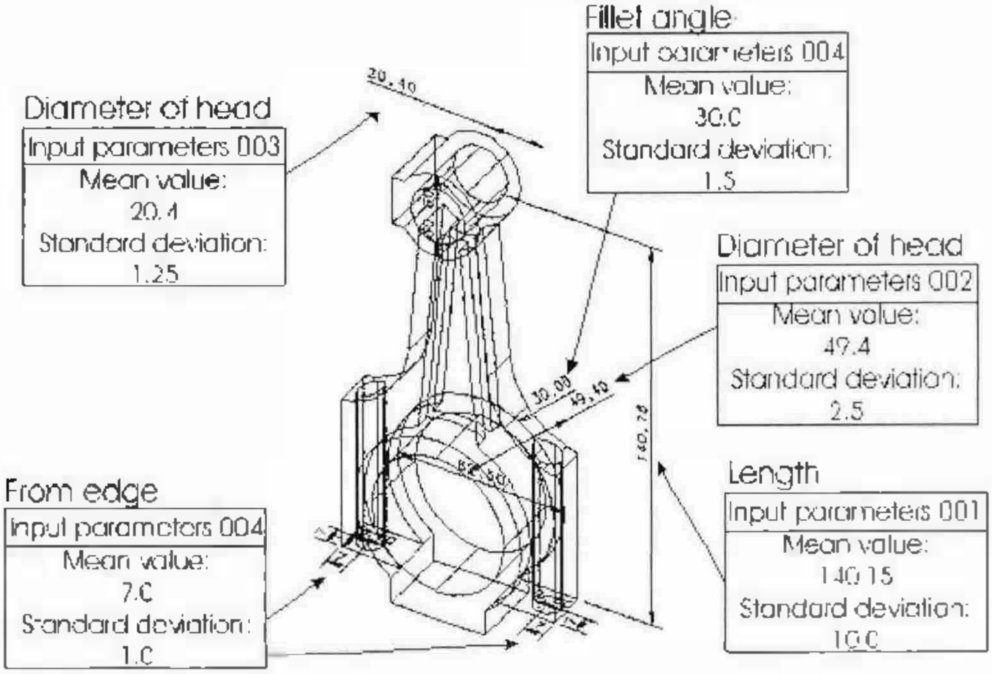


Fig. 4. Probabilistic design viewpoint of geometric analysis model.

the optimisation or probabilistic analysis data, and uses this data to create or edit macro files, which it then passes for execution to the remote process. In this way the process is more asynchronous, and the central controller could manage many remote processes simultaneously.

An example of repeatedly evaluated part geometry is shown in Figs. 3 and 4, which show the features and key parameters (values and standard deviations) respectively for a connecting rod. In order to explore the likely weight variation of the connecting rod owing to variation in the dimensions, models of connecting rods were repeatedly constructed for different values of the feature parameters in order to define a response surface for the weight of a rod, and then this response surface was used in a Monte Carlo simulation to compute the weight of the rod as a probability distribution function.

4. Conclusion

The experiments that were carried out demonstrated how feature models could be created programmatically, using a variety of interfaces to a standard CADCAM system. Initially, purely design features were used, and then it was shown how a feature-based model could be taken and extended to another viewpoint (e.g. a finite element model or geometric analysis model) using the same approaches. From these experiments, it was understood not just how to take one feature configuration and develop another viewpoint, but also in principle how different parts can be modelled using the same feature set and again further viewpoint models developed for these parts. The outcome of the work was a multi-level modelling system for the construction and execution of feature - models comprising 1) parametric design to construct generic features, 2) complete feature definitions for design, finite element analysis and geometric analysis viewpoints, 3) design by features for the automated design method for all feature types by a macro approach, operated in a distributed design environment in which different workstations in a network can create models from different viewpoints, and 4) a multi-level mapping operation for feature description and executable representation. Feature models and their geometric representations appropriate to different applications will be defined with their own sets of features, and transformed by application-specific transformation operations, e.g., adding, modifying, and removing aspects of features which need to be modified e.g., geometric description of features for geometric analysis, or idealisation/approximation for an application such as finite element analysis. This mapping is incorporated into "the mapping between the models of the multi-level mapping operation", and the automated design work demonstrated by a macro language is defined by "the mapping from feature description to executable representation". Since features are required to be generic for the flexible evaluation or transformation of geometric modelling, features are entities which can be modified by changing feature parameters which are related to geometric descriptions of features. These feature parameters are provided from parametric design. Moreover, all operations that are part of the process of creating a feature-based model and transformation of the model to different applications will be processed by the design by features technique, and this process was automated throughout the framework of this paper.

The work has shown that a framework for collaboration allowing the generation of viewpoint models for different product development stages is possible, but that in order to achieve this it is necessary to properly structure product information flow, and enhancements to CAD environments are needed, in particular a capability to attach persistent names to geometric primitives of features, if a full range of functions are to be supported. Since one of the main advantages from using feature technologies over conventional geometric modelling is the ability to associate functional and engineering information with aspects of product models these enhancements will be important if the full benefits of computer-aided engineering are to be achieved.

Acknowledgement

Department of mechanical engineering, the University of Bristol, UK previously supported this work, and continuous development of this paper is currently supported by the Brain Korea 21, Department of Mechatronics, Kwangju Institute of Science and Technology (K-JIST).

