CubeSpawn Software Architecture

Introduction
Distributed Communication Architecture
CubeSpawn Coordinator Architecture
CubeSpawn Controller Architecture

Introduction

The **Coordinator** and **Controllers** in a CubeSpawn configuration collectively form the CubeSpawn **Distributed Control System** that executes planned production orders released from an ERP system's production module. At the top level, the Coordinator module executes the order, instructing CubeSpawn pallet mover modules to route BOM inputs through **Cubes**, the machine tool work centers in a CubeSpawn configuration. The Cube's Controller module, one instance for each machine tool and pallet mover, retrieves the task's BOM and operation definitions from the Coordinator and executes the planned operation.

The Distributed Control System is central to the innovation in software that enables CubeSpawn to automate entire production orders that would traditionally require human operator intervention for many material flow and set-up tasks.

The following description of system components is visualized in **this diagram.**

Distributed Communication Architecture

In a full ERP implementation, the production module's planning engine would plan an order's production, routing inputs from the BOM through work centers in a sequence of operations to produce the finished good. The resulting planned production order would then be released to the shop floor for execution. CubeSpawn represents all or a part of a fully-automated "shop floor" whose distributed control system directly executes production orders without human intervention.

In the absence of an ERP system during development and simulation, the CubeSpawn control system input will be a synthetic, hand-coded and XML-formatted **Order Production Plan**, containing the same type of input data from an ERP production module, such as BOM input definitions and routed sequences of operations. The Order Production Plan may be directly loaded and executed by the CubeSpawn Coordinator node.

When a new Order Production Plan is loaded, the Coordinator builds internal data structures to track task states. As tasks are dispatched to CubeSpawn Controller nodes and executed on their attached machine tools, the Coordinator updates task execution state, setting state flags such as “blocked”, “pending”, “executing”, “completed”, and recording other data relevant to monitoring and control.

All communication surrounds structured state data in custom schema, mediated via MTConnect clients polling MTConnect agents running on both the Coordinator and the Controllers. In all cases, agents on one side expose views of internal state data tailored to client functions polling for changes and updating internal state on the other side, and triggering actions accordingly. These mechanisms, distributed across Coordinator and Controller nodes, form the data and execution logic structures required for loosely-coupled nodes to coordinate task execution through asynchronous, event-driven communication, with a high degree of autonomy.

The Coordinator node separately publishes a view of overall process state data suitable for graphical interfaces and external monitoring, and provides an interface for external control commands to submit new processes, stop tasks, disable machines, etc.

CubeSpawn Coordinator Architecture

The Coordinator software is the central module controlling production in a CubeSpawn configuration. It loads an Order Production Plan from ERP or a file, and manages the routing of product inputs through operations in the various Cube work centers. It presents interfaces to higher-level systems for releasing order plans, issuing control commands and reading monitoring and visualization data. Within the Distributed Control System, it interfaces with pallet mover and machine tool Controllers to coordinate production activities.

The Coordinator maintains production plan data and keeps track of the current status of each operation. It exposes a view of this data through a custom MTConnect schema, tailored for each individual Cube Controller to read its current task definitions and control instructions. It periodically polls each Cube Controller’s MTConnect agent for machine and task status and accordingly updates its internal state data to represent current overall production progress.

The Coordinator’s MTConnect agent also publishes a separate view of data for higher-level ERP, monitoring and visualization systems.

CubeSpawn Controller Architecture

The first CubeSpawn Cubes under development are mill, 3D printer and lathe CNC machine tools and pallet mover Cubes for transporting work. Each machine tool has its own Controller, responsible for accepting tasks from the Coordinator and supervises those tasks’ execution through the tool’s CNC controller.

As part of the Distributed Control Architecture, the Controller publishes “idle” or “busy” machine status, task progress and related state data through its local MTConnect agent. This data, structured with a custom schema, is polled by the Coordinator as part of control and management functions. The Controller, in turn, polls the Coordinator’s MTConnect data for retrieving task definitions and control instructions.

The Controller’s MTConnect agent also publishes CNC data in standard schema appropriate to the tool for visualization and monitoring.

Machinekit serves as the CNC controller software. The open-source Machinekit is based on code from the NIST EMC “Enhanced Machine Controller” project, a configurable CNC controller for x86 hardware proven in industrial settings. Machinekit’s Machinetalk interface and portability to ARM SOCs add extra value to the CubeSpawn application.

The Controller operates the CNC tools through Machinekit’s Machinetalk API, the same interface commonly used by Machinekit CNC (human) operator controls. The Machinetalk API exposes a complete interface to issue control commands, inquire about machine status and transmit g-code files.

The pallet mover’s Controller structure is similar to machine tool Cubes. Work transport is a new application for Machinekit, but little challenge is expected, since the pallet mover is orders of magnitude less complex than a CNC machine tool application.