Advanced MTConnect Asset Management (AMAM)

T. Trautner [1], F. Pauker [1] and B. Kittl[1]

[1] Institute for Production Engineering and Laser Technology – Vienna University of Technology, Getreidemarkt 9, 1060 Vienna, Austria

Keywords: MTConnect, asset, agent, ISO 13399, cutting tool, tool management, tool measurement

Abstract. The universal factory floor communications protocol MTConnect allows the exchange of information between shop floor equipment and software applications. Data from shop floor devices is collected by a piece of software called adapter. The adapter sends the data to a local webserver called agent, which associates the data with an information model and makes machine data available to clients in uniform MTConnect representations. The standardized MTConnect schema definition serves as a structure for modeling devices, independent components of the devices and device's design characteristics. Additional equipment associated with the manufacturing process that is not a component of a device can be modeled as an MTConnect asset. Examples of assets are objects like cutting tools, workholding systems, parts and fixtures. Apart from the fact that currently only few MTConnect adapters available on the market provide asset data, state of the art NC controllers only manage rudimentary data regarding assets such as cutting tools or workholding systems.

To tackle this problem the paper describes a network architecture for Asset Management based on the MTConnect standard and a piece of software called Asset Broker, distributing assets to multiple Machine Agents and allowing for detailed asset updates with regard to process information. The Asset Broker thereby closes the gap between MTConnect assets and current machine tools poor asset management capabilities and enables the synchronization of updated asset information between Machine Agents and software applications such as Manufacturing Execution Systems (MES), Tool Management Software (TMS) or Enterprise Resource Planning Systems (ERP).

Introduction

Initiatives like Industrie 4.0 (I4.0) and Industrial Internet of Things (IIoT) force the integration of Information and Communication Technologies (ICT) in manufacturing. One key task is machine data acquisition from a diverse set of industrial equipment on the shop floor. Therefore uniform, robust communications are part of the necessary infrastructure for modern manufacturing systems. MTConnect, as an open, royalty-free standard, allows great interoperability between both devices and software applications and is getting more important in the manufacturing domain. With MTConnect machine tools provide a unique data model of their physical structure and the data items they offer. This allows the monitoring of machine tool's process states and performing data analytics based on standardized software applications. Besides the components and data items of machine tools, a manufacturing process is related to other objects that are not a component of a machine, but also need to be addressed by the MTConnect standard, because they contain or provide viable process information. These objects can be cutting tools, workpieces (parts), NC-programs, a work order or other objects that can be created, transformed or removed within the process. To allow generating and managing such complex data models at runtime, MTConnect assets were introduced. As a consequence MTConnect assets can also change their data structure in the XML (Extensible Markup Language) representation and therefore may offer a good framework for future expansions and developments within the MTConnect standard. Several machine tool producers already offer an MTConnect adapter (e.g. Mazak or DMG [1]), but most of them only allow data access for the device and do not support MTConnect assets. Other adapters like the Okuma MTConnect adapter [2] already generate and provide cutting tool assets for agents, but lack the ability to import those assets from agents and thus may lose information added from other sources to the assets. To tackle this problem, the paper presents an approach for advanced MTConnect Asset Management. Therefore an MTConnect Asset Broker is presented which could be used for MTConnect based monitoring of manufacturing equipment and thereby allows for managing and updating assets with process data.

The first part of the paper gives a general overview of MTConnect and MTConnect assets. In the second section the MTConnect Asset Management is specified in detail. After that an MTConnect Cutting Tool Broker, which has specialized functionality for cutting tools, is described as a sample application. In the last section we give an outlook to future steps.

MTConnect

MTConnect (Manufacturing Technology Connect) is a one-way (read-only) communication standard developed by the University of California, Berkeley. MTConnect was primarily developed to provide manufacturing equipment, especially CNC machine tools, with a standardized interface, the *Bluetooth of manufacturing* [3]. This allows connecting manufacturing equipment on the shop-floor to a local area network and is so far mainly used for manufacturing data acquisition. It does not only define how data is transferred, but it also describes what is transferred based on pre-defined types. Therefore MTConnect offers a language (*schema definition*) for the exchange of semantic data between manufacturing equipment and software applications that delivers a standardized information model to provide a description of a machine tool's (*device*) physical structure through its components and subcomponents. These components are extended with data items, defined by specific types and categories. A category may be an event for discrete data items, a sample for continuous values or a condition for health and error messages, whereas types describe the meaning of a specific data item like the *Position* of an axis or the *RotaryVelocity* of a spindle. This allows the mapping of the real process data to a virtual device structure.

