USING MACHINE TOOL FEEDBACK FOR PREDICTING THE PERFORMANCE OF AS-MANUFACTURED ASSEMBLIES:

a feasibility assessment

by

Sane Wu

Bachelor of Science in Precision Instruments and Mechatronics
Tsinghua University, Beijing, China, 2000

Submitted to the Department of Mechanical Engineering in
Partial Fulfillment of the Requirements for the degree of

Master of Science in Mechanical Engineering

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2003

Signature of Author........................
Department of Mechanical Engineering
May 23, 2003

Certified by..
Professor David R. Wallace
Esther and Harold Edgerton Professor of Mechanical Engineering
Thesis Supervisor

Accepted by.......................................
Professor Ain A. Sonin
Chairman, Departmental Committee on Graduate Students

Using Machine Tool Feedback for Predicting the Performance of as-Manufactured Assemblies:
a feasibility assessment

by
Sane Wu

Submitted to the Department of Mechanical Engineering
on May 23, 2003 in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering

ABSTRACT

Failure to consider manufacturing variability in the design products can result in high cost and poor quality. This thesis presents a prototype model to predict the performance of as-manufactured assemblies. The system is built upon a network-based design-modeling environment, embedded with real-time manufacturing information feedback to provide the necessary information for part characteristics estimating. A survey is conducted to identify typical machine tool data for different machining operations and to identify estimation methods for part characteristics using available data. The feasibility assessment is made through the analysis of the various assembly-modeling improvements that could be brought about based on the estimation.

An implementation of this performance prediction model is proposed through a network-based modeling framework called DOME, which supports heterogeneous CAD modeling, and by an integration of LabVIEW, which is one of the possible interfacing methods to build connections to manufacturing in real time. The allocation of tolerances is one of the most critical interface issues between design and manufacturing in product development. Across tolerance analysis architecture, the internal relationship and communication between DOME, LabVIEW, and assembly modeling tools plugged into DOME is illustrated.

Thesis Supervisor: David Wallace
Title: Esther and Harold E. Edgerton Associate Professor

ACKNOWLEDGEMENT

I would like to dedicate this thesis to my parents, who give me love and care all my growing up. Without their support this thesis would never happen. I would also like to thank my dear big brother Walter who has always been on the same side of me, especially in my hardest time. Thanks for all their belief in me.

I also want to express my sincere gratitude to my advisor, Professor David R. Wallace, for his insightful guidance, support, and patience during my study at MIT CADlab. His knowledge and experience helped me to shape and focus my research, and for this I am very grateful. I'm also thankful to Nicola Senin, for giving me advice regarding my thesis.

Special thanks to Elaine for her time and effort spent in helping me with my work in CADlab. Thanks also go to all the CADlab members, Aubrey, Charles, Ed, Ines, Jaehyun, Kristie, Prabhat, Qing, Rainer, Tom, and Twiggy. Thank you for all the happy time spent with you.

TABLE OF CONTENTS

ABSTRACT 3

ACKNOWLEDGEMENT 5

TABLE OF CONTENTS 7

CHAPTER 1 INTRODUCTION 9
  1.1 Background 9
  1.2 Motivation 12
  1.3 Concept 14
  1.4 Thesis Objective 19

CHAPTER 2 ANALYSIS 21
  2.1 Assembly and Tolerances 21
  2.2 Cost-Tolerance Relationships 25
  2.3 Tolerance Analysis and Allocation 26
  2.4 The Concurrence of Design and Manufacturing 28
  2.5 Summary 31

CHAPTER 3 PROTOTYPE 35
  3.1 Network-Based Modeling Environment 35
  3.2 Design Procedure 39
  3.3 Manufacturing Feedback in Real Time 44
  3.4 Product Model Knowledge Base 46
  3.5 Summary 50

CHAPTER 4 MACHINING DATA INVESTIGATION 53
  4.1 Machine Tool and Machining Data 53
    4.1.1 Turning Process 53
    4.1.2 Milling Process 55
  4.2 Dimensional Accuracy 57
  4.3 In-Process Control of Dimensional Accuracy 57
    4.3.1 Mechanical methods 58

4.3.2 Optical methods 60
4.3.3 Pneumatic methods 62
4.4 Predicting Dimensional Accuracy 62
4.4.1 Prediction of dimensional tolerances 63
4.4.2 Prediction of surface roughness 66

CHAPTER 5 FEASIBILITY ASSESSMENT 71
5.1 Assemblability 72
5.2 Stackup Compensation 73
5.3 Assembly Sequencing 76
5.4 Part Mating 78
5.5 Summary 80

CHAPTER 6 IMPLEMENTATION 81
6.1 Tolerance Analysis Architecture 81
6.1.1 Tolerance Analysis Model 81
6.1.2 Control Methodology 84
6.2 Overview of DOME 86
6.2.1 Integrated Design Service Marketplace Concept 86
6.2.2 Software Prototype 89
6.3 LabVIEW Integration 90
6.3.1 LabVIEW Application 91
6.3.2 Client-Server Models for LabVIEW 93
6.3.3 Establish A Connection Between DOME and LabVIEW 95
6.3.4 LabVIEW Real-Time Module 98
6.3.5 Plug-in LabVIEW Real-Time Series DAQ Devices 98
6.3.6 Summary 98

CHAPTER 7 CONCLUSION 101

REFERENCES 105

APPENDIX: LabVIEW IMPLEMENTATION PROGRAMS 109

CHAPTER 1 INTRODUCTION

Product performance prediction, life-cycle and cost optimization have enormous implications on the business of engineering design and manufacture. Deterministic design often fails to provide the necessary understanding of the nature of manufacturing and variability (Barkan 1988; Evans 1988). It is, however, appealing because of its simplicity in form and application, but since factors of safety are not performance related there is no way by which an engineer can know whether his design is near optimum or over-conservative in critical applications. Simultaneous design with manufacturing factors being considered offers much potential in this connection, but have yet to be brought into full play due to the insufficient involving and the inadequate handling of manufacturing information. Virtually many design parameters such as tolerances, assembly sequence, and part mating features exhibit some variability that come from the manufacturing process. If part characteristics can be estimated within the acceptable precision level and product performance can be properly predicted in the design process, the simultaneous design then becomes more suitable.

1.1 Background

The traditional product development process, still used in industry, involves disjoint design, manufacturing, and assembly activities (Wolff et al. 2001). In this development process, the design engineers define the part functionalities and specify the part characteristics only based on the design principles and without the manufacturing concerns. When the design requirements of the part are passed into production, manufacture engineers are in charge of the process planning, machining operations, inspection and assembly. Because these operations in manufacturing and assembly are sequential, assemblies can only be performed after all the components to be assembled are manufactured. If a failure happens in the assembly, either the components need to be sent back to manufacture for rework, or the product and assembly need to be redesigned. As a matter of fact, in this traditional product development, the production cost is extremely high, the production cycle is quite long, and the quality cannot be ensured.

To improve product development efficiency, during the analysis of a product or an assembly, it is advantageous to attack the overall assembly first. One reason for this is that being able to have an assembly evaluation before manufacturing provides a prediction of the performance of the final assembly so that problems in assembly can be found and prevented at an early stage and rework can be eliminated to the greatest extent, and therefore the production cost can be minimized and the production cycle can be shortened. In addition, beyond the cost of production, for a product designed considering the assembled performances, it will be known at an early stage what the function and general configuration of the assembly will be, and therefore the quality analysis and control is facilitated. Additional benefits of the assembly evaluation are reduced need for inspection, fewer engineering and production-control documents, less materials handling, and probably, lower inventory levels.

The best approach for the assembly evaluation, when possible, is to have the designers themselves make the evaluation (Booker et al. 2001). This certainly has speed and accuracy advantages in that there is no need to transfer information about the design from one person to another. The time to prepare documentation and to explain the design concept to the specialist is avoided. One of the advantages of the evaluation systems, particularly those involving assembly or other aspects of manufacturability, is that they make it relatively easy for the designers themselves to carry out the evaluation. In addition to the convenience and time advantages, there is also the learning factor that benefits the designer. Designers who conduct an evaluation with a prepared system tend to learn the design principles that underlie the system.

The use of the assembly evaluation has immense influence on product design and development (Booker et al. 2001). Equally important, the manufacturing operations also play an important role in improving the production efficiency. The manufacturing process concepts are not unalterable – they are subject to change if it can be demonstrated that they are responsible for the creation of assembly problems. It is entirely possible for a design engineer, aware of the advantages of designing for manufacturability, to work independently in optimizing the product design for ease of manufacture.

The product design and development process is undergoing major changes nowadays as computer programs are being developed to do the complete processing job on the basis of either the part design, the assembly evaluation, or the manufacturing. By far the most well known approach to improving integration between these functions in the product development process is concurrent engineering. Concurrent engineering brings to the design phase consideration of many factors affecting total cost and performance throughout the life cycle of a product. The complete design-manufacturing cycle is complex, involving market analysis, conceptual design, product design and development, material selection, process planning, production, information and process control, quality and process monitoring, and cost. Concurrent engineering involves integrating the diverse functions in an organization into a process of creating a better product when viewed across the entire product life cycle. Hence, the design engineer's job fuses with the production engineer's job.

Amongst the many factors that influence the product design and development, tolerance analysis and control has a significant influence on manufacturing cost and the production cycle. Part tolerances affect the selection of manufacturing operations, the final product assembly, and the function and performance of the product (Chase and Greenwood 1988). In general, tolerances are assigned by the designers: during design stages, the design engineers determine the dimensional requirements of each part and specify the allowable variations in the dimensions of interrelated parts – tolerance. Tolerance specification is necessary because manufacturing processes are inherently imprecise in dimensioning. Especially in complex parts or assemblies, the analysis and control of part tolerances can become extremely difficult.

Individual part tolerances may be stringent because the individual small deviations can stack up to create assembly or performance problems in the assembled product. A tighter tolerance specification requires higher precision in manufacture and this may transfer to a higher cost for production (Drozda and Wick 1983). The tradeoff between tolerance and cost reflects the conflicts between design and manufacturing. By applying concurrent engineering principles, it is quite possible to balance the design requirements and the manufacturing operations.

In recent years concurrent engineering has been widely applied to design and manufacture. In product development based on concurrent engineering, all engineers involved in the development process work in close collaboration at the earliest possible stage in the development process, and information can be transferred to and shared with collaborators through intensive communication. Many design and machining data integration models have been proposed for representation, integration and communication of different engineers' viewpoints in the design and manufacturing phases. The common objective is to facilitate the evaluation for alternative design, manufacturing and assembly solutions at an early stage during production, and thus achieve a shorter production cycle and a lower production cost.

1.2 Motivation

It has been realized for many years that waiting until the product is at the end of the production line to measure its quality is not good business practice (Crosby 1969). Assembly variation, usually along with manufacturing variation, is a major contributor to poor quality and increased costs. At the same time, the cost of recovering from these problems during the late phase of production is high. This has led to an increased focus on the integration of quality into the early design stages of product development (Evbuomwan et al. 1996; Sanchez 1993).

A significant proportion of the problems of product quality can directly result from variability in manufacturing and assembly (Craig 1992). For example, when the assembly of a poorly designed and poorly made product is attempted, faults such as accumulated tolerance error, incompatible dimensions and difficult part installation become apparent. It is now being recognized that there is a need to reduce such variation at the design stage, where its understanding and control may lead to (Leaney 1996):

- Easier manufacture
- Improved fit and finish
- Less work in progress
- Reduced cycle time
- Fewer design changes

- Increased consistency and improved reliability
- Better maintainability and reparability

Decisions made during the design stage of the product development process account for a large proportion of the problems that incur failure costs in production and service. It is possible to relate these failure costs back to the original design intent where variability, and the lack of understanding of variability, is a key failure costs driver. However, the difficulties associated with identifying variability at the design stage mean that in reality those problems are detected too late in many cases, as indicated by a study of engineering change in nine major businesses from the aerospace, industrial and automotive sectors (Swift et al. 1997).

Design is recognized as a major determinant of quality and therefore cost. The designer's job is to try to capture customer expectations and translate as many of these expectations as possible to the final product. The functional requirements of the design become detailed into dimensional tolerances or into attributes of the component or assembly. The ability of manufacturing process and assembly process, by which these products are fabricated and assembled, to consistently provide dimensions within tolerance or attributes within the specification scopes reflects how well the final assembled product conform to the design. Understanding and controlling the variability associated with these design attributes then become a key element of developing a quality product.

Designers rarely fully understand the manufacturing systems where their products are fabricated, and subsequently they do not understand the variability associated with the design characteristics. Variability can have severe repercussions in terms of failure costs, appearing in production due to rework and scrap, and warranty costs when the product fails in service. There is need to try to anticipate the variability associated with the manufacturing process used to produce the final product early in the design process. The designer needs to know, or else be able to predict, the capability of the process and to ensure the necessary specification limits are adequate to avoid manufacturing defects.

However, this has previously been difficult to achieve on concept design or where little manufacturing detail exists.

Among the many attempts on the assessment of the capability of a process, numerous methods have been applied focusing on anticipating the possible results. Old methods, now still widely in use, do not consider manufacturing influences (Bralla 1999). The three common methods used to determine whether the product and process, as designed, meet the dimensional product requirements are:

1. Make a theoretical guess based on the dimensional, geometrical and kinematic variations;
2. Build hundreds or thousands of assemblies using production tools and measure the result; or
3. Simulate the design, manufacture and assembly of the product, including the 3D geometry, geometric dimensioning and tolerancing.

It has already been recognized that empirical methods are inaccurate and sometime inadequate without considering manufacturing. Especially when manufacturing processes are complex and changeable, empirical part characteristics estimation can be even worse. Statistical methods built up upon thousands of experiments are able to provide a more realistic approach but lost the efficiency. It is very important to predict the probability of successful assembly of the parts in an accurate and efficient way, so that the design specifications can be re-evaluated and modified if necessary in order to increase the probability of success and lower the associated production costs. Assembly performance prediction places special emphasis on detecting the assembly problems at the earliest stage and therefore preventing assembly failures. If implementing assembly performances prediction into the design phase, problems can be found before manufacturing instead of until the products are inspected. Hence, the costs for reworking and scrap of the products judged to be bad are avoided.

1.3 Concept

As elucidated in the above discussion, the quality and the conformance to tolerance of the product characteristics should be "designed in" and not left to the process engineer and quality engineer to be increased to the required level. In order to do this, designers need to be aware of potential problems and shortfalls within the capability of their designs. They therefore need a technique that estimates process capability and quantifies design risks.

Assembly performance prediction is a fundamental approach for detecting the design adequacy. A key objective of these methods is to provide the designer with a deeper understanding of the critical design parameters and how they influence the adequacy of the design on its performance level. The variables include dimensions, material properties and in-service loading. A key requirement is detailed knowledge about the manufacturing information involved to enable plausible results to be produced in component characteristics estimation.

1. Assembly performance prediction in product design and development

Concurrent product development (as well as sequential product development) usually consists of seven groups of activities (Prasad 1996):

- definition of goals,
- product planning,
- design,
- product process planning,
- manufacturing,
- assembly,
- inspection.

In concurrent product development there are interactions among individual groups of activities while there are no interactions in sequential product development. Track and loop technology was developed for the implementation of interactions. The type of loop defines the type of cooperation between the overlapped activities. Based on the 3-T loop representation (Winner et al. 1988), where interactions exist between three groups of activities, the product development process consists of five 3-T loops (Figure 1-1).

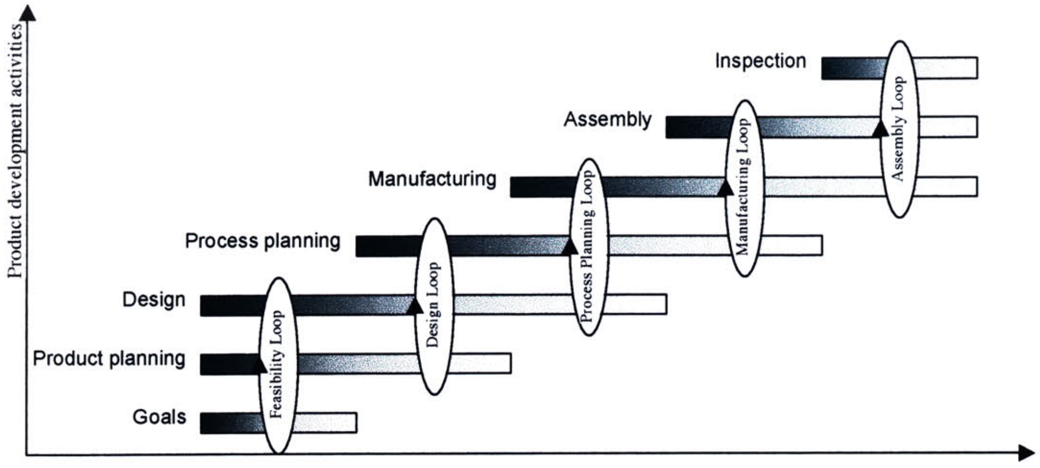


Figure1-1 Track and loop process in existing product development (Winner et al. 2002)

As illustrated in last section, the prediction of the assembly performance should be performed at the design stage.

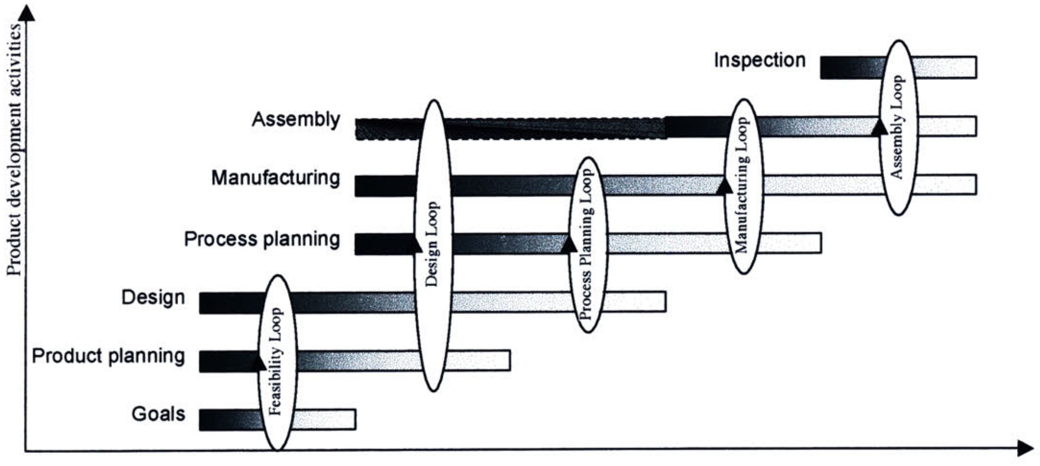


Figure1-2 Track and loop process in product development
with assembly performance prediction

On the basis of requirements and restrictions a transformation of input into output is made in each loop, as shown in the diagram of information flow in the track and loop process of product development.

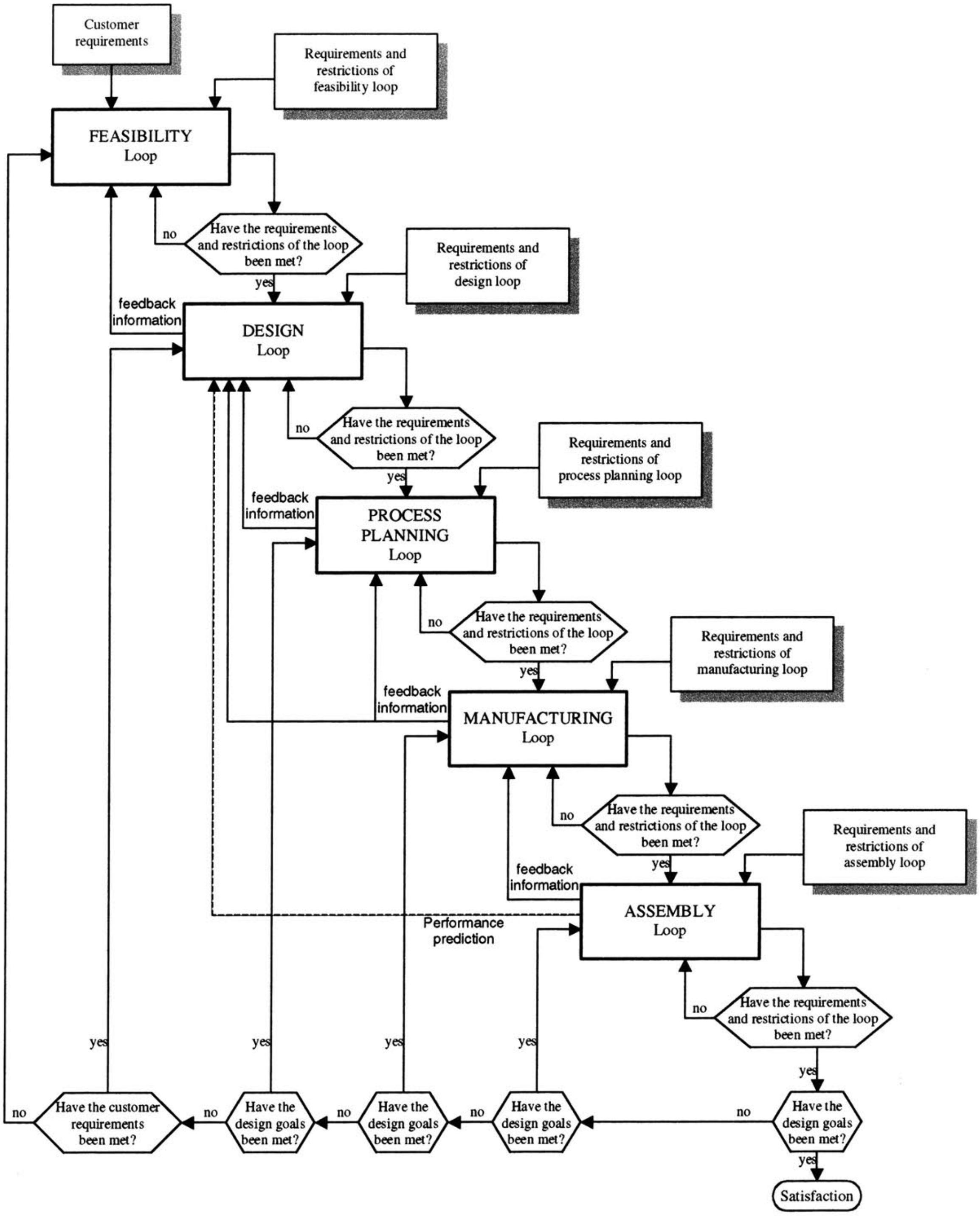


Figure 1-3 Diagram of information flow in the track and loop process of product development (Starbek and Grum 2002)

2. Realization of assembly performance prediction

Today's high technology products and growing international competition require knowledgeable design decisions based on realistic models that include assemblability requirements. A suitable and coherent assemble performance prediction methodology can be an effective interface between the customer and design. Predicting the performance of as-manufactured assemblies in the design stage requires a breadth of knowledge both in assembly and in estimating component characteristics. Assembly rules, standards and methodologies are quite important to assembly performance prediction such as in determining the assemble sequence and compensating tolerance stack-up. Equally important, it is imperative to ensure the accuracy of the estimation of as-manufactured part characteristics, as the component characteristics for an assembly are always the bases on which the assembly rules are applied and the assemblies are performed.

