A Flexible Graphics User Interface Architecture for CNC Mechanical Part Quality Measuring Application based on WPF and MVVM

Ms. Shraddha S. Nakate,
Dr. Bandu. B. Meshram,
Ms. Shubha S. Jain,
Department of Computer Science
Veermata Jijabai Technological Institute,
Mumbai, India

***Abstract* – The CNC machine is a numerical control system in which the data handling, control sequences, and response to input is determined by an on-board computer system at the machine tool. In this machine end-to-end component design is highly automated using computer-aided design (CAD) and computer-aided manufacturing (CAM) programs. In other cases, a number of different machines are used with an external controller and human or robotic operators that move the component from machine to machine. In either case, the complex series of steps needed to produce any part is highly automated and produces a part that closely matches the original CAD design. While the specific intention and application for CNC machines vary from one machine type to another, all forms of CNC have common benefits. The first benefit offered by all forms of CNC machine tools is improved automation. The operator intervention related to producing work pieces can be reduced or eliminated. Many CNC machines can run unattended during their entire machining cycle, freeing the operator to do other tasks. This gives the CNC user several side benefits including reduced operator fatigue, fewer mistakes caused by human error, and consistent and predictable machining time for each work piece. Since the machine will be running under program control, the skill level required of the CNC operator (related to basic machining practice) is also reduced as compared to a machinist producing work pieces with conventional machine tools. Here we are presenting a control for gear testing CNC machine where operator of CNC gear testing machine can provide information about which teeth to be inspected during testing with operator friendly GUI using WPF and MVVM.**

Keyword contents; real time system, CNC, Embedded system, Gear testing, user friendly GUI, WPF, MVVM

I. INTRODUCTION

Conventionally, an operator decides and adjusts various machines parameters like feed , depth of cut etc depending on type of job , and controls the slide movements by hand. In a CNC Machine functions and slide movements are controlled by motors using computer programs. By integrating a computer processor, computer numerical control, or “CNC” as it is now known, allows part machining programs to be edited and stored in the computer memory as well as permitting diagnostics and quality control functions during the actual machining. All CNC machining begins with a part program, which is a sequential instructions or coded commands that direct the specific machine functions. The part program may be manually generated or, more commonly, generated by computer aided part programming systems. All computer controlled machines are able to accurately and repeatedly control motion in various directions. Each of these directions of motion is called an axis. Depending on the machine type there are commonly two to five axes. Additionally, a CNC axis may be either a linear axis in which movement is in a straight line or a rotary axis with motion following a circular path. The most basic function of any CNC machine is automatic, precise, and consistent motion control. The most basic function of any CNC machine is automatic, precise, and consistent motion control. Rather than applying completely mechanical devices to cause motion as is required on most conventional machine tools, CNC machines allow motion control in a revolutionary manner. All forms of CNC equipment have two or more directions of motion, called axes. These axes can be precisely and automatically positioned along their lengths of travel. The two most common axis types are linear (driven along a straight path) and rotary (driven along a circular path).

There are different types of CNC machines as mentioned below:

1) CNC Plasma cutter
2) CNC Foam cutter

3) CNC Tube Bender
4) CNC Laser
5) CNC Lathe
6) CNC Woodworking
7) CNC Water Jet
8) CNC Router
9) CNC Mini Machine
10) CNC Mill

The CNC testing machine measures the automobile products like gears and hobs.

This machine consists of three main Parts

1. Gear Testing Module
2. Computer
3. Controller

They are described here

1. Motion Control – Heart of CNC

The most basic function of any CNC machine is automatic, precise, and consistent motion control. Rather than applying completely mechanical devices to cause motion as is required on most conventional machine tools, CNC machines allow motion control in a revolutionary manner. All forms of CNC equipment have two or more directions of motion, called axes. These axes can be precisely and automatically positioned along their lengths of travel. The two most common axis types are linear (driven along a straight path) and rotary (driven along a circular path). Instead of causing motion by turning cranks and handwheels as is required on conventional machine tools, CNC machines allow motions to be commanded through programmed commands. Generally speaking, the motion type (rapid, linear, and circular), the axes to move, the amount of motion and the motion rate (feedrate) are programmable with almost all CNC machine tools.

2. Telling the machine how to do – the CNC program

Almost all current CNC controls use a word address format for programming. (The only exceptions to this are certain conversational controls.) By word address format, we mean that the CNC program is made up of sentence-like commands. Each command is made up of CNC words. Each CNC word has a letter address and a numerical value. The letter address (X, Y, Z, etc.) tells the control the kind of word and the numerical value tells the control the value of the word. Used like words and sentences in the English language, words in a CNC command tell the CNC machine what it is we wish to do at the present time.

One very good analogy to what happens in a CNC program is found in any set of step by step instructions. Say for example, you have some visitors coming in from out of town to visit your company. You need to write down instructions to get from the local airport to your company. To do so, you must first be able to visualize the path from the airport to your company. You will then, in sequential order, write down one instruction at a time. The person following your instructions will perform the first step and then go on to the next until he or she reaches your facility.