With MTConnect data collection from the adapters, data processing and information serving is handled by a standardized application, the MTConnect agent. The agent is an open-source software application that implements the adapter socket communication and a Hypertext Transfer Protocol (HTTP) server [4]. This is a web server, which provides clients like software applications with all the necessary information about a system. The MTConnect agent provides different types of data exchange. Besides the polling function using HTTP GET the agent also supports event-based streaming services using network sockets. Accordingly, the communication between agent and clients is realized using established web-technologies, which provide a simple way to monitor and collect data for users and software applications. HTTP allows clients to access the agent from any location within the same local area network by connecting to the agent's endpoint, like the URL within a web browser. Also specific data items requested may be specified within the URL. The clients have the possibility to select all or just the required data items from the device model, based on specific data items or components. All data delivered by the agent is formatted in XML. This applies to the device model, as well as to the actual process data (Fig. 1).

To make a machine tool ready for MTConnect a device-specific piece of software is needed, the MTConnect adapter. This software reads out data of the device, decodes it, adds a timestamp and an identifier and then provides it to an agent. In the future the adapter should be developed and provided by the machine tool or NC control manufacturer. The adapter provides every agent connected to the adapter endpoint, with a complete data image of the current states. Subsequently transmitted data consists of changed values only. Data is sent in a specific format, the Simple Hierarchical Data Representation (SHDR), which consists of a timestamp, an identifier and the value of the data item. This reduces the amount of data exchanged between agent and adapter.

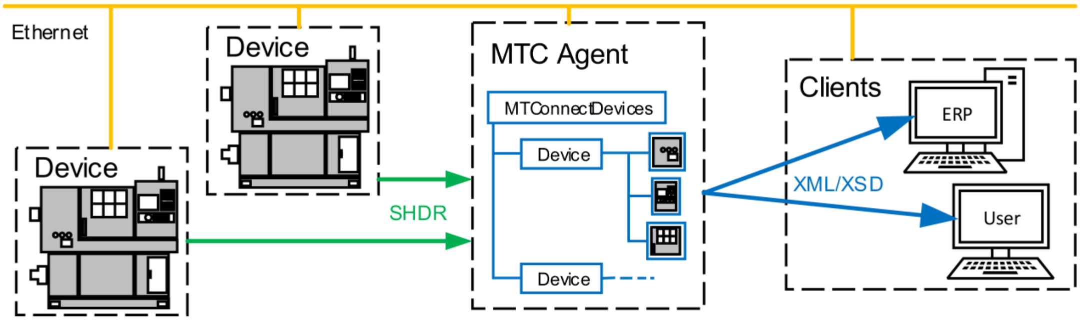

Fig. 1 MTConnect network infrastructure

MTConnect assets

The MTConnect agent does not support dynamic change or adaption of the data structure of the devices. This is caused by the fact that typically the physical structure of a device does not change very often. Some objects, like an assembly of multiple workpieces, will transform their physical structure more often than a device and require to be handled at runtime. Therefore, the standard defines the MTConnect assets. Assets are entities that are used in the manufacturing process, but are neither a device nor a component of a device. They can be associated with different devices during their lifecycle [5]. Thus they can be moved between different devices and also be extended with new elements and attributes. Currently only two types of assets are addressed, *CuttingTool* and *CuttingToolArchetype*. In the future they will be extended with parts, workholding systems and fixtures. The standard does not exactly define which objects can be an asset. The authors believe that assets can be either physical or virtual entities (e.g. NC-Programs) [6]. An instance of an asset is represented in XML-format, based on a schema definition for assets and always needs a specific identifier (*AssetID*) along with a universally unique identifier for the device currently containing the asset (*DeviceUuid*).