Part characteristics estimation chiefly consists of the acquiring and the processing of manufacturing data. First, as the estimation takes place before and during manufacturing process, real-time feedback of manufacturing information can improve the accuracy in the greatest extent. Second, the adoption of appropriate and accurate methods for accomplishment of the estimation should be cautious but flexible. With the development of computer and network, knowledge of estimation is sharable and rapidly developing. As a result, it is quite possible that designers are not necessary to have knowledge and skills for estimation, but know how to employ the methods appropriately and manage their relationship when multiple methods are applied.

Based on the discussion above, the assembly prediction asks for an environment integrating design and manufacturing, which is placed not only on the purpose of information exchange and interface support, but also on providing knowledge engineering enriched tools for simulation, optimization and visualization of design models. Some specific technical requirements are summarized below:

- Open system architecture. The architecture and class structure allow for integration of design and real-time manufacturing feedback on different types of objects, thereby making full use of the resources provided by the system. The object-oriented architecture allows for an object-oriented detailed design and implementation in an object-oriented language, gaining the advantage of easily being able to include operating system properties into real-time systems.

- Uniform software paradigm for enterprise integration. A core software paradigm supports both databases and physical processes. The local objects connected for database rules do not limit the ability of the real-time system to process ordinary core rules, thereby permitting all types of systems to communicate with each other, be it complex information systems or smaller, possibly embedded systems, which do not process mass storage and are involved in real-time operations.
- Real-time manufacturing feedback integration, which consists of:
  - broad data acquisition in real time,
  - proper and prompt data transmission, and
  - accurate data analysis.

1.4 Thesis Objective

The designer has the greatest responsibility of ensuring that the product will conform to customer requirements, comply to specification, meet cost targets and ensure quality and reliability in every aspect of the product's use, all within compressed time scales. The product design and specifications are a translation of the product features and characteristics into performance and manufacturing terms. Assessment of the assembly performances prior to and during the development and manufacture of the product can determine whether the product to be assembled conforms to the requirements of the design and specifications. If the needs of the customer have been properly translated, these assessments will predict whether or not the needs will be met.

This research looks at the feasibility of this concept. To predict the performance of as-manufactured assemblies, estimation of the characteristics of components to be assembled is required as the foundation for assembly evaluation, and correspondingly, data acquisition and methods for estimating part characteristics are both equally necessary for the accomplishment of the evaluation. A survey is needed of what kind of machining parameters, in particular, machine tool data, can be acquired real-time from manufacturing, and of what methods can accurately and efficiently accomplish the part characteristics estimation. Then, through studies of machining parameters (in particular machine tool data), what information can be attained and what kind of tolerance analysis can be done is discussed. A model suitable for real-time feedback and concurrent design improvement is introduced.

Two issues need to be discussed when implementing the assembly performance prediction: (1) the component characteristics that contribute to product assembly and (2) the production variations that affect the final assembly. These two factors are linked together because it is not valid to determine the performances of the as-manufactured product assembly simply by estimating the component characteristics from the empirical formulas with respect to conformance to specifications, without considering the possible influences generated during the manufacturing processes. In the practical cases, assembly performance prediction involves both component characteristics estimation and final assembly simulation, and therefore attention must be focused on the providing of adequate information and the adoption of calculating methods.

CHAPTER 2 ANALYSIS

In order to improve the production efficiency, it is imperative to reduce the levels of non-conformance and attendant failure costs stemming from poor product design and development. Failure costs generally make up the largest cost category within production and include those attributable to rework, scrap, warranty claims, product recall and product liability claims. This represents lost profit to a business and, as a result, it is the area in which the greatest improvement in competitiveness can be made (Russell and Taylor 1995).

In an attempt to combat high quality costs and improve product quality in general, companies usually opt for some kind of quality registration, such as with the worldwide quality assurance standards – BS EN ISO 9000. Quality assurance registration does not necessarily ensure product quality, but gives guidance on the implementation of the systems needed to trace and control quality problems. The adoption of quality standards is only the first step in the realization of quality products and also has and ambiguous contribution to the overall reduction in failure costs. A more proactive response has been to implement and support product design and development strategies focusing on the engineering of the product.

2.1 Assembly and Tolerances

The three main sources of variation in mechanical assemblies are (Chase et al. 1997):

- Dimensional variations (lengths, angles)
- Form and feature (flatness, roundness, angularity)
- Kinematic variations (small adjustments between mating parts)

The above are all closely linked to manufacturing variability depending on the characteristic associated with the product. The design of products for assembly requires careful consideration of many factors that influence the functionality and manufacturability. Amongst these many factors tolerance plays an important role in assemblability.

"Assemblability" is a measure of how easy or difficult it is to assembly a product. Tolerances affect the assemblability of a product, which in turn affects the cost of the product because of the scrap cost, and wasted time and energy. It is important to predict the probability of successful assembly of the parts so that the tolerance specification can be re-evaluated and modified if necessary to increase the probability of success and lower the production cost associated with assembly (Lee et al. 1997).

It is often necessary to consider how tolerances on individual components in an assembly are to be combined to determine the variations that will result in the assembled unit. Conversely, it may be necessary to partition an allowable assembly variation to assign the required tolerances to the individual components. The tolerance chart is a very useful tool in approaching these problems.

A component is typically produced through several machining processes in a manufacturing environment and tolerance is assigned to each operation considering the inherent relationships between them. Since a sequence of machining operations are normally required to produce a part, it is imperative to establish a relationship between these operations based on the sequence and determine the effect of each operation on the dimension of the produced part. The tolerance chart can represent a process plan that is used to establish the link between interrelated machining operations and tolerance stack up, and thus be a good graphical tool that helps in determining this relationship and calculating the tolerance stack up. However, this static tolerance charting procedure can only be used at the pre-production stage to validate the process plan, and does not take into consideration the in-process variations. The values of the tolerances were based on past experience, best guess, or anticipated manufacturing capability. This tolerance information is imprecise because it is established on the static tolerance database and by the theoretical functions; it does not take account of the manufacturing factors that have the most direct and basic influence on part characteristics, such as machining parameters, process planning, etc. There's no guarantee that a produced part will meet the design specification.

With the development of network technology, a dynamic tolerance charting through on-line process control has been proposed to avoid the disadvantage of static tolerance charting (Chen et al. 1997). In this procedure, part dimensions are measured after each machining operation and the resulting data are input to the dynamic tolerance charting system. The system validates the process plan based on the measured dimensions and reassigns tolerances for subsequent operations as needed. The process continues until all the operations of the process plan are complete. This system acts as a tool for on-line process control by integrating with an on-line dimensional measuring system, and its validation of process plans represents a significant step in the development of the on-line process control system.

Often, the use of different assembly stack models in determining the final assembly tolerance capability can generate various results in assigning and adjusting component tolerances. Many references can be found reporting on the mathematical/empirical models used to relate individual tolerances in an assembly stack to the functional assembly tolerance. The two most well known models are highlighted below. In all cases, the linear one-dimensional situation is examined for simplicity.

In general, tolerance stack models are based on either the worst case or the statistical approaches. The worst-case model (see Equation 2.1) assumes that each component dimension is at its maximum or minimum limit and that the sum of these equals the assembly tolerance. The tolerance stack equations are given in terms of bilateral tolerances on each component dimension, which is a common format when analyzing tolerances in practice. The worst-case model is:

$$\sum_{i=1}^{n} t_i \leq t_a \qquad (2.1)$$

Where:

t_i = bilateral tolerance for i^{th} component characteristic
t_i = bilateral tolerance for assembly stack.

The statistical model makes use of the fact that the probability of all the components being at the extremes of their tolerance range is very low (see Equation 2.2). The statistical model is given by:

$$z_a\left[\sum_{i=1}^{n}\left(\frac{t_i}{z_i}\right)^2\right]^{0.5} \leq t_a \tag{2.2}$$

Where:

z_a = assembly tolerance standard deviation multiplier
z_i = i^{th} component tolerance standard deviation multiplier.

Equation 2.2 is essentially the root of the sum of the squares of the standard deviations of the tolerances in the stack, which equals the standard deviation of the assembly tolerance, hence its other name, Root Sun Square of RSS model. This can be represented by:

$$\left[\sum_{i=1}^{n}\sigma_i^2\right]^{0.5} \leq \sigma_a \tag{2.3}$$

Where:

σ_a = assembly tolerance standard deviation
σ_i = i^{th} component tolerance standard deviation.

The statistical model is potentially unappealing to designers because a defective assembly can result even if all components are within specification, although the probability of this occurring may be low. The worst-case model is, therefore, more popular as a safeguard (Gerth 1997), although it has been argued that it results in tighter tolerances that are often ignored by manufacturing when the design goes into production.

From the above consideration and models, we will now look into the relationships between cost and tolerance and the influence of tolerance factors that influence the tradeoff between therefore illustrate the tolerance role in product development.

2.2 Cost-Tolerance Relationships

Ideally, engineers like tight tolerances to assure fit and function of their design. Designers often specify unnecessary tight tolerances due to the lack of appreciation of cost and due to the lack of confidence in manufacturing to produce component parts that conform to specification (Phadke 1989). However, a tightened tolerance requires higher precision and more expensive manufacturing processes, higher technical skills, higher operator attention, and increased manufacturing steps, and is more time consuming to achieve (Jeang 1995).

Manufacturers, on the other hand, prefer loose tolerances that make parts easier and less expensive to produce (Chase and Parkinson 1991). The choice of tolerance is therefore not only related to functional requirements, but also to the manufacturing cost. It has been argued that among the effects of design specifications on costs, those of tolerances are perhaps the most significant (Bolz 1981). Figure 2-1 shows the relationship between product tolerance and production cost. The cost of production increases geometrically for uniform incremental tightening of tolerances. If the truth of this relationship is accepted – and it has been found to be correct in numerous writings by different specialists in a number of countries – then a systematic approach to tolerancing will have as one of its main thrusts the maximizing of production tolerances within the framework of manufacturing process concepts.

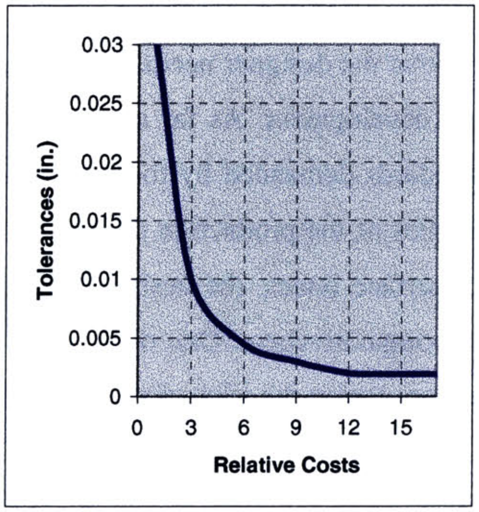


Figure 2-1 Relationship between tolerances and production costs (Drozda and Wick 1983)

The manufacturing process concepts are not, however, unalterable – they are subject to change if it can be demonstrated that they are responsible for the creation of tolerancing problems. Therefore, if a structured, systematic approach has the capability of maximizing tolerances based on an initial set of manufacturing process concepts, the technique should also support the assessment of those concepts, and thus indicate the nature of the changes that must be made in those concept decisions to produce an economically viable process/tooling package for use in production.

2.3 Tolerance Analysis and Allocation

Designers of mechanical components require a great deal of knowledge and experience to select and allocate tolerances. Traditionally, their involvement has been sequential. Product designers are primarily concerned with meeting the functional requirement of the product and they tend to assign tight tolerances to parts so as to ensure that assembly functional requirements will be satisfied. On the other hand, process designers are faced with difficult decisions. They could produce the process plans based on the tight tolerances even though the resulting manufacturing cost is high, they could enlarge the design tolerances which could lead to assembly problems, or they could ask the product designers to review the design to see whether larger tolerance could be allocated. This review could raise the final product availability but failure to consider tolerance implications at an early stage of the product design process may result in problems subsequently appearing at the production and assembly stages.

From the above, it is clear that the designer needs to be aware of the importance of the production phase in product development. As far as quality is concerned, the designer must aim to achieve the standards demanded by the specification, but at the same time should be within the capabilities of the production departments. From understanding the key design/manufacturing interface issues, the designer can significantly reduce failure costs and improve business competitiveness. One of the most critical interface issues in product development is that concerning the allocation of process capable tolerances.

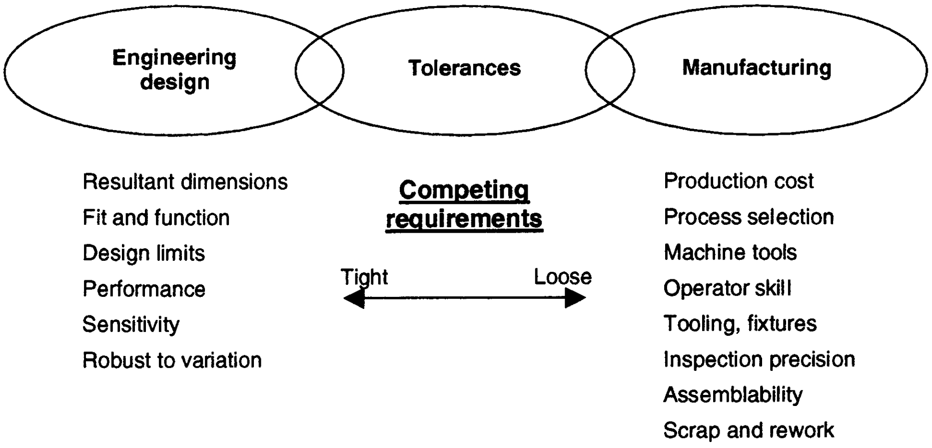


Figure 2-2 Tolerance – the critical link between design and manufacture (Chase and Parkinson, 1991)

There is probably no other design improvement effort that can yield greater benefits for less cost than the careful analysis and assignment of tolerances (Chase and Parkinson 1991). The effects of assigning tolerances on the design and manufacturing functions are far reaching, as shown in Figure 2-2. Product tolerances affect customer satisfaction, quality inspection, manufacturing and design, and are, therefore, a critical link between design and manufacturing.

Proper assignment of tolerances is one of the least well-understood engineering tasks (Gerth 1997). Assignment decisions are often based on insufficient data or incomplete models (Wu et al. 1988). The precise assignment of the component tolerances for this combined effect is multifarious and is dictated by a number of factors, including:

- Number of components in the stack
- Functional performance of the assembly tolerance
- Level of capability assigned to each component tolerance
- Component assemblability
- Manufacturing process available
- Accuracy of process capability data
- Assumed characteristic distributions and degree of skew and shift
- Cost model used
- Allowable costs (both production and quality loss)
- Tolerance stack model used

- Optimization method.

Some of the above points are worth expanding on. Tolerances exist to maintain product performance levels. Engineers know that tolerance stacking or accumulation in assemblies controls the critical clearances and interferences in a design, such as lubrication paths or bearing mounts and that these affect performance (Vasseur et al. 1992). Tolerances also influence the selection of manufacturing processes and determine the assemblability of the final product (Chase and Greenwood 1988). The first concern in allocating tolerances should be then to guarantee the proper functioning of the product, and therefore to satisfy technical constraints. In general design practice, the final assembly specifications are usually derived from customer requirements (Lin et al. 1997). The functional assembly tolerance is a specification of the design and maintains integrity with its mating assemblies only when this tolerance is realized within a suitable level of capability.

A good tolerance allocation model should maximize component tolerances for a specified assembly tolerance in order to keep production cost low (Wu et al. 1988). Any concessions to this should be in meeting the functional assembly tolerance in order to keep the failure costs low. This can only be achieved by predicting the as-manufactured component characteristics and optimizing the assemble performances through important technical tools and/or theoretical methods.

2.4 The Concurrence of Design and Manufacturing

Product design and manufacturing have competing tolerance requirements. From product design's point of view, a tight tolerance is agreeable for the assurance of a more precise assembly and thus for a better product performance. On the other hand, manufacturing prefers loose tolerance to reduce cost, simplify process planning and lower machine tool requirements. It is not an easy matter to reach a balance between the two. In traditional product development, first the designers complete their work and pass it to manufacturing. The manufacture engineers will then decide how to manufacture the product, from process planning through operation control to inspection. This appears to be weak in practice. With limited knowledge of manufacturing, the design engineer often designs products that are difficult and expensive to produce. In general, design, manufacturing and assembly have different concerns about the product. During design, the product is considered as a whole, and what being considered at the first place is the interaction of different parts and the achievement of the overall functionality of the assembled product. During manufacturing, precision in machining the parts is the main concerns. While during assembly, the establishment of the mating relationships between parts becomes very important. Usually little communication is made between design and manufacturing. This traditional sequential interaction between the design process and the manufacture process makes it very difficult and time consuming for designers to process the large amount of tolerancing information.

To improve the efficiency of tolerance control, design for manufacturing and assembly (DMFA) was carried out. One of the earliest efforts in DFMA was specifying tolerance close enough to enable interchangeability of parts (Bayoumi, 2000). The important results were not only greater manufacturing efficiencies but also uncomplicated field repairs. DFMA is often used as an optional technique, primarily by individual designers who feel strong enough about manufacturability to use it. This may be based on the designer's intuition, which, in turn, is based on experience. Thus, the designers who do the best DFMA designs are those who have had manufacturing experience. Today, DFMA is no longer optional, but, in many cases, required to be competitive. To determine a product's competitive vulnerability, ask the question: "How much better could your product line be designed for the following?" The same function, but with lower cost, better quality and reliability, better delivery, more responsiveness to customer needs, and new products developed sooner. In addition, products "fail" and go out of production because the cost is too high, quality is too low, introduction is too late, or production cannot keep up with demand. These are all manufacturability issues and therefore are very much affected by DFMA. Designers who do not take manufacturability into account cause the following manufacturing problems:

- Time to market: Non-DFMA products take longer to get into production because they are not designed for the existing processes and special arrangements must be made to be able to build them at all.

- Equipment: Non-DFMA products require more specialized equipment, which results in additional cost; delays in obtaining special production equipment can be substantial and can have a major negative impact on product introduction schedules.
- Quality: Non-DFMA products have more quality problems because they have more parts from more vendors, require more manual assembly, and may not take full advantage of factory quality control procedures that are set up for the typical processes.
- Cost: Quality problems and extra rework translate into higher manufacturing cost, especially if any defects get out to the customers.

A major obstacle to the design engineer at present is the fact that he or she is not required to be an expert in many disciplines. In addition to the usual factors of concern, new product features and proper function and quality of the product – we now, with DFM, ask the designer to ensure manufacturability. Design for all desirable attributes further expands the scope of needed skills dramatically. For many designers, this involves a redirection and refocus of approach. It is no longer satisfactory to develop a product that simply functions well and has desirable features. The kind of teamwork inherent in concurrent engineering was not – and still in too many cases is not – a normal part of the product realization process for many companies. The following summarizes four possible levels of interaction:

1. Traditional approach: "over the wall." Designers and manufacturing engineers do not communicate about the design. Design documents are transmitted to manufacturing without any prerelease review by manufacturing engineers.
2. An improvement: The sign-off procedure. Manufacturing engineers approve and accept the design *after* it is completed but before it released to production.
3. Further improvement: There is some collaboration between designers and manufacturing engineers during development of the design.
4. Current thinking: full concurrent engineering. Designers and manufacturing engineers work together on the design as a team.

The concurrent engineering process is the outcome of rapid development and enormous information exchange nowadays. It is complex, involves numerous considerations such as market analysis, concept design, material and process selection, simulations to test design strategies and manufacturing feasibility, information accessibility and integration, costing, manufacturing, and process and quality control. The objective of concurrent engineering is to avoid part features that are unnecessarily expensive to produce and make optimal choices of materials and processes. The result is improved quality of early design decisions and a significant impact on life-cycle cost of the product.

Concurrent engineering is a methodology that uses multi-disciplined teams to focus on the complete design and manufacturing sequence. It is not merely a procedure for automating design practices using computers; it is a process for evaluating product design while considering many life-cycle factors. The three basic components of concurrent engineering are (Terpenny and Deisenroth, 1992): design strategies for the design-manufacturing problem, decision aids to help in the selection of alternative design strategies, and supporting information and knowledge to accomplish the first two functions. The essence of concurrent engineering is the integration of product design and production planning into one common activity. A concurrent engineering system requires integration of these three components. Specific tools, such as computer-aided design and manufacturing (CAD/CAM), must be integrated within this system. With the right combination of hardware and software, design, manufacturing and quality control functions can work in parallel to reduce lead times and tolerance-related problems (and cost).

2.5 Summary

One of the most critical interface issues between design and manufacturing in product development is that concerning the allocation of process capable tolerances. How tolerance issue influence assembly has been studied and recent developments in tolerance control has been reviewed. The inadequacy of the worst-case approach to tolerance stack design compared to the statistical approach is evident, although it still appears to be popular with designers. The worst-case tolerance stack model is inadequate and wasteful when the capability of each dimensional tolerance is high. Some summarizing comments on the two main approaches are given below.

The 'worst-case' tolerance stack approach is characterized by the following:

- Simple to perform
- Assumes tolerance distribution on maximum or minimum limit
- Little information generated for redesign purposes
- Popular as a safeguard, leading to unnecessarily tight tolerances and, therefore, increased costs.

The 'statistical' tolerance stack approach is characterized by:

- More difficult mathematically (computer necessary)
- Assumes tolerances are random variables
- Opportunities for optimization of tolerances in the assembly
- Can perform sensitivity analysis for redesign purposes
- Can include effects of shifting and drifting of component tolerances
- More realistic representation of actual situation.

When designing a new product, it is essential to facilitate and drive the operation by an adequate product development process. The application of concurrent engineering, rather than sequential activities, has many benefits, in particular giving a reduction in the number of design changes and a reduction in the time it takes to bring the product to market. Tools and techniques have been found to effectively support the process of designing capable and reliable products.

