Fixturing System in an Information Model Based Concurrent Engineering Environment

Nilo T. Bugtai
Manufacturing Engineering and Management Department
De La Salle University-Manila
E-mail: bugtain@dlsu.edu.ph

A distinguishing aspect of concurrent engineering is the intensive information interchange between areas that are involved with the product life cycles. Shared information structures to integrate different software applications have become necessary to support effectively the interchange of information. While much work done in product model and manufacturing model, there is a need to make them able to support fixture design and planning related activities. An important part of modern design and manufacture activities is to ensure that effective manufacturing decisions are made in the early stage of the product development process. A possible effective approach in this area is the use of information models, which can provide a useful information support to a range of applications in a concurrent engineering environment. The key problem discussed in this research is the structures of information models to support the interactions between fixture planning, process planning and product design for fixturing applications. A software support environment is described which consist of three significant areas; product information, manufacturing information, and a set of fixturing data driven applications. The issues related to the data structures required for product model and manufacturing model are discussed and object-oriented solutions are proposed.

Keywords: concurrent engineering, information models, product model, manufacturing model, fixture planning, design and manufacture

1. INTRODUCTION

To make sure effective decisions are made at each stage in the design and manufacture activities, it is considered important that software tools should provide sufficient information to support the decision making of design and manufacturing engineers. Fixturing support systems therefore, need to provide information that will support all the activities in design and manufacture, which may require fixturing inputs.
The research investigations into fixture support systems can be classified broadly into four categories based on the techniques used; artificial intelligence, CAD tools, group technology, and optimizations for fixture configuration. Markus et al, [1] produced one of the first fixture design systems to make use of the artificial intelligence while CAD/CAM techniques have been used in a knowledge-based fixturing system developed by Miller and Hannam [2]. The first fixturing system developed using Group Technology was researched by Jiang et al, [3], while Trappey and Liu [4] developed a workholding verification system modeled as a quadratic optimization problem. Each of these pieces of research is useful to their own area but there is a need for the provision of a general source of information to support decision-making. This paper shows how information models can be structured to provide fixturing information to support decision making across product design, process planning and fixture planning.

Research works into fixture design systems can be also classified from the point of view of their degree of automation; namely interactive, semi-automated and fully automated. Interactive fixture design is a process where a computer is used to aid the designer by displaying the appropriate fixture elements based on his knowledge. The designer in arriving at the final fixture configuration decides the correct position of the fixture elements. A system can be said to be semi-automated if it does not require full knowledge or expertise from a designer while arriving at appropriate locating, clamping and supporting faces, points and elements. Ingrand and Latombe [5] developed a semi-automated system incorporating the expertise of a designer into the fixturing system. The determination of appropriate faces for locating, clamping and supporting can be decided automatically Markus et al, [1]. Nevertheless, the selection of appropriate points and elements for building a fixture still depends on the user’s expertise and knowledge. An automated system is one, which obtains information directly from a CAD model and makes use of the knowledge-based to decide on the appropriate fixturing points and fixturing elements. Design parameters such as orientation, stability, and deflection due to cutting forces, set-ups, tolerance relationships, assembly and interference must be considered while designing a fixture. Expert systems together with a good knowledge representation scheme were generally used in arriving at a fixture design [6].

While these various pieces of work have provided some progress towards improved fixture design systems, they have tackled only one particular issue, which is automated fixture configuration. The author agrees with Hargrove and Kusiak [7] that there are some areas worthy of investigation. In particular:

- The integration of computer-aided fixture design system with other Computer Aided Engineering (CAE) tools will be the one of the most important factors making for effectiveness and acceptance in manufacturing.
- More multi-functional automated systems can be expected in the future, which should be initiated by information models that describe the integration of shared databases.

This work presented in this paper aims to provide better support for design and manufacturing engineers through the definition of information models in a concurrent
The product model captures the information related to a product throughout its life cycle. The manufacturing model describes and captures the information about the manufacturing facility and capabilities at different levels of abstraction [10]. These information models are implemented as object-oriented databases. The engineering moderator is a specialist manager or coordinating program whose role is to drive concurrency within the MOSES system. It can provide excellent support for individual team members or groups working from particular design perspectives. Strategist applications are specialist expert applications, which assist users of the CAE system to evaluate, modify and extend the product design criteria, which are closely allied to particular design perspectives. An integration environment is required to enable these elements to work together even though they may be distributed over many computing platforms, probably located at several sites. It must also provide support for interactions and communication between applications [11].
This approach is being applied in the case of fixturing. Fixturing requires information about the machine tool, the machining process, cutting tool, fixturing process and fixtures. This information forms a part of the manufacturing model that can be access by the fixturing support applications. The fixturing system supports three activities namely; product design, process planning and fixture selection. Figure 2 shows a view of the fixturing support applications structured in the MOSES concept.