The characteristic feature of assets is that they can change their values and structure (XML elements and attributes) during life-time. Assets can be assigned to and removed from devices. When an asset is added, changed or removed from an agent, the agent will automatically generate an *AssetChanged* or *AssetRemoved* event in the devices information model. These events contain the information which asset has been changed for a specific device (*AssetID*).

ISO 13399 / Cutting Tool asset

The ISO 13399 (cutting tool data representation and exchange) [7] is a standard for representing data of cutting tools and tool holders. It was developed by AB Sandvik Coromant, the Royal Institute of Technology in Stockholm, Kennametal Inc, and Ferroday Ltd. The primary objective is to provide a mechanism for describing cutting tools, independent from any particular system. Thus it can be used for data exchange in a neutral format but also for archiving and sharing a product database. The usage of the ISO 13399 standard will simplify the exchange of data for cutting tools. This is why in the latest specification of MTConnect cutting tool assets are represented with a structure similar to the ISO 13399.

As mentioned before the assets currently specified in the standard are *CuttingTool* (CT) and *CuttingToolArchetype*. These types contain the cutting tool definition from ISO 13399 in XML (ISO 10303-28) or EXPRESS (ISO 10303-21) format. In the beginning this cutting tool definition was part of the CT asset but in Version 1.3 it was defined separately in the archetype asset to make sure that also CTs without a complete ISO 13399 definition can be used. The CT asset may now have a reference to an archetype [8].

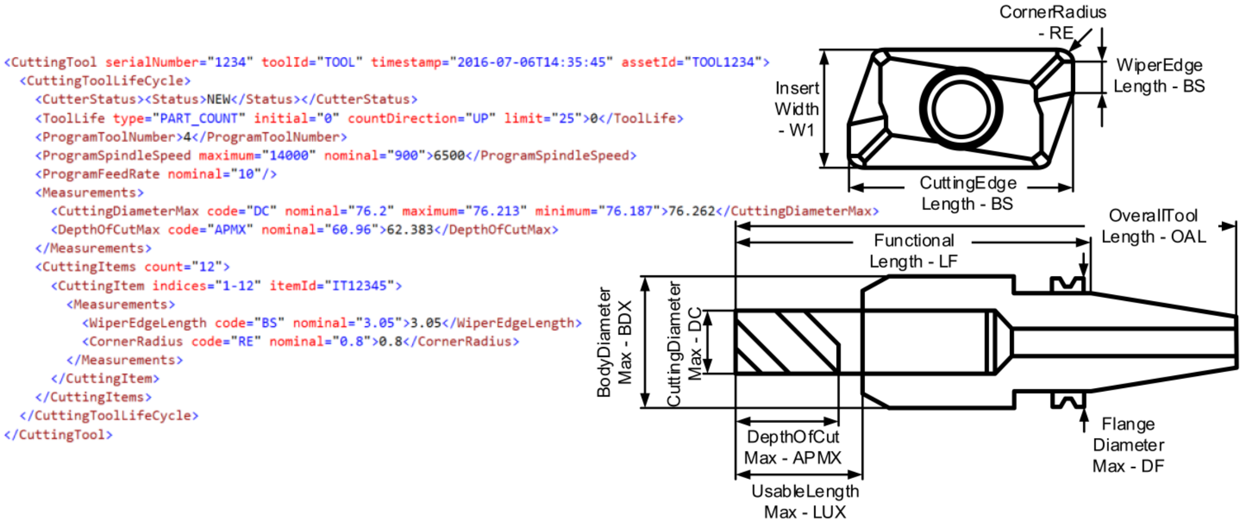

Fig. 2 MTConnect cutting tool including ISO 13399 measurement codes [8]

The data model of a CT in MTConnect is limited to information that is essential to a manufacturing process. Therefore the term CT in MTConnect always refers to the complete assembly of cutting items, the tool item and adaptive items [8]. Thus lengths and other measurements always refer to this complete assembly (Fig. 2). Specific measurements for the tool item and adaptive items may only be given with their static nominal values within the *CuttingToolDefinition* of the archetype. In contrast the measurements for specific cutting items may be included within the CT asset, as cutting items have an essential impact on the production process. Therefore the implementation of the ISO 13399 standard into the MTConnect CT asset is simply an adaption of the ISO measurement codes into the terminology of MTConnect. This information will be added in the XML structure below the *Measurements* element. Measurements may also be added for single or multiple cutting items.

