CASE HISTORY

SUMMARY

- Describes the requirements and selection procedure for a documentation management system adopted by a large multinational manufacturing company
- Provides information about the system's features and operation, implementation, and benefits

A Successful Documentation Management System Using XML

MATS BROBERG

This case history describes a large manufacturing company's need for a documentation management system, the requirements that were defined for the system, and the solution that was selected to meet those needs. The article then provides information about the solution and its implementation.

BACKGROUND AND REQUIREMENTS DEFINITION

FLIR Systems was established in 1978 and is an industry leader in the area of civilian and military infrared cameras and programs for image analysis (see Figure 1), with more than 30,000 systems delivered. The Swedish division of FLIR Systems is located in Danderyd just outside of Stockholm and exports products to nearly 60 countries. Examples of typical applications include predictive maintenance, non-destructive testing, research and development, and civilian and military security monitoring.

Problems with previous documentation workflow

I was hired at FLIR Systems in February 1999 to manage production of end-user documentation for infrared cameras and image analysis programs, as well as purchasing and coordination of translations. At that time, the Swedish portion of FLIR Systems' product portfolio was limited, and production volumes were relatively small. Additionally, the company had relatively few requests from subsidiaries, agents, and customers for translation of documents into local languages. As a result, our needs could be handled by programs such as Adobe FrameMaker without much trouble. The manuals were written in English and were usually translated into only French, German, Spanish, and Italian.

Early in 2001, this situation changed drastically. As a result of strategic decisions regarding hardware and software within the group, the marketing division and the development division began to work in a significantly more modular manner. It became easier for product managers to specify new variations of a given product platform, so their interest in focusing on new customers and applications grew. Along with this new focus, the demands from subsidiaries and agents for translations into other languages also grew. Simultaneously, demands for significant reductions in lead-times for translations were also voiced.

However, these new trends within the company created extensive problems regarding user documentation. Along with an initial drastic increase in product variation with increasing demands for shortened lead times, we identified a number of problems that a future manual-processing system should solve.

- FrameMaker was not Unicode-compliant and offered support only for Western European (ISO Latin-1 character map) and Asian languages. Eastern European languages were not supported and required cumbersome workarounds. FrameMaker has supported Unicode for import/export of XML files since version 7.0, but not writing. Full Unicode compliance was regarded as essential.
- Processing of conditional text and images was implemented in a manner that did not support complex Boolean and nested conditions. Support for Boolean and nested conditions would minimize duplication of information.
- An excessively large part of the writing process was

Manuscript received 7 April 2004; revised 4 June 2004; accepted 5 June 2004.

Figure 1. ThermaCAM P65.

interrupted by desktop publishing work in Adobe FrameMaker.
- FrameMaker's proprietary file format was not considered "future-proof."
- Because of language-specific formatting requirements, we needed to finalize formatting of translated manuals at the translation agency. This step was significantly expensive.

Based on these problems and FLIR Systems' vision for the future, we decided to purchase a new system to manage user documentation.

Selection of a new documentation solution

The requirement specifications were partly a result of the problems with Adobe FrameMaker that we had perceived, and partly due to our ideas about how a documentation solution should function. The discussions revolved around the following points.

- To make the documentation future-proof, a non-proprietary data format (SGML or XML) should be used for source information.
- The primary output format should be print-ready PDF files, but expansion options for different output formats—HTML, WML, and so forth—should also supported.
- The documentation solution should, to the greatest degree possible, make use of programs with open source code.
- Unicode should be fully supported.
- The writing process should not be interrupted by the need to manipulate page breaks, paragraphs, font formatting, and so forth.
- Processing of conditional text and images should be visually clear and support complex Boolean and nested conditions
- The documentation solution should be specified and designed in such a way that no new technical writers would be needed despite the dramatically increased number of product variations and increased demand for translations.
- The formatting of covers, title pages, tables of contents, lists of figures, bodies, and indexes should be completely automated, able to be performed at night in batches.
- We should be able to format an individual manual or a batch (for example, all manuals within a product platform or all manuals in one language).
- All images, source texts, and configuration files should be subject to a version management system.
- Granularization of the source texts—that is, the breaking up of the documentation into smaller parts—should be supported to make re-use and translation easier.
- A user interface for initiating formatting and translations should be implemented.

