CIRP Industrial Technical Paper

Authorship approach and production method for machine-readable manufacturing data standards

Tim Shinbara (3) [1*] Russell K. Waddell [1], Moneer Helu (3) [2]

1 The Association For Manufacturing Technology, McLean, VA, USA
2 Engineering Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA

*Corresponding author. Email address: tshinbara@amtonline.org

Abstract:
Standards to represent and manage manufacturing data are proliferating as industry rapidly digitizes. The existing global standards development paradigm for manufacturing data needs to shift to digital tools and workflows that produce machine-readable, interoperable standards rather than static, rigid, and siloed documents. We propose a general approach for machine-readability, and specific production method and tools to affordably produce standards and implementation artifacts. The approach leverages Redmine, Overleaf, and GitHub for collaboration and issue tracking, SysML/XMI and Cameo Enterprise Architecture for information modeling, and LaTeX, custom scripts, and PDF for publishing. The MTConnect standard is used as an example, as well as the MTConnect - OPC UA companion specification.

Keywords:
Standards Development, Information Modeling, Systems Architecture, MTConnect, OPC UA

1. Background

Smart manufacturing is a booming industry as production and engineering functions rapidly adopt modern software, computer networking, and IT systems. As a result, standards related to manufacturing data have proliferated. There are two broad classes of manufacturing data standards: domain specific, wherein the standard is only applicable to manufacturing; and, domain non-specific, wherein the standard is applicable to multiple industries including those outside manufacturing. Across both classes, most standards offer unique functionality, and most standards also have at least some overlap with other standards. Users want to select best fitment to purpose for each functionality, which in turn creates a need for modular, interoperable standards. Standards development organizations (SDOs) can accomplish this interoperability objective by striving to produce machine-readable standards in addition to or instead of human-readable standards.

2. A standards development approach for interoperability

Standard developing organizations (SDOs) have enthusiastically embraced some approaches to working with remote teams. They should take this further and embrace new technology on an ongoing basis to increase speed and agility.

Existing processes and procedures across developers establish a clear hierarchy from company-level, to national consensus, to international consensus. These methods and the standard development organizations that sanction and referee them are excellent at: identifying stakeholders and gathering input; consensus building among stakeholders; and, creating transparency. Procedures governing voting and approval, meetings, documentation, how to raise and resolve issues, review, etc. enable industry, national, and international standards.

These functions often depend on documentation because they engage a large and diverse pool of stakeholders. SDOs have universally embraced digital workflows for document sharing and remote meetings but have not kept pace with current tools. The existing management process is based on single users checking in and checking out static documents, diagrams, and implementation examples. This creates stability for multi-author materials by creating a single source of the truth that is edited only in sequence. It is important that final documents be frozen snapshots in time, but adopting that approach for in-process materials dramatically slows the development process. Collaboration tools allowing simultaneous authors are widely available and already widely used for authoring standard documents (e.g., Google Docs, MS Word, Overleaf and other LaTeX editors). Sequential editing does solve a real problem of collaborative authorship and document management, but that is now an outdated approach considering the technology now available for simultaneous editing.

Applying rigid sequential processes to living draft documents slows down standard development by favoring a legacy process over tools that are better fit to purpose. An alternative approach that yields dramatic improvements is to continually reevaluate the production methods and tools of standard development against best-in-class technology.

3. Production method and tools for machine readability

There are several gaps in current standards production methods and tools that can easily be filled. Online issue tracking should be tied more closely to final releases by way of templates and rules. Information modeling should use standard languages and formats. Publication should take advantage of simple automation.

3.1. Issue tracking

Online issue tracking has been borrowed from software development and adopted relatively widely. Tools such as Mantis, Redmine, Bugzilla, and Jira among many others provide user-based access control to a central, online, always-on repository of work items. These serve as master checklists for contributors to the standards. By nature, teams developing standards are nearly always remote and benefit from all-hours access to proposals and in-process work. For international standards, language requirements are dictated by sanctioning bodies, but easy access eases the translation burden and technical contributors can work in their preferred language through a later point in the development, adoption, and approval process.