Advanced MTConnect Asset Management

As most of current machine tool's MTConnect adapters are not able to deal with MTConnect assets, this paper presents a piece of software, which manages assets on the shop floor. It acts like a Broker between agents of different devices and distributes their assets. This software also provides services for updating assets with process relevant data gathered from the devices and delivering condition messages about the asset to the device's agent. So the Broker needs client as well as server functionality. To get process data and actual assets the Broker acts as a client requesting data from an agent. Computing the retrieved information allows the Broker to create an updated version of the asset that contains the transformations made to the asset within the machining process. Lastly the Broker also acts as an adapter for the agents, which allows adding, updating and removing assets (Fig. 3).

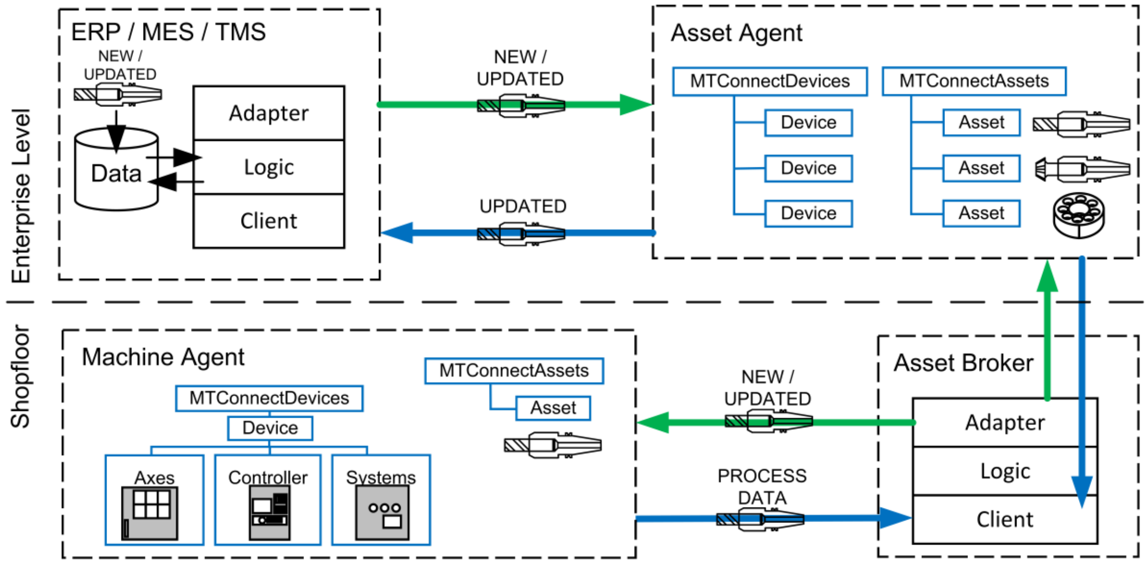

Fig. 3 Overview of the MTConnect Asset Management paradigm

Asset Source System

Caused by the fact that the Asset Broker can only update existing assets, the Advanced MTConnect Asset Management system needs an instance which can create new assets. Assets can be created by several entities on different layers in the company. Mostly an ERP, MES or TMS creates new assets, because it manages manufacturing orders and shopfloor resources. These systems need an interface to create or update assets that correspond to the MTConnect standard and provide them to agents. They also need an interface to retrieve changed asset data from those agents to update the source system.

Asset Agent

With an interface to a source system assets can be created, updated and delivered to an MTConnect agent. It is possible to connect this source system interface with several Machine Agents and provide them directly with their specific assets. If the device associated to the asset is unknown to the source system, the asset needs to be associated to a default virtual device like a storage area or container. An MTConnect agent called Asset Agent could contain a simplified model of all shop floor devices as well as the virtual devices and handle the communication with the Asset Source System. This way the Asset Agent is a specialized agent containing only assets and always representing the current assets contained in the source system.

