Connected Factory Exchange

Issues to be Discussed

12-JAN-2017

John Walls – Aegis Software

# Common Terms for Industry and Manufacturing Concepts

Before namespace and message definition can begin, we will need to agree upon terms for various manufacturing and industry concepts. This is necessary because some of this terminology will be built into the namespace and message naming.

|  |  |
| --- | --- |
| Description | Proposed Term(s) |
| Materials in a factory are typically managed by storing and transporting them in some form of container (reel of SMT components, zip-lock bag of parts, tote, box, tray, cart, bin, etc.). Regardless of the container type, software systems work with materials by assigning each unique instance of material a unique identifier (UID). Part Number, Quantity, Lot Number, and all other attributes of that material instance are then related to that material UID.  We will need a term to use throughout the standard to represent this concept of a singular material instance. When a reel of SMT components, for example, is loaded onto a feeder, we will need a term to represent that reel of components. It cannot be “reel” because that is too specific. It could be a tray, a tube, a bag of bulk parts, etc, etc. | HandlingUnit  MaterialHandlingUnit  MaterialUnit  MaterialInstance  ??? |
| The products that are built within the factory are also typically managed and tracked using a unique identifier or UID. What term will we use represent a single instance of a product that is being built/assembled/tested etc. | ProductionUnit  Unit  UnitUnderProcess  ??? |
| A physical location or station where work of any kind (assembly, test, inspection, etc. etc.) is performed on production units. This could be both an automated robotic station, or a station where humans perform work. | WorkCenter  Workstation  ProcessStation |
| A physical location where materials are stored. This could be an entire stock room, a KanBan shelf, an automated material storage and retrieval system, an automated SMT reel storage tower, etc. | Stock  MaterialStock  MaterialStorageUnit |
|  |  |
|  |  |
|  |  |
|  |  |

# Namespace Organization

1. Separate Commands and Events (synchronous vs asynchronous message types) into two separate root namespaces? Or mix the two within one singular namespace root?

# Namespace Names

|  |  |
| --- | --- |
| Namespace |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Naming Conventions for Messages

1. Synchronous Messages
   1. Single parameter for all commands (Request object)? Or multi-parameter style?
   2. Request / Response suffix? Command / Response? Other?
   3. Different convention for commands vs queries or requests?
2. Asynchronous Messages
   1. <Target><Action>? Eg. <Unit><Processed> = “UnitProcessed”
   2. Other convention?

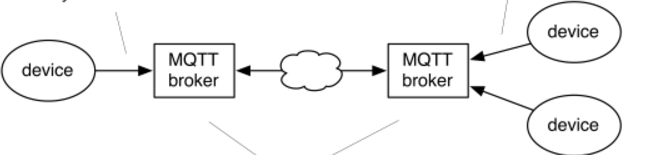
# CFX Terminology and Definitions

1. What term will be used to represent a single “participant” in a CFX network? (endpoint, CFX endpoint, device, CFX device, etc, etc…) Consider endpoints aren’t always “devices” really. An endpoint could be a software system (like ERP) that is participating in a CFX network.
2. What exactly is a “single device”? Consider a Fuji NXT. Is each module a device? Or does a singular contiguous block of modules represent one device? Does it matter?
3. We also need to consider “grouped” endpoints. A single line controller might serve as a singular endpoint managing multiple “devices” or machines within a single production line or even multiple lines. Maybe this does not matter. Perhaps each unique “device” within the mesh just has a unique URL, allowing multiple devices to share the same IP address or hostname. For example:
   1. cfx://endpoint1/device1 (Hostname = endpoint1 = 192.168.1.1)
   2. cfx://endpoint2/device2 (Hostname = endpoint2 = 192.168.1.2)
   3. cfx://endpoint2/device3 (Hostname = endpoint2 = 192.168.1.2)

So… device2 and device3 are both managed by a single line controller (endpoint2)

# Transport Related Questions / Issues

1. Are synchronous (REST) communications always direct “point-to-point”? Broker or any intermediary involved? For Internet situations, may not be feasible to “expose” / “publish” all devices in a factory to the public Internet. Reverse proxy perhaps? Use single public IP to publish all devices, and use HTTP “HOST” header in REST commands?
2. Does MQTT Support any kind of “discovery” mechanism? Do we need to explicitly incorporate endpoint discovery into the content / messaging structure?
3. Does MQTT Support bridging / federation for multi-site subscriptions?



# Miscellaneous

1. Date / Time format? ISO 8601?
2. State Model for machines / work centers?