In similar manner, a manual CNC programmer must be able to visualize the machining operations that are to be performed during the execution of the program. Then, in step by step order, the programmer will give a set of commands that makes the machine behave accordingly.

Though slightly off the subject at hand, we wish to make a strong point about visualization. Just as the person developing travel directions MUST be able to visualize the path taken, so MUST the CNC programmer be able to visualize the movements the CNC machine will be making BEFORE a program can be successfully developed. Without this visualization ability, the programmer will not be able to develop the movements in the program correctly. This is one reason why machinists make the best CNC users. An experienced machinist should be able to easily visualize any machining operation taking place.

Just as each concise travel instruction will be made up of one sentence, so will each instruction given within a CNC program be made up of one command. Just as the travel instruction sentence is made up of words (in English), so is the CNC command made up of CNC words (in CNC language).

We have developed a control for CNC gear testing machine computer using WPF and MVVM design pattern through which we are communicating with controller using CNC programming to give the functionality to our gear testing CNC machine to set specified teeth for measurement and so we can test accuracy of manufactured gear according standards.

II. WINDOWS PRESENTATION FOUNDATION(WPF)

WPF is an abbreviation for *Windows Presentation Foundation*. Physically, it's a set of .NET assemblies and supporting tools. It's intended to provide a unified API for creating rich, sophisticated user interfaces on Windows XP and Windows Vista, Windows 7.

WPF combines the good things from web development, such as style sheets and a markup language for declarative UI, with good things from Rich Internet Applications, such as scalable vector graphics, animation, and media support. These good things are wrapped up with the good things from traditional Windows development—things like strong integration with the OS and data binding. In WPF, these concepts are strengthened and unified, even all that does not capture the full extent of WPF. It has other facets, such as support for 3D drawing, advanced typography, and portable documents similar to PDF.

WPF is also a *unified* API. Many of the things you are able to do in WPF, you could do before. However, doing them all in one application was extremely difficult. Not only does WPF enable you to bring these disparate features together, but it provides you with a consistent API for doing so. WPF is just one part of a larger picture. Three additional libraries were also released as part of .NET 3.0. All four of these libraries have the same intent of providing a consistent, unified API for their domain.

The Sibling Libraries of WPF:

- WCF: Windows Communication Foundation is focused on messaging. This API greatly simplifies all sorts of networking and communication tasks. It covers everything from web services to remoting to P2P and more.
- WF: A powerful library for building workflow enabled applications. It utilizes a markup language for declaring workflows in an application, and thus prevents workflow from becoming hard-coded. It also makes it very easy for developers to create custom workflow tasks.
- CardSpace: The least famous of the four libraries, CardSpace provides a common identification system that can be used by desktop applications, web sites, and more.

The immediate predecessor to WPF is Windows Forms, the graphical API available to developers in .NET 2.0 and earlier. Windows Forms provides a managed wrapper for accessing the graphical functions of the traditional Windows API.

III. MVVM: MODEL-VIEW-VIEW MODEL: DESIGN PATTERN FOR WPF

There are popular design patterns that can help to tame this unwieldy beast, but properly separating and addressing the multitude of concerns can be difficult. Sometimes we use complicated design patterns, which require writing a lot of code because the UI platform in use does not lend itself well to a simpler pattern. What's needed is a platform that makes it easy to build UIs using simple, time-tested, developer-approved design patterns. Fortunately, Windows Presentation Foundation (WPF) provides exactly that. As the software world continues to adopt WPF at an increasing rate, the WPF community has been developing its own ecosystem of patterns and practices. In this article, I'll review some of those best practices for designing and implementing client applications with WPF.

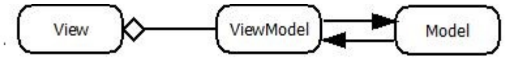
Why to use MVVM with WPF:

Once a developer becomes comfortable with WPF and MVVM, it can be difficult to differentiate the two. MVVM is the lingua franca of WPF developers because it is well suited to the WPF platform, and WPF was designed to make it easy to build applications using the MVVM pattern (amongst others). Many aspects of WPF, such as the look-less control model and data templates, utilize the strong separation of display from state and behavior promoted by MVVM.

The single most important aspect of WPF that makes MVVM a great pattern to use is the data binding infrastructure. By binding properties of a view to a ViewModel, you get loose coupling between the two and entirely remove the need for writing code in a ViewModel that directly updates a view. The data binding system also supports input validation, which provides a standardized way of transmitting validation errors to a view.

Two other features of WPF that make this pattern so usable are data templates and the resource system. Data templates apply Views to ViewModel objects shown in the user interface. You can declare templates in XAML and let the resource system automatically locate and apply those templates for you at run time.