Asset Broker

The Asset Broker plays three basic roles in the network: The first role is to act as a client, so the Broker can connect to the Asset Agent and the Machine Tool Agents. Thereby the Broker gets new assets from an Asset Agent and monitors process data and assets from the Machine Tool Agents. The second role includes computing and logic capabilities. These functions are responsible for aggregating the the assets with process data from the machine tools, based on an asset-type specific logic. The last role of the Broker is to act as an adapter that allows agents to connect to the Broker and collect the new assets with updated process data. To ensure that asset data is sent to the appropriate Machine Tool Agent the Broker uses the HTTP PUT method. This method was presented in the MTConnect agent to allow clients to upload assets or data items to a particular agent [4]. Thereby the Broker can systematically send and manipulate assets and data items in specific agents. The major advantage of using the presented Broker for asset management is that the communication and data processing is based on the MTConnect standard. No additional information and proprietary information models are needed. The Broker only needs the IP-addresses / endpoints of all agents to manage. For security reasons the Machine Tool Agents also need the IP-address of the broker to enable the PUT mechanism only from this specific source.

The Asset Broker continuously monitors the *AssetChanged* events from the Asset Agent and retrieves the changed or new asset when an event is triggered. Based on the associated device, the Asset Broker is able to distribute the asset to the corresponding Machine Tool Agent via a HTTP PUT request. Furthermore the Asset Broker is monitoring the states of the machining processes on the machine tools. This allows the Broker not only to enrich the asset data models with additional information or to update existing values, but also to provide asset specific data items for the agent to facilitate the state or condition of an asset. It may use condition data items from the agents device model to present the health state of the asset caused within a specific component. If a Machine Tool Agent provides a unique identifier for the currently used assets like *ToolAssetId* or *PartAssetId*, the Broker may move an asset from another device (like from the virtual device for assets with an unknown associated device) to the device utilizing the asset.

Sample Application

The approach presented in this paper has been demonstrated through the application of the paradigm to the management of cutting tools. Therefore an interface for data exchange between a Tool Management System (Zoller Tool Management Solutions) and an MTConnect agent for cutting tools was developed. This Tool Agent is a specialization of the Asset Agent presented in the previous chapter. The TMS and its database were primarily used as storage for all tool data. Access to the database was realized using a web service provided by the TMS. Using this service the interface can request new or measured tools and offer them in an MTConnect conform cutting tool format through an integrated adapter. Beside the allocation of tool data the interface allows monitoring of the agent's assets, so changes on the agent side can be adopted in the TMS. Thereby the Tool Agent always contains a representation of the TMS data.

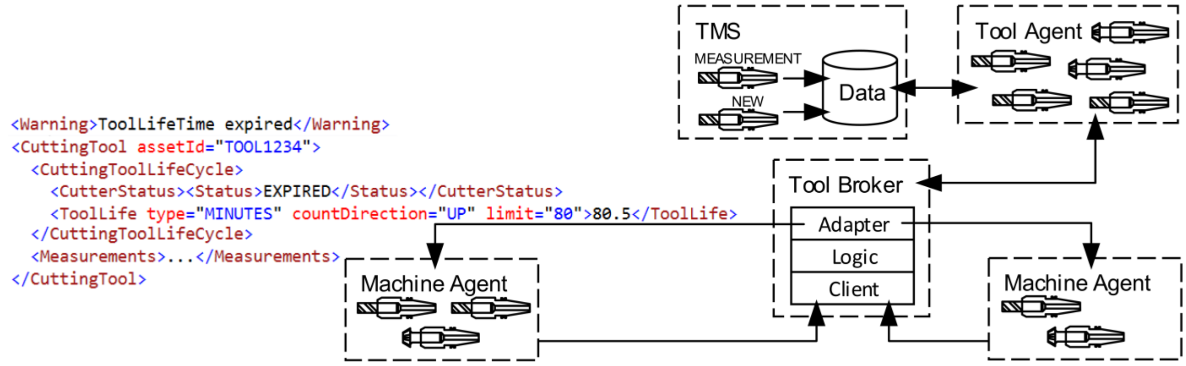

Fig. 4 Functional overview of the AMAM for cutting tools

At this point the Tool Broker distributes the CT assets from the Tool Agent to the Machine Agents, based on the *AssetChanged* and *AssetRemoved* events within the Tool Agent's devices and thereby synchronizes their data. Thus the Machine Agents also contain the latest image of the measured or new CT (Fig. 4).

