Siemens PLM Software

PLM-driven content management

Aligning engineering design and technical publications support functions: Ensuring product launch success

www.siemens.com/teamcenter

white paper

By using product lifecycle management (PLM) to integrate today's engineering design and technical publications domains, companies can dramatically improve the way in which content is managed across a product document's lifecycle. With this in mind, Siemens PLM Software provides Teamcenter® software's content management solution to enable engineering design and technical publication groups to share information retained in different system repositories, re-use content as often as possible and automate functions and processes common to both domains.

PLM Software

Answers for industry.

SIEMENS

Table of contents

Executive summary

The implementation of structured authoring initiatives and other technology advances has significantly improved the lifecycle associated with managing today's product and service documents – especially with respect to reducing production time and managing resources effectively. However, additional progress can be made by recognizing the intersecting relationships between engineering and technical publishing and capitalizing on their synergy to improve multiple business processes, including:

- *Content authoring*
- *Content editing/updating*
- *Document publishing*
- *Graphics management*
- *Document translation*

To gain these additional improvements, many of today's most innovative companies are using product lifecycle management (PLM) to integrate their engineering design and technical publications domains at multiple levels by:

- *Sharing information retained in multiple systems*
- *Re-using product and service content*
- *Automating functions and processes common to both domains*

Teamcenter software provides a PLM-driven environment that streamlines today's technical publications processes through dynamic publishing techniques. These streamlined publication processes enable technical documents to be developed in concert with the product development process.

Teamcenter content management capabilities address issues associated with traditional technical publications processes, including concerns that these processes:

- *Take too much time*
- *Have difficulty reflecting the latest engineering changes*
- *Inhibit extensive re-use of content and graphics*
- *Require "heroic" efforts to meet product delivery dates*
- *Fail to meet multiple language requirements*
- *Fail to publish timely content in all of today's required delivery formats*

Many of these issues result from the fact that the content for engineering design and technical publications traditionally has resided in disparate and discrete authoring systems and organizational domains. Teamcenter's content management solutions overcome this isolation by providing workflow, version control and relationship management capabilities that link product documents with their associated parts in an assembly.

PLM-driven content management directly relates XML content instances to a product's parts – thereby synchronizing the product and its documentation even when product changes arise. The relationships between product parts and content ensure that critical path documents that depend on engineering data flow will be completed without imposing unnecessary overhead (typically required when engineering and technical publication teams work with different systems in isolated environments).

In a PLM environment, the product definition is managed in one location – a logical repository that serves as the environment's authoritative information source regardless of configuration or whether manufacturing or as-sold BOMs are being referenced. Teamcenter-driven PLM environments provide a single logical authoritative source of product definitions and linked documents that can be manipulated by user-initiated workflow and data management capabilities to integrate engineering, manufacturing and technical publications.

Keys for Success

Companies can eliminate the isolation that separates technical publications groups from their engineering design/development counterparts by effectively using XML in both environments.

To facilitate effective collaboration, the technical publication group's system of choice and engineering design group's system of choice must use the same workflows and process automation environment.

Moving XML into an engineering environment – through the use of PLM-driven content management – is crucial to integrating the processes and information flows common to both environments, while at the same time delivering improved productivity and cost savings.

Recent evolution of technical publications

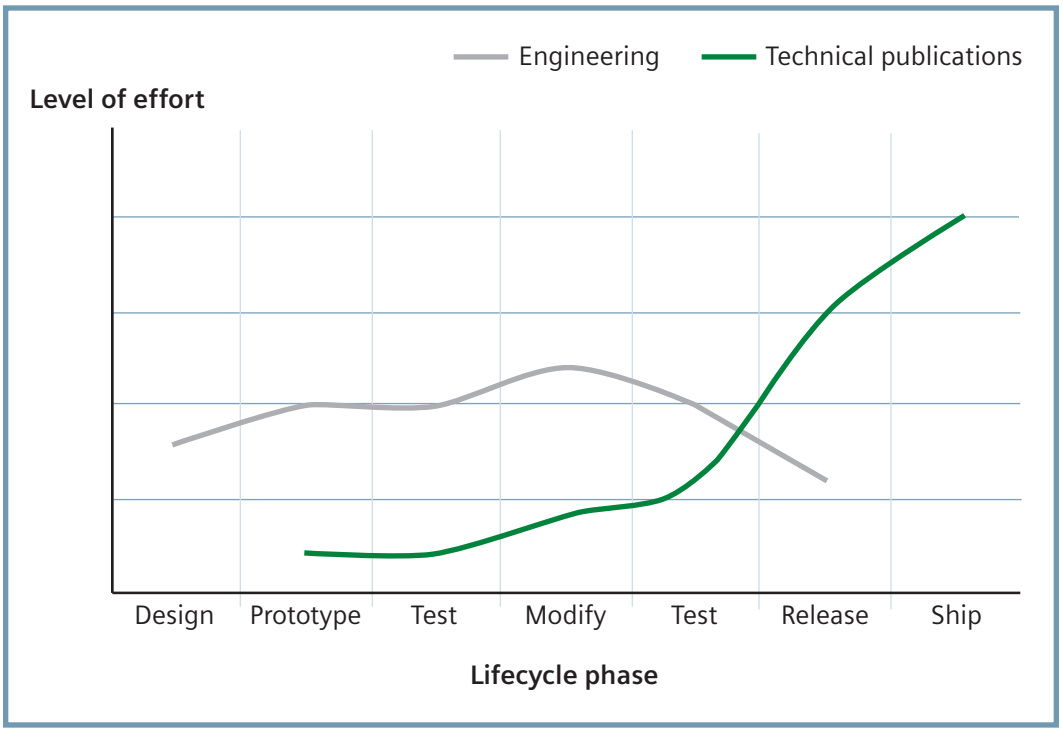
Over the last two decades, engineering design and manufacturing have experienced sweeping technology changes. Working relationships between these two organizational domains have been integrated through the use of interdisciplinary engineering software that recognizes and capitalizes on intersecting roles that each domain plays in the product lifecycle.

