Chapter XII
Function Block–Enabled Integration

ABSTRACT

Function blocks are an IEC (International Electro-technical Commission) standard for distributed industrial processes and control systems (IEC 61499, 2005). It is based on an explicit event driven model and provides for data flow and finite state automata-based control. Based on previous research, function blocks can be used as the enabler to encapsulate process plans, integrate with a third-party dynamic scheduling system, monitor process plan during execution, and control machining jobs under normal and abnormal conditions. They are also considered to be suitable for machine-level monitoring, shop-floor execution control, and CNC control. Combination of STEP-NC and Function Blocks can be seen as a “natural marriage”. This is because the former provides an informationally complete data model but with no functionality, whereas the latter can embed intelligence and provide functionality in the data model for a more capable CNC regime. This chapter introduces the function block architecture which has been implemented in two types of integrations. The first brings together CAD, CAPP, and CAM. The key is to embed machining information in a function block system that is based on the concept of machining features. The second integration connects CAM with CNC. This is in fact an open CNC architecture that is function block driven, instead of G-code driven.

FUNCTION BLOCK STRUCTURE

Being an atomic distributable and executable control function unit, a function block instance can encapsulate a part of machining process data (e.g. slot roughing, pocket finishing, and
hole drilling, etc.) for a given machining feature. It comprises of an individual, named copy of the data structure specified by its function block type, which persists from one invocation of the function block to the next. Figure 12.1 illustrates both the basic (left) and composite (right) function blocks.

A function block, especially the basic function block, can have multiple outputs with internal state information hidden. This means a function block can generate different outputs even if the same inputs are applied. This characteristic is of vital importance for automatic cutting condition generation/modification, after an NC program has been downloaded to a CNC controller, by changing the internal state of a function block (a machining process). For example, an NC program for the same pocket roughing can be used by two different milling machines with different cutting conditions, simply by adjusting the internal state variable of the function block instances. Similar to object-oriented definitions, a function block type can be considered as a class, and a particular function block is the instance of that class. For example, a pocket milling function block can be used for either roughing or finishing depending on the message received. However, different from the object-oriented approach, the behaviour of a function block is controlled internally by a state machine whose operation can be represented by an execution control chart (ECC). Each basic function block is an atomic unit for execution. Composite function blocks may require a multi-threaded concurrent execution model due to the complex event flow. The event flow determines the scheduling and execution of machining operations specified by the algorithms (methods) in basic function blocks. It can also provide signals to maintain or change the internal state variables. In terms of machining process encapsulation, basic function blocks encapsulate both the data and functions of machining process, whereas composite function blocks only encapsulate basic function blocks.

Figure 12.1. Function block structure ©2006, IEEE used with permission
Green Computing as an Ecological Aid in Industry
www.igi-global.com/chapter/green-computing-ecological-aid-industry/69373?camid=4v1a

Addressing Privacy in Traditional and Cloud-Based Systems
www.igi-global.com/article/addressing-privacy-in-traditional-and-cloud-based-systems/105484?camid=4v1a