Online issue tracking is not universally adopted by SDOs but is on a clear growth trajectory. Among those that use it, there are still unnecessary manual steps between issue tracking and standard authorship that can be removed with a combination of business processes and technology. All online issue trackers contain machine-readable content, as each individual issue consists of a database record with various fields and field data. Contributors are largely ambivalent as to the rules for format requirements for submissions, and international SDOs with large contributor bases already have strict formatting rules for acceptable submissions. Using strict templates in issue tracking and work proposals will allow document publishing tools (e.g., MS Word, LaTeX, text editors) to consume content without manual formatting and editing. To understand the full potential of machine-readable content submissions, it is also important to understand how information models should be most efficiently developed, managed, and expressed.

3.2. Information modelling

Manufacturing data standards each express their own information model. New proposals for inclusion in a standard need to consider the existing information model and how proposals will change or extend it. Systems engineers and systems architects use the UML or SysML languages and associated editors and development tools to manage information models that will be consumed or referred to by other systems and processes. Attempting to express an information model without a modeling language requires that consumers (human or machine) simultaneously learn how a model is expressed and what the model actually is. This is an unnecessary and artificial barrier to understanding that severely hinders standards interoperability.

The information modeling interoperability problem will only get worse as manufacturing data standards mature and their information models grow in scope and complexity. Larger, more complex models will increasingly see machine-readability take precedence over human-readability. Adopting existing modeling languages and tools is currently the only path forward for these standards to scale up and meet the needs of manufacturers, but it also allows faster and easier publication of human readable documents that serve current needs.

3.3. Publishing

Publishing a standard combines content (text, tables, figures, media) and formatting. All standards in the manufacturing data management space are published to be human-readable. The predominant standards publication workflow is: word processor (e.g, MS Word, Google Docs, LaTeX) to PDF. Text is created in the word processor. Tables and figures are created in the word processor or exported from another tool and embedded as images. Media are exported as image files and embedded (in MS Word, Google Docs, LaTeX) or referenced (in LaTeX). Final documents are released as flattened PDFs to provide a single snapshot in time that cannot be altered by users.

Content creators interact with content at its source, whereas editors interact with the content once it has already been formatted for publication. MS Word and Google Docs are designed at human editors manipulating a final document. The LaTeX document preparation system is designed to allow a human programmer to manipulate a set of files and the content they contain via rules, metadata, libraries, objects, and scripts. Individual documents in MS Word and Google Docs also consist of a set of files and formats, and can be manipulated programmatically, but in practice are rarely used in this fashion.

Both online issue tracking systems and UML/SysML models contain machine-readable content by design. Minimizing variation and human intervention between these two content creation tools, as well as between content creation and publication will also minimize the overall work effort. Publication requirements are known and static and should be carried through and communicated back to the content origins. The requirements can be communicated actively through procedures and rules, and passively via templates and configurations.

4. Use case: MTConnect

The MTConnect Standard (ANSI MTC1.4/2008) is developed by the MTConnect Institute. That SDO has partially deployed the workflow presented here as shown in Figure 1. Issues are managed in online issue tracking with Redmine. Information modeling is done with SysML. Sparx Enterprise Architect and NoMagic Cameo Enterprise Architecture were both explored for production, and NoMagic Cameo Enterprise Architecture was ultimately deployed for this purpose. Issues in Redmine are templated and automatically formatted for publication using Overleaf for simultaneous collaboration on LaTeX documents. Official releases are flat PDFs.

This method was first deployed to publish "OPC 30070-1: OPC UA for MTConnect, Part 1: Device Model Release 2.00.00" with the OPC Foundation. That companion specification was developed between October and January 2019 and released June 2019, used Mantis for issue tracking (OPC Foundation maintained), UML for information modeling, and Overleaf for publication. The official release is a flat PDF.