During fall 2001 and spring 2002, 15 suppliers of documentation systems were invited to make a presentation. Some of these companies supplied documentation systems that had been completely developed internally, and others were resellers of internationally well-known system solutions. About one-third of the suppliers presented system solutions that could manage page-based technical documentation but that focused more on Web publishing. The others offered system solutions that were specifically aimed at page-based technical documentation with a focus on either user documentation or on catalog production. The bids of the various companies included adaptation of their systems to the specific requirements of FLIR Systems. The bids ranged from $20,000 to $800,000 USD.

Very early in the specification work, the question of the degree of content granularization became a core issue. A high degree of granularization would drastically increase translation savings and the ease of content re-use. But linguistic problems naturally accompany any translation system of this type. FLIR Systems' manuals contain both strict procedural sections and more descriptive sections, and, in certain cases, an expository style, as in the chapter about the history of IR technology. These require various language styles for obvious reasons. Excessively rigid granularization would not be conducive to stylistic differences in the granules.

In the final phase of the specification work, a decision was made to accept a degree of content granularization in which each XML file corresponds to one chapter in a master manual. Because each paragraph within an XML file

can be identified with a unique ID, this means in practice that the real degree of granularization lies at the paragraph level. We found, however, that re-use at the paragraph level seldom occurs.

It was also clear that the authoring environment looked very different in the various system solutions. Some suppliers offered traditional SGML/XML editors, while others offered editing environments in which the writing act did not appear to be the focus—that is, editors in which the writer would build the documentation by searching for existing and previously used words, sentences, and paragraphs in a database, rather than writing new paragraphs. Because the writing act was defined as a core issue during the system specification, we decided to focus on traditional SGML/XML editors.

IMPLEMENTATION OF THE DOCUMENTATION MANAGEMENT SYSTEM

After a comprehensive evaluation of the solutions for which bids were submitted, an analysis of the current needs of FLIR Systems, and an attempt to extrapolate current needs to reflect what we thought the documentation situation would look in 5 to 10 years, a contract for $50,000 USD, including 160 hours of consulting, was signed with Excosoft in mid-November 2002.

Excosoft is a smaller consulting and software development company with several decades of experience in system solutions for information processes. Their customers include FMV (the Swedish Defense Material Administration), Ericsson, Wärtsilä (a provider of ship power plants), Saab, and Flextronics (an electronics manufacturing services provider). Their product portfolio includes programs and system solutions for processing SGML- and XML-tagged information, as well as Web-based document- and message-processing systems.

We decided to introduce a fundamentally new way to manage user documentation during an extremely project-intensive period within FLIR Systems. At the same time that existing documentation needed updating and a number of new product variations were being produced, management communicated a strong desire to increase the number of languages into which the manuals would be translated. For these reasons, from an implementation point of view, there was no time when existing FrameMaker documents could be frozen for later conversion to XML in one procedure and then processed further in the documentation system delivered by Excosoft.

The solution was for Excosoft's consultants to implement and adapt the FLIR Documentation Management System (FLIR-DMS) gradually, with work continuing while manuals were being maintained and updated in both FrameMaker and the FLIR-DMS. Implementation was complicated by the need to initiate a comprehensive translation assignment of updated manuals that had already been converted to XML.

Despite these difficulties, we converted all technical documentation to XML by mid-December 2002 and initiated translation work just prior to the end of the month. At that point, large portions of the FLIR-DMS were still not fully developed but had reached a stage that made it possible to begin translation assignments critical to operations. For example, we had done no work yet on the stylesheet that would define the visual appearance of the manuals or on the import functionality that was supposed to automate processing of XML files when they were delivered by the translation agency. We completed those portions of the FLIR-DMS in early January 2003. Toward the end of January, FLIR-DMS was, in principle, completed, and deliveries of print-ready PDF files to the print shop could begin on 1 February—about two and a half months after FLIR Systems had signed the contract with Excosoft.

DESCRIPTION OF FLIR-DMS

FLIR Systems documentation management system consists of four distinct modules.

- Excosoft XML Client is the authoring environment in which the user works while creating or maintaining documentation.
- ExcoConf is the version management system that checks the version and version history of each XML file and each image, and from which the user checks out the documents that will be worked with in Excosoft XML Client.
- ExcoForm is the environment in which the user works when one or more manuals will be formatted into print-ready PDF files.
- ExcoTran is the environment in which the user works when a translation assignment will be initiated and XML files will be exported to the translation agency. It is also used to re-import finished XML files from the translation agency.