Product lifecycle management (PLM) has driven these initiatives by providing these domains with a common platform for effectively integrating otherwise isolated information assets and streamlining cross-discipline tasks across the enterprise. This integration has enabled companies to accelerate their product development cycles, improve product assembly, and target their product designs to highly selective market segments. Equally important, these changes have increased the need for more product support documentation while raising its level of complexity.

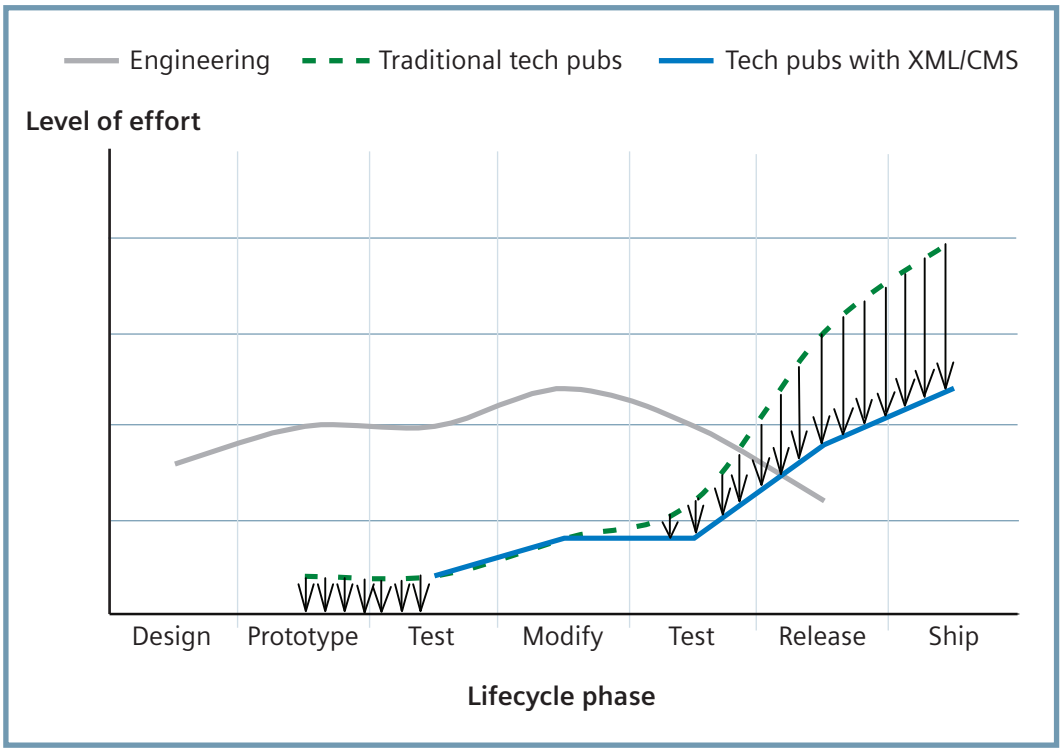
Role of technical publications

The need to support more complex product documentation has raised new challenges for today's technical publication groups, including the need to:

- Develop content to support a variety of documentation deliverables, including traditional hardcopy manuals, CD-based publications and online documents in both page-based and interactive formats
- Simultaneously publish documents in multiple languages to support today's global marketplace
- Release product and service documentation on time (i.e., when the product is ready to ship) while accommodating distributed teams that need access to current versions of the same content
- Accurately reflect late breaking engineering changes in the document's content, even when these changes occur within days of the product ship date



Traditional relationships between engineering design and technical publications in the product lifecycle



Influence of structured content on the relationship between technical publishing and engineering design

- Avoid "heroic" publication production efforts that incur overtime costs, which in turn jeopardize the product's profit margins

The accompanying diagram illustrates the traditional relationship between engineering design and technical publications during the product lifecycle.

Facing the "last hurdle"

Historically, technical publications groups have been viewed as the "last hurdle" in product development innovation (see appendix for more details). While publication services and deliverables remain on the critical path to product shipping, engineering design decisions and engineering change orders (ECOs) are still managed outside the technical publications' system and its related workflow.

To reduce the time lag between product release and the document ship date, most publication groups start document development early in the product design cycle. As a result, technical publication groups engage in significant rework as the product design is repeatedly refined.

In essence, technical publication is still largely a follow-on process. While providing a necessary function, technical publications groups are largely isolated from other disciplines in the product lifecycle. They invariably use separate automation systems and often rely on non-integrated processes to handle change notices that materialize late in the product lifecycle.

However, diverse development paths no longer offer sufficient justification for keeping engineering design and technical publications in separate computing environments. Today's common change management processes and the configuration management capabilities of PLM now provide companies with a holistic environment for managing both product and documentation development. This holistic environment is especially adept at reducing the risk of inconsistent or out-of-date technical documentation while shortening the length and complexity of the publications cycle.