We must promote the use of tolerance stack approaching methods with integrated manufacturing knowledge, through user-friendly platforms in order to design capable products. It is entirely possible for a design engineer, aware of the advantages of concurrent design and manufacturing to work independently in optimizing the product design for assuring of assembly. With the prediction based on concurrent systems, the design, manufacturing and assembly specialists advocate a team approach in the application. Essentially, this means that the design and manufacturing people work together to gain the benefits of manufacturing knowledge and experience that the designer may not have and to ensure that the product is both functional and under sound assembly conditions.

CHAPTER 3 PROTOTYPE

The assembly performance prediction models that are preventive in nature are generated through constructing a distributed network-based design-modeling environment. A critical factor in the configuration of the environment is an understanding of the internal relationships among all participating components. An integration of real-time manufacturing information feedback into the environment is significant in estimating the as-manufactured part characteristics. A knowledge base associated with analytical methodologies is a helpful augmentation in amplifying the ability to carry out assembly performance prediction.

It is the objective of this chapter to present a model for effective product development. It is based on a generic framework on which will be constructed the tools and techniques seen as the most beneficial in the design of capable and reliable products. Given the diversity as well as the complexity of the product development processes, no single set of activities or steps can be defined that will be appropriate to all processes due to the dependence on so many subjective variables. Therefore, for the appropriate and effective placement of the tools and techniques, it is necessary that in design procedure each generic process with defined functional stages be presented. Also discussed are some of the important peripheral issues regarding the gathering of manufacturing information and the interfacing methods.

3.1 Network-Based Modeling Environment

By applying concurrent engineering principles to product development, the manufacturing process control for parts and assembly performance analysis in design can be solved simultaneously, as discussed in Chapter 2. Errors and critical situations for assembly might be mostly eliminated. Implementing concurrent engineering means integrating product design, production planning, and manufacturing into the concurrent engineering environment, in which early intermediate development results are discussed and coordinated. Taking advantage of network development, design and manufacturing conflicts can be solved simultaneously. Distributed modeling allows different product development experts to work collaboratively, and solve the design to assembly problems as early as possible. The model is faster and more precise and efficient than a general model in which tolerance information acquisition and design are separated.

The design-modeling environment can be broken down into a series of functioning units that impact the product development process from design to manufacturing. There are the commonly involved units such as the creation of concept model, the assignment of design specifications, quality analysis and control, and assembly evaluation in the product design phase, as well as material engineering, machining operations, process planning, and inspection in the manufacturing phase. These units are either intimately or indirectly linked by the fact that the output of one unit becomes the input for another unit. The quality of the final product is dependent not only upon the capability of each functioning unit, but also upon the unit processes working together. Therefore, the configuration of the design-modeling framework involves the creation of a physical understanding of each process by itself as well as the influence of each unit process upon subsequent unit processes.

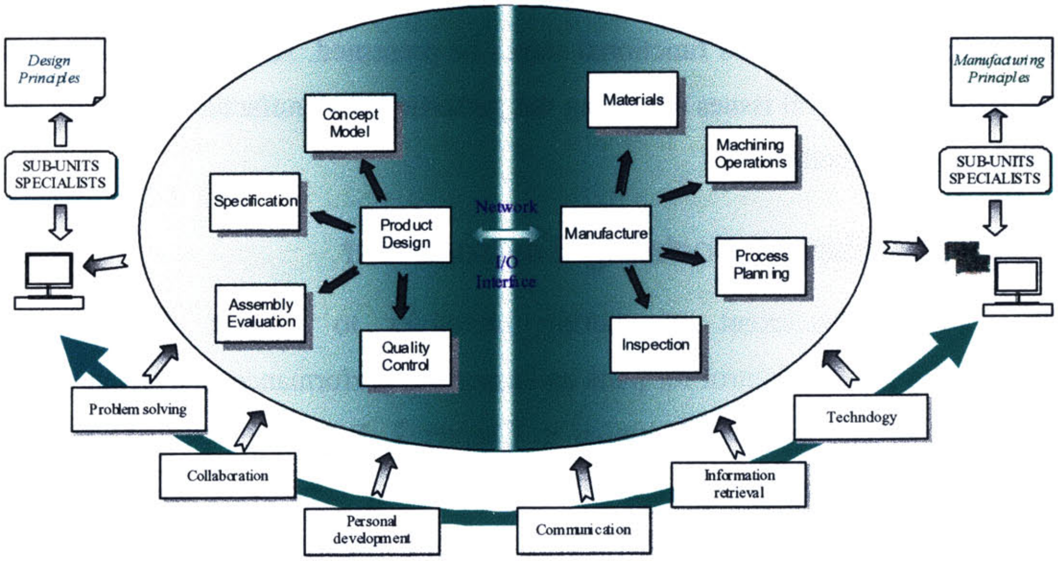


Figure 3-1 Product development framework

Normally different product development experts need to communicate their models, and for this reason the framework shown in Figure 3-1 should provide the functionality of integrating the different points of view by different people. Experts do not need to have the knowledge of all functioning units, but only of all the data necessary to their task unit. Therefore the framework should in first place been deployed to construct a distributed network-oriented modeling environment that can perfectly satisfy the requirements. The distributed character should allow many designers participate and collaborate during the product development process. Designers, engineering resources, models and activities are not centralized in one location, but are distributed among many departments. Integrated is another important character of the environment. Integration is the means, methodology, and practice of combining component parts of a system through interfacing, and aggregating them into a cohesive set. Such work is typically done through understanding and preserving the interrelationships and the interactions between the various components.

A distributed integrated system facilitates processing efficiency and offers conceptual as well as structural advantages. Firstly a central manager unit is absolutely necessary in determine the relationships of different participating units, and control and mediate their processing sequence. Secondly the interfacing part is quite potential to be explored deep into. Detailed constructions are discussed below:

1. Central control and mediation: Relationships of different functioning units are determined here through analyzing the outputs and inputs of the units. Priorities are given to each participated unit and sequences are generated in which the units will be processes. All calculations should be running on the back so that the security issue can be assured.
2. Interface with manufacturing: This is the part that acquires machine parameters' feedback during manufacturing. The real-time characteristic of the model assures more precise tolerance information. Especially when the components to be assembled are manufactured in different production branches in which the machining parameters are completely different, implementing this interface to

each of the manufacturing branches is the assurance of concurrent design and manufacturing.

3. Interface with designer: Drawing data can be easily extracted from a CAD database; in the meanwhile the real-time acquired machining data can also be input to tolerance database. Many software tools have both design functions and assembly simulations. The implementation of this software into the distributed network-based modeling environment provides the designer the opportunity of checking if the designed product performance is satisfactory, simulating the final product assembly, and compensating for stack up if needed at this early phase of design.

The system should also integrate multiple operations research techniques, database, and knowledge bases to jointly and concurrently help resolve multiple production system design issues. Thus, the system should have the following characteristics:

- Model for product representation. All information needed for all analyses is integrated and represented in this model. Different views or subsets of the model may be selected (transparently) to support different types of analysis.
- Adaptable functionality. Modeling and analysis capabilities, including the set of model building constructs, data and knowledge editors, analysis controllers and result processors, must be adaptable to different production systems or the particular systems of interests.
- Transparent information transfer. Information generated by one type of analysis is automatically and transparently captured and stored for later use by other types of analyses.
- Gateways to external databases and devices. Needed information residing outside of the system is accessible to support model construction and analysis specification as well as for other purposes such as project management and classical corporate information processing activities.

A number of tasks should be supported by the engineering system, such as problem solving, communication, collaboration, information retrieval, etc. These tasks should be fulfilled while the user performing design activities. The common purpose of performing these tasks is to provide services to the designer through user interfaces and utilities. The important aspects of the user introduction and utilization of the engineering system are that the user should,

- be in ultimate control,
- be creative,
- have overview, and
- have understanding.

The activities of the user, or the designer, are crucial to the establishment of design services that should be provided with the system, and should be further studies. The design procedure is the basis to the development of the product design environment based on the engineering system framework discussed in this section. An exploration in the relationships among design, manufacture, and assembly prediction towards design activities is necessary to the development of the prototype.

3.2 Design Procedure

Designers should look on the assembly or subassembly as a means to reduce the cost of production. Product designers, when designing parts for assembly, should visualize how the parts are put together. This in itself will help ensure that they consider design alternatives that facilitate assembly. The designer should understand as well the assembly method to be used and know what tools, fixtures, and gauges will be used during assembly.

The product design and specifications are a translation of the product features and characteristics into performance and manufacturing terms that engineers can understand. The concept of assembly performance prediction is to acquire manufacturing information before the production of the part and send them to assemble model, so that designers can look on the assembly or subassembly as a means to reduce the cost of production. The information flow in the system is shown in Figure 3-2.

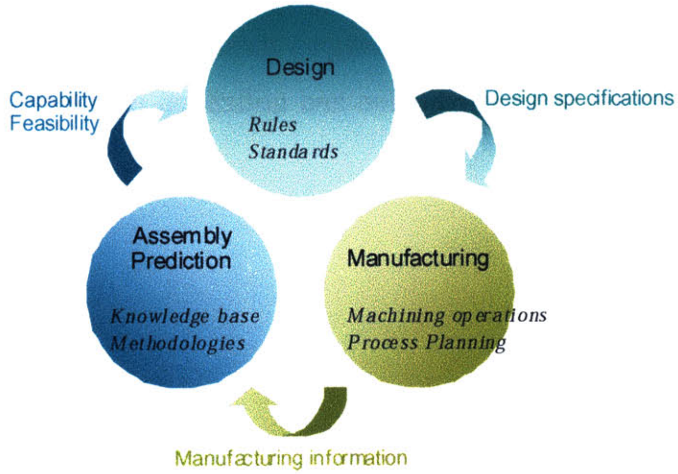


Figure 3-2 Information flow

The assembly performance prediction environment will be based on product modeling, interfacing techniques, graphics and database/knowledge-base techniques. Application modules will be based on engineering knowledge and vocabulary. Those will be used in combination according to the designers or planners way of working and thinking. Figure 3-3 illustrates this concept. Designing in this environment means that the designer will develop his design in a stepwise way, in the form of product model knowledge, base in the computer, being a representation of achieved new knowledge. In this environment, it is the designer of the planner, etc. that is both the creative thinker and final decision maker.

The whole system concept is built upon distributed system components running in a network of computers. As have been mentioned, many of the system components are built on the concept of the product model knowledge base. It is something like an "intelligent" product documentation system. Other system components are built on AI system techniques, like for instance a rule based system for assembly design. We can also see the need for more conventional programs integrated with an "intelligent" documentation, which can be reasoned upon from the problem point of view and in view of achieved result.

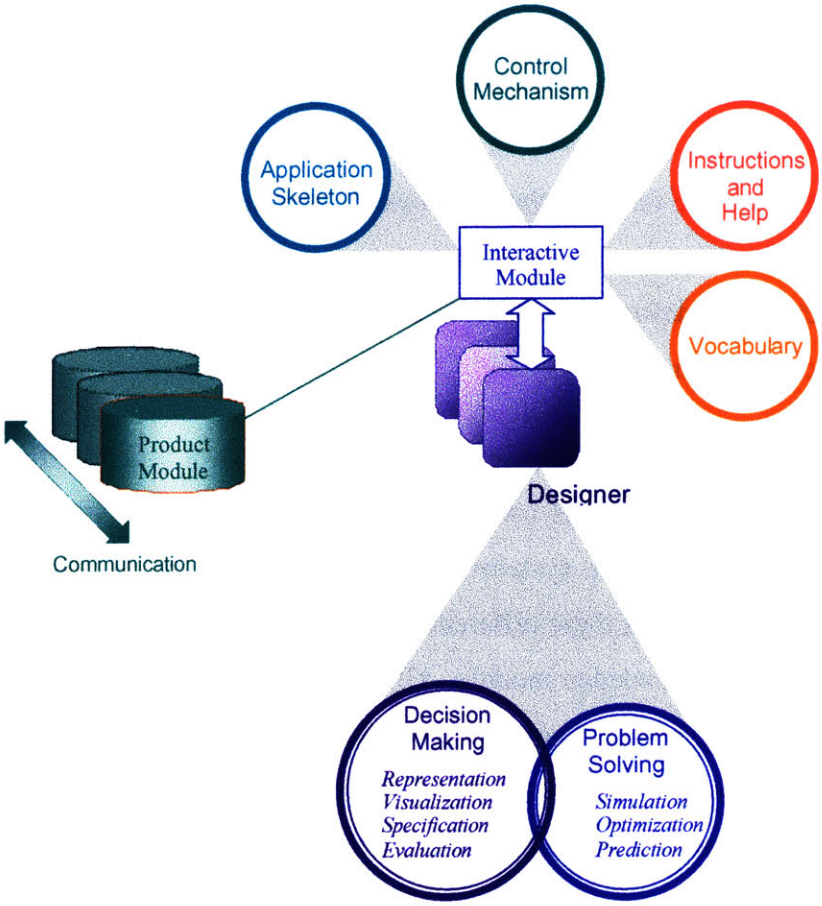


Figure 3-3 Product design environment

The problem of realizing the product model knowledge base is very large, but also the potential for benefit is substantial. The whole system concept is built on a very modularized approach where system components have well defined bounded tasks, within a limited problem domain, where the knowledge can be captured and where a well-defined application vocabulary can be specified. There must be some terms in common between the vocabularies of the different application modules. They will be defined to different levels of detail – abstraction levels. Through these common engineering terms the control system will make it possible to distribute the task to a specific module for deep reasoning and conclusion – taking overall constraints into account – being defined by other modules or the designer. In order to keep the designer in control her must be well informed and have functions for control. He should be able to define constraints, manipulate them and visualize them. In the engineering "vocabulary" there is a number of established symbols for dimensioning and tolerancing, for the defining max or min torque and forces, etc., or relations like greater or smaller than. These symbols will be advantageous to use as a graphics vocabulary in a system. The designer must also be able to stay in control by asking for verification like: "Check if these constraints can be fulfilled simultaneously", etc. In the framework, there should be a number of knowledge bases and expert systems of different kinds, for standards, for material selection, for designing different machine elements and for different components like robots, screw drivers etc.

In the systems' framework the user is working through a user interface which is an interaction module. Through his own interaction module he can view and consult the knowledge bases and expert systems in the framework, in an intermixed working situation. He works towards these different systems through an inference engine, which is supported by a meta knowledge module. All consulted expert systems knowledge bases and 'intelligent" programs etc., are giving the meta knowledge module their vocabulary, telling the meta knowledge module therefore the vocabulary of the whole system is available in the thesaurus with synonyms, etc.

The application skeleton is a kind of expert system for doing certain designs. It can for instance be a support tool for designing hydraulic cylinders, or for designing hydraulic valves. It can guide the designer in doing his work and can give some advice on making analyses that are important to do at a certain stage, or advice the designer to consult a certain specific knowledge base or expert system. In the order phase the application skeleton can be a much more rigid expert system, having its own knowledge base of knowing exactly what should be captured from the other expert systems or knowledge bases in the systems framework as a whole.

The help and instruction module is very important. It should give general help and instructions and also work as an educational too. This means that the application skeleton has more deep knowledge in a specific subject or that the meta-knowledge module "knows" where it can be picked up, from different modules in the system framework, to make it available for he education module. The education module should work on the knowledge that is available in the total system, but according to certain education strategies.

In order to make this a reality a well-defined engineering vocabulary has to be defined, with well defined semantics in specific problem domains and knowledge bases. With the interaction history logged in the product model knowledge base, the vocabularies of achieved results and the actual problem situation can be matched against the vocabularies of the whole system. Specific questions can then be related to the intention of the designer through achieved results and the actual problem at hand. This gives the system relevant facts and constraints in searching for the relevant "help information".

The information control gives the user support in controlling all activities going on in the system. It knows what kind of questions that have been sent out to different modules to be answered and how they relate to the result being produced by the designer or planner. The control system controls with process to start up and stop. It controls when, how and what to input and output. Apart form that, the control system have a lot of tasks to perform:

- Interpret design language
- Interpret design intent
- Control information flow
- Monitor questions/answers
- Control relations between product data administration and used processes and databases/knowledge bases
- Control inference/matching of data/questions and answers with data (1) in the product model and (2) in the model of the problem situation
- Utilize meta knowledge/abstractions, related to the problem and application domain,
- Control search and generate searches strategies
- Control communication of data
- Control communication for collaboration between users in the environment

- Have an overall application related control for different domains/cases/scenarios according to the problem solving strategies for the type of design
- Control design documentation in the product models (1) that it is complete and (2) done according to defined rules
- Control that verification is done and according to given rules
- Control design changes
- Generate design history with definitions and applied rules related to specifications and justifications.

The control system should just be a tool for the user, the design or engineer using the system. The user should always have the ultimate control. The control mechanism helps in integrating distributed units. In the software scope, each of the programs running for a unit process is separated from every other and encapsulated, and only ordered strictly by the control mechanism in the integrated environment. Any communication between two programs must rely on translation by the control mechanism. A configuration is provided by the control mechanism to the users, which enables users to readily add, edit, or delete programs. When a new package is to be integrated into the integrated environment, only the control mechanism need to be modified, while other programs are kept unchanged. The most significant fact is that successfully developed software can be applied wherever needed.

3.3 Manufacturing Feedback in Real Time

The term real-time is used to indicate that the computation is triggered by external events and that the results of the particular calculation may depend upon the value of the variable "time" at execution, or upon the time taken to execute the computation (Bennett 1984). The real-time computer is required to communicate and interact with a number of peripheral devices in order to carry out its process-related assignments. In a complex real-time system, i.e., one with many different computer- and process-related peripheral devices, more attention must be paid to the interconnection of computer and devices.

The principle architecture of the real-time system must be designed and evaluated according to three basic predicates of real-time control:

- Flexible and scalable: The architecture must allow for implementation that operates on a single process system up to implementations that make use of the capabilities offered by advanced operating systems; real-time operating systems especially offer a wide range of inter-process communication and parallel execution of processes, often with specific shedding strategies. The architecture of the interface to local applications must allow for implementation by being linked to the application process or by a connection based on the server/client concept where each application can connect to the shell, without the need for link actions.
- Lean: Real-time systems are often embedded systems without mass storage, such as a disk drive. The architecture must enable the management rules and data in the memory of the computer system. Certainly, the possibility to apply disk storage or even make the use of a database management system must be included.
- Enable deterministic timing: A clear architecture with obvious message paths and the possibility to follow the exact actions, which are taken to perform certain functions, is required. Additionally, some functions to enable actual measurement of execution times may be helpful.

The manufacturing feedback needs to be hardware oriented because it is necessary to understand its role in providing these facilities listed below:

1. Object-oriented functions. These functions can provide comprehensive and meaningful services only if they are defined in terms of concepts which are a level above the primitive functions of address blocking, block relocation and access checking.
2. Linkers and loaders for a shared environment. The use of shared regions by tasks in a real-time system is extremely common. While calls from within tasks can be used to map external shared objects, it is far more convenient to let the loader and linker handle these functions.

Based on the above discussion, a real-time data acquisition and information transmission model is created as shown in Figure 3-4.

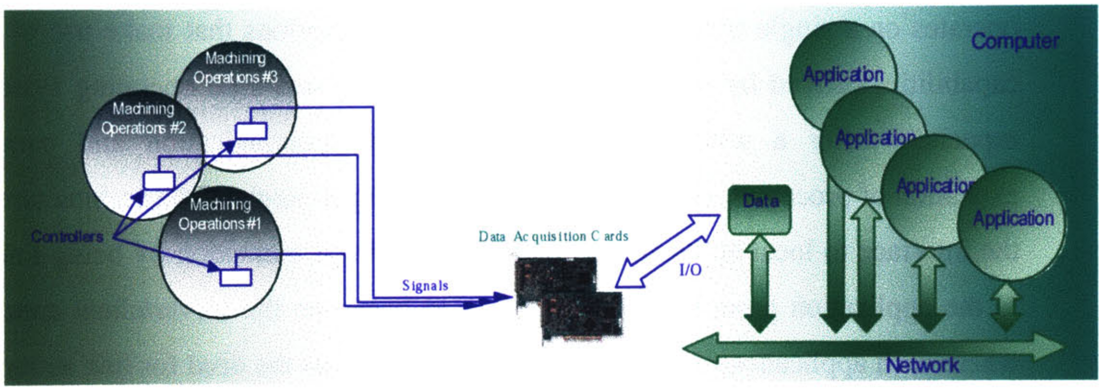


Figure 3-4 Machine data acquisition and transmission in real time

Assume that in different manufacturing branches components of a product be fabricated at different machine stations and in a serious of machining operations. In each operation there exists the manufacturing uncertainty, therefore the machining data should be acquired and organized by the controller of that operation. The data are useful information for part characteristics estimation and many further applications in product design. At the interface between manufacturing and design modeling, a data acquisition card, which has the functionality of receiving digital signals, recognizing signal types, and sending out real-time calls to computers, is adopted.

The major advantage of this real-time machining information feedback model is that it realizes the automation of data acquisition and transmission between manufacturing and design, from hardware to software. Its use conforms to the concurrent engineering idea, and is one of the most important steps in the development of prototype of the product performance prediction of as-manufacture assemblies.

3.4 Product Model Knowledge Base

When utilizing computers to enhance human capability and intellectual capacity, the knowledge representation in the computer is the key issue. The representation of knowledge about a product in design and in the production-planning phase for manufacturing and assembly should be in a form of a product model knowledge base. It manufacturing and assembly should be in a form of a product model knowledge base. It can be looked upon as a sort of "intelligent" product documentation, where "intelligent" means a representation system that can explain its own contents and utilize them (Kjellberg and Wingard 1989).

The product model in the computer is a genetic representation of a number of relevant properties of a product or concept of a product. Properties are represented in an integrated and consistent way. It is important to realize that a good modeling of a product must not only represent the shape and form – the geometry – of a product but also non-geometric attributes such as weight, material, performance, and manufacturing methods. In addition, it must include the design process to allow the early development and evaluation of the product.

How the product should be manufactured is not represented in our product model. There are references to how products should be manufactured. It will be specified and detailed in the process planning. Different kinds of defined features will reference different methods of manufacturing, which can be either specific, or more general. What we concern here is the final results coming out from manufacturing and assembly, which should remain unchanged despite the manufacturing methods.

In order to make the product model into a knowledge base, we must keep the explicit definitions and the applied advice in the form of conclusions from consulted design rules, manufacturing or assembly rules etc. The knowledge represented must also be possible for the product modeling system itself to interpret to a certain level. In order to fulfill these demands, the product model must be defined through well-defined vocabularies for product function, solution principles, form and dimensions, etc., taking the application area into account.