Authoring environment

Excosoft XML Client is the XML editor that comprises the authoring environment in FLIR-DMS. This powerful editor was developed by Excosoft for processing SGML or XML-tagged information. It does not differ significantly from the market's leading XML editors in regard to the number of functions, but is primarily characterized by two basic concepts—portals and expandable links.

The portal concept means that each user can have his or her own start page in Excosoft's XML Client, where all relevant information for that user is accessible. This information may include, for example, data such as a project document, links to reference information on the Internet or links to other internal or external files. Starting from the

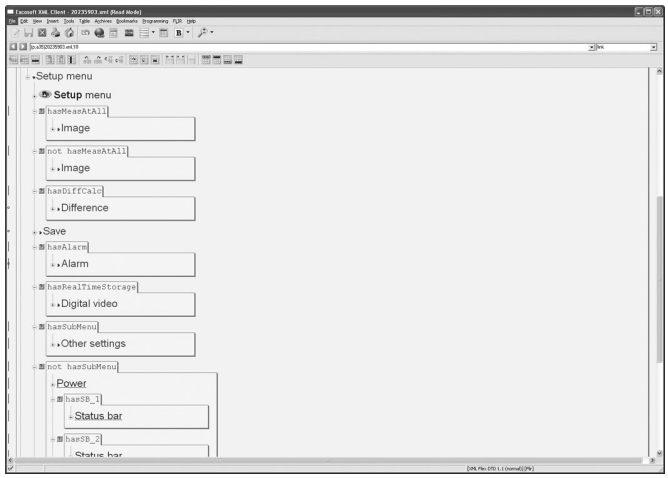

Figure 2. Screenshot of Excosoft XML Client, with sections in a manual, some of which are surrounded by conditions or nested conditions.

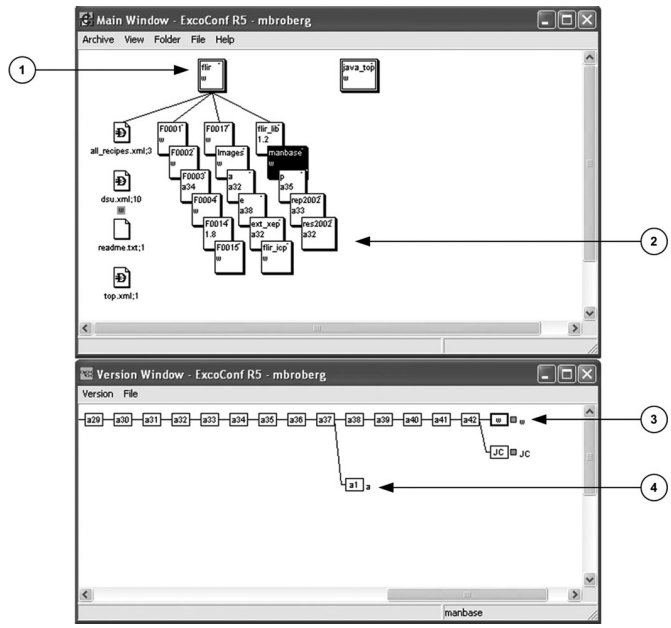

Figure 3. Version and Configuration Management System in ExcoConf showing 1) top parental folder; 2) child folders; 3) development line of the top parental folder; and 4) branches from the top parental folder development line.

portal, users can then navigate—using these expandable links—through the file structure of the version management system in the same way that they navigate on the Internet using hyperlinks. In the case of FLIR Systems, at the top level of the portal in the FLIR-DMS there are only three expandable links. By expanding these links, the author has access not only to the many thousands of XML and image files included in the inventory of manuals but also files for system maintenance, adaptation of menus and dialog boxes, style sheets for formatting the XML-tagged information into PDF, and so forth (see Figure 2).

Version and configuration management system

Both the XML and image files that are part of the stock of manuals and the system files that deal with adaptation of the system undergo version management in a graphic version management tool called ExcoConf. This version management tool is available in two different models—one using file-based version management and the other using version management in an SQL database. The former model was implemented for the FLIR-DMS and was used for 18 months. During the spring of 2004, the company converted to the SQL-based version management system.