▶ Commonality between product and documentation lifecycles

During the past two decades, significant technology improvements have enabled multiple disciplines in the product lifecycle to work together more effectively. Engineering data management and standards for metadata management have improved communications between engineering teams and shortened the engineering production cycle. Content technologies and their related standards have enabled publication groups to quickly and consistently produce higher volumes of technical documents – as well as facilitate content exchange with business partners and government oversight groups without the need for rework.

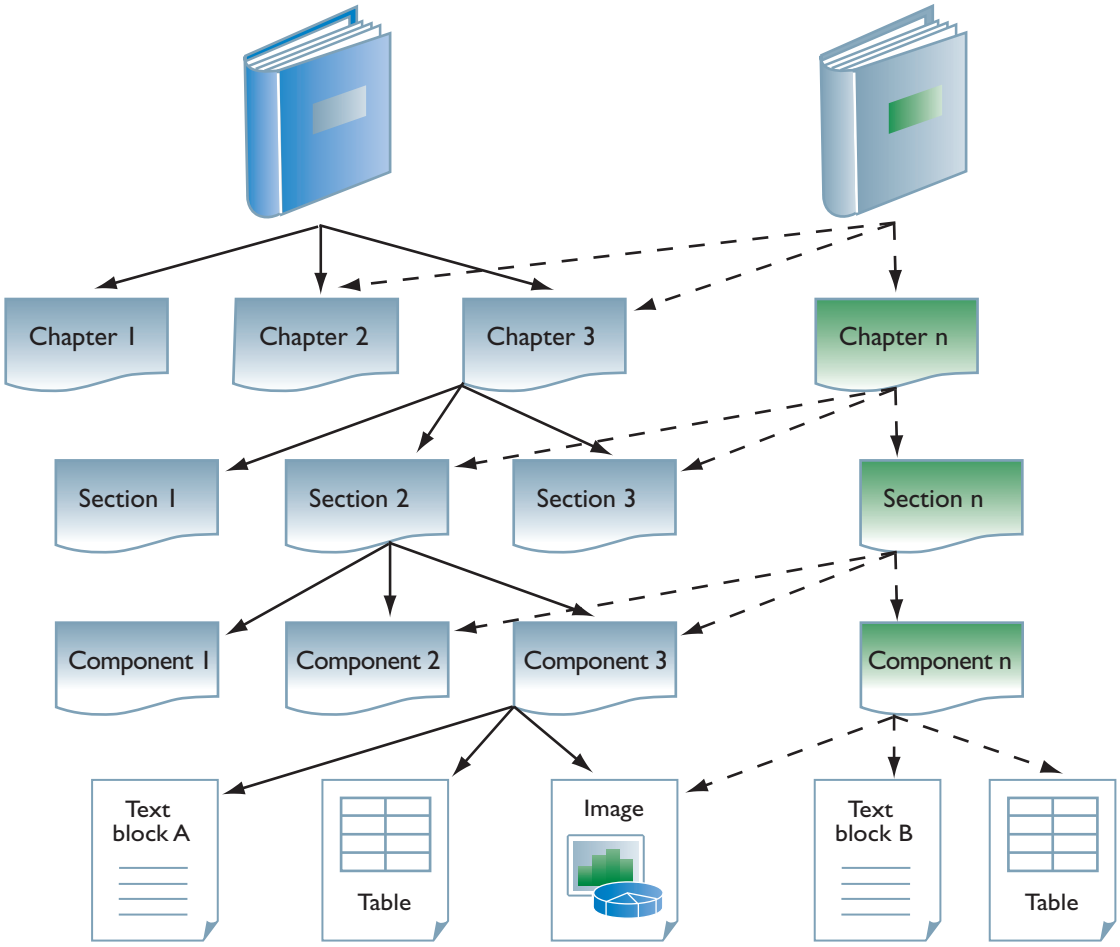
A close look at the publishing process and documentation lifecycle reveals similarities and dependencies with today's engineering processes and the product lifecycle. Technical publication groups have direct ties to the engineering design discipline in at least three ways:

- Source material
- Parts data and its related descriptive content
- Technical review/feedback process

