Deborah S. Ray
and Eric J. Ray
Editors

Maintenance Procedures for a Class of Warships: Structured Authoring and Content Management

William P. Hall

This column examines emerging technologies of interest to technical communicators to help them identify those that are worthy of further investigation. It is intended neither as an endorsement of any technology or product, nor as a recommendation to purchase. The opinions expressed by the column editors and author are their own and do not represent the views of the Society for Technical Communication or the Tenix ANZAC Ship Project. All URLs and site contents were verified at the time of writing.

Many technical writers are exploring tools that produce or work with XML/SGML structured text as possible real-world "structured information" solutions to improve author efficiency, to manage large amounts of content, and to retrieve and process information with speed and focus. Although some highly experienced people in the industry have argued strenuously that structured authoring and content management offer little of value to technical writers, others are already replacing word processing tools with ones that can readily parse and process semantic document content and are implementing structured content development processes.

In this case history, I describe how structured information has helped to solve real content development and management problems in the pivotal area of Integrated Logistics Support within Tenix's ANZAC Ship Project to build 10 ANZAC frigates for the Australian and New Zealand navies. I joined the ANZAC Ship Project just after the prime contract to supply the ships was signed, and I cut my teeth on the range of commercial documentation issues. For the last 8 years, I have been developing and maintaining our environment for authoring and managing *maintenance routines*—that is, user maintenance instructions to keep the ships in service at a high level of availability for combat duties.

Our initial authoring environment for maintenance routines used WordPerfect merge tables to structure text and used merge/macro processing to extract and format a variety of deliverables. Unique applications developed in this environment provided us with several wins, as we single-sourced more than 20 kinds of output documents and reports from one set of minimally formatted files. Even though this structured environment was very good in some respects, it had a number of fundamental limitations that became increasingly evident as the number of ship-sets of documents increased.

From 1995 through 1998, we reviewed many alternatives for replacing WordPerfect. Our most cost-effective choice proved to be a state-of-the-art generic authoring and content management system: Adobe FrameMaker+SGML combined with RMIT University's Structured Information Manager (SIM)—a native XML database and indexing system supported by a powerful Web server. Beginning in the second quarter of 1999, we successfully implemented this system for maintenance routines, and we delivered a complete class-set of documents from SIM in October 2000. We are currently extending SIM to cover new classes of documents and will seek to eventually bring most project life-cycle documents into the new environment.

Using SIM has already reduced some 10,000 ship-specific maintenance routines to around 2,000 for the whole class. Other gains include substantially reducing the revision time for individual documents and replacing yearly deliveries of complete ship-sets of documents with deliveries limited to net changes, cutting delivery requirements by more than 95 percent.

In this case history, I detail the following aspects:

- Typical documentation cycle and scope for aerospace and defense projects

- General problems and challenges in developing documentation for aerospace and defense projects
- Problems specific to the ANZAC ship project
- Document authoring environment and process in place prior to implementing the structured authoring system
- Interim authoring solutions explored
- Requirements outlined for the structured authoring system
- Project successes
- Success factors in implementing the structured authoring system

As you'll see, many issues Tenix faced would be found in any large project management situation, and some would apply to almost any technical writing environment, thus indicating that structured information could potentially be valuable in a number of technical writing environments.

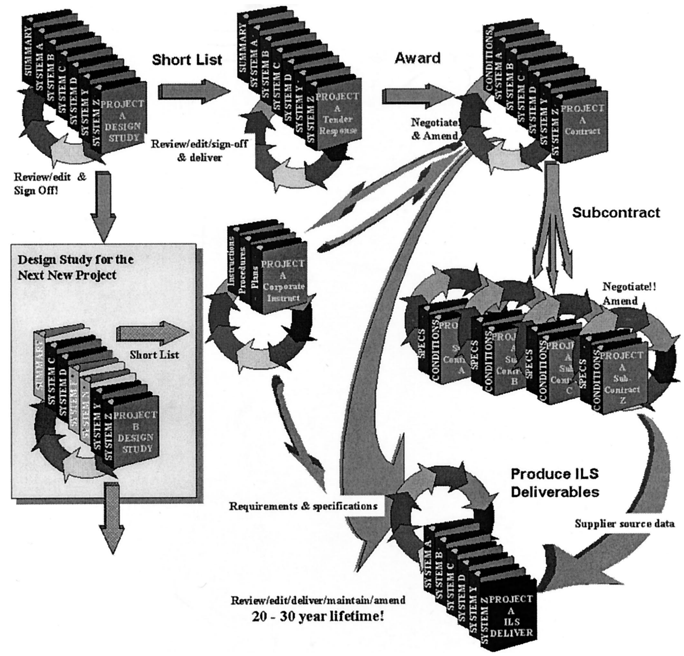

Figure 1. Information flow through the documentation cycle of a large aerospace or defense project. The development cycle results in literally hundreds of thousands of pages of specifications, contracts, and documentation, developed over several years.