A typical characteristic of ExcoConf is that files are not managed at the file level but at the folder level. The advantages of this approach become clear when the stock of files is very large. In the FLIR-DMS, for example, nearly 10,000 XML and image files undergo version and configuration management. Approximately two-thirds of these files have been added since the implementation of the FLIR-DMS, while the other one-third is document files that were converted from Adobe FrameMaker or image files that were imported from the structure of files included in previous documentation. By gathering files in logical clusters that often fall together with current projects within FLIR Systems, these 10,000 files are organized into 15–20 folders that undergo version management in various separate development lines. These folders in turn form a logical cluster that is included in the FLIR folder that has also undergone version management in its own development line. Each release in the top parental folder development line (the FLIR folder) defines exactly which release in the development lines of each subfolder is correct for that point in time. Figure 3 illustrates this hierarchy.

ExcoConf, as a version and configuration management system, is completely integrated functionally with Excosoft's XML Client. Without needing to check out and open documents, one can browse all files in ExcoConf from the main portal in Excosoft's XML Client. When users need to edit files, they are asked whether the file should be checked out. When the file is later closed or simply left by navigating to another file, the original file is checked in again.

Formatting environment

ExcoForm (see Figure 4) is used when XML-tagged information will be formatted to print-ready PDF files. In this

Figure 4. ExcoForm dialog box.

enviroment, the user can choose to format a specific manual or all manuals within an entire model group in one or more languages. There is also an option to format all manuals in the complete FLIR-DMS ("All models") or to define explicitly which manuals will be formatted ("Explicit"). In the latter case, a matrix is displayed in a new window where the desired manuals can be selected by checking them off.

ExcoForm also offers the user a number of other functions. By selecting "Include conditions," all conditionally processed text and images are included, and a frame around the conditional text or image—together with the conditional term—is printed out in the manual. This function is extremely useful when a manual must be proofread by a project group because it generates a manual that includes all permutations of function descriptions and images for that model group.

FLIR Systems camera manuals contain a very large number of screenshots from camera and PC programs, and these screenshots are included as illustrations in the manuals, where they are described and referred to. When a manual must be formatted to a specific language, these screenshots—together with other images—are made available to the formatting engine. When the user selects "Check screenshot translations," a text file generated after formatting gives a warning if the localized versions of these screenshots are missing.

Image file references in the XML file have previously been assigned attributes that identify whether certain screenshots contain information that can be localized—that is, screenshots that must also be created in other languages. Because each image file has a two-digit signal code in the file name that identifies the language for the image, and the configuration information for a specific manual defines in which languages the manual must be made available, the function can check whether one or more localized screenshots are missing by using a control in the file structure in ExcoConf. In this way, the text file becomes a powerful aid for the user to ensure that no screenshots are missing.

The "Difference" function is selected when manuals for updated translations must be generated for validation by FLIR's local agents and subsidiaries. When a translation update is implemented, the user imports the files from the translation agency to the FLIR-DMS, the user selects the "Difference" and "Include conditions" functions, and the manual is formatted to PDF. Blue borders are added around the new information at the same time. Agents and subsidiaries thereby get a clear indication of what must be verified and do not need to verify the entire manual.

If the user selects "Blue links," the table of contents, list of figures, cross references, and index words in the resulting PDF files are marked in blue and underlined in a way that many users associate with hyperlinks on the Internet. This type of PDF file is used as online help in FLIR Systems PC software. Links in PDF files are clickable even after normal formatting, but when they are marked in blue, the links are more obvious in an online situation.

For traceability and quality assurance, FLIR Systems' official PDF files are created only from the frozen folders. PDF files are created from folders in a working state only for testing, verification, or proofreading. To avoid misunderstandings about which are official manuals and which are working manuals, the notice "DRAFT—FOR INTERNAL USE ONLY" is regularly included in the footer, of working draft manuals. The function can be removed by deselecting "Watermark."

The ExcoForm dialog box also has two windows in which information is logged successively during formatting. The upper window logs the different steps that occur during formatting, such as inputting target documents, configuration files, and so forth. In the lower window, warnings and error messages are logged to signal the user when certain problems must be resolved.