A very important question is the information content of the product model. The listings below give a brief indication of the amount of, and the different kinds of information, which we want to represent in the product model knowledge base. It puts a lot of demands on different kinds of representation.

- Product
    - Name
    - Type
    - Properties
    - Functions
    - Feature List: Form — Dimensions - Symmetry
    - Sub Assemblies
        - Names of Components
        - Type
        - Properties
        - Function
        - Main Part
        - Has Connecting Joints
            - Properties
            - Function — tightness
            - Form — Dimensions — Symmetry
            - Assembly Demands — Constraints
            - Manufacturing Demands
        - Part Clusters
            - Name
            - Type
            - Properties
            - Function
            - Form — Dimension — Symmetry
            - Joints
    - Material Ref. Properties
    - Law Constraints
    - Service Demands
    - Assembly Methods Reference
        - Assembly Planning
            - Assembly Sequence
            - Assembly Methods and Tools
        - Part Name
            - Type
            - Main Properties & Functions
            - Form — Dimension — Symmetry
                - Dimension — tolerance
                - Surface Type
                - Surface Properties
                - Help Geometry
                - Form Feature
                    - Function
                    - Constraints
                    - Dimension — tolerance
                    - Functional Surfaces
                    - Ref. Manufacturing Constraints
                    - Ref. Assembly
                    - Help Geometry

                - Materials
                - Geometry Specifications
                - Feature List
    - Material Ref. M-Properties
    - Ref. Manufacturing Methods & Constraints
    - Ref. Assembly Methods

```
     Other Spec.
-------------------------------------------------------------
```

In establishing a knowledge base, the product structure tree – which represents the topology of the product – is dynamic. The user would be provided with an interactive model. This character of the knowledge base should assure that the user reach to and trace with the intermediate and final data of the product.

Within the product-modeling framework we also want to represent in a formal way the requirement specifications. Before starting the development of a new product or a product program, we have to capture requirements and develop a requirement specification. This specification should include all overall demands in a well-known terminology and when possible, the demands should be quantified. So when talking about models, we also want to represent the requirement specification in a formalized way, utilizing a well-known vocabulary and through that be able to relate to the more detailed product specifications. Within the product modeling framework we therefore want to represent in a formal way the following specifications and models:

- Requirement specification
- Product specification & models
  - Specification of properties
  - Functional specifications
  - Manufacturing specifications/possibilities and constraints
  - Economical specifications/what it may cost related to quality and quantity
  - Service specifications
- Modeling functional concepts
- Modeling of physical principles
  - Quantitative models
  - Qualitative models
- Modeling principle solutions (or solution concepts)
  - Modeling the principle of the solution
  - Modeling its main form and dimensions

- Detailed models
    - Subassemblies
    - Structure
    - Parts
        - Form/dimension/tolerance
        - Form features
        - Joints.

It is important to have a methodology for efficiently capturing and structuring knowledge about a design and its design process as well as for defining engineering methods and procedures. A possible improvement is to construct the product modeling in the knowledge base into categorized component databases. It shall be noticed that each component database may support specific internal relationships, which are difficult to describe in a network scheme because there are no common facilities for identifying related elements. Besides this,

- Component databases may modify information independently of any notion of central control
- Information elements updates in one of them may not have a counterpart in another, and hence there are no corresponding changes in cross-database.

3.5 Summary

This chapter has outlined the main concepts and techniques associated with constructing the framework for assembly performance prediction. Past work on distributed object-oriented modeling environment has made it feasible to manage different and comprehensive function units, enable the independent running of the units, and transfer and share information. A key component in the integration of real-time manufacturing feedback, played as the role of data resource for part characteristics estimation. This integrated prediction system facilitates processing efficiency and offers conceptual as well as structural advantages. This new conceptual design framework can server as a universal configuration to develop high-performance intelligent systems for many complicated applications.

One of the problems in part characteristics estimation is the applying of the adequate methods to accomplish the estimation. The product model knowledge base provides to the designer definitions and modules, and enhances the data classification and processing to a higher integrated level. Where this product data needs to be modeled, the knowledge base can aid the most efficient techniques to make estimations of the part characteristics. The effective use of models in the product design process can also be increased by embodying more comprehensive methods and tools in future modeling environment.

With the real-time feedback information provided to the environment, information can be transferred to the design models before production manufacturing, and the entire assembly's performance can thus be predicted. For example, component tolerances could be dynamically adjusted in the design stage in relation to one another rather than applying rigid tolerance requirements for each part. Different process capabilities for different parts may compensate for each other. In the next chapter, a survey will be conducted in manufacturing information, and particularly in machine tool data.

CHAPTER 4 MACHINING DATA INVESTIGATION

Manufacturing variations influence assembly variations. Each machining operation creates a feature which has certain geometric variations compared to its nominal geometry. In machining, various factors such as deformation of the workpiece and tool, vibration, thermal deformation, inaccuracies of machine tool, etc., affect the machining accuracy. Some of these factors are dependent on the selection of cutting parameters. For a limited number of machining processes, deterministic models have been developed to provide quantitative mappings between the cutting parameters (such as cutting speed, feed, and depth of cut) and machining accuracy (such as surface finish and dimensional accuracy). In this chapter we survey the current researches in machining data.

4.1 Machine Tool and Machining Data

Of the many variables affecting the machining operation, the machining tool is one of the most critical for specific machining operations. As far as the machine tool is concerned, we have variables in cutting tool geometries, tool holders, cutting speeds, feeds, lubrication, admissible cutting forces, horsepower, vibrations, deflections, and thermal expansion (Kronenberg 1966). Great efforts are being made today throughout the world to find out what is going on at the cutting edge of the tool. In this section the relationship between machine tool data, and surface finish and dimensional tolerances will be illustrated through turning and milling processes.

4.1.1 Turning Process

Turing is a machining process in which a workpiece is held and rotated about its longitudinal axis on a machine tool called a lathe. Cutting tools mounted on the lathe are fed into the workpiece to remove material and thus produce the required shape. The turning process has been studied throughout the years with the major parameters affecting turning process well defined. Although there is a gamut of variables that affect the turning process, it is generally agreed that the cutting speed, feed and depth of cut are a few of the most significant. Other parameters such as tool geometry, tool condition, and workpiece material also greatly influence the performance of the process.

Operating variables

Operating variables are the many factors that influence any turning operation. The three major ones are cutting speed, feed rate, and depth of cut.

The formation of the machined surface is due to two motions. These motions are the cutting speed, which is the speed of the work material past the cutting tool edge, and the feed speed, which is the speed of the cutter along the workpiece axial direction. *Cutting speed* refers to the rotational speed of the lathe spindle and workpiece, and can be expressed in revolutions per minute (rpm). Cutting speed constants are influenced by the cutting tool material, workpiece material, rigidity of the machine setup, and the use of cutting fluids. For turning and most other machining operations, however, the cutting speed is generally given in surface feet per minute (spm) or meters per minute (m/min), which is the rate at which the workpiece surface moves past the cutting tool.

The machine movement that causes a tool to cut into or along the surface of a workpiece is called feed. *Feed rate* is the rate at which the tool advances along its cutting path. There are two ways to describe the feed rate. One is in terms of how far the cutter advances in one revolution of the workpiece, and the units are inches or millimeters per revolution (ipr or mm/rev). Another way to describe feed rate is with the more typical definition of speed as inches or millimeters per minute (ipm or mm/min).

Depth of cut is the thickness of the layer of material removed from the workpiece surface (the distance from the uncut surface to the cut surface), expressed in inches or millimeters. When turning cylindrical workpieces, the diameter is reduced by twice of the depth of cut.

Surface finish and tolerance requirements

Surface finishes of 20-50 μin. (0.51-1.27 μm) are the practical limits that can be expected from turning operations when using well-maintained lathes and tools (Judson, 1976). Smoother surface finishes, to 1 μin. (0.025 μm) or less, however, can be produced, particularly with precision machines and diamond cutting tools, but several cuts are generally required, resulting in increased manufacturing costs.

Dimensional tolerances that can be maintained in turning vary depending upon the machine and operating parameters used, the workpiece, setup rigidity, and other variables. Practical limits for production applications, with machines and tools in good condition, range from ±0.001″ (0.03 mm) for workpieces having diameters of about 1/4″ (6.4 mm) or less to ±0.03″ (0.08 mm) for diameters of 4″ (102 mm) or more. Closer tolerances to ±0.000050″ (0.00127 mm) are often maintained, but maintaining these tolerances generally requires the use of more precise machines and results in higher manufacturing costs.

Operating parameters for turning

The parameters of cutting speed, feed rate, and depth of cut control both the metal removal rate and tool life in turning (surface finishes and dimensional tolerances). A change in any one of the three parameters has an equal effect on the metal removal rate; however, each parameter has a different effect on tool life.

Depth of cut. The consensus of most authorities in metal-cutting is that the best method of increasing the metal removal rate is to use the deepest cut possible. Depth of cut, however, is limited by the amount of stock to be removed, power capability of the machine, rigidity of the setup, tooling capability, surface finish and accuracy requirements, and sometimes by the shape of the workpiece.

Feed rate. It is generally recommended that the largest possible feed rate be used to obtain a higher production rate and a lower power requirement per volume of stock removed. Increases in feed rates, however, are limited by the ability of the machine tool, cutting tool, workpiece, and setup to withstand the cutting force, as well as by the surface finish required on the workpiece. – *Surface finish.* The feed rate used and the nose radius on the cutting tool have the greatest impact on the surface finish produced in turning. Increasing the nose radius or reducing the feed rate improves the surface finish.

4.1.2 Milling Process

Milling is a machining process for removing material by relative motion between a workpiece and a rotating cutter having multiple cutting edges. A characteristic feature of the milling process is that each milling cutter tooth takes its share of the stock in the form of small individual chips.

Milling methods

The major milling methods are peripheral and face milling; in addition, a number of related methods exist that are variations of these two methods, depending upon the type of workpiece or cutter, such as end milling, etc.

Operating parameters for milling

There are so many variables influencing milling that it is difficult to predict results reliably. These variables include the size and shape of the workpiece; the material from which it is made (its machinability, physical properties, and condition); the kind of milling operation to be performed; the type of cutter used (its material and geometry); the machine employed (including its condition); the rigidity of the setup; and the production rate, tolerance, and surface finish requirements. Width and depth of cut, as well as the tooth entrance angle, also have a profound effect.

All these variables, notwithstanding, an attempt must be made to establish at least initial parameters for any milling operation. These parameters include power and force requirements, cutting speed, feed rate, depth of cut, and cutting fluid to be used. Optimum parameters cannot be established, however, and truly reliable predictions of results cannot be obtained until the milling operation is actually performed and trial parameters are tested.

Feeds and speeds for milling

The two factors, in addition to depth and width of cut, that have the greatest effect on productivity and forces acting upon the cutter, machine, workpiece, and fixture in milling are the feed rate and cutting speed. They therefore influence the surface finish and dimensional tolerances.

4.2 Dimensional Accuracy

It is difficult to achieve high rates of material removal and obtain good surface finish and tolerance in the same cutting operation. Separate 'roughing' and 'finishing' operations are required.

Often of primary interest, are the finished part's dimensional accuracy and surface qualities, which can be required in the face of very demanding productivity goals. Dimensional accuracy is important for components that are used in precision mechanisms and high performance machines. Surface quality is important where aesthetics, wear resistance, paint and coating adhesion and lubrication retention is concerned. In a turned part, surface quality and dimensional accuracy are directly affected by the relative radial displacement between tool and workpiece that arises during cutting.

Tolerances can be defined as being concerned either with physical sizes of features on a part or with the geometric characteristics of those features. Among the geometric characteristics, profile of a surface, position, concentricity, and symmetry are handled within tolerance control of sizes.

Surface finish is a term used to describe the general quality of a workpiece surface. It consists of roughness, waviness, lay, and flaws. Surface finish and dimensional tolerances usually go hand in hand. Bearing surfaces and locating surfaces usually require close dimensional and surface finish control for proper functioning and for ensuring that functional dimensions are maintained throughout the useful life of the workpiece.

4.3 In-Process Control of Dimensional Accuracy

In-process control refers to the methods of control part accuracy during machining process. The meaning of "in-process" (*in situ*) measurement is the measurement of workpiece size, while the component is located on the machine tool and preferably while the machining process is in progress. The available technologies for "in-process control" of machined products, prior to the completion of the machining process, are various, particularly in relation to turning, milling, and cylindrical grinding. The techniques developed for controlling accuracy of the workpiece, while *in situ*, are described in this section using representative examples of current technology.

Automated inspection methods are the solutions to the increasing demand by customers for near perfection in the quality of manufactured parts and a desire to decrease the costs involved. These inspection methods should be able to control the accuracy of the part during the machining process and provide the required feedback information for the necessary corrections.

4.3.1 Mechanical methods

A mechanical method of "in-process measurement" can be defined as one in which the measuring transducer, operates in mechanical contact with the workpiece, although the actual signal may be electrical or pneumatic.

1. Direct methods.

Caliper type A typical caliper type contact gauge consists of a simple scissors caliper with non-rotating circular contact pads. The instrument can be set to measure over a range of diameters. Various types of caliper gauges, such as scissors and C calipers, and substantial descriptions with a wide range of experiment results have been performed by Tillen (Tillen 1964).

In this method, the contact pads or jaws are in continual rubbing contact with the workpiece. It is attached to the machine bed on its own slide so that it can be rapidly withdrawn and returned to the measuring position in a repeatable fashion. The rear gap of the scissors is bridged by a sensing element, which can be pneumatic or electrical transducer. The caliper is set with respect to a circular setting master. It is possible to derive an electrical signal with both types of transducer, which can be used to control the machining process such as grinding and turning.

Calipers are a popular method of direct measurement, often used in grinding. This is because grinding is naturally suited to this type of measurement, as it is a multiple point cutting process, removing small amounts of material from a workpiece with a fine surface texture and on a good quality machine tool. The wear of contact heads is the main drawback of this method, particularly if it is applied to a turning process. Furthermore, these must be set with respect to a setting master each time the workpiece size is altered, causing an increase in lead times.

Friction-roller type The friction-roller method measures the perimeter of workpiece by counting the number of revolutions of the measuring roller for one or more complete revolution of the workpiece (Nokikiv and Makarevich 1961; Ivanov and Mel'nichuk 1964). The friction-roller type instrument uses a roller contacting the peripheral surface of the turning workpiece. Each revolution of the roller produces a number of pulses. Reference pulses are also derived once per revolution of the workpiece, and are used to stop and start the counter, so that it reads the number of measuring pulses every revolution.

In this method, accuracy varies with the type of material. Moreover, the application of the friction roller is restricted to rigid workpieces, due to the high pressure applied by the roller. The instrument, in general, gives better result with larger diameter workpieces than smaller ones, although the resolution is unaffected by workpiece diameter (Lee 1963).

2. Indirect methods.

Probe type A probe in mechanical contact with the workpiece is used to determine the actual size of workpiece. For the gauging process, the probe is moved towards the workpiece and deflected by the contact. The co-ordinate value of the point of touch makes it possible to determine the workpiece radius provided the position of the axis of rotation is known.

The measuring probes are mounted in tool holders and deposited in the tool turret of the machine during cutting. One such existing system is the well-known Renishaw touch probe (Anon. 1984).

The significant advantage of this system is that the same sensor can be employed to measure both the internal and the outer diameter or workpiece lengths. With this system, however, the accuracy of the measurement is subject to the accuracy of an individual machine's positioning system and also the traceablility of measurement cannot generally be achieved.

4.3.2 Optical methods

An optical method of in-process measurement is defined as one is which the transmitter module produces and emits a light, which is collected and photoelectrically sensed through the object to be measured, by a receiver module. This produces the signals that are converted into a convenient form and displayed as dimensional information, by the electronics process.

The chief advantages of optical methods are that no physical equipment is required in the vicinity of the workpiece, and information on the diameter deviation is conveyed by means of light from the workpiece to a suitable detector. The optical system can usually be arranged to enable the desired distance between the source or the detector and the workpiece to be determined.

The use of optical methods increases the rate of dimensional data acquisition. Particularly in recent years, modern photoelectric devices and advanced computers have been developed, thus making the optical system the most convenient methods. In addition, the availability of He-Ne lasers resulted in the practical use of optical methods.

1. Direct methods.

In direct optical methods, the diameter of the object is generally measured by interrupting the light emitted from a transmitter, and by detecting this light electronically, to obtain electronic signals, so that this basic data can be converted into dimensional readings.

Scanning light beam Since this technique, in general, uses laser beams for the measurement process, it is also called a canning laser beam. It basically employs a transmitter module that emits a high-speed scanning laser beam, generally by means of a combination of a mirror and a synchronous motor. The object to be measured interrupts this beam, and produces a time-dependent shadow. This shadow is electronically detected by a receiver, and converted into dimensional readings by a control unit. An instrument has been built for measurement of a rod diameter using a scanning laser beam technique (Binks 1971). Due to recent development in optics, submicron resolution can be achieved with this technique (Anon. 1990).

Machine vision The use of machine vision systems, for inspection, is an exciting area, which holds the promise of significant improvements in both the productivity of the inspection process, and the quality of the resulting product. First attempts have been made using these type of techniques for "in-process measurement" particularly in Russia. Vikhman (Vikhman 1963) proposed a system in which a conventional light source is used and in image of the workpiece is focused on the measuring grid on the face of a television tube. Later, a more advanced technique using a charge coupled device (CCD) and a microcomputer has been developed (Takesa *et al.* 1984). The technology of machine vision inspection is one in which advancements and refinements are continually being made. Further improvements in vision technology will allow improved resolution.

Light gauging The commercially available prototype instrument by Novak that uses a light-gauging method is capable of measuring diameters up to 280 mm with a resolution of 0.001 mm (Novak 1981). This instrument can operate on the machine tool or be handled by a robot. It is based on He-Ne laser. The laser beam is divided into two measuring beams and a reference beam by the beam divider. The measuring beams are first reflected towards the workpiece and then back, to be detected by a detector. The parallelism of the measuring beams does not depend on the accuracy of the mechanical parts which guide the optical components.

2. Indirect methods.

Indirect optical methods are defined as those in which the diameter of the workpiece is indirectly measured by means of features of the light reflected from the workpiece surface. In general, the light beam is projected on to the workpiece, and reflected onto a photodetector or any electronic device. Any change in the diameter or workpiece, results in a change in the location of the image or a change in the location of the focusing point on the workpiece. Hence, the workpiece diameter can be measured by determining the relation between one of these changes.

Light focusing The focusing point of an incident beam on the workpiece surface that produces maximum light intensity, is detected by a photo-detector. A change of workpiece diameter yields a deflection of the focusing point and leads to less reflected light intensity or, a change in the place of reflected light image on the photo-detector. The sensor output is then used to operate a servo-mechanism, which controls the tool position to maintain the diameter within a prescribed level.

Light-spot detection The finish size of a workpiece is continuously monitored by detecting a displacement of a laser spot, which is reflected from the workpiece surface.

Light sectioning The method introduced by Shiraishi and Sato (Shiraishi and Sato 1990) can be considered as a light-sectioning method. This method not only controls the dimension of workpiece, but also it makes a smoother surface, by removing surface irregularities during machining.

4.3.3 Pneumatic methods

A pneumatic method can be defined as one in which the location of surface is measured, by means of an air jet stream directed against the surface to be measured, and by obtaining a signal from the variation in back pressure in the air feed line. In other words, the measuring system measures a pressure drop in the gap between the air gauge and workpiece, and converts it into an electrical signal.

The advances of the pneumatic methods are: unaffected by the workpiece material; high speed gauging and control; and application over a wide temperature range. Using these advances, gauging instruments can be built for application to in-process measurement. However, the main disadvantage of the pneumatic methods, in the case of "in-process measurement" of workpiece diameters in turning, is the small gap between the sensors and the workpiece.

4.4 Predicting Dimensional Accuracy

Dimensional accuracy is one of the primary concerns of the manufacturing process. As products become more sophisticated and precise, the requirement for manufacturing accuracy increases. Thus it is important to integrate tolerance analysis into design and manufacturing to achieve a higher success of assembly. Notable efforts have been made toward tolerance analysis in CAD and CAM.

4.4.1 Prediction of dimensional tolerances

1. Cutting Force distribution and its influence on dimensional accuracy

Cutting force has a significant influence on dimensional accuracy due to tool and workpiece deflection. Machine error due to the cutting force originates from tool deflection and workpiece deflection. Not only the (accurate) cutting force, but also the dynamic cutting force should be evaluated in order to accurately investigate the influence of cutting force on the part quality.

For the cutting force prediction, several models based the theoretical assumptions and experimental observations have been developed and reviewed by Smith and Tlusty (Smith and Tlusty 1991). Besides, there are a few enhanced cutting force models that have been developed in the last decade.

The influence of cutter static deflection due to cutting force on the machining error is investigated by Kline, Sutherland, and Devor (Kline et al. 1982; Sutherland and Devor 1986). The machining error was predicted using the cantilever beam theory for the cutter deflection and the finite element method for the workpiece deflection. In these references, however, the dynamics in the tool/workpiece system was neglected.

The influence of the tool/workpiece system dynamics on surface generation was investigated by Montgomery (Montgomery and Altintas 1991). The kinematics of the cutter and workpiece vibrations was modeled. Zhang introduced the effect of random vibrations on the surface roughness in turning. These vibrations were shown to occur due to the material. Ismail presented a mechanistic model for surface generation in peripheral milling, which includes the effect of cutting vibrations, run out and flank wear.

An improved theoretical dynamic cutting force model of peripheral milling was presented by Cheng and Liu (Liu et al. 2002). They derived analytical differential equations by which the cutting force can be approximated by definite integrals with numerical integration. The cutting force coefficients in these equations are dependent on workpiece material, cutter material, cutter geometry, cutting edge radius, and friction characteristics between the workpiece and the cutter. In order to obtain values for the coefficients, some previous measured cutting forces are considered. Yucesan and Altintas have presented a detailed description of their experimentally measured cutting forces in the peripheral milling of a titanium alloy (Yucesan and Altintas 1994). These experimental results were used to verify a different cutting force model, and it is appropriate to use them to verify this improved cutting force model. Cheng and Liu chose the same cutter, workpiece material and cutting conditions for their simulation as in the cutting test conducted by Yucesan and Altintas. They demonstrated that predicted cutting forces are very good approximations to the measured cutting forces, and the improved dynamic cutting force model can be used to predict the cutting forces accurately.

2. Tolerance analysis in setup and fixture planning

Among many factors that affect accuracy in the manufacturing processes, parts' setup and fixturing is very critical. The tolerance analysis in setup and fixture planning is broad and includes datum selection, position and geometric tolerancing, error analysis and control strategies.