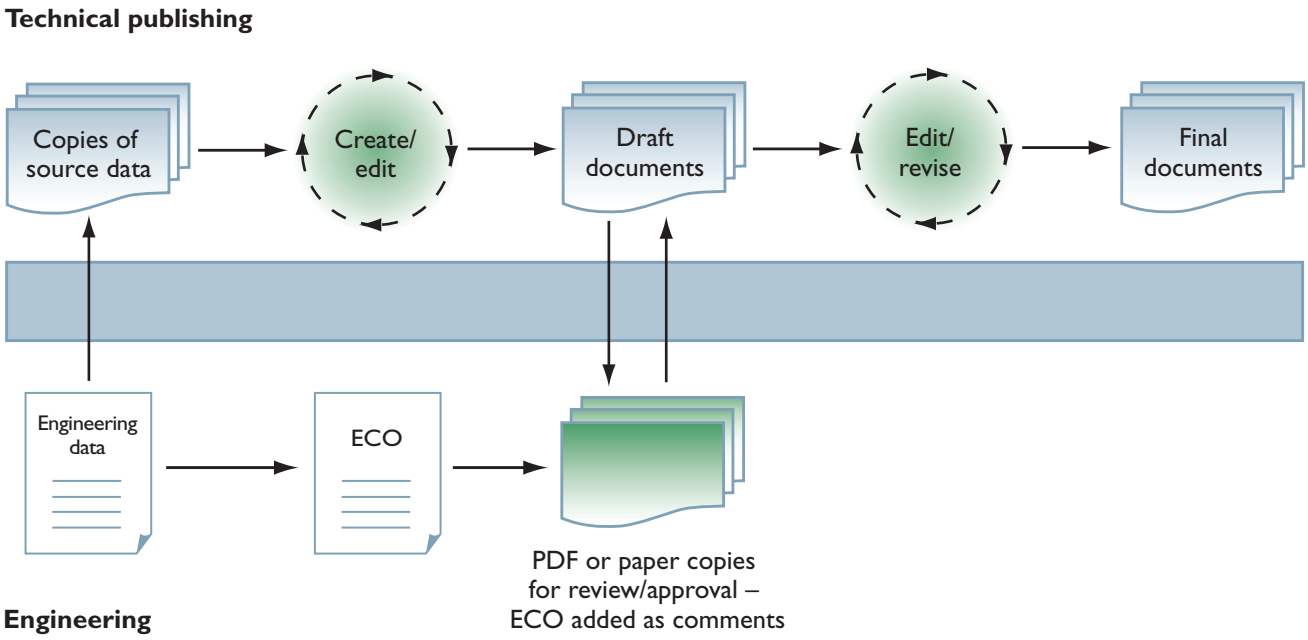
Functional system requirements

Technical writers deal with the same complex parts and assembly constructs as their engineering counterparts. A series of related technical publications is comprised of reusable "parts" (e.g., content and graphics) that can be assembled into documents, such as training materials, illustrated parts catalogs and operation manuals. By looking closely at these manuals, it is easy to see common graphics, operational procedures, and descriptive paragraphs.

Graphics often are used in the product's support documents as well as in documents that describe the product's related configurations. For example, the steps and graphics that describe a carburetor's maintenance procedure might also appear in the owner's manual for a particular model of coupe or sedan offered by the same automaker. To support these models, an engineer must be able to track the assemblies where the carburetor is used, while a technical writer must be able to track all documents where the carburetor is described.



Typical parts and assemblies for a technical document



Typical product information flow between engineering and technical publications groups

Expanding on this example, the coupe owner's manual might include a cover photo of the coupe, while the sedan's owner's manual might include a photo of the sedan. Each manual's text content will be different depending of whether the automobile in question is a two-door or four-door model. This configuration-sensitive content has metadata that classifies the content as being appropriate for a specific document. In essence, this metadata is applied to the content's "parts" to indicate when it should be added to the document's "assembly" (in much the same way that a physical product's parts and assemblies are managed in an engineering-driven PDM environment).

Both the engineering and technical publications environments use repository software to meet the product's state and lifecycle tracking/reporting requirements. Both environments need to manage access rights. Both need to track contributor changes to the parts and assemblies that they manage. Both need to manage risk and facilitate IT efficiency by reducing redundant or duplicate parts and providing a consistent product view.

In addition, both environments require the following key functionalities:

- Workflow/lifecycle management
- Change management
- Security controls, including International Traffic in Arms Regulations and Export Control (ITAR)
- Program execution management
- Configuration management
- Relationship management

In the best case, companies have traditionally provided these functionalities to their engineering design and technical publications groups through multiple repository applications – and in worst case, through directory structures and generic tracking mechanisms that are manually maintained.

Source material

Output from logistics, mechanical and software engineering groups usually provide the source materials used by technical writers to create document content. In most cases, source material is "thrown over the wall" to the technical publications group so it can be reworked. Draft documents are created in the technical publications environment, where source materials can be revised, rewritten or copied/pasted into complete documents. Subsequently, these drafts are thrown back over the wall for technical review by the engineering group.

Concurrently – and in many cases, unbeknownst to the technical publications group – design changes take place in the engineering environment. These changes are provided to technical publications in the form of new/revised content or ECO copies. In turn, the technical writing team updates the in-process documentation accordingly.

Relationship between data and content

There is a direct relationship between a product's parts/assemblies and the text/graphics that describe it within a technical document. This relationship is so close and complete that many publications group use part numbers as the metadata for classifying their content.

The engineering design changes that affect mechanical, electrical and software models/codes also drive the changes that are made against a technical document's content and graphics. In addition, while ECOs, their related discussions and their authorizations/approvals are tracked in the product engineering and development environment, technical publications groups have a similar requirement to monitor/manage the annotations and approvals associated with product's documentation content.

Technical review/feedback

Usually, the responsibility for validating/verifying the content of a technical document falls to the engineering staff. To handle the review/feedback functions associated with the validation/verification process, engineering groups generally work with page-based displays or hard copy documents.

Engineers sometimes review multiple configurations of the same document content in the same way they review designs that relate to multiple product configurations.

▶ Today's next major improvement opportunities

Technology advances and the implementation of structured authoring have facilitated significant production time and resource management improvements with respect to creating technical document content. Additional progress can be made by capitalizing on the common relationship between engineering design and technical publications. Specifically, the engineering design and technical publication groups can be effectively integrated by:

- Sharing information retained in multiple systems
- Re-using product and service content
- Automating functions and processes common to both disciplines

Shared systems

Both the engineering design environment and the technical publications environment have similar requirements for managing information in a sharable repository. In addition, engineering changes to the product design act as prompts for the technical publication group to initiate content changes against the product's related documentation.