References

[1] MG-IT Coordonnateur: Leon, J. C. (1998), "Contribution to a multi-views, multi-representations design framework applied to a preliminary design phase," *IDMME '98, 2nd International Conference on Integrated Design and Manufacturing in Mechanical Engineering*, Compie-France, May 27-29.

[2] Shah, J. J. and Mantyla, M. (1995), *Parametric and Feature-based CAD/CAM*, Wiley-Interscience, New York, 1995.

[3] Jha, K. and Gurumoorthy, B. (2000), "Automatic propagation of feature modification across domains," *Computer-Aided Design*, **32**, 691-706.

[4] Hoffmann, C. M. and Joan-Ariyo, R. (2000), "Distributed Maintenance of Multiple Product Views," *Computer-Aided Design*, **32**, 421-431.

[5] Bronsvoort, W. F., Noort, A., van den Berg, J. and Hoek, G. F. M. (2001), "Product development with multiple-view feature modelling," *CD-ROM Proceedings of the IFIP Conference on Feature Modelling and Advanced Design-For-The-Life-Cycle Systems*, 12-14 June, Valenciennes, France.

[6] Bidarra, R., van den Berg, E. and Bronsvoort, W. F. (2001), "Web-based collaborative feature modelling," *In Proceedings of Solid Modeling '01 - Sixth ACM Symposium on Solid Modeling and Applications*, 6-8 June, Ann Arbor, MI, USA, Anderson, D.C. and Lee, K. (Eds.), ACM Press, NY, pp. 319-320.

[7] Huifen, W., Youliang, Z., Jian, C., Lee, S. F. and Kwong, W. C. (2003), *Feature-based collaborative design, Journal of Materials Processing Technology*, **139**, 613-618.

[8] Li, W. D., Ong, S. K., Fuh, J. Y. H., Lu, Y. Q. and Nee, A. Y. C. (2003), "Feature-based design in a distributed and collaborative environment," *Computer-Aided Design*, Article in press.

[9] Lee, K.H. (2002), *Engineering design representation by feature-based design in design automation - multiple viewpoint dependent models in product development*, Department of Mechanical Engineering, University of Bristol, UK, PhD Dissertation.

[10] Pratt, M. J. and Anderson, B. D. (2001), "A shape modelling applications programming interface for the STEP standard," *Computer-Aided Design*, **33**, 531-543.

[11] Choi, G. H., Mun, D. and Han, S. (2002), "Exchange of CAD Part Models Based on the Macro-Parametric Approach," *International Journal of CAD/CAM*, **2**(1), 13-12.

[12] Sistla, R., Dovi, A. R. and Su, P. (2000), "A distributed heterogeneous computing environment for multidisciplinary design and analysis of aerospace vehicles," *Advances in Engineering Software*, **31**, 707-716.

[13] Dabke, P., Cox, A. and Johnson, D. (1998), "NetBuilder: an environment for integrating tools and people," *Computer-Aided Design*, **30**(6), 465-472.

[14] Lee, K. H., Kaymaz, I. and McMahon, C. A. (2001), "The use of distributed viewpoint-dependent feature-based modelling and the response surface method in design assessment," *International Conference on Engineering Design ICED01*, Glasgow, August 21-23.

[15] Lee, K. H. and McMahon, C. A. (2003), "Characteristics of a multi-viewpoint feature-based design automation environment," *International Conference on Engineering Design ICED03*, Stockholm, August 19-21.

[16] Suh, N. P. (1990), *The Principle of Design*, Oxford University Press, New York.

[17] De Kraker, K., Maurice, D. and Bronsvoort, W. F. (1995), "Multi-way feature conversion to support concurrent engineering," *Solid Modelling '95, third symposium on solid modelling and application*, Salt Lake City, Utah, May 17-19.

[18] Kugathasan, P. (1998), *A feature-based approach to design information management - Multiple viewpoint dependent models for the product introduction process*, Department of Mechanical Engineering, University of Bristol, UK, PhD Dissertation.

KwangHoon Lee is currently involved in the department of Mechatronics, KwangJu Institute of Science and Technology (K-JIST) as a post-doctorial fellow. He received a PhD from department of Mechanical Engineering, University of Bristol, UK, 2003, and MSc from University of Surrey, Guildford, UK, 1997. His research interests include PDM/CPC, virtual engineering, design collaboration, feature-based geometric modelling, freeform features, and computer graphics.

Kwan H. Lee received his M.S. and Ph.D degree at North Carolina State University in 1985 and 1988 respectively. He worked as an assistant professor at Northern Illinois University from 1988 to 1994. He is a professor in the Mechatronics Department at Kwangju Institute of Science and Technology (K-JIST) since 1995. His research interests include Surface modeling, Texturing (Immersive Media Modeling), Reverse Engineering and Computer Aided Inspection, and Rapid Prototyping. Currently he is the director of the Immersive Media Research Center at K-JIST and focuses on his research in immersive modeling.

Chris A. McMahon is a Reader in the University of Bath, which he joined in 2002 from the University of Bristol. In his early career as an engineer he worked in the railway industry and with a consulting engineering company specialising in IC engines. He teaches and researches in engineering design and computer-aided design. He is interested in many aspects of design and computing, in particular how computer aids can assist design in the organisation and management of the information used in design.

KwangHoon Lee

Chris A. McMahon

Kwan H. Lee