An approach to including tolerance analysis into setup and fixture planning for precision machining was presented by Huang and Gu (Huang and Gu 1994). Their work focused on classification and calculation of errors, tolerance transformation and synthesis based on precision machining considerations.

Setup and fixturing are two essential activities in the machining process. Generally speaking, setup concerns how to orient a workpiece at the machine table, while texturing considers how to locate and hold the workpiece. They are tightly coupled activities in the machining processes. A setup would not stand if there were no fixtures available to support it; a fixture would not be configured if no setup orientations were given. The bottom line for setup and fixturing is to ensure the stability and precision of machining processes.

To meet the accuracy requirement, one might not fixture a workpiece on some surfaces, although they are feasible from a holding point of view. The guideline for setup and fixture planning is tolerance requirements.

Tolerance analysis involves checking all possible deviations in dimensions and geometric forms due to machining, setup and fixturing against the design requirements. It is an important part of ensuring machining accuracy. Tolerance analysis should also yield a proper guidance to the selection of setup and design of fixture configurations. In addition, tolerance analysis in setup and fixturing requires more investigations on the error sources and their characteristics.

At present, most automated setup and fixture planning systems do not support proper tolerance analysis. In many cases, they do not rely on some heuristic rules and simple tolerance checks. To make a planning system adaptable to various situations, it is clear that tolerance analysis is needed to provide the system some "deep" knowledge to guide the planning.

During an operation cycle, errors caused by setup and fixture can be classified as: *setting error,* due to the accuracy limit in measurement and adjustment of the locators; *datum feature error,* because the datum faces of the workpiece may always exist with some degree of imperfection; *clamping error,* because of the deformation of the workpiece; *forced error,* caused by the cutting force and torque; *tool wear,* due to the cutter wear; *machine error,* due to the accuracy limit of machine movement and positioning; etc. Industrial experience indicates that it is impossible to eliminate these errors. It is possible, however, to control them within an allowable range through proper setup and fixturing. According to the characteristics of the errors classified above, different methods are developed to calculate setting error, datum feature error, clamping error, forced error, tool error, and machine error, etc.

4.4.2 Prediction of surface roughness

The roughness of machined surfaces can be expressed in terms of a single factor or index. The main factors contributing to surface roughness are cutting speed and feed (Boothroyd 1975).

Additional factors that commonly contribute to surface roughness in practice are:

- the occurrence of chatter or vibrations of the machine tool;
- inaccuracies in machine tool movements such as the movement of the saddle on a lathe;
- irregularities in the feed mechanism;
- defects in the structure of the work material;
- discontinuous chip formation when machining brittle materials;
- tearing of the work material when ductile metals are cut at low cutting speeds; and
- surface damage caused by chip flow, etc.

1. A genetic algorithmic approach for optimization of the surface roughness prediction model

Due to the widespread use of highly automated machine tools in the industry, manufacturing requires reliable models and methods for the prediction of output performance of the machining process. The prediction of optimal machining conditions for good surface finish and dimensional accuracy plays a very important role in process planning.

Process modeling and optimization are two important issues in manufacturing. The manufacturing processes are characterized by a multiplicity of dynamically interacting process variables (Azouzi and Guillot 1998). Surface finishing has been one of the most important considerations in determining the machinability of materials. Surface roughness and dimensional accuracy have been important factors in predicting machining performance of any machining operations. The predictive modeling of machining operations requires a detailed prediction of the boundary conditions for stable machining.

Most surface roughness prediction models are empirical and are generally based on experiments in the laboratory. Generally these models have a complex relationship between surface roughness and operational parameters, work-materials and chip-breaker types.

Optimization of machining parameters increases the product quality to a great extent. An effort has been made to estimate the surface toughness using experimental data by Suresh and Rao (Suresh et al. 2002). Others have attempted to optimize the surface roughness prediction model using a genetic algorithm (GA) approach.

Since turning is the primary operation in most of the production processes in the industry, surface finish of turned components has a greater influence on the quality of the product. Surface finish in turning has been found to be influenced in varying amounts by a number of factors such as feed rate, work material characteristics, work hardness, unstable built-up edge, cutting speed, depth of cut, cutting time, too nose radius and tool cutting edge angles, stability of machine tool and workpiece-setup, chatter, and use of cutting fluids. In Suresh and Rao's work, experimental results were used for modeling using response surface roughness methodology (RSM). The RSM is practical, economical and relatively easy to use and it was used in lots of research. It is a collection of mathematical and statistical techniques that are useful for modeling and analyzing problems in which response of interest is influenced by several variables, and the objective is to obtain the response. Experimental data was utilized to build a mathematical model of the first-order and a second-order model by the regression method.

This mathematical model was taken as objective function and was optimized using a genetic algorithmic approach to obtain the machining conditions for the required surface finish. Genetic algorithms are search algorithms for optimization, based on the mechanics of natural selection and genetics. The GA approach, used to optimize using the mathematical model, was found to be the most useful technique for research. The simplicity of the operation and computational efficiency are the two main attractions of the genetic algorithm approach.

A detailed survey was carried out to find out how machining parameters affect surface roughness. Based on this, the four parameters of speed, feed, depth of cut, and nose radius of the cutting tool were selected for Suresh and Rao's experimentation. The experimental results were modeled using RSM and respective first-order and second-order models were developed. The models were analyzed based on regression coefficients and an appropriate model was selected for optimization. The machining parameter levels were fed to the GA program. The GA program uses different types of crossover and mutation operators to predict maximum and minimum values of surface roughness. This GA approach provides optimum machining conditions for corresponding, given maximum and minimum values of surface roughness. This approach is quite advantageous in order to have the range of surface roughness values, and their corresponding optimum machining conditions, for a certain range of input machining parameters. The predictive capability of GA could also be incorporated for automatic monitoring, in order to plan operations. The surface roughness prediction model takes into account cutting speed, feed rate, depth of cut, cutting tool, nose radius and their interactions. With the known boundaries of surface roughness and machining conditions, machining could be performed with a relatively high rate of success with selected machining conditions.

Suresh and Rao's two-stage effort of obtaining a surface roughness model by surface response methodology, and optimization of this model by generic algorithms, has resulted in a fairly useful method of obtaining process parameters in order to attain the required surface quality. The methodology of prediction models, along with optimum machining conditions, can be used in computer-aided process planning, and computer-aided manufacturing. The application of the GA approach to obtain optimal machining conditions will be quite useful in the production of high quality goods with tight tolerances by a variety of machining operations, and in adaptive control of automated machine tools.

2. A fuzzy knowledge-based system for predicting surface roughness in finish turning

Predictability of surface roughness in finish turning is a major requirement in component manufacturing, as this would influence the acceptability of the manufactured part based on the achievable consistency, hence contributing to increased productivity and reduced cost of manufacture. The well established feed and tool nose radius-based relationship for predicting surface roughness (i.e. $R_a=32.1\ s^2/r_\varepsilon^2$, where s = feed and r_ε = tool nose radius), does not give any correlation to the experimentally obtained surface roughness and the experimental values are always by about 60-80% higher than the predicted values in finish turning (Jawahir et al. 1990).

3. On the prediction of surface roughness in turning using artificial neural networks

The aim of this work is to use the technique of artificial neural networks, with a back-propagation routine, for the prediction of surface roughness of workpieces produced by the turning process. The vibration level during machining, and the after-machining tool-flank wear were measured and the surface roughness parameters of turned specimens were assessed.

Several attempts were performed to predict the surface roughness in the turning operation by using either the mathematical models or the artificial neural network technique. All proposed prediction models take into account cutting speed, feed, cutting tool nose radius, and their interactions.

El-Sonbaty and Megahed's work is an approach for predicting the surface roughness, in the turning operation, using the feed-forward artificial neural networks technique trained with the back-propagation routine (El-Sonbaty and Megahed 2000). The inputs to the neural network of surface roughness prediction are cutting speed, feed, depth of cut, nose radius, pre-tool-flank wear, and vibration level.

CHAPTER 5 FEASIBILITY ASSESSMENT

After the investigation made in the previous chapter about manufacturing information acquiring and data processing, there follows the question, what improvements can be brought to assembly to assist product design and development through assembly performance prediction? In this chapter, we look at the problem of handling the task of assembly at the design phase, analyze the factors that influence assembly performance, and assess the feasibility of the assembly performance prediction idea based on different assembly concerns.

An assembly is defined as a configuration of parts of known geometries subject to tolerances in the pose, dimensions, and mating relations among part features. The design of products for assembly requires a careful consideration of the many factors that influence the functionality and manufacturability (Sanderson 1999). While stability and relative precision of part positions are often essential for the functional performance of assemblies, these same requirements may make the product difficult to manufacture. At the same time, dimensional clearance among parts is essential to create paths for assembly operations; fine motion strategies may utilize contact between parts to guide the assembly motions, and often the addition of fixtures and supports may be necessary to maintain planning of part designs and subassembly groupings; and also, assembly sequencing is critical to efficient and reliable manufacturing processes.

In practice, part geometries are not controlled precisely during manufacturing, and there is also uncertainty in each positioning operation. These uncertainties may be represented by tolerance specifications in the parts and the assembly relations. Investigating these tolerance uncertainties is important to the understanding of the difficulty of the assembly, and may be critical to the determination of a sequence feasible to the compensation of tolerance stackup. A given sequence may cause the tolerances among parts to propagate and accumulate as the assembly operations proceed, and the accumulated tolerances may result in an infeasible operation for some percentage of the assemblies produced. Tolerance is a representation of a stochastic geometry of the parts and positions, so the resulting analysis and reasoning is inherently probabilistic. The question is: even if an assembly sequence is feasible for the *nominal* geometry of the parts, what is the *probability* that, in practice, tolerances may accumulate to make the resulting assembly infeasible? Are there some tolerance stackup compensation works that will increase, or guarantee, the probability of successful assembly? How should the design of product assembly and the tolerancing of the parts be changed to improve the assemblability? When is it cost-efficient to loosen the tolerance on parts to improve the reliability of manufacturing?

5.1 Assemblability

Assemblability can be evaluated based on tolerances and adjustable displacements. An adjustable displacement is a functionally permitted free space between two mating features, which can be used for compensating tolerances. The assemblability of a product is computed in terms of a multi chain by incrementally solving the parallel chains. We consider both the functionality and assembly sequence constraints in the evaluation. The result of the computation is a statistical measure of the product assemblability, which can be used by the designer to evaluate and to optimize the tolerance allocation. In addition, this measure can be used for evaluating assembly sequences.

The tolerances of parts affect the assemblability of a product because small errors can propagate and accumulate in the product, creating larger pose errors and assembly problems. An assembly is typically composed of many parts that are assembled together through their mating features. The errors can propagate through these mating features.

Tolerances in an assembly can be compensated through using adjustable displacements, which are the permitted clearances. *Clearance* is a free space between the two mating features that allows one mating feature to move freely within the space with respect to the other mating feature. A *mating feature* is a feature (or surface) of a part, which plays a functional role in an assembly, for example, to satisfy geometrical constraints specified by mating constraints, assembly sequence constraints, or functional constraints. An *adjustable displacement* is a permitted free space defined by the clearance and the functionality requirement of the two mating features. For example, in order for a product to function to its desired level, because of some mechanical constraint, the cylindrical surface of a round peg must not be in contact with the cylindrical surface of a round hole.

The assemblability of a product design is defined as the probability of successfully assembling the varying parts that are within the tolerance specification (Lee et al. 1997). By definition, tolerances work against the assemblability, and adjustable displacements work for the assemblability. Consequently, the assemblability measure is the probability of errors that can be compensated by adjustable displacements. For example, a parallel chain is assemblable if two adjustable displacement ellipsoids of two serial chains of a parallel chain intersect for given instances of tolerances. This is because the intersection area denotes the area where the two mating features of serial chains can be assembled successfully by bringing their poses to anywhere in this area.

5.2 Stackup Compensation

Given a drawing of a component and the raw material and told to make one piece, a manufacturer will machine the part and upon inspection will find that it conforms very closely to the mean dimensions on the drawing. This is a result of machining from one feature to another, zeroing out each completed feature, and using that zeroed out condition as the datum to machine the next feature. Tolerance stackup is bypassed.

In production planning for quantity runs, however, the part cannot always be machined dimensionally as shown on the drawing, so datum surfaces must be set up by the production engineer based on a selection of locating surfaces for fixturing and on cutting tool design layout decisions. As a result, the problem of tolerance stackup occurs.

The principle of tolerance stackup – or buildup of tolerances, or accumulation of tolerances – is, simply, addition or subtraction of length dimensions by the adding of tolerances on the individual lengths. However, because the assembly variations usually come from the manufacturing variations, the tolerance stackup problems appeared in assembly could not be easily solved through simply applying the tolerance stackup principles to the component nominal dimensions. For years tolerance stackup problems have been one of the main causes in design for assembly and assembly evaluation problems. The widespread and growing use of NC machines, when they can be applied, has reduced the extent of the tolerance stackup control problem by eliminating manual control of machine decisions affecting the cuts and by reducing the number of location surface changes and the attendant fixturing required by non-NC machining. In general, it has also improved the control of size and the control of geometric characteristics of part features. However, not all the tolerance stackups are eliminated through using NC machines.

Whether a part is made completely to print in one operation or routed over a series of machines, some of which may be CNC, the process engineering must be capable of recognizing that a tolerance stackup situation has been created which will affect the tolerances assigned to the machining cuts. When tolerance stackup problems must be handled, the earliest, quickest, and most foolproof way is by the use of the tolerance chart.

Whether the tolerance chart is built manually or by a computer program, it is only built after all the initial engineering decisions have been made concerning the process. These decisions include:

1. The sequence of operations to be performed.
2. The machine selection for each operation, based on its capacities and known accuracies.
3. The dimensioning patterns for the cuts to be made in each operation.
4. The selection of the locating surface to be used in each operation.
5. The kind and type of tooling to be used in each operation to control geometric characteristics such as squareness, parallelism, concentricity, symmetry, etc.

Once these decisions have been made, and possibly subjected to critiquing by tool engineers, and the master mechanic, etc., a tolerance chart can be constructed to generate the dimensions and tolerances required by each process cut. Properly constructed, the tolerance chart will verify that the following criteria for economical production have been satisfied:

1. Within the framework of the process/tooling decisions, as much as possible of the blueprint maximum tolerance has been allocated among the in-process cuts, which results in the maximum possible tolerance being assigned to each cut in the process.
2. The minimum and maximum stock removals on secondary cuts are practical and acceptable to the shop.
3. Every tolerance assigned is equal to and preferably larger than the estimated process capability for the cut in question. Since the relationship between the working tolerance and the process capability has a direct bearing on the frequency of tool changes or adjustments, many companies have in-house rules for the working tolerance to be 1.5-2.0 times the process capability value.

During the process of building the tolerance chart, it may become obvious that one or more of the initial process/tooling decisions results in assigning an impossibly tight tolerance to an in-process dimension. When this happens, it is necessary to change these decisions to satisfy the criteria for economical production.

Since all these decisions are still in the paper stage, that is, no tooling has yet been designed; no great time or dollar loss will occur if a process change is required. However, failure to respond to the clear signals from the tolerance chart will result in the problems on paper being transferred into iron on the shop floor.

In the above discussion, the tolerance chart is constructed after the initial engineering decisions being made, and the generation of tolerances is based on either past experience or theoretical calculations. In this process, the manufacturing variation is involved in such a way that the tolerance of a part is corrected by a mean value or with a range. This leads to a rough result in tolerancing. Some in process charting for compensating tolerance stackup can improve the accuracy through acquiring the geometry information of the part after each machining process and adjusting the tolerance compensation dynamically. It is obvious that the earlier and quicker way in the acquiring of manufacturing information will generate more accurate data for the assembly estimation, i.e. for tolerance compensation, and so definitely benefit design process and reduce cost.

Product performance prediction provides a quickest way for getting the stackup information. The machine tool data feedback in real time provides the necessary base to improve dimensional accuracy, and with appropriate methods tolerances can be estimated. Stackup can be simulated through assembly modeling software, and compensation can be performed.

5.3 Assembly Sequencing

Process planning in general, and specifically assembly planning, is the topic that has attracted much research effort. In the context of assembly planning much work has been undertaken to develop specific representations to support assembly planning and sequence evaluation.

Assembly planning consists of two major activities: assembly modeling and assembly sequence planning. Assembly modeling starts with CAD design model of a product, and generates a set of connections between product components, and a set of functional precedence constraints between these connections. These sets are aimed at assisting the generating of an optimal or a best assembly sequence. Assembly sequence planning on the other hand, is the procedure of generating an optimal or a best assembly procedure of generating an optimal or a best sequence to assemble the product, given its modeling information.

Assembly sequence is the most basic information contained in an assembly plan, which lists an order in which the parts can be assembled to produce assembly. The determination of the proper assembly sequence to be used is critical, because it affects a number of aspects of the assembly process, such as definition of subassemblies, selection of process equipment, design of special tools and fixtures, etc. In addition, sequence issues highlight assembly machine and tooling design problems, such as part approach directions, tolerance buildup due to prior assembly steps, access for grippers, stability of subassemblies, number of tools needed, tool change requirements, and so on. This means that the assembly sequence must be known to the parts designer very early in the design process. By contrast, the usual practice has been to delay consideration of assembly sequence until after the parts are designed and fabrication methods have be chosen. Since different fabrication methods cost different amounts and are capable of making parts to different tolerances, these choices, if made without assembly process knowledge, can render an assembly sequence unrealizable.

A major obstacle for assembly process automation and its integration with other processes such as design and scheduling, is the assembly sequence planning problem. It can be simply stated as how to find the best or optimal sequence to assemble a product, given its design characteristics and CAD data. The automation of this problem proved to be very complicated, and is still performed manually in most assembly firms, thus consuming human power and time. Many planners were introduced in research, most of which (1) seek exact solution, (2) perform a part-by-part search procedure, or sometimes (3) are restricted to linear solutions, where one part is assembled at a time.

It is desirable to select a satisfactory sequence from the set of all feasible sequences. A typical product can have a very large number of feasible assembly sequences and there is a need to develop a systematic and efficient method to evaluate all of the available alternatives and choose the best one, considering the available resources and facilities. To determine the feasibility of an assembly sequence, it is necessary to analyze various assembly constraints.

The specification of tolerances is an integral part of product design since tolerances directly affect the assemblability, as well as the functionality, manufacturability, and cost-effectiveness of the product. Assembly planners in the past have assumed nominal dimensions for product components for generating the feasible assembly sequences. These sequences, however, may not be feasible in practice due to propagations and accumulations of tolerances.

Assembly planning is a process, manual, semi-automatic, or fully-automatic, for generating one or more valid assembly sequence. Moreover, a valid assembly sequence must guarantee part interchangeability such that the product can be assembled with the components made within the design specification, which includes dimensions and tolerances. For example, an assembly operation may not succeed if the tolerance accumulation at a mating feature of a subassembly is large such that there is no way to compensate for the accumulated tolerance, but to modify the feature at an assembly line, if it is possible. However, it is not uncommon to encounter an assembly problem after the assembly components have been manufactured. In which case, either the designer of an assembly product has a problem or one or more assembly components are manufactured over the tolerance limits. Modifying an assembly component at an assembly line can increase the cost of the product and can delay production rate, or may invalidate functionality constraint of the product specified by a designer. Therefore, a robust assembly planner should evaluate assemblability of a product in terms of tolerance propagations to generate valid assembly sequences as early as possible.

In product performance prediction system, assembly sequences can be evaluated before production process starts. The real-time integration of manufacturing information and the estimation with specified assembly sequences provides the designer a nice environment in evaluating his work and comparing alternative designs. This makes the system to be a robust assembly planner in sequencing area.

5.4 Part Mating

Any time a part is designed, the accuracy of its manufacture must be specified. Some of its surfaces are important to its function, so the designer states tolerances on them for this purpose. However, grip and jig surfaces deserve to be toleranced as well so that assembly can take place with confidence that the parts will mate properly. There already exists a large body of theory on how far parts can be misaligned from each other and still be assembled. Thus the designer must see that the surfaces on which one part rests and the other is grasped are made accurately enough. Naturally, depending on the assembly sequence, the resting and grasping surfaces will be different.

Two families of tolerancing schemes – parametric and geometric – have been developed for industrial use. Parametric tolerancing is based on ordinary dimensions, and comes in three flavors: worst-case limit tolerancing, statistical tolerancing, and vector tolerancing (Requicha, 1984). Geometric tolerancing was developed to ameliorate some intrinsic weaknesses in parametric tolerancing.

The set of mating relations between parts in an assembly is an implicit representation of the position of each part. The part-positioning problem involves resolving the mating relations to determine the positions of all parts and thus the final configuration of the assembly. Various types of design-analysis application require a capability for computing part positions on the basis of functional requirements. These applications include kinematic analysis, tolerance analysis, assembly-sequence planning, and robot-path planning.

Assembly drawings generally are supplemented with an assembly process plan that specifies and assembly sequence and some feature-mating and -joining conditions. Some modern CAD systems do better, by providing means to declare feature mating in hierarchical graph structures (Allen, 1993). However, Figure 5-1 shows that feature mating by themselves are not enough; mating often must be ordered in the same way that components of datum systems are ordered. The underlying message here is that assembly specification requires more attention than it has received to date.

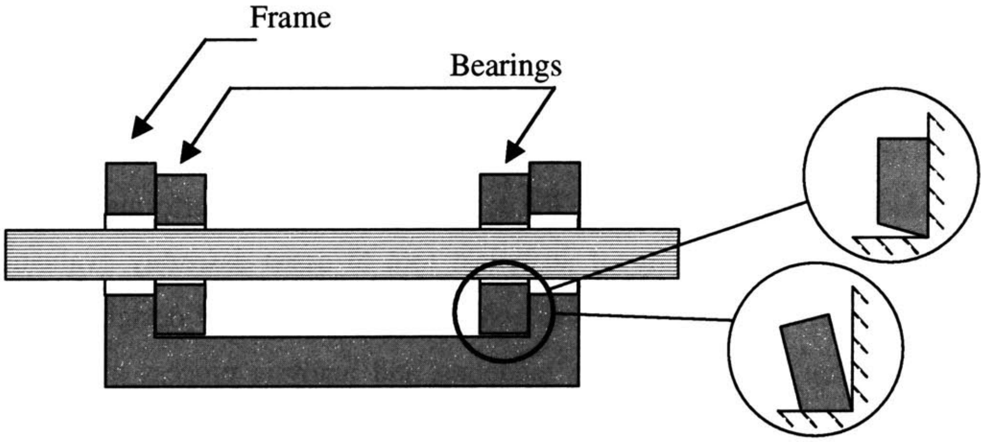


Figure 5-1 Feature mating

To bring a see in feature mating problems in another functionalities that the prediction of product performance can provide. The visual assembly evaluation tool provides an analytical environment to deal with part features and solve mating problems. Geometric tolerance can help. For example, the two situation in Figure 5-1, if geometric tolerance can be estimated from machining data, the design will be able to predict whether the various problem sin mating will be happen after real manufacturing.