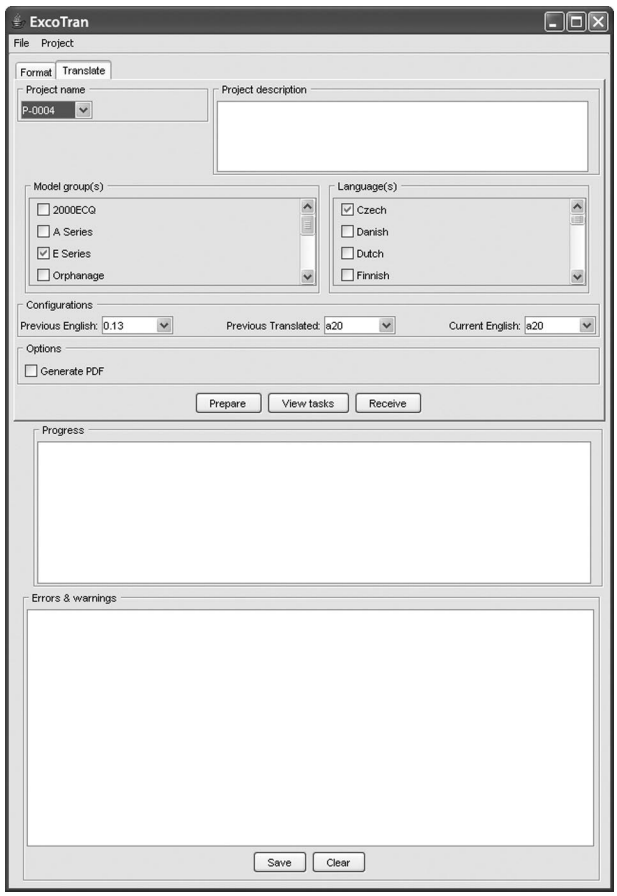

Figure 5. ExcoTran dialog box.

The error may result from an image file for which there is no image reference, no value specified for variable data, or the fact that the formatting engine does not find the correct hyphenation algorithm for a specific language. From this perspective, formatting from a working folder is an effective way for the user to verify that all relevant data is present before performing the final formatting. The logged error messages can be saved to the user's desktop as a text file and can thereby function as a checklist when the user prepares a comprehensive overnight formatting.

Translation environment

When a master manual is frozen and a decision has been made to translate this master manual into certain target languages, such projects will be processed via the ExcoTran dialog box for translation projects (see Figure 5). Updates of existing translations to ensure that they match a new version of a master manual are also processed using this dialog box.

In cases where no previous translations of a master manual exist, the translatable information flows through the following steps.

1. A new translation project is created by selecting "New" under the Project menu.
2. Details about the translation project are written in the "Project description" box.
3. Product family and target language are selected.
4. The revision letter for the most recent existing master manual in the source language is indicated in the "Current English" box.

After the user has taken these steps, a number of commands are automatically executed by the system.

1. A folder called "tranproj" ("translation project") is created, and the file that defines the user settings that have been chosen for the translation project is saved in that folder.
2. A second folder called "trandir" ("translation directory") is created for folders for respective target languages, as well as a delivery folder ("del") intended for the translation agency.
3. The system extracts the translatable information from the relevant files in ExcoConf and saves these extractions in the respective target language folder.

During the step when a translation project is created, a specific filtering action also occurs from the extractions that will be sent to the translation agency. The variables that are included in the source files, such as product name, company name, software strings, and so on, are protected from translation by providing them with a special XML tag ("notranslate"). Mathematical formulas are also protected with similar tags. When the translation agency has performed the translation, the agency returns a folder named "imp" (for "import"). This folder is placed at the same level as the translation project folder ("del"). Then the user starts ExcoTran and clicks the "Receive" button. An opposite flow is then initiated; the extractions are reintroduced into the respective source files, and these are copied back to ExcoConf.

The normal routine for FLIR Systems is that all manuals must be verified by a subsidiary or agents prior to release, and this requirement entails generating preliminary PDF files with the new information is highlighted after the XML files have been imported. If a subsidiary or an agent requests a change in a master manual, the correction is performed in a second step by the translation agency, and the XML files are imported once again.

However, for ongoing translation projects, translations usually already exist for the documentation of a specific product family. When a translation project is initiated, the system automatically determines whether a translation currently exists. If an existing translation is identified, a delta analysis is performed between the most recent version and the present

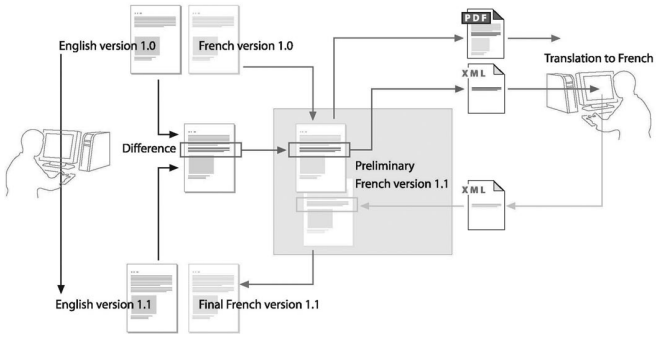

Figure 6. Delta translation workflow.

version using an algorithm. Figure 6 shows a typical flow.