Today's most innovative companies recognize that the publication process makes considerable contributions (in the form of both design and after-market documents) to the product lifecycle. Equally important, they realize that the "parts" and metadata that comprise these documents are directly related to the product's traditionally defined parts.

Many companies are making the move to consolidate and reduce the hardware, software and legacy customizations that account for a large portion of their overhead spend.[1]

By moving all of their product data into a single PLM environment and enabling entitled users from all disciplines to work with the same system, companies can reduce the number of systems they employ and cut their related overhead.

Similarly, a single PLM system improves cross-discipline communications while providing an integrated product definition that all product support teams can use to understand the impact of approved design changes. Equally important, take-to-market risk is reduced and time-to-market schedules are improved as companies no longer need to depend on isolated information silos, manually-maintained tracking mechanisms or individual knowledge workers who may leave a company's workforce vulnerable to their retirement.

Content re-use

By combining the shared information capabilities required by both engineering and technical publications, today's companies are positioned to move closer to a true solution for re-using their product-related intellectual assets.

Manufacturing companies typically value part-centric information. Information is developed as the part passes from conception through refinement and testing, deployment, maintenance and obsolescence. As this happens, that information is incorporated into technical documentation. In theory, any information regarding a part should have a relationship to the part. In practice, engineering and publication teams copy/paste and rewrite that information several times over during the life of the part.

For example, suppose Ed from Engineering has come up with the concept for an improved widget. Typically, he writes up a proposal that includes the rationale and requirements for a new assembly. Then, he gets approval to proceed. Given the nature of this process, Ed's company has text and product requirements that describe the assembly before Ed even begins the details of the design. Companies often store and manage this early information in a requirements management database (or lacking that, they might retain this information in an emacs or MS Word or Excel file).

At this point, Ed might begin the design by creating, combining or copying/pasting numerous versions of numerous CAD files. The design now has expanded to include parts that are contained or used by the assembly. For example, a larger engine component might contain or use Ed's new assembly. In addition, relationships between CAD/CAE files that support the manufacturing of the new part also can be created.

At this point, Larry from Logistics joins the product development process by creating maintenance, repair and operations information for the widget. This logistics support analysis record (LSAR) data, which is stored in yet another database, includes requirements and information about other factors, such as mean-time-between-failure (MTBF) and data used internally by the service organization. At the same time, Larry also writes instructional maintenance procedures that include each procedure's related steps. In essence, Larry's work describes or supports the part. The company maintains the relationship between

[1] According to the Gartner Group, consolidating as few as six small servers into a pair of large machines provides total cost of ownership savings of 35 to 40 percent (primarily achieved by reducing internal support costs). *Gartner Group Study K-LAN-308.*

Ed's work and Larry's work by using the same part number when referring to the information they produce. As a best practice, this information should be stored in the same system.

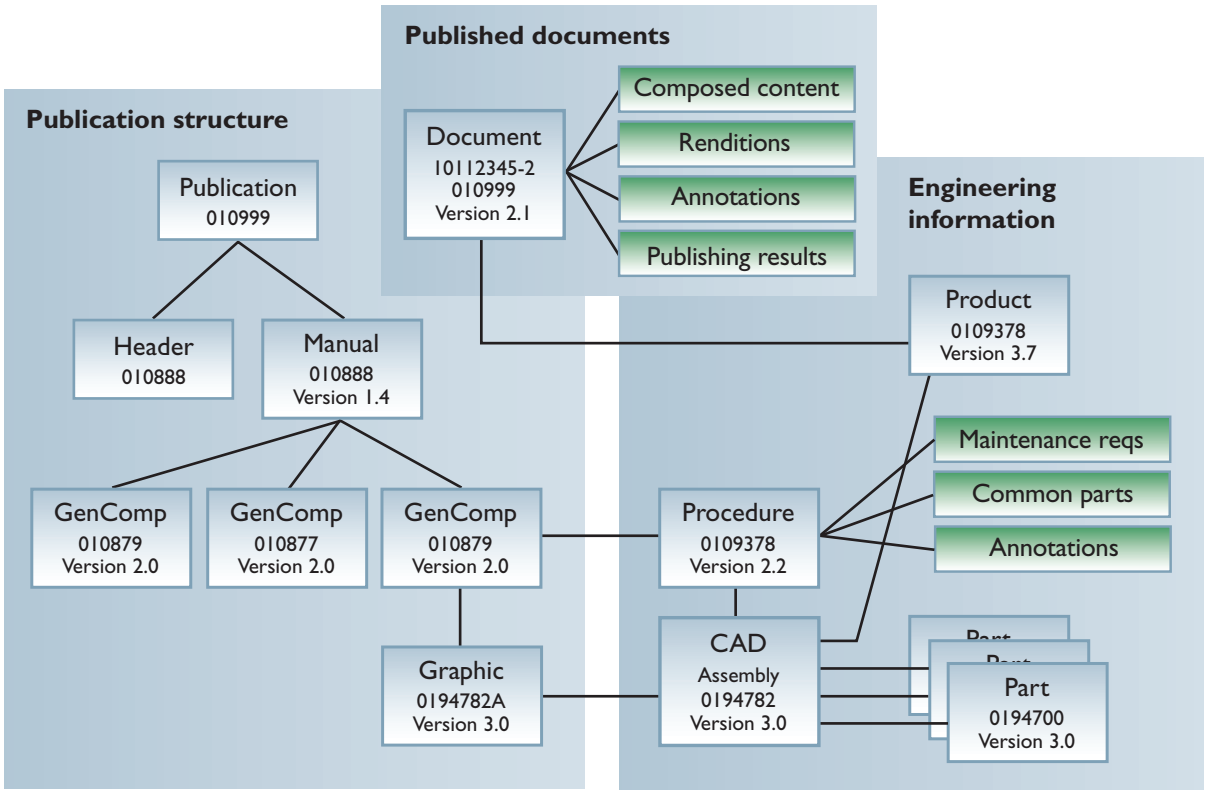
Now, Arthur from Technical Publications gets involved in the product lifecycle. He is responsible for creating all of the consumer and services documentation that needs to be shipped with Ed's assembly. Arthur requests access to all of the information that has been created to date by Ed and Larry, including engineering diagrams, the manufacturing BOM, and logistics information. At this point, Arthur might identify documentation from previous projects that relates to the parts in the assembly – and then start to work.

