[*Translating and the Computer 26*, November 2004 [London: Aslib, 2004]

Connecting Writers and Translators - XML-based Content Management of Product Documentation at Autodesk

Mirko Plitt
Autodesk
San Rafael, CA, USA / Neuchâtel, Switzerland

mirko.plitt@autodesk. com

Initial situation

Autodesk is a leading CAD[1] software company and its products include extensive documentation, mainly in the form of online help and printed manuals. Autodesk products are localised into up to twenty languages, and part of the product localisation is the translation of documentation that ships with the product.

For most of its products, Autodesk traditionally follows documentation creation and translation processes similar to those employed by many other software companies. The vast majority of the documentation is created in unstructured Adobe Framemaker, some also in Microsoft Word or directly in HTML. Single-sourcing of online help and printed documentation is typically achieved through complex conversion processes e. g. using WebWorks Publisher in conjunction with Framemaker and other tools developed in-house.

[1] Computer-Aided Design – Autodesk is particularly known for its AutoCAD software.

On the translation side, English source files are converted, translated/updated and converted back using the Trados Translator's WorkBench. The actual translation work and much of the related engineering (WebWorks conversion, online help compilation and print layout work) is typically outsourced to localisation agencies. Translation memories are managed on a per-project and per-language basis and are part of the final project deliveries from the agencies to Autodesk.

Review cycles as well as changes and updates imposed by the nature of software development add a significant amount of complexity to the above processes. Such cycles and changes often involve a variety of actors and although generally planned and scheduled, they can require last-minute modifications to the English and the translated documentation.

In such situations, many of the tasks that make up the creation and translation of the product documentation have to happen concurrently where intuitively they would be sequential. As in most real-world localisation scenarios, this ultimately leads to an ever decreasing offset between writing and translating, with the ultimate aim to remove the so-called "localisation delta".

Implications

The two project dimensions mainly affected by limitations of the above situation are time and cost, essentially because the different processes involved are not sufficiently robust to be automated without affecting quality.

More concretely, the manual activities that drive the cost and time in an actual project context, whilst clearly offering more potential for automation than the core writing and translation tasks, are layout work (DTP – Desktop Publishing), online help engineering and testing, and the numerous conversion steps.

These activities have to be repeated for each language – without demanding extensive linguistic knowledge – so that when looking at the entire cost structure of writing and localizing the documentation (rather than, as is often the case, looking at both separately) a language multiplier will turn these activities into drivers of the project cost, again despite the fact that they are not per se linguistic.

But these implications also affect the authoring side. A typical software development schedule requires weekly online help builds which in the traditional process can be time-consuming and error-prone. As a result, the numerous hand-offs between the different actors involved in the documentation creation (e.g. editors, indexers, testers, designers, marketing) require a considerable amount of manual coordination as do hand-offs to localisation.

At the same time, these activities are on the critical path for English and for translated deliveries so that any project acceleration will be limited by their constraints. In addition, these activities constitute major limitations for process scalability and the sharing of resources (people/know-how and tools).

Also, the localisation process imposes restrictions to the content creation process when late changes need to be made. Finally, the exchange of content between products, as well as the re-usage of content between different versions of a product, can be difficult under the conditions described above.

Content management vision

Starting from the above analysis a cross-functional team of authors and localisers at Autodesk developed a shared vision of a single system and common processes that would address writing and localisation needs likewise. The system would allow to reliably automate repetitive tasks whilst offering all the flexibility required to produce the high-quality product documentation that Autodesk's customers find useful and attractive.

The vision consisted of a central repository for all the content, not just English, a workflow engine that would be used by both writers and localisers, the possibility to automatically generate the desired output (help and print formats), and a fully integrated server-side translation memory. Furthermore, the system should as much as possible be built around standards, in particular in the area of XML[2].