5.5 Summary

Variability must be accommodated in design and controlled in production, and mechanisms for doing this are woven through the entire production system. Assembly evaluation is one of the most important activities among them. It is quite frequently done with inadequate data from past experience and theoretical guess, or through building thousands of assemblies in production line and measuring the results. Hence, assembly is an area that accounts for a large part of the manufacturing cost. Consequently, there is a lot of interest in improve the assembly evaluation activities. It is desirable to automate and computerize this evaluation activity. The idea of predicting the performance of as-manufactured assemblies comes out with beneficial in improving the overall efficiency.

To provide a design framework that supports product definition from an assembly viewpoint, the design process has to be supported at the assembly level, rather than at the component level. This suggests an assembly modeling environment that can capture part characteristics and provide the designer appropriate assembly evaluation activities. The product performance prediction idea is feasible based on the discussion in this chapter. It is able to help the designer in evaluating the assemblability of a design, compensating tolerance stackup for an optimal design, conducting sequence planning to compare design alternatives, and detecting and solving potential problems in part mating. This predicting environment 1) will help to relieve the designers to low-level performance-oriented tasks, 2) will incorporate a variety of analytical tool to assist assembly evaluation, 3) will support multiple types of analysis to address a wide variety of design questions, and 4) will ensure that the tasks in the modeling and analysis process are performed in a methodologically correct manner.

CHAPTER 6 IMPLEMENTATION

Predicting the performance of as-manufactured assemblies at the design stage relies on the presence of a network-based modeling environment, the integration of real-time manufacturing information feedback, and the adoption of effective assembly evaluating tools, as discussed in Chapter 3. As tolerance is a critical link between design and manufacturing in product design and development, an implementation of product performance prediction will be carried out across tolerance analysis architecture. First, an overview of a distributed object-oriented modeling environment (DOME) will be given. LabVIEW will then be introduced and integrated to DOME functioning as the connection to manufacturing. The internal relationship between DOME and the LabVIEW plug-in will be explored through the data transfer between them. A brief introduction to NI Card will also be presented.

6.1 Tolerance Analysis Architecture

Among the many progresses that can be made with product performance prediction to improve product quality and reduce production cost, as discussed in Chapter 5, tolerance analysis and control is an excellent example of an implementation model. Tolerance is a critical factor in product design and development. It plays an important role in assemblability; at the same time, tolerance is the critical link between design and manufacturing. A well-defined tolerance analysis model will not only be a good starting point for production, but also it can be used as the basis for illustrating the internal relationship between various components throughout the product performance prediction system.

6.1.1 Tolerance Analysis Model

Because tolerance analysis is so important to product design and development, it is necessary to shed more light on this model. An engineering design must perform properly in spite of dimensional variation. To achieve this, engineering design requirements must be expressed as assembly tolerance limits. The designer must assign limits to the gaps, clearances and overall dimensions of an assembly critical to performance. Assembly tolerance limits are applied to the statistical distribution of the assembly variations predicted by tolerance analysis to estimate the number of assemblies that will be within the specification.

Manufactured parts are seldom used a single parts. They are typically used in assemblies of parts. The dimensional variations occurring in each component part of an assembly accumulate statistically and propagate kinematically, causing the overall assembly dimensions to vary according to the number of contributing sources of variation. The resultant critical clearances and fits that affect performance are subject to variation due to the stackup of the component variations. Tolerances are added to engineering drawings to limit variation. Dimensional tolerances limit component size variations. Geometric tolerances are added to limit the form, location or orientation of individual part features. Assembly tolerance specifications are added to limit the accumulation of variation in assemblies of parts to a level dictated by performance requirements.

In the implementation of product performance prediction, the major concern is assembly tolerances, because tolerance analysis is carried out during part characteristics estimation for as-manufactured assemblies. Designers are the creative thinkers and also the performers of assembly evaluation in our system. They need to control more than just gaps and clearances in assemblies. Orientation and position of features may also be important to performance. To be a comprehensive design tool, a tolerance analysis system must provide a set of assembly tolerance specifications that will cover a wide range of common design requirements.

Tolerance analysis promotes concurrent engineering by bringing engineering requirements and manufacturing capabilities together in a common model, as illustrated in Chapter 2. In the implementation, the engineering modeling and analysis are integrated with a CAD system to create a practical tool for product and process development. This quantitative design tool can predict the effects of manufacturing variation on performance and cost in a computer-based design environment. As shown in Figure 6-1, tolerance analysis brings production capabilities and performance requirements together in an engineering model that can be easily understood. It provides a common meeting ground where design and manufacturing can interact and where the designer can quantitatively evaluate the effects of their decisions. Thus, the implementation across tolerance analysis architecture fulfills the need for concurrent engineering and provides a tool for improving performance and reducing cost.

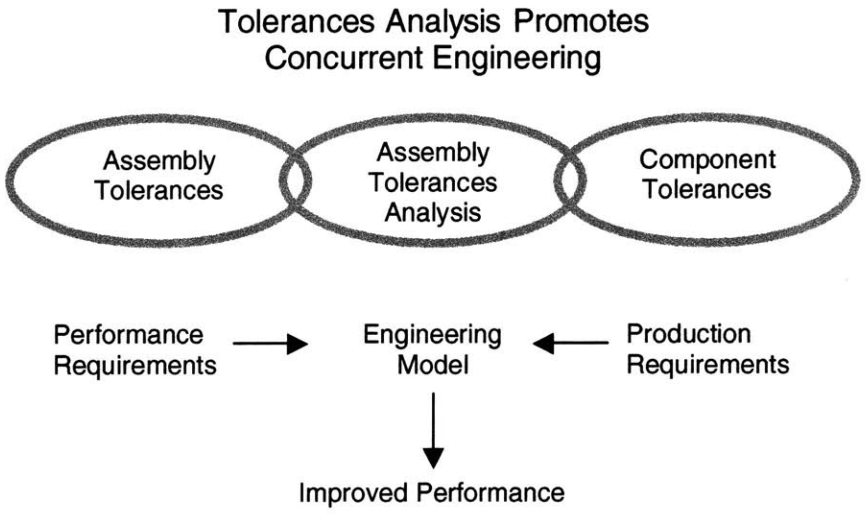


Figure 6-1 Tolerance analysis promotes concurrent engineering

The ability to model a system is a fundamental skill for effective engineering design or manufacturing systems analysis. Unfortunately, few engineers know how to construct variational models of assemblies beyond a 1-D stack. This is primarily because the methods have not been established. There is little treatment of assembly modeling for tolerance analysis in engineering schools or texts. Until engineers learn how to model, tolerance analysis will never become widely used as other CAD tools. A consistent set of modeling procedures, with some guiding rules for creating vector assembly models, allows for a systematic approach that can be applied to virtually any assembly. The steps in creating a model are:

- Capture fundamental requirements for proper representation
- Specify dimensional and geometric requirements to the component design
- Determine assembly strategies
- Generate assembly sequence

- Represent tolerance stack requirements
- Assign tolerance allocation.

To construct a tolerance analysis model in assembly evaluation based on part characteristics estimation, the methodology for information control and tolerance analysis control is critical. After the steps are taken in creating a model as listed above, what information should be brought into the analysis to help evaluation, and what type of process should be appropriate for the information should be our next concern.

6.1.2 Control Methodology

The primary contribution of the implementation is to propose an approach to tolerance analysis, to enable tolerancing to be addressed at successive stages of design in an incremental, continuous ongoing fashion. The proposed approach integrates three design-related domains.

1) Design activities at successive stages of design.
2) Assembly models for tolerancing that evolve continuously during the design process.
3) Methods and best practices for tolerance analysis and synthesis.

Tolerance control methodology is the cornel element of the implementation. The building of methodology directly influence efficiency. A flow chart for the tolerance stack methodology is shown in Figure 6-2. Elements of stackup methodologies and process selection methodology should be used in order to provide a complete solution to the assembly stack problem. Additionally, an understanding of geometric tolerancing, process capability indices and selection of key characteristics is useful.

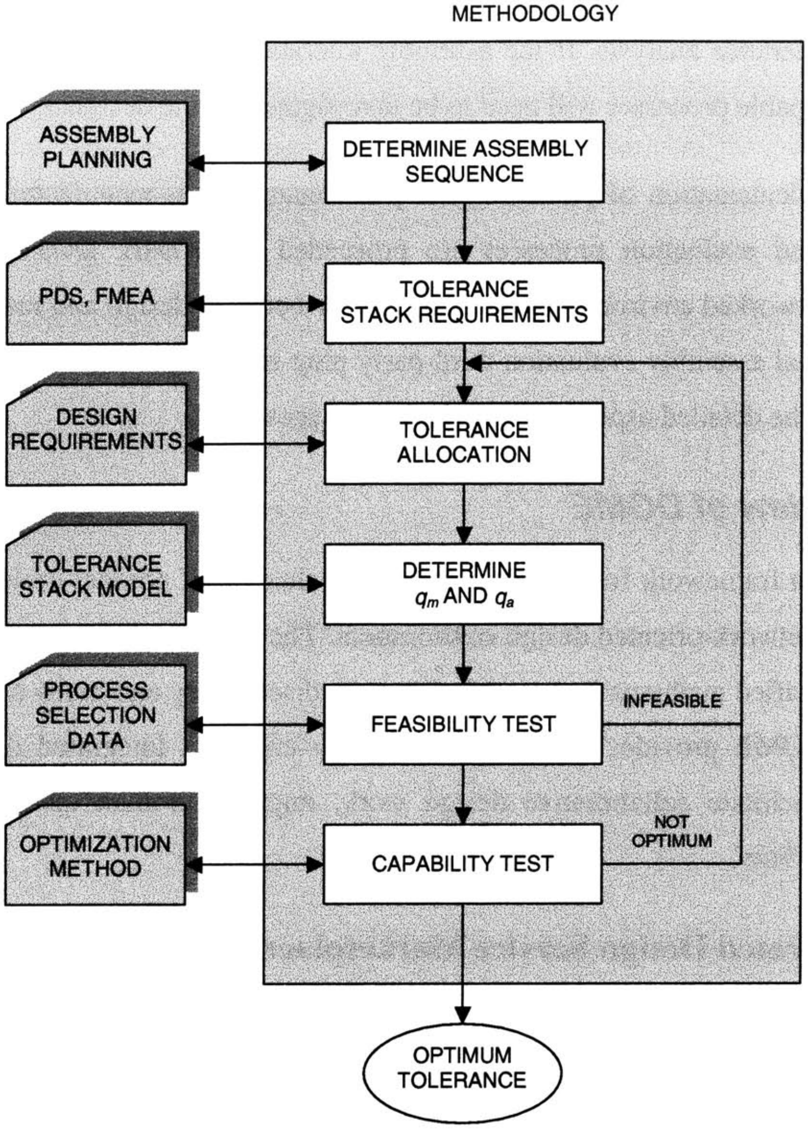


Figure 6-2 Elements of the tolerance analysis methodology.

Initially, it is recommended that an assembly sequence for the tolerance stack design be developed and that any customer specifications be noted, typically the final assembly tolerance and potential failure modes. Once an assembly tolerance has been assigned and the level of capability determined, design tolerances for each component in the stack can be assigned. These are then sent to manufacturing for process planning and machining operations selection and machining data are returned back to design as soon as there is a solution generated for the manufacturing of a component. The use of networked environment and real-time machining data feedback greatly speeds up the time for a feedback. In assembly evaluation, the machining data are categorized and processed to perform a tolerance analysis. If the assembly tolerance cannot be met by optimization, then more capable processes will need to be investigated by the designer.

In the implementation of predicting the performance of as-manufactured assemblies, all design and evaluation processes are proceeded in DOME. DOME provides the necessary networked environment for data transfer between design and manufacturing. It also has virtual assembly evaluation third-party plug-ins as the Solidworks and I-deas. In next section the detailed aspects of DOME will be presented.

6.2 Overview of DOME

DOME is a framework for the modeling and evaluation of product design problems in a computer network-oriented design environment. The framework is intended to integrate designer-specified mathematical models for multi-disciplinary and multi-objective design problems. DOME provides the ability to rapidly construct integrated design problem models to facilitate collaborative design work, improve product quality, and reduce development time.

6.2.1 Integrated Design Service Marketplace Concept

The integrated design service marketplace concept is embodied by three main elements. The first component is providing the ability of *publish* integrative services so that they are widely accessible. (1) Information system architectures that allow distributed users in different environments to participate, without investing significant training time into systems, tools or modeling languages outside of their own expertise. (2) Systems that allow each design participant to use the tools, representations, simulations, heuristics, or models that are most suitable within their domain. (Abrahamson *et al.* 1999) Individuals with particular product development expertise need to be given the capability to create models or model components hose services are readily accessible by others, without requiring them to have additional knowledge outside of their traditional domain specific tools.

The publishing mechanism is similar to web publishing. If one is creating a document is a word processor, one can save it as HTML and then place it on a server for access by others with appropriate permissions. Likewise, model service publishers will enable users to define interfaces to their modeling capabilities. This concept can be demonstrated using the MIT CADlab's DOME software prototype in application to a Ford door moveable glass system.

Model owners use simple DOME publishing programs to define interfaces that will mediate how other users will interact with their models. A publisher is a standalone program or macro specific to a third party application. It allows users to transparently create metadata defining desired service interfaces for their models and the types of DOME objects that will embody these services. Then, model owners use a web browser to log into a DOME model server and use special wrapper objects to make their published services available over the Internet. A wrapper is an object written as software plug-in to DOME for third party applications. It interprets the metadata generated by a corresponding publisher to create a front-end service interface constructed using standardized DOME objects. It also manages the back-end communication between these DOME objects and the third party application.

Once a model is published, other users can interact with the model through its service interface, just as most object-oriented programming languages selectively expose certain attributes and methods while protecting inner workings. DOME objects (also called modules) facilitate the object-oriented structuring of design modeling services. In this way each design participant can use their modeling tool of choice without requiring others to do the same. Any tool can be used to create simulation or model content. Publishing applications then allow interfaces to be created so that model services can be accessed and operated through a web-browser. Design participants use publishers to transparently construct service interfaces for their models using a collection of standardized DOME objects. Participants are free to include any services they feel are appropriate for their model.

The second component is to create a mechanism for subscribing to published services and integrating them to build system models. (3) Flexibility to allow for the spontaneous and robust growth, extension, change, revision and reuse of integrated models, tools, or resources to solve evolving or new problems. (4) Incorporation of a seamless mix of detailed models and incomplete or approximate models to support both top down and bottom up design. (5) Accommodation of both tight and loose collaboration, ranging from close colleagues to customers or supplier, while respecting a diverse set of intellectual property and synchronization needs.

Further, one can imagine that the system integrator could then encapsulate the integrated model and offer these services to yet other participants working on yet larger systems. Users can subscribe to these interfaces and use them in their own environment, similar to the idea of channels on the Internet that allow users to observe certain content and receive feedback when it changes. However, the fundamental difference in DOME is that the mechanism is bi-directional, and an array of actions can be taken in response to changes, from email notification to triggering a sequence of simulations.

The mechanisms for publishing, subscribing, and synthesizing relationships provide the underpinnings of a *service marketplace* for the producers and consumers of product development models and data. The marketplace should facilitate the matching of producers and consumers, and the linking and augmentation of services to create new system models. Each participant in the marketplace brings expertise and formal representation in the form of data and models, ranging from an individual offering finite element analysis to an application engineer offering a catalog of electric motor simulations. System integrators are able to flexibly define and alter relationships between different services derived from different software applications and residing on different DOME servers on the fly, without hard coding software connections.

The third and final component of the concept is the introduction of tools to support the interrogation and management of system models. (6) Provide the ability to explore a design solution space, elicit trade-offs between participants and goals, and monitor design evolution in both a manual and an automated fashion. (7) Methods to extract and analyze both explicit and implicit model or organizational structure, information flow, and resource behavior. Although the mechanisms for connecting and integrating models have been demonstrated, a number of issues remain. For example, how does one understand the structure or behavior of the resulting models? How does one select solutions once a large number of options become available? What are the mechanisms that drive publishing and subscribing in a marketplace?

6.2.2 Software Prototype

A software prototype called DOME has been developed in the MIT CADlab to test the design service marketplace. Early work on the concept is described in work by Pahng (Pahng *et al.* 1998) while the underlying object formalism is described in work by Senin and Wallace (Senin *et al.* 2003). There are objects that provide standardized DOME services corresponding to data types ranging from engineering quantities to files. Relation objects define service relationships between other objects, and manager objects coordinate the firing of relation objects within their scope. An application program interface (API) is provided so that third party software wrapper objects plug into the system. Wrapper objects provide a mapping between the interface of proprietary software and standardized DOME service objects. There are numerous objects to support design exploration: selection of services from catalogs; decision or tradeoff analysis; model structure analysis; and model optimization. Finally, the publishing concept and mechanisms are detailed in work by Borland (Borland and Wallace 2000; Borland *et al.* 1998). The structure of the prototype system used for concept testing is illustrated in Figure 6-3.

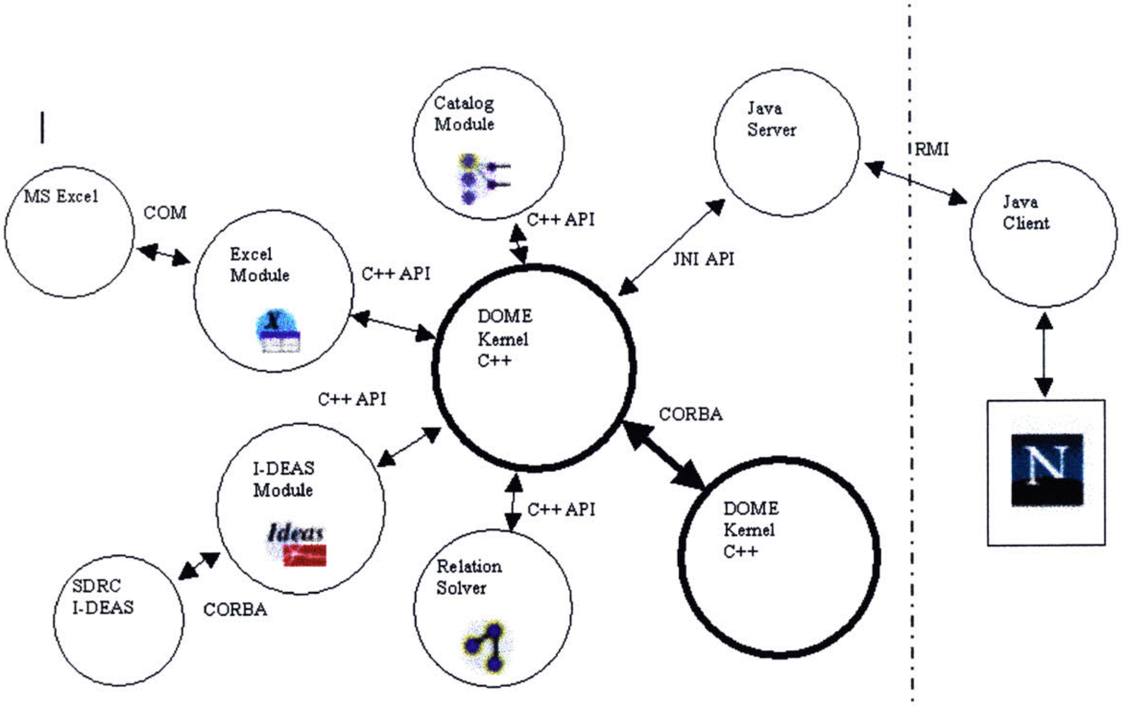


Figure 6-3 Architecture of the DOME prototype system (Wallace *et al.* 2000)

The main elements are: the DOME server kernel that provides the underlying mechanisms for creating and interconnecting DOME models; and a variety of objects that plug into this kernel, providing links to external applications such as Excel and I-deas or particular functions such as catalog selection or model integration capabilities. Finally a browser-based client provides a graphical building and evaluation environment.

6.3 LabVIEW Integration

LabVIEW (an acronym for Laboraory Virtual Instrument Engineering Workbench) is a software tool that lets you build "virtual instruments" (VIs) using its graphical programming language, "G". LabVIEW is developed and sold by National Instruments (http://www.ni.com). LabVIEW has been around over 10 years, and is used by thousands of developers worldwide.

The whole concept of *virtual instrumentation* has been to create more powerful, flexible, and cost-effective instrumentation systems built around a PC using software as the engine and interface. A VI can easily export and share its data and information with other software applications since they often reside on the same computer.

A unique, high-power electron microscope has just been purchased by an internationally funded research agency. Although the electron microscope is located in CA, it is being made available to researchers in Russia. The Russian scientists do not need to travel to this facility, since they can control the settings, run experiments, and retrieve images remotely from specimens thanks to an Internet-enabled system.

6.3.1 LabVIEW Application

The preceding stories illustrate the possibilities of combining virtual instrumentation and the Internet. What kind of applications is possible when you leverage network and Internet technology into your systems? What advantages are there to Internet-enabling your text lab or manufacturing process? The answer usually falls into one or more of four categories (see Figure 6-4):

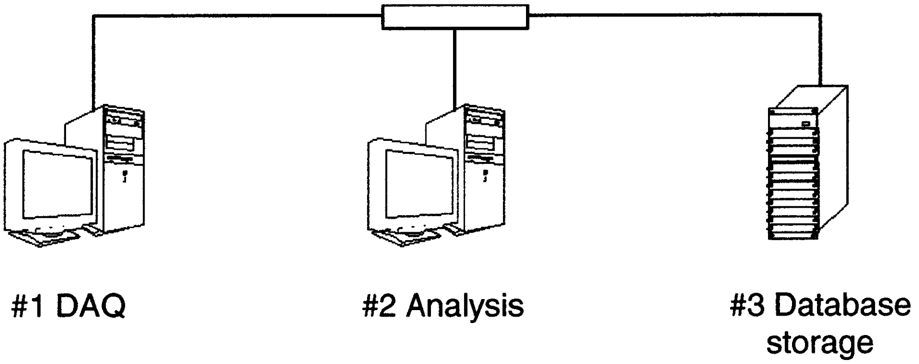


Figure 6-4 Categories for networked instrumentation.