Typically, Arthur might copy/paste from other documents, rewrite content as needed, reorient certain illustrations, and edit procedural steps to accommodate both new users and experienced users. When this is done, he will save all of this work in the technical publication group's information repository. After this, Arthur might generate a draft PDF and submit it to Ed and Larry for their review/approval. In many cases, Arthur will be informed that design changes have been made, which will require revisions to his document.

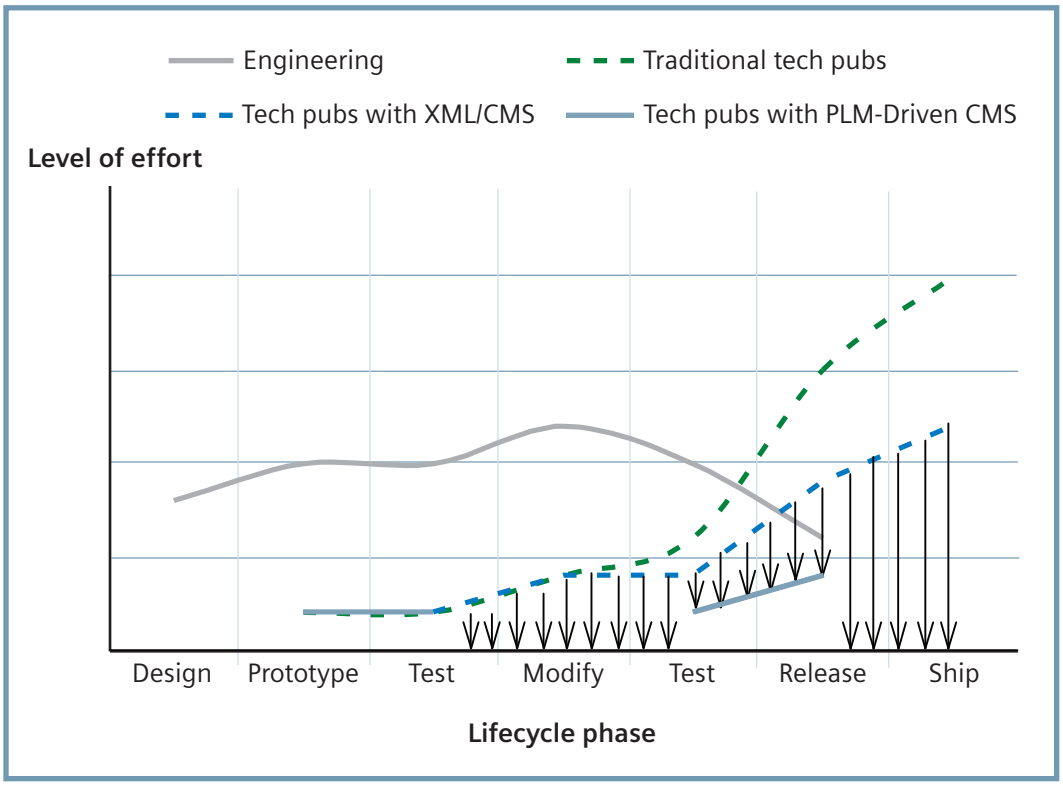
Before Arthur is done, he probably will have created several documentation versions that describe or support the part. Moreover, some of these versions might be a version of the work previously done by Ed and Larry. Since traditional publications approaches do not link the documentation to its source data, publications teams often add time, cost and even risk (in the form of inaccurate or out-of-date content) to the process of producing technical support documentation.

However, imagine how that workflow would change if engineering and technical publication team members were working in the same Teamcenter environment. Arthur would only need to change/edit content that directly relates to new or change product information. A single PLM-driven repository with today's latest illustration software would enable Arthur to identify – and in some cases, to programmatically produce – new graphics whenever the engineering group updates the source CAD/CAE files.

Equally important, Arthur would receive automatic notification about which specific part or which logistics information is affected by the design change. This notification capability is particularly valuable since it frees technical writers from having to research the design change and determine its impact manually. Instead, edits to the source data would be retained between ECOs.