The differences between the most recent master manual in the source language and the present version are identified. These differences may be, on the one hand, functional in character, such as deleted information, moved information, changed image references, changed conditions, and so forth. On the other hand, the differences may be linguistic in character, such as newly composed text.

When the XML structures in a master manual are identical between the source language and target language, the algorithm can handle differences of a functional character without requiring the intervention of the translation agency. A paragraph that has been deleted or moved in an English manual (source language) can therefore be deleted or moved automatically in a Spanish manual (target language). The same applies to image references, conditions, and so on.

If completely new text has been added in the source language, this text is introduced into the existing master manual in the target language. A PDF file of this composite manual is generated—containing, for example, both Spanish target language text and English source language text—and this file and the actual extracts that must be translated are sent to the translation agency.

When the translation is finished, the translated extracts are imported and placed in the correct locations in the respective master manual in the target language.

The reason this method of managing translations was implemented in the FLIR-DMS was that a significant portion of the ongoing work with user documentation at FLIR Systems involved functional revisions rather than linguistic revisions. Because the products are constructed in a modular manner, new products are continually being created with functions that have already been described. Because FLIR Systems' manuals are translated in as many as 17 languages, significant effort had been required to enter these functional revisions manually in all master manuals in all languages. As the system now works, the algorithm manages the functional revision work, while the translation agency manages the purely linguistic work.

SOLUTIONS TO SPECIAL PROBLEMS

It is important here to describe a few of the problems surrounding the documentation that the company, together with Excosoft, has solved through a number of special functions.

Naming conventions for XML files and image files

As mentioned previously in this article, approximately 10,000 XML and image files are processed in the version and configuration management system. Such a large stock of files requires a special naming convention in which one no longer describes the file with a descriptive file name but instead uses a system of incremental number series combined with certain types of signal codes. In the case of FLIR Systems, with documentation that will be translated to a substantial number of languages we crystallized a number of naming conventions early on.

Document files are signaled using the prefix *2*, naming is continued with a five-digit incremental number series *nnnnn*, and it ends with a two-digit signal code for language version. A typical name for a document file might thus be *20254303.xml* in which the prefix *2* indicates that it is a document file, the next five digits *02543* are the incremental number, and the signal code *03* indicates that the language is English.

Image files are signaled using the prefix *1*, naming is continued with a five-digit incremental number series *nnnnn* (separate from the number series for document files) and ends with a two-digit signal code for language version. A typical name for an image file might thus be *10546011.tif*, in which the prefix code 1 indicates that it is an image file, the next five digits *05460* are the incremental number, and the signal code *11* indicates that the language is Russian.

This naming convention has several advantages:

- The user does not need to create new file numbers for document or image files when the only difference from the existing document or image files is the language.
- The user can immediately identify the language version of a document or an image by looking at the file name.
- When the XML files come from translation, the image references can be changed simply by automatically changing the two-digit signal code using a script.
- Cross-references in the XML files that are sent for translation can be changed in the same way when they are returned.

When formatting a Spanish manual, for example,

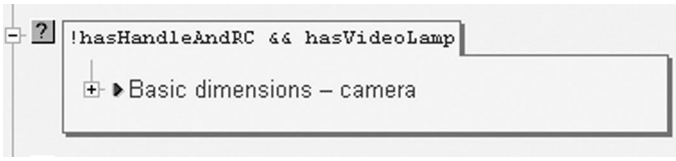

Figure 7. An example of a Boolean condition. The section "Basic dimensions—camera" will be included in the final manual if the both the condition "!hasHandleAndRC" (which means "the camera does not have a handle and a remote control") and the condition "hasVideoLamp" (which means "the camera has a video lamp") are defined as "true" in the configuration file for the manual in question.