In addition to the WPF features that make MVVM a natural way to structure an application, the pattern is also popular because ViewModel classes are easy to unit test. When an application's interaction logic lives in a set of ViewModel classes, you can easily write code that tests it. In a sense, Views and unit tests are just two different types of ViewModel consumers. Having a suite of tests for an application's ViewModels provides free and fast regression testing, which helps reduce the cost of maintaining an application over time. A simple figure that tells you exactly what **MVVM** is:



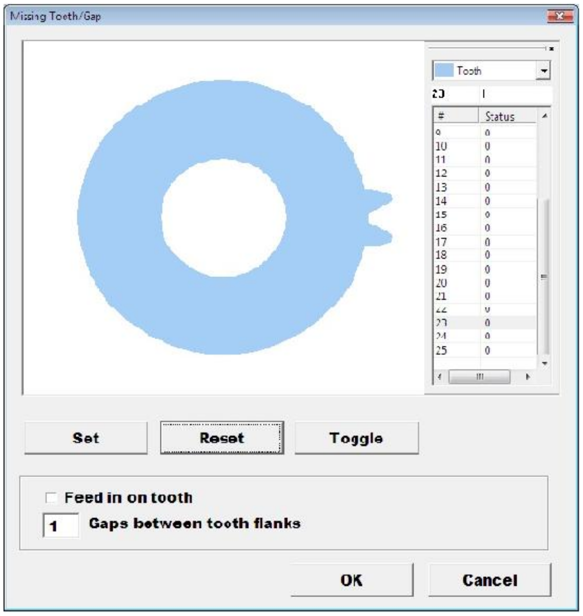
- ***View***: The View holds a reference to the ViewModel. The View basically displays stuff by **Binding** to entities in the View Model.
- ***ViewModel***: The ViewModel exposes Commands, Notifiable Properties, and Observable Collections to the View. The View Binds to these ViewModel entities/members
- ***Model***: The Model is your data and/or application objects that move data while applying Application Logic. If you have a Business Layer, then you might not need this.

The ViewModel is the most significant in the entire pattern as it is the glue that sits between the View and the Model and binds both of them together.

IV. CONTROL FOR GEAR TESTING

There will be graphical as well as manual representation of Gear tooth information for measurement.

Figure: UI of Software Control for Gear testing Machine



23 I

#	Status
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0

☐ Feed in on tooth

1 Gaps between tooth flanks

VI. CONCLUSIONS

Hence we have developed software control using WPF and MVVM design pattern to select teeth in testing gear for CNC machine.

REFERENCES:

[1] C. L. Liu And James W. Layland "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment" Project MAC, Massachusetts Institute of Technology Jet Propulsion Laboratory, California Institute of Technology IEEE Vol 20 No.1 1973

[2] GC Buttazzo, G Lipari, M Caccamo "Elastic Scheduling for Flexible Workload Management" - IEEE 2002 Vol 54 computer.org

[3] C.Steiger, H.Walder and M.Platzner, "Operating Systems for Reconfigurable Embedded Platforms:Online Scheduling of Real-Time Tasks" IEEE Nov. 2004 Vol 53 Issue: 11

[4] Krzysztof M. Sacha "Measuring the Real-Time Operating System Performance" 1068-3070/95 1995 IEEE

[5] AbouGhazaleh, N. Mosse, D. Childers, B. Melhem, R. Craven, M. "Collaborative Operating System and compiler power management for real time applications" Dept. of Comput. Sci., Pittsburgh Univ., PA, USA IEEE 2003 Vol 56 10.1109/RTTAS.2003.1203045

[6] C. Centioli, F. Iannone, G. Mazza, M. Panella, L. Pangione, V. Vitale, and L. Zaccarian "Open Source Real-Time Operating Systems for Plasma Control at FTU" IEEE VOL. 51, NO. 3, JUNE 2004

[7] Scordino, C.; Lipari, G "A Resource Reservation Algorithm for Power Aware Scheduling of Periodic and Aperiodic Real-Time Tasks" Volume: 55 , Issue: 12 Digital Object Identifier: 10.1109/TC.2006.190 Publication Year: 2006

[8] The Performance and Energy Consumption of Embedded real time operating systems

[9] Ketan Kotecha and Apurva Shah "Adaptive Scheduling Algorithm for Real-Time Operating System" 978-1-4244-1823-7/08/ 2008 IEEE

[10] Fabricio Rusu-Banu and Yingxu Wang "Formal Description Of Time Management In Real-Time Operating Systems" 1-4244-0038-4 2006 IEEE CCECE/CCGEI, Ottawa, May 2006

[11] Caccamo, M.; Buttazzo, G.C.; Thomas, D.C. "Efficient Reclaiming in Reservation-Based Real-Time Systemswith Variable Execution Times" IEEE Volume: 54 , Issue: 2 Digital Object Identifier: 10.1109/TC.2005.25 Publication Year: 2005 , Page(s): 198 - 213