Relationship of engineering information and publication structure to published document



Targeted technical publication efforts restricted to product and engineering changes

Functional and process automation

Besides defining the relationships between all parts and all of the documentation that supports them, Teamcenter facilitates new possibilities for automation, including:

- Automated change impact notification (based on where-used queries)
- Rapid documentation for product variants (configuration-driven document builds)
- Incorporation of other engineering source materials

Assemblies and system configurations use and re-use the same parts. Teamcenter enables engineers and technical writers to determine (by query) what parts have changed and where those parts are referenced. In turn, this capability allows companies to establish an automated process for notifying engineering groups and technical publication groups about the impact of engineering changes on supporting documentation.

Teamcenter solutions allow publication groups to relate documentation content directly to part and assembly configurations – and store this information with appropriate metadata. This enables them to assemble customer-specific publication content. Documents can be built on the basis of part and assembly options, language or any other criteria that is relevant to today's product management needs. These capabilities are especially valuable for facilitating the rapid publication of product variant documents.

Once the relationships between engineering data and technical documentation are established, companies can leverage Teamcenter to automatically incorporate other engineering source materials directly into documentation content. For example, part data can be imported directly into illustrated parts catalogs or maintenance procedures – with the extra bonus of programmatically identifying relevant illustrations immediately after engineering changes are identified.

Teamcenter's ability to relate change impact to notifications about specific follow-on publication activities is especially valuable in publication environments that require the localization of product content. Similarly, PLM-driven environments are adept at enabling publication groups to build documents that reflect customer-specific or release-specific configurations.

Moving forward

As long as product information is managed in discrete and disparate systems, companies face the risk of publishing inconsistent product documentation, delivering it late or incurring unnecessary costs across the publications cycle.

Shared environments that directly relate XML content instances to parts maximize their ability to re-use component designs and document topics. In these environments, information is created (i.e., entered) only once (at its source) and is re-used throughout the product lifecycle. The relationships between product parts and document content ensures the timely completion of critical path documentation (which is dependent on engineering data flow) without imposing additional overhead that would otherwise arise from engineering and publications teams working in different systems.

Under Teamcenter, the product definition is centrally managed by a repository that functions as the environment's authoritative information source. This central information source pertains to all product information regardless what configurations or lifecycle states (manufacturing or as-sold BOMs) are being referenced. The environment's shared workflow and data management schemes facilitate the implementation of systematic and repeatable processes.

By leveraging a PLM-driven environment, Teamcenter's content management solution enables companies to manage their technical documentation in the same system they use to manage their product data. This Teamcenter solution supports all DTDs while providing modular options for specialized functionality required in the DITA and S1000D standards. In addition, the ability to add direct relationships between product data and document content instances – and use dynamic drivers to update the latter based on changes to the former – is just around the corner.

▶ Appendix: A brief history introducing today's publications standards

Role of SGML

SGML – the Standard Generalized Markup Language – is both a language and an ISO standard for describing information embedded within a document. Today's highly popular HyperText Markup Language (HTML) is based on the SGML standard. SMGL was introduced in the mid-1980s by IBM technical writers who were struggling to manage their documentation more efficiently.

Essentially, SGML enables an author to create reusable chunks of content that subsequently can be assembled into a document on the basis of such factors as its product configuration, serial number or customer-related attributes. SGML manages document content in the same way that parts are managed relative to the higher-level structures of which they are comprised, such as assembles or product systems.

Almost immediately, publication groups and other user communities understood and accepted these "structured languages" as a means for improving document production. Similarly, the U.S. Department of Defense (DoD) leveraged structured languages to exchange information with subcontractors in the air transportation and telecommunications industries. Equally important, rules governing these structures and their content-related metadata were established in document type declarations (DTDs) that became the norm in these user communities.

Along these lines, the use of SGML was incorporated into the Computer Aided Logistics Support (CALS) initiative begun by the DoD in 1985. The DoD was primarily concerned with facilitating the integration of the digital information that pertained to weapons systems acquisition, design, manufacture and support. Originally established in the computer mainframe era, structured language initiatives were formalized as MIL-SDBK-59B, mandated in 1988, and affected maintenance documents, training materials and interactive electronic technical manuals (IETMs) – each of which had their own standards and DTDs.

In Europe, the Association Europeanne Des Constructeurs De Material Aerospatiale (AECMA) developed its own standards for IETMs (AECMA 1000D) and for illustrated parts catalog and provisioning (AECMA 2000M). Similarly, NATO Headquarters in Brussels began to address CALS in 1991.

The NATO CALS Organization was formed in 1994 with a memorandum of understanding between 12 NATO countries. Military and civilian personnel from the United Kingdom, Spain, France, Germany, Italy, Norway and the United States began serving full-time onsite at NATO Headquarters. These member nations performed studies, conducted workshops and developed projects that were managed by the NATO CALS Program Office at NATO Headquarters. The NATO CALS Data Model Version 3.0 grew out of these efforts and was published in May, 1998.

The NATO CALS Data Model is based on three standards:

- MIL-STD-1388 for logistical support analysis records (LSARs)
- ACEMA 1000D for IETMs
- AECMA 2000M for illustrated parts catalog and provisioning

The Air Transport Association of America (ATA) and telecommunication industry soon developed similar standards and DTDs. These conventions enabled subcontractors and their collaborators to organize and normalize documentation in accordance with common structures that facilitate easy interchange and communications, as well as improved production.

Recently, several industries began to accept and implement standards, such as S1000D in the aerospace and defense (A&D) and complex consumer packaged goods (CPG) sectors. These standards reflect functionality that has been derived from several historical standards, including editing and publishing conventions that support today's large systems. In addition, the DITA standard now is accepted by a growing number of small and mid-sized businesses – as well as by A&D and CPG companies – as modular or topical approach for creating and re-using publications content.