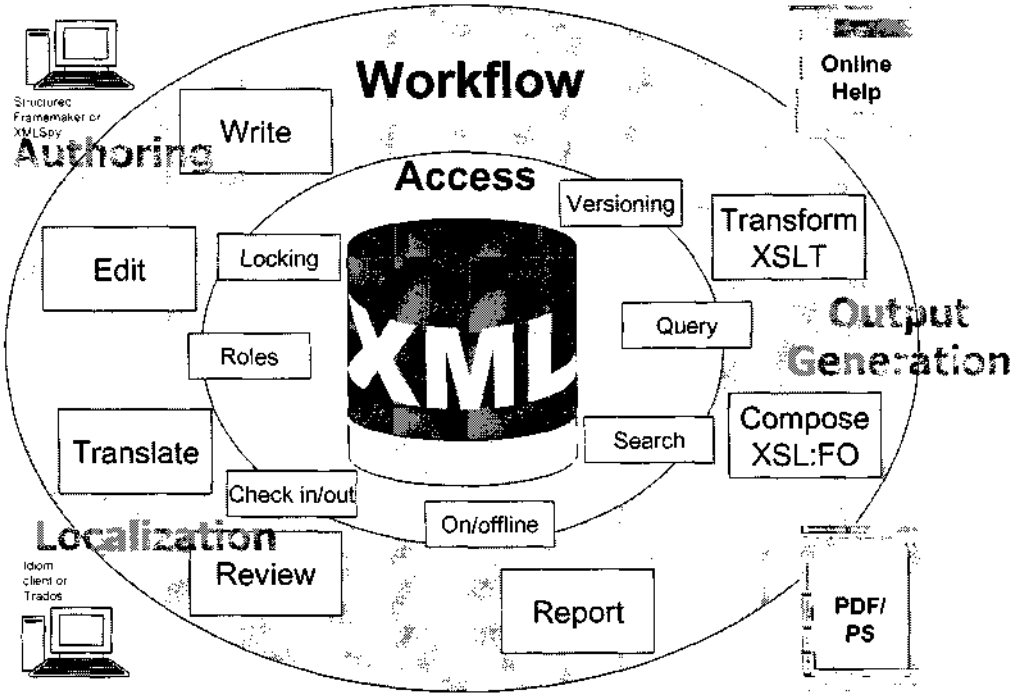
At the same time, the vision was that such a system would be sufficiently modular that those of Autodesk's many documentation processes proven to already be efficient would at the most need to be adapted but not have to undergo radical changes.

[2] eXtensible Markup Language - sort of an intelligent and efficient HTML.

Many aspects of this vision appear obvious in today's technology context, although not very few organisations seem actually to have implemented systems that are based on such principles. What is original about Autodesk's initial CMS[3] vision, however, is that content authoring and translation processes were considered as equal parts of a whole from the very start.

The proposed system was to cover the following areas:

- Content creation and update (interface to editing tools, layout-neutral authoring)
- Content storage (versioning, search and query, archiving, check-in/check-out)
- Workflow (creation, editing, approval, update, translation, publishing)
- Localisation (integration with translation memory databases, central terminology)
- Security (roles-based access rights)
- Reporting (budget, schedule, progress, metrics)
- Publishing (automated generation of printed, online and other formats)



[3] Content Management System - a rather vague label for a variety of more or less different concepts. Most systems that claim to do content management and which are actually being used in a real-world context are probably those designed to manage complex websites. Systems addressing more specifically documentation processes were traditionally referred to as document management systems but this term seems now to be falling out of favour.

Benefits

Autodesk's team of authors and localisers expected the following benefits from a Content Management System as described:

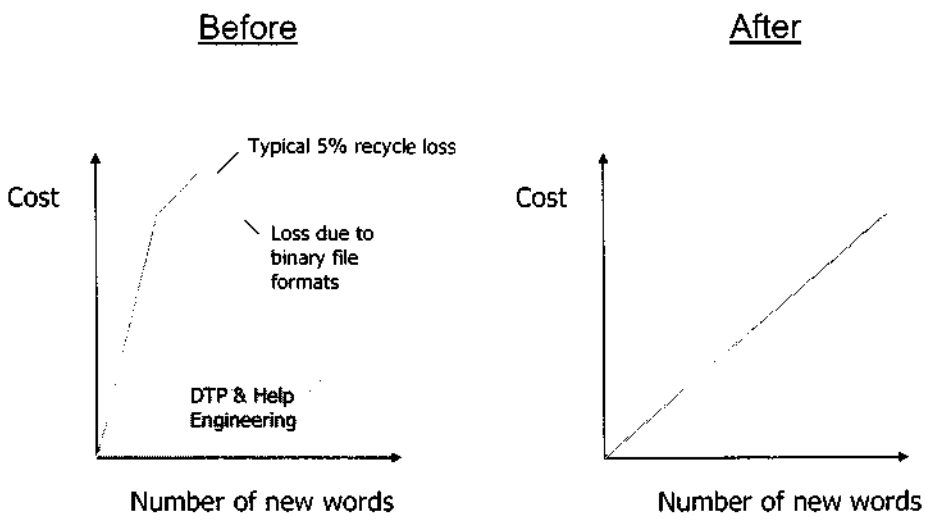
- Significant return on investment thanks to internal and external savings
- Shortened localisation delta, especially for last-minute changes which tend to threaten release dates
- Protection of intellectual property thanks to centralised data storage, access control, versioning, check-in/out and locking
- Process robustness and scalability thanks to the elimination of manual breakpoints in the file handling processes, self-documented deterministic workflows and Translation Memory management
- Further enhanced product quality thanks to improved document-internal and external consistency (across documents and with software)
- Possibility for ad-hoc documentation assembling
- Possibility for ad-hoc output format changes
- Easier sharing of resource (people and tools)
- Easier reporting on scope, progress and budget

Savings

Prior to launching into the CMS project an exhaustive analysis of documentation cost and savings potential was carried out which confirmed that even a pessimistic CMS implementation scenario involving a high initial investment would offer a compelling ROI (return on investment), essentially thanks to the following savings: elimination of manual layout of translated printed documentation, elimination of manual engineering and reduction of testing of localised online help.

In addition, the preliminary analysis showed that an increased efficiency of the translation process would generate savings by both reducing the project coordination effort and the pure translation scope.

Documentation localisation cost curve before CMS (traditional translation memory-based process) and after the CMS implementation (integrated server-side translation memory):



Project approach

At project start the overarching goal was set to reduce costs by streamlining and automating the entire product documentation life cycle from authoring through localisation.

Rather than developing an internal solution, Autodesk decided to look for a CMS vendor whose solution would satisfy the goal the cross-functional team had identified. Based on the vision, goal and expected benefits the team created an exhaustive list of selection criteria and use cases which were submitted to different vendors. Although the initial selection of vendors was probably not exhaustive, it became quickly clear that there were only very few solutions that corresponded to the vision of a common system for authoring and localisation.

The vendor to be selected had therefore to commit to the development of those aspects of the system that were insufficient for either half of the content life cycle. Because of the potential impact on the ROI, the vendor selected by Autodesk was the one with the strongest localisation approach, while offering a credible concept for the integration of authoring features.

A representative pilot project was carried out in parallel to the development of the missing features. Due to business constraints the pilot had to be a real production project, so the availability of a robust fail-over plan was important. As part of the pilot, metrics were gathered to confirm the assumptions regarding expected benefits and savings.

Autodesk's cross-divisional and cross-functional CMS team agreed on selection criteria during the summer of 2003 and the subsequent vendor selection process was completed in the autumn. The pilot project started shortly before the end of the year and was successfully completed in April 2004. In parallel to the pilot, a cross-divisional effort to convert existing documentation to XML was undertaken, and Autodesk-specific DTDs[4] were developed, largely inspired by the DITA[5] architecture.

The rollout to production of the final CMS began in June 2004, with a first major project starting production in July for which final English and localised project deliveries were produced beginning of October. By the end of October, the major part of the product documentation of the three Autodesk divisions involved in the CMS effort was migrated into the system, as well as the corresponding translation memories.

It appears that the timeline of Autodesk's CMS implementation was dramatically shorter than the one other companies implementing comparable content management solutions typically plan for. The main advantage of such aggressive rollout timing is the accelerated return on investment, but it also helps maintain the project momentum which in turn helps drive the change within a complex organisation.

Autodesk's system

Autodesk chose to start from a system which would generally be categorised as globalisation management system (GMS), i.e. a system that complements an already existing but monolingual content management system or content repository to manage the localisation workflow. However, there tend to be important overlaps between features offered by a mature GMS and a monolingual CMS, which is a factor that Autodesk decided to take advantage of.