MTConnect Version 1.5 was subsequently released in December 2019 with a process that varied slightly from that of the OPC companion specification. The companion specification was a complete overhaul, with a normative information model developed in UML and most content for publication generated directly from the UML model. No UML was completed for MTConnect Version 1.5, although a SysML model was completed and will be released accompanying Version 1.6. SysML was used because of the information modeling requirements and knowledge of the language in the MTConnect Standards Committee. Mantis was not tightly integrated to the publication process for OPC, whereas MTConnect issue tracking in Redmine was strictly formatted to allow content to drop directly into Overleaf for publication.

Figure 1

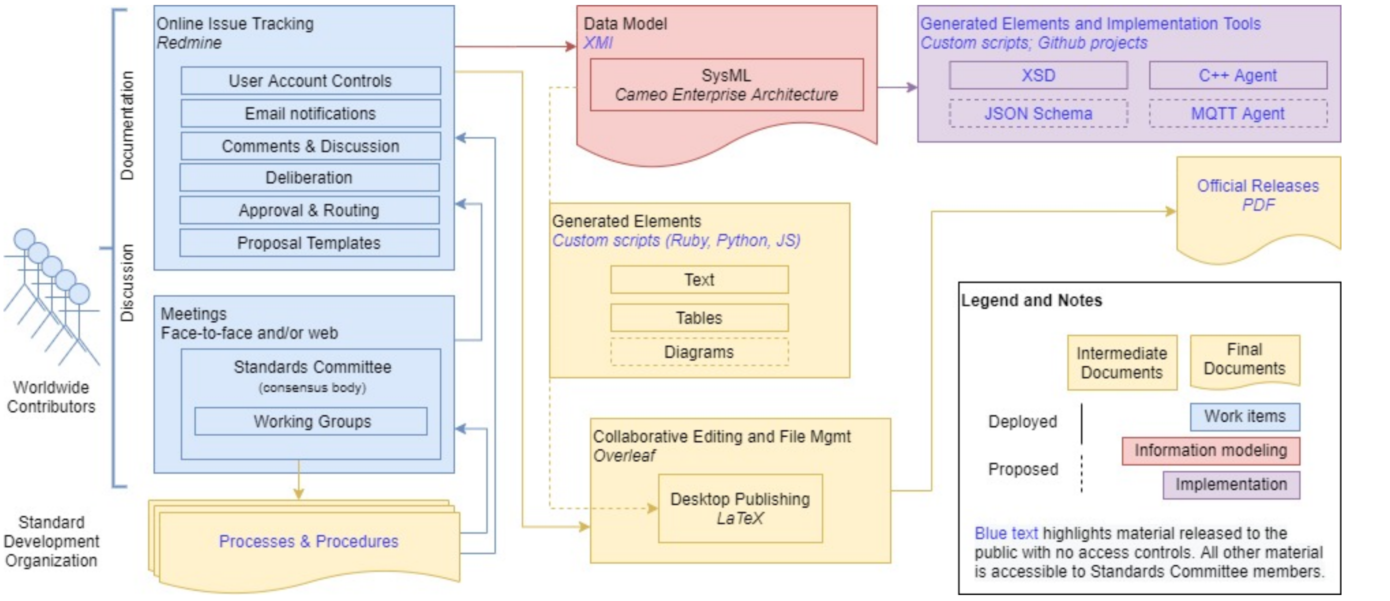

Figure 1. MTConnect Use Case: The standard development process for MTConnect adheres to sanctioned procedures developed in accordance with the ANSI Essential Requirements. The process also incorporates tools and rules to make the content machine-readable. The end-to-end process will develop across several releases.

5. Conclusion

Machine-readability will be required for manufacturing data standards to interoperate with one another. A method to produce and manage machine-readable content with existing commercial software and tools is presented, as well as specific examples of releases by the MTConnect Institute and OPC Foundation, two accredited standards development organizations. This method takes advantage of the existing global standards development infrastructure, speeds standards development in the current human-readable paradigm, and removes obstacles to interoperability and global-scale standards development and deployment.

6. Disclaimer

This work represents an official contribution of NIST and thus is not subject to copyright protection in the United States. Identification of commercial systems in this paper are for demonstration purposes only and does not imply recommendation or endorsement by NIST. Nor does it imply that these commercial systems are necessarily the best available for the purpose.