Equipped with the CT assets, the Broker can pursue its task of monitoring process data and applying it to the CT assets. For this purpose the Broker searches the Machine Agent's information model specifically for particular types which are relevant for the CT's life cycle and requests the agent to stream new data as soon as it becomes available. These types include *RotaryVelocity*, *AxisFeedrate*, *ToolAssetId*, *ToolNumber*, *Execution*, *PartCount* and *AssetChanged*.

The data item *ToolAssetId* allows identifying the tool currently being used. If the Machine Agent doesn't provide this data item or it only offers the tool number and not an ID the Broker has to rely on a static *ToolNumber* and search for the asset associated with the device and containing this *ProgramToolNumber*. With this information, the Broker requests the current CT from the agent and enriches it with the computed process data: *CutterStatus* may be set from a *new* or *measured* state into a *used* state, if the tool has not been used before. Process data like *PartCount* or the *Execution* state of a machine tool allow the Broker to calculate the remaining *ToolLife* and *ItemLife* of the CT. Life time can be expressed by the actual time of usage (*minutes*), the count of parts produced (*part_count*) or the degeneration (*wear*) of the CT. The time of usage is calculated in a simplified way by adding up the time in which the tool has been used and the execution state of the control was *active*. The number of parts produced is calculated based on the tools used and the *PartCount* given. Exceeding the lifetime or the warning limit of the *ToolLife* the Broker provides a *system*-type fault or warning condition containg the expired CT in the error message.

Besides data modifications in the asset the Broker also monitors current process speeds like *AxisFeedrate* and *RotaryVelocity* and matches them with the maximum allowed values for *ProcessFeedrate* and *ProcessSpindleSpeed* given by the CT. Exceeding those target values may provide a warning for a particular axis.

Conclusion

The presented concept of an Asset Broker to manage MTConnect assets has facilitated the provision of cutting tool data based on the MTConnect standard and filled the gap of current machine tool's adapters with respect to the handling of CT assets. For future implementations *AssetType*-specific applications must be defined. For example, *Part (workpiece)* assets are identified using the type *PartAssetId*. Considering this asset in the sense of smart products, it would gather and provide information about itself during the production process and its whole lifetime. Therefore process-specific data, such as process time, duration and any condition states in a particular device may be added to this asset.

Future implementations of the CT Broker may provide a tool agent interface (panel) for the operator or other applications to allow handling of tools. This will include associating the tools with new devices, changing the *ProgramToolNumber* and defining the *Location* in a tool changer or the station of the CT. Furthermore, an implementation of an MTConnect client interface into the machine tools may allow taking advantage of the cutting tool assets, like recognizing an expired CT or adjusting to the given measurements.

Acknowledgment

This paper was realized within the scope of the Manufacturing Analytics System (MAS) project, in response to the Call Pro Industry 2015 funded by the Vienna Business Agency.

References

[1] B. Edrington, B. Zhao, A. Hansel, M. Mori, and M. Fujishima, "Machine Monitoring System Based on MTConnect Technology," *Procedia CIRP*, vol. 22, pp. 92–97, 2014.
[2] Okuma America Corporation, "Okuma MTConnect Adapter - Software User Manual." 2015.
[3] D. Edstrom, *MTConnect: To Measure Is To Know*. Virtual Photons Electrons, LLC Ashburn, VA, 2013.
[4] W. Sobel, "MTConnect C++ Agent Version 1.3.0.0," *GitHub*, 2016. [Online]. Available: https://github.com/mtconnect/cppagent.
[5] W. Sobel, "MTConnect Standard Part 4 – Assets v1.3." 2014.
[6] T. Trautner, "Sensor- und Steuerungsintegration mit MTConnect," Vienna University of Technology, Vienna, 2016.
[7] ISO 13399-1:2006, "Cutting tool data representation and exchange -- Part 1: Overview, fundamental principles and general information model." 2006.
[8] W. Sobel, "MTConnect Standard Part 4.1 - Cutting Tools v1.3." 2014.