The research issues being pursued in this work are:

- What information is needed in the product model and manufacturing model and how should be structured?
- How should fixture planning, process planning and product design support applications interact with these information models?

This paper focuses on the first issue, providing a view of the information structures required and how they can be used to provide information support to design and manufacture activities.

3. OBJECT-ORIENTED APPROACH

3.1 UML Methodology

The author has used the Unified Modeling Language (UML) of Rational Rose [12] to help explore this research work. In order to have a well-implemented software system, it is necessary to have a good understanding to the problem that leads to the design of the information models that emphasized on the correct and effective structuring of a software system as well as the definition of the relationships and interactions between the systems. The designed models help to reason about the structure of the system and provide a requirement to implement.
The object-oriented approach to software development is based on modeling objects from real world (e.g. machine tool, clamps, locating devices) and offers several advantages to conventional approaches. Such advantages are better to understand the requirements, better handling of the complex systems, smaller systems through the re-use of common mechanisms as well as leading to fewer complexes, easy to enhance and maintain software systems.

3.2 MOSES Manufacturing Model

The manufacturing model describes and captures the information about the manufacturing situation of a company in terms of its manufacturing facility and capabilities at different levels of abstraction [13]. Three entities can be regarded to be basic elements in the definition of any manufacturing environment: resources, processes and strategies. The relations and interaction among them defines the manufacturing environment of a company. Manufacturing resources are all the physical elements within a facility that enable product manufacture e.g. production machinery, production tools, etc. A description of the resources based on their physical properties and functional composition allows the capture of their capabilities.

Being able to represent resource capability enables the support of design decisions e.g. designs for manufacture and manufacturing functions e.g. process planning. Manufacturing processes are those processes carried out in a facility in order to produce a product. Strategies are decisions made on the use and the organization of resources and processes. The manufacturing model as shown in Figure 3 has been structured into four levels based on a de-facto standard (i.e. factory, shop, cell, and station) and modeled in UML class diagram, implemented in an object-oriented database. These levels of abstraction provide manufacturing information for all hierarchical and functional activities within a manufacturing enterprise.

![Figure 3 – Object Oriented Representation of MOSES Manufacturing Model Using UML Class Diagram](image-url)
In defining a representation of the fixturing process as a manufacturing process, we must recognize that it is a different type of process from machining. It’s not a mass reducing process like machining but it is related to machining, as machining requires fixturing process to be used before it can itself be performed. The fixturing process involves a number of sub-processes. These are being locating a workpiece and clamping a workpiece. Figure 4 illustrates the class diagram for Manufacturing Processes showing in particular the structures for mass reducing processes and mass conserving processes as well as illustrating the relationships between machining and fixturing. It also clearly shows the relationship between both machining and fixturing processes and the resources required achieving them.

**Figure 4 - Manufacturing Processes Using UML Class Diagram**

These resources can also be captured with the manufacturing resource class structure as shown in Figure 5. This illustrates the different manufacturing resources such as, production machinery and production tools, which can be represented, and their relations are defined. By defining these interrelationships, it is possible to define how machine tools use cutting tools and workholding tools in order to perform particular machining operations.
4. INFORMATION STRUCTURES TO PROVIDE FIXTURING SUPPORT

4.1. Product Model Structures

Product modeling is accepted as an important part of data exchange (STEP) and data sharing in integrated environments. A Product Model can be considered to be a computer representation of a product, which should hold a complete depiction of the information concerning a product. The product model therefore becomes a source and repository for information for all applications and allows information to be shared between the users and software components of the CAE systems. When we consider the fixturing process in terms of product model data are two questions to be answered. What product information should be stored in the product model? What should be the structure of this information?