Improved documentation production

As defense contractors and large OEMs began implementing SGML/XML, software vendors started providing structured editing applications to support these standards. This synergy spurred the growth of structured content – which, in turn, led to accelerated documentation production times.

The use of desktop publishing applications (DTPs) had given authors both content and formatting controls, which ironically had impeded the content authoring process. More than 20 years of study has shown that writers who work in traditional publication environments spend over 60 percent of their time trying to manage their content and just 40 percent of their time actually writing.[2]

As writers began using the techniques of structured content (e.g., to create/edit documents in chunks), they significantly reduced the time they spent formatting bullets, margins and headings. Instead, they applied formatting rules after the document was assembled and just before it was published.

Other new technologies were adopted to support automated formatting for both electronic and hardcopy documents. These formatting techniques have constantly improved during the past two decades as industry and government regulation has increased to include compliance requirements for document structure, format, function-specific nomenclature and even page numbering schemes.

Growing acceptance

Throughout the late 1980s and early 1990s, SGML continued to gain business and government acceptance. However, the overhead associated with implementing newly added capabilities inhibited small to medium sized businesses from adopting its technology.

By the mid-to-late 1990s, eXtended Markup Language (XML) had become increasingly popular due to its internet authoring and exchange capabilities. This successor to SGML continues to enable users to author structured content (without focusing on format) while providing several significant advantages, including:

- Removing various SGML complexities
- Enabling the use of both DTDs and schemas for structure
- Facilitating a more modular standard

Where SGML contains rules for content classification, linking, and relationships within itself, XML provides the user with options by breaking its functionalities into additional related standards (such as Xpath and Xlink) – and formatting standards (such as XSLT/XSL-FO), which are written in easy-to-understand XML.

Since the late 1990s, the increased use of XML and related standards has led to new authoring and formatting applications, as well as the integration and development of modules for using XML standards and content imported in established publishing applications such as MS Word, Quark and InDesign.

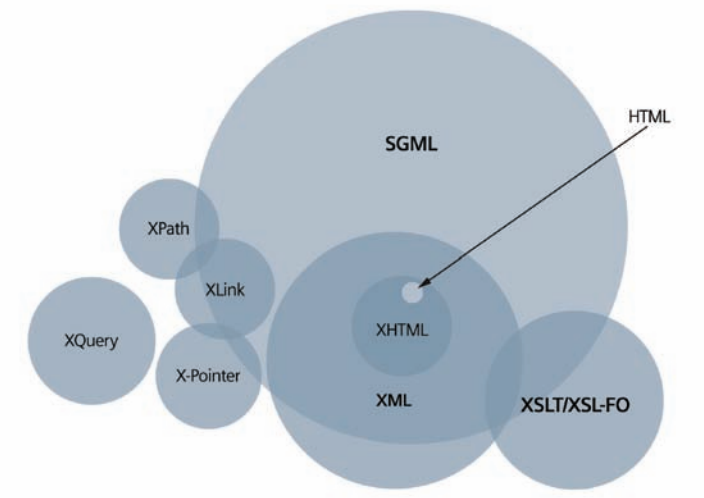
Traditional document management systems began to layer on XML-aware modules to support an XML document's various components. In addition, content management systems – comprised of repositories with bridges to editing and publishing applications – were developed to provide management and workflow functionality for the publishing process.

As XML software applications became more available, additional industries began to adopt XML. For example, few medical device and pharmaceutical companies worked with structured content before 2000. However, beginning in 2007, new drug application documents will have to be prepared for submittal in XML (per regulations specified by the U.S. Food and Drug Administration and its Asian and European counterparts).[3]

The increased availability and affordability of application software also has encouraged small to medium-sized businesses (with technical publication departments of five to ten team members) to adopt XML technology with the prospect of seeing a reasonable return on their investment.

[2] *ZAPTHINK, as reported by CNET NEW.COM* January 23, 2003: "Producers of content in the enterprise spend over 60 percent of their time locating, formatting and structuring content and just 40 percent of their time actually creating it."

[3] Since October 31, 2005, labeling submissions to CDER must appear in SPL format (see 21 CFR 14.50(l)(1)(i) and (l)5) and 314.71(b); see also Memorandum 32 to Docket Number 92S-0251). In addition, annual export submissions must contain labeling content in SPL format (see 21 CFR 314.81 (b)(2)(iii). *Guidance for Industry Indexing Structured Product Labeling,* U.S. Depart and Health and Human Services Food and Drug Administration, Center for Drug Evaluation and Research (CDER), March 2007.



Comparing conceptual relationship of XML and its related standards to SGML and HTML

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 5.5 million licensed seats and 51,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software's open enterprise solutions enable a world where organizations and their partners collaborate through Global Innovation Networks to deliver world-class products and services. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

SIEMENS

Division headquarters

United States
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
972 987 3000
Fax 972 987 3398

Regions

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
800 498 5351
Fax 972 987 3398

Europe
Norwich House Knoll Road
Camberley, Surrey
GU15 3SY
United Kingdom
44 (0) 1276 702000
Fax 44 (0) 1276 705150

Asia-Pacific
Suites 6804-8, 68/F Central Plaza
18 Harbour Road, Wan Chai
Hong Kong
852 2230 3333
Fax 852 2230 3210

8/08