

Basic Concepts

Protocols

Each equipment has its own communication protocol and, as an Equipment Integration stakeholder, the first step is understanding the communication protocol and its specification. There are three main concepts about a protocol that are critical to an equipment integration success: **Events**, **Commands** and **Properties**.

Event

An event is something you want to be notified of, in the lifecycle of the machine. The events available vary according to the protocol and equipment under analysis. They can be as simple as the creation of a file, or changes in the content of a file, or a change on a value of the equipment.

Example:

A material is placed in a resource, and a change is detected, indicating that a material is present.

```
graph LR
  A[start] -->|Material placed in the resource| B(Resource)
  class A,C mermaid_start
```

This triggers an event that notifies a system (MES) that is waiting for this information.

```
graph LR
  A[start] -->|Material placed in the resource| B(Resource)
  B --> |Event: Material placed| C(MES)
  class A mermaid_start
```

Property

A property is a value that holds a specific meaning in the context of the machine, and like events the properties available also vary according to the protocol and equipment under analysis. A property can be, for example, the name and size of a file that is sent along an event notification triggered by a file change, or even the state of a machine informing if it is running or not. An event may have several properties.

Example:

When the system (MES) receives the notification informing the material placement, it also wants the value of the property temperature at the time the material is placed.

```
graph LR
  A[start] -->|Material placed in the resource| B(Resource)
  B --> |Event: Material placed <br> Material ID=123 <br> Property Temp=40°C| C(MES)
  class A mermaid_start
```

Command

A command is an action sent to the equipment that may or may not have a reply, as for example the command to an equipment to start processing, which may have a success or failure reply, or the command of deleting a file. The commands available vary according to the protocol and equipment under analysis. Commands serve to interact with the equipment from an external application.

Example:

After the system is notified about the material placement, and knows the temperature, it sends the command to the equipment for it to start the material processing, and changes the property machine state to processing.

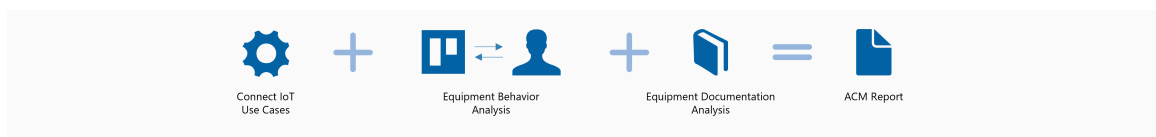
```
graph LR
  A[start] -->|Material placed in the resource| B(Resource)
  B --> |Event: Material placed <br> Material ID=123 <br> Property Temp=40°C| C(MES)
  C -->|Command: Start Process <br> Property Machine State: Processing| B
  class A mermaid_start
```

In this scenario MES can perform validations knowing which machine is being used, the material being processed (and all the context of that material), such as validating that the material should be processed in the equipment and, for example it can run when the machine is at 40°C.

Process Validation - Automation Capability Management (ACM)

All the information about a protocol is usually provided on the device documentation, where there is usually a section dedicated to the protocols properties, events and commands. But relying on this information only to carry out an Equipment Integration, is not enough, because the business development also depends on the actual device behavior, not only on to the information provided by the documentation.

To support this operating approach, there is an Automation Capability Management (ACM) process that can be performed. This process refers to an analysis, from which results a report stating the scenarios that will be covered (use cases) and what we need to be aware of whilst the development (when things happen, what to read, what to send, etc.).



This saves time testing the machine. It is a common scenario to have very limited reserved machine time to perform validation tests. So it is useful to have a breakdown of the integration scenarios and of the pre-requirements of each. This way, you can have a systematic problem checking approach before embracing a full integration and be confident that the assumptions you will be doing in the equipment integration matches the real behavior.

ACM Practical Examples

This example, is based on a real world scenario, where a machine to start the production process needs to have its recipe chosen. Using the Connect IoT application, a logic will be designed so the recipe is automatically chosen through a command, as soon as MES receives the information about the material that is about to be processed.

With the ACM of the equipment in the shop floor, one will notice that for the recipe to be chosen in the equipment, its state must be a specific State A, and no other. If the resource is not on state A, choosing a recipe is not possible.

By consulting the equipment documentation, the only information is that the Command X should be used to select the production recipe, and only with this information the business logic would not work as expected.

Another example of the importance of analyzing the behavior of a machine while in a production setting is, for example in a file integration, not just knowing the file structure and type, which normally is given in the machine documentation, but also when the file is generated, when information is appended to the file, if the file is deleted or moved at the end of the process and the physical moments that match these changes.

These scenarios are common, and they are the main driver for validating and assessing a machine in real life.

Documentation usually informs the user about the messages, its information and format but it doesn't enlighten the reader about the circumstances under which said messages are valid and when they are produced.