[4] Document Type Definitions which describe the structure of a given XML instance.
[5] Darwin Information Typing Architecture for technical documentation. For an introductory overview, visit e. g. http://xml.coverpages.org/dita.html

The GMS Autodesk selected was Idiom WorldServer. The solution that Autodesk and Idiom then commonly designed was built onto the GMS base with its integrated translation memory, translation editor and workflow engine. The additional features included a specific editing interface for authors and a central content repository consisting of a native XML database (X-Hive/DB) for the entire content – English and translated. The editing interface consists of an interface to Adobe Framemaker which is essentially used as XML editor, with the addition of a cross-reference browser that allows authors to manage cross-references to any referenceable content within the XML repository. The XML content can be edited with any other editor, if desired, but only the Framemaker interface currently offers a non-manual way of selecting and editing such cross-references.

Furthermore, WorldServer's existing preview feature was enhanced to allow the generation of the delivery formats Autodesk products include, i. e. mainly HTML and compiled online help and different flavours of PDF[6]. In Autodesk's CMS implementation, the XSL[7] transformation of the XML content – retrieved from the native XML repository through XQuery queries – into the final delivery format is carried out using tools like Saxon, Xalan and XT, the rendering into PDF being done with RenderX XEP. All the output generation steps are fully automated, with some limitations for the help compilation process which is restricted by encoding issues around Microsoft's HTML help compiler.

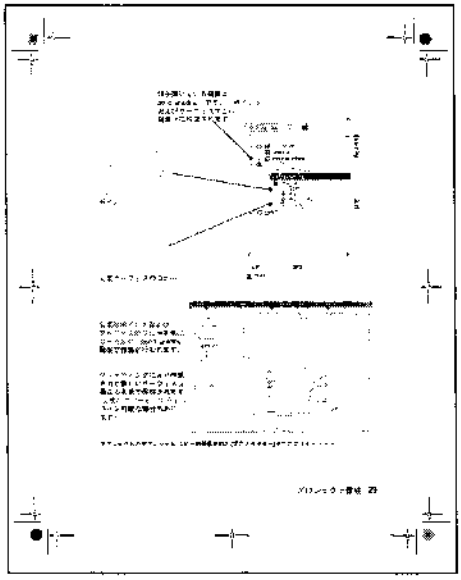
As mentioned above, the return on investment of the entire CMS implementation at Autodesk depended to a large extent on the ability to fully automate the generation of localised deliveries, parting from the XML content structured according to the DTDs developed by Autodesk divisions to meet their specific needs. The build processes had therefore to be global and capable of taking into account the language (or "locale") specific aspects of hyphenation, sorting, and more specifically the generation of indexes etc.

Developing XSL processes that satisfy these criteria is not trivial, but the quality of the compiled help files as well as the PDF generated with Autodesk's CMS speaks for itself.

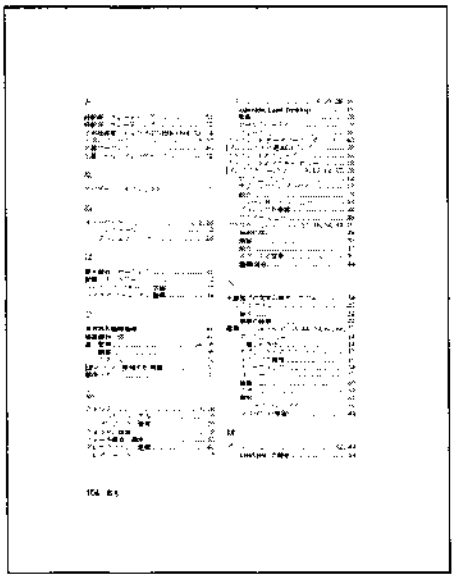
[6] Portable Document Format—the format read by the Acrobat Reader software which has also become a de-facto standard in the printing industry

[7] eXtensible Stylesheet Language – an XML "dialect" which allows to tranform (convert) a specific type of XML into other formats, typically a different type of XML or HTML.