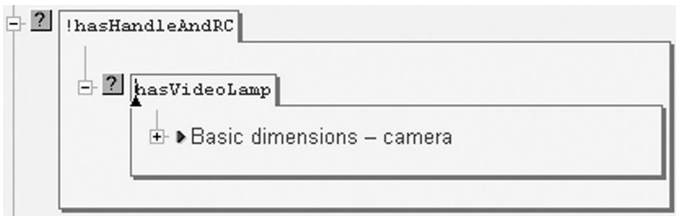

Figure 8. The same condition in a nested format.

built-in formatting logic causes the system—as a last resort—to refer to an English screenshot if it does not find a screenshot with the correct language signal code. This rule builds on the assumption that it is better for the customer to find a screenshot in the manual that is in the wrong language than for the screenshot to be missing completely.

Conditional text and image processing

FLIR Systems' cameras and PC programs contain a very large number of functions, and can be said to contain sub-quantities of all functions on a number of product platforms. A master manual in the FLIR-DMS for a specific product platform describes all possible functions and technical data that can exist for that product platform. Individual manuals for a respective product can be created by surrounding described functions or images in the FLIR-DMS with what is called a condition. If a product platform, for example, has a function that consists of a laser pointer—as in FLIR Systems' camera model ThermaCAM P65—the description of this laser pointer is surrounded in the master manual with the condition "hasLaserPointer." When the manual for the P65 must be formatted, the condition "hasLaserPointer" is checked against a special configuration file. The file specifies whether the condition "hasLaserPointer" is "true" or "false" for that model, and the system is thus able to determine whether the description should be included in the manual.

Conditions can also be managed as Boolean or nested (see Figure 7 and Figure 8). For example, if the writer is describing a product platform it may be that certain models have a handle with a video camera at the forward end, whereas other models have only a handle but no video camera. For obvious reasons, no cameras can be sold with only a video camera and no handle. If this situation is managed in a Boolean manner, the condition "hasHandle && hasVideoCamera" will surround the description for the video camera. If one instead manages the condition in a nested manner, the condition "hasVideoCamera" will be first in the description for the video camera, while outside this condition there will be another condition, "hasHandle." The description of the video camera in both of these cases will be included in the final manual if both the condition "hasHandle" and the condition "hasVideoCamera" have been defined as "true" in the configuration file.

Recipe files

When it is time to format a manual, either for a test, verification, and proofreading, or for official delivery to the print shop, subsidiary, or agents, the formatting is done using a configuration file, which the FLIR-DMS calls a recipe. Each manual has its own configuration file that contains four separate tables:

- Ingredients
- Features
- Variables
- Publication Numbers

The Ingredients table specifies the file name for the XML document in which all links to each XML chapter going into the master manual are found. The formatting engine uses this document to find the files that will be included in the formatting of a specific manual. This table also specifies the file name of the image—usually a product photograph—that will be on the cover of the manual.

The Features table contains all conditions that are found in the master manual. This table specifies "true" or "false" for each of the conditions. If the condition is "true," the corresponding information or image is included in the manual, while it is excluded if the condition is "false."

Each final manual contains a number of variables—that is, variable information—that will not be hard coded in the running text but will be checked from another location. For example, this may be a product name, certain technical data that is changed often, or a copyright date. The values for these variables are retrieved before formatting from the Variables table.

Each manual, including different target language versions of the same source language manual, has its own publication number. This number corresponds to the part number of other parts or products and is the manual's unique identifier for purchases, inventory, picking lists,

and so forth. The publication number is taken from a register and is specified in the Publication Numbers table. When a manual is formatted, this number is retrieved and is formatted as a number and as a barcode (Code 39) on the cover, as well as a number on the title page and in the page footers of the manual.

Input of PC and camera strings

The products that FLIR Systems manufactures contain a very large number of software strings—on the order of tens of thousands of units. The infrared cameras use internal programs that are comparable to more advanced varieties of the programs used in consumer digital cameras. The PC programs developed for image analysis also contain a large number of software strings. These software strings are found in dialog boxes, on menus, and on buttons and toolbars.

The traditional flow regarding translation of software strings and manuals was a serial flow. First the software strings were translated, and when these strings had been translated by the translation agency and verified by agents and subsidiaries, they were implemented in camera or PC programs, after which the manuals were translated. The instructions to the translators of the manuals included a notice that software strings (which were indicated in the manual using semibold fonts) were already translated and that the translations should be taken from the attached string document. With tens of thousands of software strings, however, it was impractical to perform quality assurance. Translators often translated software strings in manuals as if they were part of the running text without knowing or realizing that a verified translation of the string already existed. Thus, certain texts in the manual that referred to screenshots used incorrect terminology and did not correspond to what the user saw in a specific dialog box or menu. Another problem was that the serial translation flow took too much time.