1. Remote monitoring

In a remote monitoring application, a process can be observed from another location on the network. The observation is done with a *client* while the process runs on a server. In a pure remote monitoring scenario, the client cannot give any feedback or provide any inputs to the server process.

2. Remote control

A remote control application usually includes the same capabilities as a remote monitoring system, but also allows the remote user (the client) to send some data, messages, or inputs back to the server process.

Another example of remote control application could be applications through web. The client is a Java applet inside a web browser that communicates over the network with the server. Instead of just passively observing the server process, the remote user can manipulate the controls on the web browser to control the VI.

Remote control applications often require more thought and design considerations, since the capability to affect a process is influenced by many more factors.

3. Collaboration

"Computer systems are intended to support and enhance the activities and tasks that people perform. Unfortunately, most computer systems have been developed with the view that the user performs activities and tasks in isolation. Realistically, people perform many activities and tasks in collaboration with others. The design of computer systems should reflect, support, and enhance the natural ways that people collaborate to accomplish work."

Leverage the capabilities of the Internet along with software-based instrumentation is one of the more powerful uses of remote virtual instrumentation. In a collaboration application, multiple uses from remote sites can use a client program to communicate and share information not only with the server process, but the aware of and share information with each other as part of the communication.

4. Distributed computing

A distributed computing application simply refers to a software process that runs on more than one computer or machine. Distributed computing is a way of sharing computing and hardware resources to accomplish a task that might be burdensome, inconvenient, or impossible to perform on one machine. An example of distributed computing might be a system that collects data, sends the data over the network to another computer to be stored in a database, while a third machine retrieves data from the database and performs some intensive computational analysis on it. By distributing these processes over three machines, each process can run in a more efficient manner.

6.3.2 Client-Server Models for LabVIEW

The client-server model is a common model for networked applications. In the client-server model, one set of processes (called *clients*) requests services from another set of processes (*servers*). A server usually waits for a client to initiate a connection, and in many cases, a server can service multiple clients at a time. Although clients and servers can use many different ways to communicate with each other (and can even reside on the same machine or the same program), we will focus on TCP/IP as the underlying communication path.

In LabVIEW, you can build custom client-server systems, where both the server and the client application are written in LabVIEW. We will first look at the generic template for a client and for a server and then illustrate specific examples in LabVIEW of each.

1. The VI Server

The VI Server gives you the capability to access features programmatically in LabVIEW either using the VI Server functions in a block diagram or through an ActiveX control. VI Server is much more than just some type of networking server built into LabVIEW. The VI Server functionality is really a way of introducing *object-oriented* programming into LabVIEW. With VI Server, you can programmatically:

- Load a VI into memory, run it, and then unload the VI without the need to have it statically linked as a subVI in your block diagram.
- Dynamically run a subVI that gets called at run-time, by only knowing its name and connector pane structure (this is known as calling a VI by reference).
- Change properties of a particular VI, such as the size and position of the front panel window, whether it is editable, etc.
- Make LabVIEW windows move to the front of the screen.

- Call a subVI from the block diagram, without waiting for it to finish executing (one of the few places you can get away with not obeying the normal dataflow paradigm.).

What do these features have to do with network applications? It can provide network transparency, which means you can do all of the mentioned manipulation of a VI or of LabVIEW itself *on another machine across the network* in just the same way as if you were on your own machine. This means that, for example, you could have a data acquisition VI running at remote sites, while your local analysis VI gets information from the remote machines without having to write any special networking or using the TCP/IP functions.

The VI Server exposes its functionality in LabVIEW through block diagram functions. It also allows its functionality to be accessed in Windows from eternal programs through an ActiveX automation client (e.g., a Visual C++ program or macro) and from a remote LabVIEW VI over TCP/IP. Figure 6-5 illustrates this architecture.

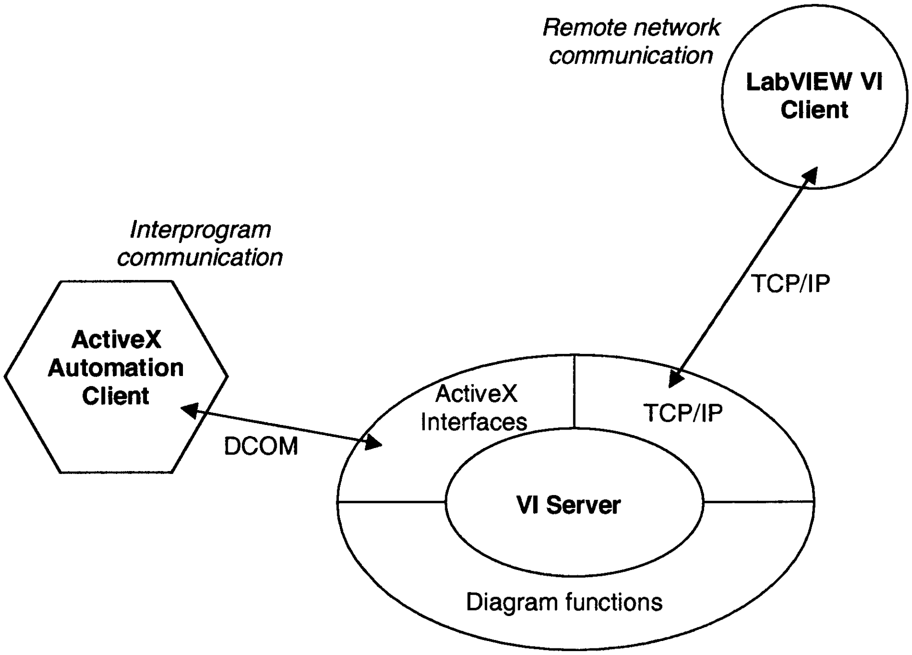


Figure 6-5 VI Server functionality

2. LabVIEW and ActiveX

ActiveX simply represents the component-based technology from Microsoft, relying on COM. The main use of ActiveX is to create ActiveX controls, which can run anywhere on a network with application that supports ActiveX.

LabVIEW for Windows has strong support for ActiveX functionality; LabVIEW can function both as an ActiveX server (meaning other applications can access LabVIEW properties and methods, such as running a VI and getting results) and an ActiveX client (meaning LabVIEW can run and control other applications). LabVIEW's block diagram functions allow you to read and write properties, call methods, and handle events from ActiveX components. LabVIEW can have ActiveX containers on the front panel where you can place the ActiveX control you wish into them; you can then manipulate this ActiveX control through the block diagram.

LabVIEW can expose properties and methods of the LabVIEW application itself and of specific VIs to other ActiveX-enabled applications (e.g., Microsoft Excel, Visual C++, Visual Basic, etc.). It does this through the VI Server interface.

6.3.3 Establish A Connection Between DOME and LabVIEW

The function of a Web Server program is to receive requests from Web browsers and to act on those requests. The "action" normally is to return a specified document to the browser. In any event, the requests are simply communicated across the Internet in strings of text characters, the responses back to the browser are also strings of characters sent back via the Internet. The Web standard for network communications is TCP/IP. LabVIEW comes equipped with programs that handle TCP/IP communications.

A way of displaying a flow chart for a Web Server is shown in Figure 6-6.

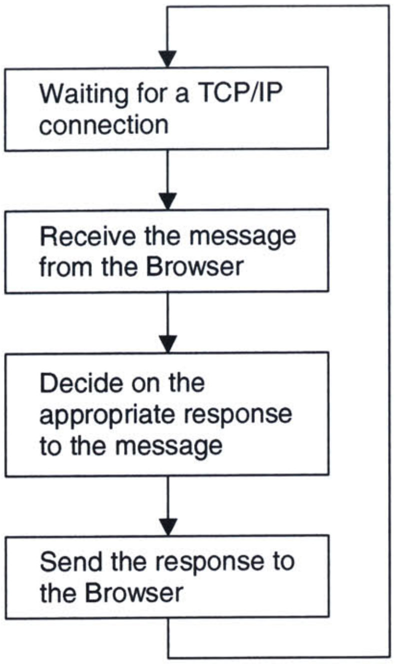
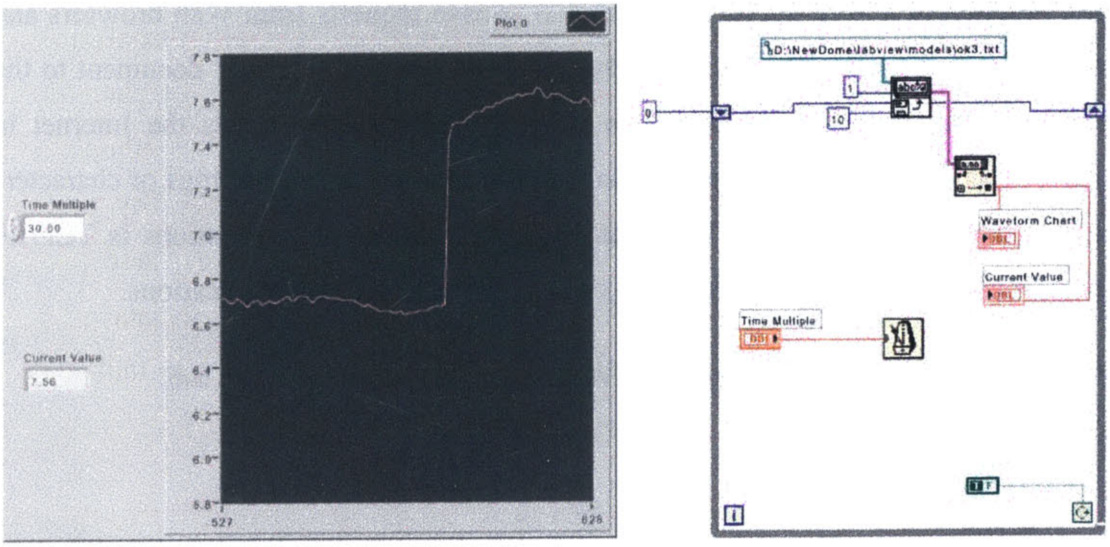


Figure 6-6 Flow chart of a Web Server

In the LabVIEW Web Server, the function in the top box is carried out in the "Web Server" VI; the function in the second box is carries out in a subVI named "Receive Command"; the functions of the lower two boxes are carried out in a subVI named "Generate Reply." The front panel and diagram of Web Server are shown in Figure 6-7.


Time Multiple 30.00

Current Value 7.56

Figure 6-7 Front panel and block diagram of LabVIEW Web Server

1. ActiveX communication

At the most basic level, ActiveX is a description of an object-based programming model developed by Microsoft based on COM (Component Object Model) standard and used for much of the modern Windows architecture.

Built on top of the ActiveX technology are many implementations; consider the two most popular (see Figure 6-8). First, ActiveX controls are reusable software components you can embed in various compatible environments (known as containers). These controls implement custom user-interface objects as well as programming functions such as database access and data analysis.

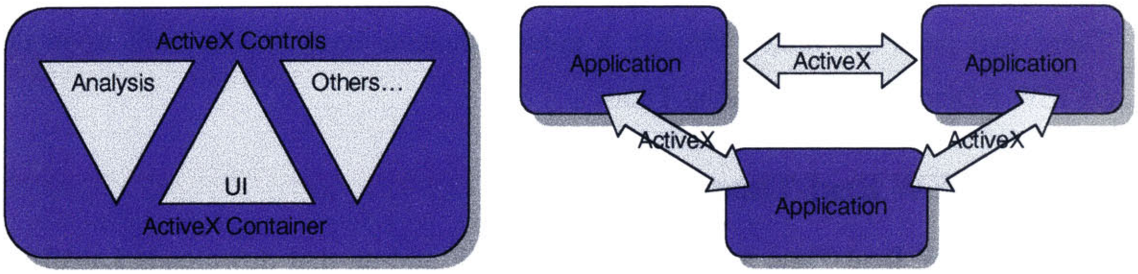


Figure 6-8 ActiveX Implementations

The second is ActiveX automatons, which provides easy communications among independent programs or tools. This technology runs in a client/server configuration where one tool, the automation server, provides services to any automation client in the OS. The client application makes calls to any functions, commonly called methods in the ActiveX realm, which the server exposes. A client can also change any exposed properties to modify the server's current state and future operations. Using these exposed properties and methods, a client can communicate with a server, send data and call functions, resulting in an integrated application made up of two or more independent programs or tools.

2. Communication architecture

Given its fully computer-based implementation, the laboratory environment described can be easily expanded for remote manipulation. The main concept in turning the locally controlled setup into a remotely controlled one consists of moving the user interface away from the experiment. Two distinctive parts result: the remote client and the local server.

- The remote client is a computer equipped with the user-interface functionalities. The client software, which allows the users to observe and to act on the remote experiment, is an executable application compiled for the target platforms using the Application Builder.
- The local server is the computer located near the real process and equipped with the hardware interface to the sensors and actuators. The server software receives the client commands and transmits them to the real process. It also returns the state of the real process to the client.

These two parts are linked through a communication layer that is built upon the LabVIEW 5 Distributed Computing Tools.

6.3.4 LabVIEW Real-Time Module

The LabVIEW Real-Time Module is a LabVIEW add-on that downloads LabVIEW applications onto embedded hardware for deterministic real-time performance. Users can debug these applications using the traditional LabVIEW development environment under Windows and then download the applications to run under a real-time operating system embedded on Real-Time Series processor-based measurement hardware.

6.3.5 Plug-in LabVIEW Real-Time Series DAQ Devices

The LabVIEW Real-Time Series DAQ devices are embedded processor boards with standard National Instruments DAQ devices as daughter cards. The Real-Time Series DAQ device, along with LabVIEW Real-Time and NI-DAQ, provides and easy-to-use system for real-time applications. The processor board contains a microprocessor that runs real-time, embedded LabVIEW Real-Time applications.

6.3.6 Summary

The current version of LabVIEW Distributed Computing Tools permits highly interactive remote manipulation to be performed inside an institution equipped with an intranet. This requires only minor changes when a local solution already exists. Therefore, it is well suited for live demonstrations conducted by somebody or for experimentation carried out by students from a computer room. In this way, practice time is unrestricted. The same approaches for doing something using remote facilities can be applied to home or even overseas connections.

The high-level networking capabilities of LabVIEW enable a user to implement virtual instruments for remote manipulation in a very efficient manner, both from a time and resource point of view. Compared with experimentation in virtual reality, remote manipulation on real processes is easier to implement and more versatile. In face, adding or selecting another physical setup does not involve the elaboration of complex mathematical models and graphical representations.

Finally, remote experimentation is not limited to education. In research and industry, remote accesses also represent an interesting opportunity to meet the growing need of scientists who wish to share unique or expensive equipment, and to enable support engineers to operate immediately at customer facilities.

CHAPTER 7 CONCLUSION

This thesis has discussed a prototype network-based system for predicting the performance of as-manufactured assemblies. The essential concept is to predict the assembly performance at the early design stage of product development. The system is based on a distributed design-modeling framework, embedded with real-time machine tool data feedback for estimating characteristics of the as-manufactured parts. By surveying various machine tool data for different machining operations and researching the possible methods for part characteristics estimation using the available data, this assembly performance prediction prototype model has been shown to be both possible in principle and practically achievable. Database facilities for estimating part characteristics are potential auxiliary tools for product design activities.

The creation of an assembly performance prediction model is central to this framework. Prediction of the performance of as-manufactured assemblies is a way to help designers optimize design specifications. The model can be used to predict assembly performances for different designs and evaluate how well solution alternatives meet stated design objectives or designer preferences. This framework is intended to integrate manufacturing data for part characteristics estimation with assembly performance prediction in a distributed design environment. This will allow detailed design changes to propagate effects to manufacturing level considerations and, correspondingly, allow manufacturing level changes to influence design details.

An implementation of DOME and LabVIEW is proposed for the realization of the assembly performance prediction prototype. DOME provides a distributed network-based environment for collaboration. LabVIEW performs data acquisition and transmission, and enables real-time data acquisition from manufacturing. Through an assembly tolerance analysis model, the internal communication between DOME and LabVIEW has been illustrated.

One of the major contributions of this research is that the network-based design modeling prototype is a significant step on the way to integrating manufacturing information and assembly prediction knowledge into product design. As long as the manufacturing branches are connected to the network system, the designer is able to estimate the part characteristics directly from the machine tool data feedback and, therefore, predict the performance of as-manufactured assemblies prior to manufacturing. Such an assembly performance predicting implementation in product design offers a number of advantages over conventional product design. First, the decisions and alternative solutions for product design to meet stated design objectives can be made prior to manufacturing, based on the assembly performance prediction, and therefore, the product life-cycle and production cost are both reduced to the greatest possible extent. Second, the estimation of part characteristics is more accurate than with the statistical and empirical methods. This is due largely to the third advantage: that the machining parameters for assembly performance prediction are acquired completely in real time, and the newest developments on part characteristics estimation can be incorporated into the system. Finally, design data are shared and communicated among all parties concerned immediately after they enter the system.

The prototype system described in this paper is network-based and distributed. There is sufficient flexibility in the design of the framework in terms of both the methodology and the implementation. This system for integrating acquired machining data can connect with, and efficiently integrate data from, different manufacturing branches with different machines. Extensions can be made to the prototype in order to generalize it for even wider integration of different machining operations in various manufacturing branches. This resolves the otherwise serious limitation that different manufacturers have different standards for and definitions of machining parameters.

Further developments are possible in several directions. First, the critical factors for a successful assembly are worth a deeper investigation. To achieve a more thorough and complete prediction of assembly performance, how these factors affect assembly should be classified and systematized. In the current prototype model, it is not quite clear how these factors influence each other: whether one influences another or they exert their effects in parallel. The inner relationships among these factors should be explicitly incorporated at the framework-level to strengthen assembly prediction.

In addition, the database for estimating part characteristics is an auxiliary resource for assembly performance prediction. The estimation of part characteristics involves machining data and requires a substantial amount of knowledge and skills from the design engineer. The implementation of such a database for machining data analysis, with encapsulated estimation methods provided, facilitates assembly prediction and therefore allows engineers with limited manufacturing knowledge to do design work. The undoubted benefits include reduced analysis time, less analysis error. This implementation, however, is complicated by the fact that the machining parameters for various manufacturing methods differ considerably, different machining operations may have different focuses, and new methods for estimating part characteristics are always being developed.

The manufacturing factors that influence part characteristics are many. The prototype discussed in this thesis predicts the performance of as-manufactured assemblies only on the basis of real-time machining data feedback. It works chiefly in the restricted realm where real-time feedback is focused on machining parameters. What else could be used for even better estimation, and what methods could be applied, merits substantial investigation. When extension or integration is made in directions like this, additional data analysis and estimation models will need to be developed.

Although the assembly performance prediction has not yet been put into practice for testing and verification, the basic functionality and performance have been demonstrated within the prototype system. No serious obstacles have been identified that would limit its real-world industrial applications.

REFERENCES

Abrahamson, S., D. R. Wallace, N. Borland, and N. Senin (1999). "Integrated Engineering, Geometric and Customer Modeling: LCD Projector Design Case Study." *Proceedings of the ASME DT Conferences*, DETC/CIE-8780, Las Vegas, Nevada.

Allen, G. (1993). "Tolerances and Assemblies in CAD/CAM Systems", Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology.

Anon. (1984). "3-Dimensional Touch Trigger Probes for Machining Centres and Lathes." *Commercial 16-Page Brochure*. Renishaw Electrical Ltd. New Mills, Wotton-under-Edge, UK.

Anon. (1990). "World's Smallest Laser Scan Micrometer." *Commercial 6-Page Brochure*. Keyence Corporation. UK.

Azouzi, R. and M. Guillot (1998). "On-line Optimization of the Turning Using an Inverse Process Neurocontroller." Transactions of the ASME: Journal of Manufacturing Science and Engineering **120** (February) 101-107.

Barkan, P. (1988). "Simultaneous Engineering." Design News **44** A30.

Bennett, S. (1984). "Construction of Real-Time Software." Real-Time Computer Control, S. Bennett and D. A. Linkens. London, UK: P. Peregrinus on Behalf of the Institution of Electrical Engineering.

Bergman, B. (1992). "The Development of Reliability Techniques: A Retrospective Survey." *Reliability and System Safety* **36** 3-6.

Binks, S. D. (1971). "The Development of a Laser-Operated Scanning Rod Gauge." *Measurements & Control* **4**(April) T49.

Bolz, R. W. (Ed.) (1981). *Production Processes: The Productivity Handbook*, 5th Edition. New York: Industrial Press Inc.

Booker, J. D., M. Raines, and K. G. Swift (2001). *Designing Capable and Reliable Products*. Oxford; Boston: Butterworth-Heinemann.

Boothroyd, G. (1975). Fundamentals of Metal Machining and Machine Tools. Washington: Scripta Book Co.

Borland, N., H. Kauffman, and D. R. Wallace (1998). "Integrating Environmental Impact Assessment into Product Design: a Collaborative Modeling Approach." *Proceedings of the ASME DT Conference*, DETC/DRM-5730, Atlanta, Georgia.

Borland, N. and D. R. Wallace (2000). "Environmentally-Conscious Product Design: a Collaborative Internet-Based Modeling Approach." *Journal of Industrial Ecology* **3**(2) 33-46.

Bralla, J. G. (Ed.) (1999). Design for Manufacturability Handbook, 2nd Edition. New York: McGraw-Hill.

Chase, K. W. and W. H. Greenwood (1988). "Design Issues in Mechanical Tolerance Analysis." *Manufacturing Review* **1**(1) 50-59.

Chase, K. W. and A. R. Parkinson (1991). "A Survey of Research in the Application of Tolerance Analysis to the Design of Mechanical Assemblies." *Research in Engineering Design* **3** 23-37.

Chase, K. W., S. P. Magleby, and J. Gao (1997). "Tolerance Analysis of Two- and Three-Dimensional Mechanical Assemblies with Small Kinematic Adjustments." Advanced Tolerancing Techniques. H. Zhang. New York: Wiley Interscience.

Chen, X., S. H. Cheraghi, and R. Arupathi (1997). "Dynamic Tolerance Analysis of Multi Operation Manufacturing Process." Proceedings of the 1997 IEEE International Symposium on Assembly and Task Planning, Marina del Ray, California, 54-59.

Craig, M. (1992). "Controlling the Variation." *Manufacturing Breakthrough* **1**(6) 343-348.

Crosby, P. B. (1990). Cutting the Cost of Quality. The Quality College Bookstore.

Deisenroth, M., J. P. Terpenny, and O. K. Eyada (1992). "Design for Competitiveness: Case Studies Developed in Concurrent Engineering and Design for Manufacturability." Design for Manufacture ASME DE-Vol. 51 29-34.