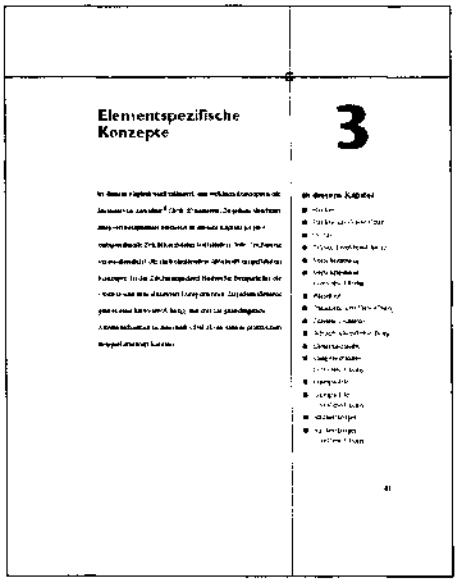
Output examples



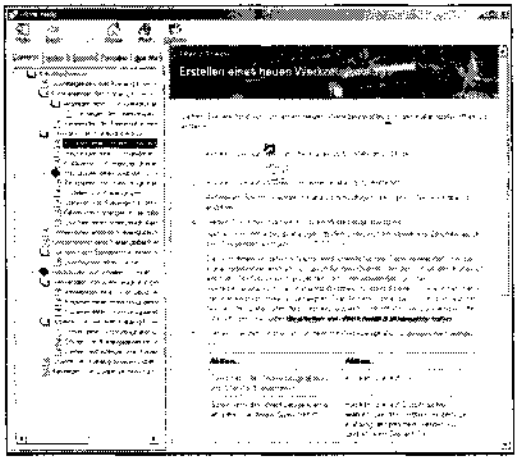
Japanese print PDF page with SVG[8]



Japanese online PDF index



German online PDF chapter title



German online help

[8] Scalable vector graphics – a type of XML for vector and mixed vector/raster graphics.

Challenges

The main interest of sharing Autodesk's CMS experience with others who may be starting from similar scenarios as Autodesk's initial situation and who also see the benefits of a solution integrating content creation and localisation aspects, is to increase awareness of the challenges such an effort implies.

The following is a list of some areas which are likely to require special attention in most implementations of a similar CMS solution:

- Change management and management of expectations. Processes in the context of technical documentation and its localisation seem not to have undergone radical changes for a relatively long period of time. Also, writing and translating are often perceived, and in many aspects rightfully so, as crafts, and not industrial and technical activities. This adds considerably to a natural resistance in people to technological change. To help alleviate this factor during the CMS rollout, Autodesk decided to wherever possible keep familiar interfaces, like Framemaker and the Trados translation client, and to design workflows that mimic pre-CMS work methods.
- DTD management. One advantage of a native XML database is its flexibility with regards to the structure of the XML it contains. There are no technical restrictions to the number of different DTDs that can co-exist within the same repository. However, the XML being at the heart of the CMS, changes to its structure affect all parts of the system, including translation memory coverage, the integration with Framemaker, the functionality of cross-references, the ability to share content and the XSL build processes - to name but a few. Autodesk had to introduce a DTD change control process to prevent unplanned and untested changes from breaking the system's functionality.
- Training. Autodesk's CMS team felt it was important to use, as much as possible, non-proprietary technology to make it easier to train people or find people familiar with those technologies. However, the number and complexity of new and even emerging technologies involved combined with production time constraints, an aggressive rollout schedule and sometimes resistance to change turn the thorough training of the key actors into a crucial factor of success.
- Ownership of global processes. Within the context of teams who are used to work to some extent in disconnected and isolated processes taking on ownership of processes involving other teams and mutual dependencies is not straightforward. This can easily become a critical factor for the development of global XSL and XSL-FO stylesheets.

Conclusion

After six months of working in production, the first conclusions of Autodesk's CMS implementation can be drawn.

The implementation effort turned out to be more demanding than was expected, mainly due to the combination of the pressure of an aggressive rollout schedule with the increased need for robust processes *around* the CMS.

However, Autodesk's experience with a single CMS for the entire product documentation life cycle from authoring to localisation clearly shows that the expectations in terms of benefits, including with regards to the expected savings, will be met.