At this stage, the author defined the product information requirements for fixturing by extending the general data structures defined by Borja [14] to include fixturing. The general data structure of product information is shown in Figure 6. The figure shows a product specification and product definition which are to be stored in the product model. The specification describes the requirement, which the product must achieve. The product definition describes the ways in which the specification may be achieved and includes the product geometry, dimension and material and also captures the manufacturing plans, machine tool and set-up, defining operation and fixturing information.
4.2 Manufacturing Model Structures

The fixturing process cannot be considered independently of the machining process. Thus, any representation of information related to fixturing must also be related to machining. In a similar way fixturing resources also relate to machining resources in terms of the machine tool and cutting tool being used. An examination of the requirements for fixturing a product and the identification of the key aspects of related manufacturing information led to the general structure illustrated in Figure 7. This shows a general view of manufacturing information for fixturing and how they it should be structured in a manufacturing model.
5. POPULATING INFORMATION MODELS TO PROVIDE FIXTURING INFORMATION

This section describes how the data models can be populated with information and provides a simple example of how that information can be used with respect to a particular workpiece to identify possible work-holding methods. Figure 8 illustrates the production resources part of the model being populated with example workholding tools. The class diagram for WorkholdingTool, which shows a GeneralPurposeFixture is_a subclass of WorkholdingTool, leads to devices such as a chuck and a vice. Also, ModularFixture is_a subclass of the class WorkholdingTool, which leads to the representation of a variety of standard off-the-shelf tooling such as tooling plates, locating, clamping, and supporting elements.
Figure 8 - Populating Production Resources with Particular Workholding Tools

Figure 9 illustrates the representation of the fixturing process; in particular the locating process and the clamping process. The class diagrams show a LocatingProcess and ClampingProcess is_a subclass of FixturingProcess. Locating is a process of positioning the workpiece relative to the fixture and guiding the fixture relative to the cutting tool. Locating the workpiece to the fixture is done with locators and guiding devices, such as feeler gages does guiding the fixture to the cutting tool. Clamping is a process of holding the position of the workpiece in the fixture. The primary devices used for holding a workpiece are clamps, which have the functions to hold the workpiece against its locator and prevent movement of the workpiece during machining.
6. SUMMARY

In this paper a new approach has been presented which has proposed a fixturing system within information supported design and manufacture. The rationale behind this approach is to provide better support for design and manufacturing engineers through the use of information models in a concurrent engineering environment. The following contributions are made in this work:

- The fixturing information structures needed within product and manufacturing models have been identified.
- An object-oriented method has been used to illustrate how the fixturing information models can be applied to a particular workpiece to provide information support to fixturing process in conjunction with the appropriate available fixturing resources.

The authors will pursue further work to explore how fixture planning, process planning and product design support applications should interact with the information models to provide fixturing information to support decision making across these processes. The experimental fixturing system will be implemented using the object-oriented programming language C++ and SQL language (Structure Query Language) of the UniSQL/X object-oriented database system [15].

Acknowledgement

The research undertaken by the author was funded by the Department of Science & Technology and De La Salle University - Manila, Republic of the Philippines.
References


About the author:

**Dr. Nilo T. Bugtai, PhD** is an Associate Professor and currently the Chair of the Manufacturing Engineering and Management Department of De La Salle University-Manila. He holds a degree of Doctor of Philosophy in Manufacturing Engineering at Loughborough University in England, U.K. and received his Master of Science in Manufacturing Engineering at De La Salle University-Manila.

Dr. Bugtai was a recipient of the Doctorate and Master of Science scholarships from the Department of Science and Technology. He was visiting fellow at the University of Pisa, Italy and a Tan Chin Tuan research fellow at Nanyang Technological University in Singapore. Dr. Bugtai Exchange Faculty at the University of Southern Queensland, Australia working in the Mechatronics Laboratory. He was a recipient of the British Council scholarship grant on Low Cost Automation Course at Loughborough University, England U.K. and a US-ASEAN scholar on Computer Integrated Manufacturing Course at the Asian Institute of Technology in Bangkok, Thailand.

His areas of research expertise include: Computer-Aided Engineering, Computer-Aided Design & Manufacture, Computer Integrated Manufacturing Systems, Integrated Information Systems, Concurrent Engineering, Decision Support Systems, Product Development & Realization and Biomedical Engineering. Dr. Bugtai has completed several researches and published technical papers in the international/national journals and proceedings. He is a senior member of Society of Manufacturing Engineers, Dearborn Michigan, USA.