Drozda, T.J. and C. Wick (Eds.) (1983). Tool and Manufacturing Engineers Handbook, Vol. I, Machining, 4th Edition. Dearborn, MI: Society of Manufacturing Engineers.

El-Sonbaty, I. and A. A. Megahed (2000). "On the Prediction of Surface Roughness in Turning Using Artificial Neural Networks." Current Advances in Mechanical Design and Production, VII. M. F. Hassan and S. M. Megahed. Amsterdam; Oxford: Pergamon Press.

Evbuomwan, N. F. O., S. Sivaloganathan, and J. Webb (1996). "A Survey of Design Philosophies, Models, Methods and Systems." Proceedings of the Institution of Mechanical Engineers, Part B **210**(B4) 301-320.

Gerth, R. (1997). "Tolerance Analysis: A Tutorial of Current Practice." *Advanced Tolerancing Techniques*. H. Zhang. New York: Wiley-Interscience.

Huang, X. and P. Gu (1994). "Tolerance Analysis in Setup and Fixture Planning for Precision Machining." Proceedings of the 4th International Conference: Computer Integrated Manufacturing and Automation Technology, Troy, New York.

Ivanov, B. N. and V. A. Mel'nichuk (1964). "Measuring the Diameter of Cylindrical Articles by the Method of Running a Roller Over Their Surfaces." *Measurement Techniques* **9** 772.

Jawahir, I. S., N. Uddin, and C. K. Jun (1990). "Development of Finish Turning Inserts: Present Trends and Future Prospects." Surface Engineering. S. A. Meguid. London: Elsevier Publishers.

Jeang, A. (1995). "Economic Tolerance Design for Quality." *Quality and Reliability Engineering International* **11** 113-121.

Judson, T. W. (1976). Product Design Turning and Milling, SME Technical Paper MR76-902.

Kjellberg, T. and L. Wingard (1989). "Product Design, Planning and Design of Manufacturing Systems Based on Product Model Knowledge Bases." Optimization of Manufacturing Systems Design. D. L. Shunk. Amsterdam; New York: North Holland.

Kline, W. A., R. E. DeVor I, and A. Shareef (1982) "The Prediction of Surface Accuracy in End Milling." Transactions of the ASME: Journal of Engineering for Industry **104** (August) 272-278.

Kronenberg, M. (1966). Machining Science and Application; Theory and Practice for Operation and Development of Machining Processes. Oxford; New York: Pergamon Press.

Leaney, P. G. (1996). "Design for Dimensional Control." *Design for X – Concurrent Engineering Imperatives*. G. Q. Huang. London: Chapman & Hall.

Lee, R. L. (1963). "Turning Railway Wheel Sets." *Machinery* **192** 144.

Lee, S., C. Yi, and R. Suarez (1997). "Assemblability Analysis with Adjusting Cost Evaluation based on Tolerance and Adjustability." *Proceedings of the 1997 IEEE International Symposium on Assembly and Task Planning*, Marina del Rey, California, 103-108.

Lin, S.-S., H.-P. Wang, and C. Zhang (1997). "Optimal Tolerance Design for Integrated Design, Manufacturing and Inspection with Genetic Algorithms." *Advanced Tolerancing Techniques*. H. Zhang. New York: Wiley-Interscience.

Liu, X. W., K. Cheng, D. Webb, and X. C. Luo (2002) "Prediction of Cutting Force Distribution and its Influence on Dimensional Accuracy in Peripheral Milling." International Journal of Machine Tools & Manufacture **42**(7) 791-800.

Montgomery, D. and Y. Altintas (1991). "Mechanism of Cutting Force and Surface Generation in Dynamic Milling." Transactions of the ASME: Journal of Engineering of Industry **113** (May) 160-168.

Nokikov, N. I. and B. K. Makarevich (1961). "Automatic Measuring of Dimensions in Turning." *Measurement Techniques* **6** 431.

Novak, A. (1981). "Sensing of workpiece Diameter, Vibration, and Out-Off-Roundness by Laser-Way to Automate Quality Control." *CIRP Annals* **30**(1) 473-476.

Pahng, F., N. Senin, and D. R. Wallace (1998). "Distributed Object-Based Modeling and Evaluation of Design Problems." *Computer-Aided Design* **30**(6) 411-423.

Phadke, M. D. (1989). *Quality Engineering Using Robust Design*. Englewood Cliffs, NJ: Prentice-Hall.

Prasad, B. (1996). "Integrated Product and Process Organization." Concurrent Engineering Fundamentals, Vol. I. Upper Saddle River, NJ: Prentice Hall PTR.

Requicha, A. A. G. (1984). "Representation of Tolerances in Solid Modeling: Issues and Alternative Approaches." Solid Modeling by Computers: from Theory to Applications.

Russell, R.S. and B.W. Taylor (1995). Production and Operations Management: Focusing on Quality and Competitiveness. Englewood Cliffs, NJ: Prentice Hall.

Sanchez, J. M. (1993). "Quality in design." Concurrent Engineering: Automation, Tools, and Techniques. A. Kusiak. New York: Wiley-Interscience.

Sanderson, A. C. (1999). "Assemblability based on maximum likelihood configuration of tolerances." IEEE Transactions on Robotics and Automation **15**(3) 568-572.m

Senin, N., D. R. Wallace, and N. Borland (2003). "Distributed Object-based Modeling in Design Simulation Marketplace." *ASME Journal of Mechanical Design*. **125**(1) 2-13.

Shiraishi, M. and S. Sato (1990). "Dimensional and Surface Roughness Controls in a Turning Operation." *Transactions of the ASME: Journal of Engineering for Industry* **112** (February) 78-83.

Smith, S. and J. Tlusty (1991). "An Overview of Modeling and Simulation of the Milling Process." Transactions of the ASME: Journal of Engineering for Industry **113** (August) 169-175.

Starbek, M. and J. Grum (2002). "Concurrent Engineering in Small Companies." International Journal of Machine Tools & Manufacturing **42**(3) 417-426.

Suresh, P. V. S., P. Venkateswara Rao, and S. G. Deshmukh (2002). "A Genetic Algorithmic Approach for Optimization of Surface Roughness Prediction Model." International Journal of Machine Tools & Manufacture **42**(6) 675-680.

Sutherland, J. W. and R.E. Devor (1986). "An Improved Method for Cutting Force and Surface Error Prediction in Flexible End Milling Systems." Transactions of the ASME: Journal of Engineering for Industry **108** (November) 269-279.

Swift, K. G., M. Raines, and J. D. Booker (1997). "Design Capability and the Costs of Failure." Proceedings of the Institution of Mechanical Engineers, Part B **211**(B6) 409-434.

Takesa, K., H. Sato, and Y. Tani (1984). "Measurement of Diameter Using Charge Coupled Device (CCD)." *CIRP Annals* **33**(1) 377-382.

Tillen, R. (1964). "Controlling Workpiece Accuracy." *Product Engineering* **43** 499.

Vasseur, H., T. R. Kurfess, and J. Cagan (1992). "A Decision-Analytic Method for Competitive Design for Quality." *Proceedings of ASME Design Automation Conference, Advances in Design Automation* **1** DE-Vol. 44-1 ASME 329-336.

Vikhman, V. S. (1963). "Application of the Television Computing Technique in Automatic Control of Dimensions." *Measurement Techniques* **11** 904.

Wallace, D. R., S. Abrahamson, N. Senin, and P. Sferro (2000). "Integrated Design in a Service Marketplace. " *Computer-Aided Design* **32**(2) 97-107.

Winner, R. I., J. P. Pennell, H. E. Bertrend, and M. M. G. Slusarczuk (1988). "The Role of Concurrent Engineering in Weapons System Acquisition." IDA Report R-338. Alexandria, VA: Institute for Defense Analyses.

Wolff, V., J. F. Rigal, P. Ghodous, and M. Martinez (2001). "Design and Machining Data Integration." Proceeding of the 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation, Antibes-Juan les Pins, France, 483-491.

Wu, Z., W. H. Elmaraghy, and H. A. Elmaraghy (1988). "Evaluation of Cost-Tolerance Algorithms for Design Tolerance Analysis and Synthesis." *Manufacturing Review* **1**(3) October 168-179.

Yucesan, G. P. and Y. Altintas (1994). "Improved Modeling of Cutting Force Coefficients in Peripheral Milling." International Journal of Machine Tools & Manufacture **34**(4) 473-487.

APPENDIX: LabVIEW IMPLEMENTATION PROGRAMS

LabviewIncludes.h

```
//interface for the LabviewIncludes class.
//
//////////////////////////////////////////////////////////////////////

#ifndef DOME_LABVIEWINCLUDES_H
#define DOME_LABVIEWINCLUDES_H

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

//#include <stdlib.h>
//#include <string.h>
//#include <iostream.h>
//#include "windows.h"
//#include "stdio.h"
//#include "conio.h"

#import "D:\\Program Files\\National Instruments\\LabVIEW
6.1\resource\\LabVIEW.tlb" no_namespace \
        raw_dispinterfaces \
        raw_native_types \
        named_guids


#include "DomePlugin.h"
using namespace DOME::DomePlugin;

#include "TypeConversions.h" // str
using namespace DOME::Utilities::TypeConversions;

#define LABVIEW_ERROR1(msg) throw DomeException("Labview Error",msg)
#define LABVIEW_ERROR2(msg) throw DomeException("Labview
Error",__FILE__,__LINE__,msg)
#define LABVIEW_ERROR LABVIEW_ERROR2

#define LABVIEW_DEBUG_SWITCH true
#define LABVIEW_DEBUG(s) if (LABVIEW_DEBUG_SWITCH)
std::cerr<<"Labview Debug: "<<s<<std::endl;

#endif // DOME_LABVIEWINCLUDES_H
```

LabviewModel.h

```
//interface for the LabviewModel class.
//
//////////////////////////////////////////////////////////////////////

#ifndef DOME_LABVIEWMODEL_H
#define DOME_LABVIEWMODEL_H

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "windows.h"
//#include "stdio.h"
//#include "conio.h"
//#include <stdlib.h>
//#include <string.h>
//#include <iostream.h>

#include "LabviewData.h"

namespace DOME{
namespace LabviewPlugin{

class LabviewModel
{
public:
	friend class LabviewData;

	LabviewModel(string fileName);
	virtual ~LabviewModel();

	LabviewReal* createReal(string controlName);// throw(DomeException);

	bool isModelLoaded();
	void loadModel() throw(DomeException);
	void unloadModel();

	void execute();
//	void run();
//	void abort();
	void setOutputControl(string outControlName);
	void offOutputControl();
```

```
private:
        string _filename;
        _Application* _lvApp;
        VirtualInstrument* _pVI;
        vector <LabviewData*> _data;

        void _createConnections() throw(DomeException);
        void _destroyConnections() throw(DomeException);

        //vector<string> _outputControlVector;
//      string _outputControl;
//      bool _outputC;
};

} // namespace LabviewPlugin
} // DOME

#endif // DOME_LABVIEWMODEL_H
```

LabviewData.h

```
//interface for the SolidworksData class.
//
//////////////////////////////////////////////////////////////////////

#ifndef DOME_LABVIEWDATA_H
#define DOME_LABVIEWDATA_H

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "LabviewIncludes.h"

namespace DOME {
namespace LabviewPlugin {

class LabviewData
{
public:
	LabviewData();
	virtual ~LabviewData();

protected:
	friend class LabviewModel;
	virtual void connect(VirtualInstrument* pVI) throw(DomeException);
	virtual void disconnect();

	VirtualInstrument* _pVI;
};

class LabviewReal : public LabviewData, DomeReal
{
public:
	LabviewReal(string controlName);
	virtual ~LabviewReal();

	double getValue() throw(DomeException);
	void setValue(double value) throw(DomeException);

private:
	BSTR _bcontrolName;
	string _controlName;

	void connect(VirtualInstrument* pVI);
```

```
        void disconnect();
};


} // namespace LabviewPlugin
} // DOME

#endif // DOME_LABVIEWDATA_H
```

LabvieModel.cpp

```
//implementation of the LabviewModel class.
//
////////////////////////////////////////////////////////////////

#include "LabviewModel.h"
#include "conio.h"

namespace DOME {
namespace LabviewPlugin {

LabviewModel::LabviewModel(string filename)
{
	_filename = filename;
	_lvApp = NULL;
	_pVI = NULL;
//	_outputC = false;
}

LabviewModel::~LabviewModel()
{
	unloadModel();
}

bool LabviewModel::isModelLoaded()
{
	if (_lvApp != NULL && _pVI != NULL) return true;
	return false;
}

void LabviewModel::loadModel()
{
	try{
		if (isModelLoaded()) return;

		CoInitialize(NULL);
		CLSID clsid;
		HRESULT hr = CLSIDFromProgID(L"LabVIEW.Application", &clsid);
		if(FAILED(hr)){
			unloadModel();
			LABVIEW_ERROR("LabviewModel::loadModel:
CLSIDFromProgID failed.");
		}

		IUnknown* pUnk = NULL;
```

```
        hr = CoCreateInstance(clsid, NULL, CLSCTX_LOCAL_SERVER,
IID_IUnknown, (void **)&pUnk);
        if(FAILED(hr)){
                unloadModel();
                LABVIEW_ERROR("LabviewModel::loadModel: Labview not
registered properly.");
        }


        hr = pUnk->QueryInterface(DIID__Application, (void **)&_lvApp);
        pUnk->Release();
        if(FAILED(hr)){
                unloadModel();
                LABVIEW_ERROR("LabviewModel::loadModel: Unable to
QueryInterface.");
        }

        //convert _filename to BSTR
        wchar_t wfilename[512];
        mbstowcs(wfilename, _filename.c_str(), 512);
        BSTR bfilename = ::SysAllocString(wfilename);
        wchar_t wpassword[20];
        mbstowcs(wpassword, "", 20);
        BSTR bpassword = ::SysAllocString(wpassword);
        try {
                _pVI = _lvApp->GetVIReference(bfilename, bpassword, false);
        } catch (...) {
                unloadModel();
                ::SysFreeString(bfilename);
                throw DomeException(__FILE__, __LINE__, "Unable to open
labview model.");
        }
        if (FAILED(_pVI)) {
                unloadModel();
                ::SysFreeString(bfilename);
                LABVIEW_ERROR("LabviewModel::loadModel: Unable to open
LabVIEW model.");
        }
        ::SysFreeString(bfilename);
        ::SysFreeString(bpassword);

        //_lvApp->BringToFront();
        _pVI->FPWinOpen = true;
        //pVI->ShowFPOnCall = TRUE;
        _createConnections();
```

```
		LABVIEW_DEBUG("Labview model "<<_filename.c_str()<<"
successfully loaded.");
	} catch (...) {
		throw;
	}
}

void LabviewModel::unloadModel() {
	_destroyConnections();

	if (_xlBook != NULL)
	{
		try {
			#ifdef XL_2000
				_xlBook->PutSaved(NULL, VARIANT_TRUE);
			#else
				_xlBook->PutSaved(VARIANT_TRUE);
			#endif
			hr = _xlBook->Close();
		} catch (...) {
			EXCEL_ERROR("ExcelModel::unloadModel: PutSaved() or
Close() failed.");
		}
		if(FAILED(hr)) {
			EXCEL_ERROR("ExcelModel::unloadModel: PutSaved() or
Close() failed.");
		}
	}

	//_lvApp->AutomaticClose = 0;
	_pVI->FPWinOpen = false;


	if (_pVI != NULL) _pVI->Release();


	if (_lvApp != NULL)
	{
		HRESULT hr = _lvApp->Quit();
		if(FAILED(hr)) {
			LABVIEW_ERROR("LabviewModel::unloadModel: Unable to
quit Labview.");
		}
	}

	if (_lvApp != NULL) _lvApp->Release();
```

```
        CoUninitialize();

        _lvApp = NULL;
        _pVI = NULL;

        LABVIEW_DEBUG("Labview model "<<_filename.c_str()<<" successfully
unloaded.");
}

void LabviewModel::execute() throw(DomeException)
{
        LABVIEW_DEBUG("Entering LabviewModel::execute: press any key to stop");
        _pVI->Run(true);
        if(_outputC) {
                double result;
                LabviewReal* real = new LabviewReal(_outputControl);
                while(!_kbhit()) {
                        result = real->getValue();
                        cout<<result<<endl;
                }
                _getch();
                delete real;
        } else {
                while(!_kbhit()) {}
                _getch();
        //}

        _pVI->Abort();

        LABVIEW_DEBUG("Leaving LabviewModul::execute");
}

void LabviewModel::execute(string valueName = NULL, int freq = 1)
throw(DomeException)
{
        LABVIEW_DEBUG("Entering LabviewModel::execute: press any key to stop");
        _pVI->Run(true);
        int count = 0;
        while(!_kbhit()) {
                count++;
                if(count%freq==0)
                        cout<<
        }
        _getch();
        _pVI->About();
```

```
	LABVIEW_DEBUG("Leaving LabviewModul::execute");
}

void LabviewModel::run()
{
	LABVIEW_DEBUG("Entering LabviewModel::run: press any key to run vi");
	while(!_kbhit()) {};
	_getch();

	_pVI->Run(true);
}

void LabviewModel::abort()
{
	LABVIEW_DEBUG("Entering LabviewModel::abort: press any key to stop vi");
	while(!_kbhit()) {};
	_getch();

	_pVI->Abort();
	LABVIEW_DEBUG("Labview model "<<_filename.c_str()<<" successfully
stopped");
}

void LabviewModel::setOutputControl(string outControlName)
{
	_outputControl = outControlName;
	_outputC = true;
	//_outputControlVector.push_back(boutControlName);
	LABVIEW_DEBUG("OutputControl is ON: Controlname
'"+_outputControl+"'!");
}

void LabviewModel::offOutputControl()
{
	_outputC = false;
	LABVIEW_DEBUG("OutputControl is OFF!");
	//_outputControlVector.erase(_outputControlVector.begin(),
_outputControlVector.end());
}

LabviewReal* LabviewModel::createReal(string controlName) throw(DomeException)
{
	LabviewReal* real = new LabviewReal(controlName);
	_data.push_back(real);
	return real;
```

```
}

void LabviewModel::_createConnections() throw(DomeException)
{
	try
	{
		for (int i=0; i<_data.size(); i++)
		{
			_data[i]->connect(_pVI);
		}
	} catch(...) {
		throw;
	}
}

void LabviewModel::_destroyConnections() throw(DomeException)
{
	try
	{
		for (int i=0; i<_data.size(); i++)
		{
			_data[i]->disconnect();
		}
	} catch(...) {
		throw;
	}
}

} // namespace LabviewPlugin
} // DOME
```

LabviewData.cpp

```
//implementation of the ExcelData class.
//
//////////////////////////////////////////////////////////////////////

#include "LabviewData.h"

namespace DOME {
namespace LabviewPlugin {

LabviewData::LabviewData()
{
	_pVI = NULL;
}

LabviewData::~LabviewData()
{
  disconnect();
}

void LabviewData::connect(VirtualInstrument* pVI) throw(DomeException)
{
	if (pVI == NULL) LABVIEW_ERROR("LabviewData::connect: pVI is null.");

	_pVI = pVI;
}

void LabviewData::disconnect()
{
	_pVI = NULL;
}

LabviewReal::LabviewReal(string controlName)
{
	_controlName = controlName;
	_bcontrolName = NULL;
}

LabviewReal::~LabviewReal()
{
	disconnect();
}

void LabviewReal::connect(VirtualInstrument* pVI) throw(DomeException)
{
```

```
        LabviewData::connect(pVI);
        //convert _filename to BSTR
        wchar_t wcontrolname[512];
        mbstowcs(wcontrolname, _controlName.c_str(), 512);
        _bcontrolName = ::SysAllocString(wcontrolname);
}

void LabviewReal::disconnect()
{
        ::SysFreeString(_bcontrolName);
        LabviewData::disconnect();
}

double LabviewReal::getValue() throw(DomeException)
{
        VARIANT result;
  try {
           result = _pVI->GetControlValue(_bcontrolName);
  } catch (...) {
     VariantClear(&result);
     LABVIEW_ERROR("LabviewReal::getValue");
  }
        VARIANT doubleResult;
  try {
          VariantInit(&doubleResult);
          VariantChangeType(&doubleResult, &result, NULL, VT_R8);
          double value = doubleResult.dblVal;
          VariantClear(&result);
          VariantClear(&doubleResult);
          return value;
  } catch (...) {
     VariantClear(&result);
          VariantClear(&doubleResult);
     LABVIEW_ERROR("ExcelReal::getValue: not real value!");
  }
}

void LabviewReal::setValue(double value) throw(DomeException)
{
        if (_pVI == NULL)
     LABVIEW_ERROR("LabviewReal::setValue: pointer is null.");
        VARIANT vInput;
        vInput.vt = VT_R8;
        vInput.dblVal = value;
        try {
                _pVI->SetControlValue(_bcontrolName, vInput);
```

```
	VariantClear(&vInput);
	   } catch (...) {
	VariantClear(&vInput);
			LABVIEW_ERROR("LabviewReal::setValue: cannot set to
"+str(value)+".");
	   }
}

} // namespace LabviewPlugin
} // namespace DOME
```

test.cpp

```
#include "LabviewModel.h"
#include "LabviewData.h"
#include <string>
#include "conio.h"

int main(int argc, char* argv[]) {
        string filename ="D:\\NewDome\\labview\\models\\tt_try2.vi";
        using namespace DOME::LabviewPlugin;

        try {
                LABVIEW_DEBUG("Creating LabviewModel.");
                LabviewModel* lv = new LabviewModel(filename);

                cout<<"Press any key to continue"<<endl;
                while(!_kbhit()) { };
                _getch();

                LABVIEW_DEBUG("Creating LabviewReal.");
                LabviewReal* real1 = lv->createReal("Time Multiple");

                LABVIEW_DEBUG("Loading model.");
                lv->loadModel();
                LABVIEW_DEBUG("Getting value.");
                double tt = real1->getValue();
                LABVIEW_DEBUG("Value is: "<<tt);
                LABVIEW_DEBUG("Setting value.");
                real1->setValue(500.0);
                tt = real1->getValue();
                LABVIEW_DEBUG("Value is set to "<<tt);

                lv->setOutputControl("Current Value");
                cout<<"Press any key to run"<<endl;
                while(!_kbhit()) { };
                _getch();




                LABVIEW_DEBUG("Executing...");
                lv->execute();

                LABVIEW_DEBUG("Unloading Model.");
                lv->unloadModel();
                LABVIEW_DEBUG("Finished.");
```

```
        } catch (DomeException e) {
                LABVIEW_ERROR(e.what());
        } catch (...) {
                LABVIEW_ERROR("Unknown exception caught in test.");
        }
        return 0;
}
```