Chapter 43
Software Development Tools to Automate CAD/CAM Systems

N. A. Fountas
School of Pedagogical and Technological Education (ASPETE), Greece

A. A. Krimpenis
School of Pedagogical and Technological Education (ASPETE), Greece

N. M. Vaxevanidis
School of Pedagogical and Technological Education (ASPETE), Greece

ABSTRACT

In today's modern manufacturing, software automation is crucial element for leveraging novel methodologies and integrate various engineering software environments such Computer aided design (CAD), Computer aided process planning (CAPP), or Computer aided manufacturing (CAM) with programming modules with a common and a comprehensive interface; thus creating solutions to cope with repetitive tasks or allow argument passing for data exchange. This chapter discusses several approaches concerning engineering software automation and customization by employing programming methods. The main focus is given to design, process planning and manufacturing since these phases are of paramount importance when it comes to product lifecycle management. For this reason, case studies concerning software automation and problem definition for the aforementioned platforms are presented mentioning the benefits of programming when guided by successful computational thinking and problem mapping.

1. INTRODUCTION

Current manufacturing challenges impose the need for innovative solutions to facilitate production process by minimizing time while simultaneously maintaining high quality. Modern industrial manufacturing systems involve CAD, CAPP, CAM and several other engineering software modules which provide flexibility during a product's lifecycle management. Due to the fact that software vendors intend to satisfy all aspects of industrial domains, they provide large suits of software options which may confuse end users and result to longer processing time.

DOI: 10.4018/978-1-5225-3923-0.ch043

Software automation allows the creation of programming entities capable of overcoming existing limitations occur when applying generic solutions. By building complete automated modules using programming, software systems may be rapidly customized to either fit specific industrial tasks or collaborate with other applications so as to take advantage of the data exchange capability.

To capture such needs, engineering software vendors provide access to their software through the establishment of opened Application Program Interfaces (APIs). That is, users and software developers may create their own macros and add-in programs by taking advantage of available automation objects and routines through available programming languages such as C++ and Visual Basic and JavaScript.

Advancements of software automation solutions are employed mainly to product design, finite element analysis and process planning phases. Notable contributions include the development of programming functions, macro commands and scripts which perform automated design of machine elements such as cutting tool profiles (Ćuković, et al., 2010), work holding / fixturing devices (Farhan, et al., 2012) and product geometries along with their special features (Lamarche & Rivest, 2007 ; Wayzode & Wankhade, 2013; Wayzode & Tupkar, 2013). Wikström (2011) proposed a methodology for combining three dimensional geometry models of various fuel tank systems assembled in combat aircrafts. The methodology calculates fuel surface location and center of gravity and is based on the orientation of the fuel acceleration vector and the amount of fuel. Basic steps involve the automated volume analysis. For each orientation the fuel body is discretized to a predetermined number of volumes and numerical values for each volume, fuel surface location and center of gravity location are automatically recorder. On their efforts to predict surface errors when machining thin walled parts, R. Izamshah et al. (2011) needed to automate simulations for modeling solids, material removal processes and structural analysis operation, through the use of macros and Visual Basic. Reddy and Brioso (2011) developed a methodology where tedious and repetitive processes are automated while designing an industrial robot lower arm. Further on, their model is structurally evaluated in ANSYS®. Finite element analysis process is automated by the use of the programming languages Python and JavaScript. Furthermore a user interface is created using Microsoft Excel with Visual Basic.

Aiming at rapidly producing setup sheets for the CNC machining of industrial products research efforts have also been conducted to automate the process planning stage. Harik et al., (2008) developed an entire platform to facilitate process planning for aeronautical parts using the main application development framework namely *CAA-RADE* of *Dassault Systèmes*® CATIA®. *CAA-RADE* is a C++ based development environment native for CATIA® and provides a low-level access to most of CATIA's features allowing thus; the creation of embedded commands, toolbars, dialogs etc. To overcome limitations found employing this development platform, Fountas (2008) moved towards the automated process planning by creating support functions using Visual Basic as the main programming application. Jeba Singh and Jebaraj (2005) presented an automatic environment within a commercial CAD/CAM software to generate optimal process plans for determined objectives with respect to factory environment for modeled components. In their work, data exchange is achieved between a feature-based design system and spreadsheet software applications, through Visual Basic programming. Ispas et al., (2007) applied C++ programming application to automate specific utilities for the formulation of a solution for improving the machining precision by the machine tool's error minimization in terms of geometrical errors, kinematic errors, static/dynamic load errors, thermal, etc. Deb et al., (2011), developed a methodology of machining operations selection using Artificial Neural Networks (ANNs). Process planning is conducted by a modular expert system fully integrated with an automated data extraction system which obtains data from CAD to feed process planning modules in a fully automated environment. Krimpenis (2008) con-

Related Content

The Human Role in Model Synthesis

Steven Gibson (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications (pp. 81-102).*

www.irma-international.org/chapter/the-human-role-in-model-synthesis/192873

Optimization of the Angle of Twist of Propeller Using Modified Flower Pollination Algorithm

Kok Meng Ong, Pauline Ongand Chee Kiong Sia (2018). *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering (pp. 299-327).*

www.irma-international.org/chapter/optimization-of-the-angle-of-twist-of-propeller-using-modified-flower-pollination-algorithm/206755

On Software Architecture Processes and Their Use in Practice

Perla Velasco-Elizondoand Humberto Cervantes (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications (pp. 1610-1631).*

www.irma-international.org/chapter/on-software-architecture-processes-and-their-use-in-practice/192938

Exceptions in Ontologies: A Theoretical Model for Deducing Properties from Topological Axioms

Christophe Jouis, Julien Bourdaillet, Bassel Habiband Jean-Gabriel Ganascia (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications (pp. 61-81).*

www.irma-international.org/chapter/exceptions-ontologies-theoretical-model-deducing/62435

Object-Oriented Cognitive Complexity Measures: An Analysis

Sanjay Misraand Adewole Adewumi (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications (pp. 917-940).*

www.irma-international.org/chapter/object-oriented-cognitive-complexity-measures/192907