To counteract these problems, a parallel flow was introduced for translations of new software strings and manuals. Each occurrence of a software string in a manual text was replaced by a dynamic variable. The variable for the English string *Image* could be, for example, *N0001.Image*. The prefix is a unique identifier that indicates from which string library the string has been taken. This prefix is necessary because the string *Image* may be translated differently depending on the context in which it is used. With a prefix that principally corresponds to namespace in the world of XML, *N0001.Image* can coexist with *N0002.Image*.

If the English string value for a variable is missing translations—something that always happens at the beginning of a translation cycle when translation of the software strings has not yet begun—the unique syntax (such as *N0001.Image*) is input instead into the manual that will be formatted as a reference for the translators. Because the latter part of the variable is the string itself in English (in this case *Image*) it will be easy for the translator to understand the significance of the variable to the surrounding text that must be translated.

This method of processing software strings offers the following important advantages.

- Translations of software strings and manuals can be performed in a parallel instead of serial manner, which saves time in a translation project.
- The translator never needs to differentiate between a software string in the manual and the running text.
- Because variables are protected when XML files are exported to the translation agency, the syntax of the variables is not disturbed, and the correct string values will be retrieved when the final official formatting is performed.

BENEFITS

Focus on the writing act

If the user has previously worked in a page-based program where the appearance of the document corresponds to that of the finished manual, it may seem somewhat strange to work in an XML editor which completely separates the document content from its final form. My experience, however, is that users become accustomed to the new environment very quickly and soon see its advantages. Writers can focus completely on the writing act and do not need to stop regularly to manipulate font and paragraph formatting, page breaks, image placement, and so on. The focus on the writing act seems to lead to a more consistent use of the language and a clearer arrangement of the material.

Traceability and quality assurance

Because all XML and image files undergo version management in ExcoConf, and their inherent relationships are defined in the different development lines, just about any historic version of one or more manuals can be traced. This fact is significant if the company needs to verify when a specific critical piece of information was introduced into the documentation. Because all versions of all files are saved, it is also possible for the user to re-use previous function descriptions when needed.

Benefits of a non-proprietary format

XML is a non-proprietary data format—that is, the format itself is not owned by an individual company. As a result, general interests rather than private interests steer development surrounding XML, and that fact is positive from a user perspective because there is a large quantity of software and associated standards that support the format. It is also an extremely compact format in terms of file size.

TABLE 1: TECHNICAL INFORMATION ABOUT FLIR-DMS

Version management	ExcoConf
XML editor	Excosoft XML Client
Pre-formatting	ExcoForm
XML parser	Xerces, from Apache Software Foundation
XSLT processor	Xalan, from Apache Software Foundation
XSL-FO formatting engine	XEP, from RenderX
Personnel required	One person manages all user documentation
Number of unique products managed	64
Number of manuals managed	More than 300 manuals (growing by approximately 20% per year)
Number of languages used	Up to 17 languages
Formatting speed	11,400 pages/hour
Number of page formattings per revision of all manuals for the two largest product families	Approximately 30,000–40,000 pages
Number of page formattings per year of all manuals for the two largest product families	Approximately 150,000–200,000 pages
Number of page formattings per year of all manuals for all product families	Approximately 250,000–300,000 pages
Number of manuals delivered per year	7000–8000

Lower internal costs

FLIR Systems' calculations indicate a savings of approximately 75% to 80% of the costs of managing the same quantity of information after conversion to XML. These savings can be attributed in part to the powerful functions that have been implemented in the FLIR-DMS, such as processing conditional text and images, and automatic formatting, and in part to the previously described workflow and focus on the writing act as a result of the implementation.

Lower external costs

With a largely automated translation flow, the external costs for translation have also decreased dramatically. A conservative estimate indicates a savings of approximately 25% to 30%.

Technical information

Table 1 provides information about the system and its performance. T**C**

MATS BROBERG is technical documentation manager at FLIR Systems, a global leader in the area of civil and military infrared cameras. He has worked at FLIR Systems since 1999 and has been responsible for changing the technical documentation and translation workflow to XML. His previous job experiences include a position as technical writer at Optovent AB, a medical technology company specializing in real-time respiratory rate and apnea monitoring, and as an independent consultant in graphic design, commercial printing, and publishing. He holds an MA degree from Uppsala University. Contact information: mabr@flir.se.