THE DOCUMENTATION CYCLE AND SCOPE FOR AEROSPACE AND DEFENSE PROJECTS

Producing documentation for aerospace and defense systems is a major business. Figure 1 illustrates the cycle of document production used for most large and technically sophisticated projects. Document types authored during the cycle include design study proposals, tender and contract specifications, internal standards and procedures, and technical manuals and maintenance documentation needed to support the product through what may be a 25- to 50-year life-span in service. Authors of these documents include professional technical writers, legal experts, engineers, logistic support analysts, and skilled maintenance personnel.

For government-funded projects, documentation is produced in three major phases:

1. Pre-contract The client develops requirements, and suppliers bid to provide them.

- **Design studies** Just to develop the proposals for a large project requires potential suppliers to draft several versions of commercial and design documents over a several-year period as the client establishes in detail what capabilities they seek. The client may fund some of this work through design study contracts with several suppliers.
- **RFT** The client then drafts and issues a formal Request for Tenders (RFT) document package describing specific requirements, which may fill several to many looseleaf binders.
- **Tendering** Bidders have two to six months to prepare a truckload of response documents detailing how they will meet requirements and their prices for doing so.
- **Contract negotiation** Once the client selects a preferred tenderer, tender documents are extensively reworked over several more months of negotiation to turn tender documents into a contract to be signed by both parties. The final prime contract typically fills dozens to hundreds of looseleaf binders.

2. Detailed design and subcontract flowdown Once the prime contract is complete, the major documentation tasks begin. The selected supplier will then complete the detailed design and production plans, which, in turn, require dozens to hundreds of large- and medium-sized subcontracts to be finalized for the supply of major systems and components that go into the design.

- **Subcontracts** To form subcontracts, prime contract clauses are used as precedents to flow many of the terms and conditions (and associated risks) down to subcontracts with these lower-tier suppliers. It may take several months to more than a year to agree on major subcontracts before the design can be finalized. The prime contract and major subcontract documents remain live for amendment for at least the life of the production phase (10 or more years for a major project). Each subcontract normally fills five or more binders.
- **Contractual reporting** Defense contracts typically require the supplier to deliver a number of major documents during the detailed design and subcontracting phase to detail plans as they are finalized and to document progress against the plans.

3. Support Documentation

Authoring of support documentation can begin once the component supply contracts and the product design are finalized.

- **Equipment manuals** For a ship or airframe, each of several hundred separate systems and items of equipment comprising the individual hull or airframe and related software has to be documented with its own detailed technical manual. Such manuals are often several hundred to more than a thousand pages in length. System and equipment suppliers may provide acceptable generic manuals for their products, or manuals may have to be authored specifically for the project.
- **Planned maintenance documentation** Several thousand individual equipment-related maintenance and overhaul procedures are needed to cover planned maintenance activities. Because maintenance requirements depend crucially on the client's usage and upkeep requirements for the products, the prime contractor normally produces the maintenance documentation.
- **Operating manuals** Higher-level system and global operating manuals are then written to cover lower-level technical and maintenance documents.

With minor exceptions, the entire suite of technical documents must be completed during the 1- to 3-year initial production phase before the first product is delivered to the client. After first delivery and for as long as the product remains in production, documents must be revised to reflect engineering changes and in-service operational experience. Once the supply contract is completed, the client will then be responsible for maintaining documents through the additional 25- to 50-year in-service life of the product to reflect refits and system upgrades.

To date, most authors in the first two phases of the project have engineering, legal, and marketing experience. Some would argue that technical writers should do much of the actual writing. The major drawbacks to that approach are crushing deadlines and management concerns over additional delays resulting from involving technical writers in authoring, review, editing, and signoff cycles, which would still have to include engineering and legal staff.

THE GENERAL PROBLEMS AND CHALLENGES OF DEVELOPING AEROSPACE AND DEFENSE DOCUMENTATION

Aerospace and defense projects are very costly. Most of the project cost is allocated to designing and producing hardware and software; however, 5 to 10% of the total acquisition cost for the project may relate to documentation. This amount includes costs for drafting the client's own request for tender, prime and subcontractor costs for unsuccessful bids amortized across those won, and costs of drafting and maintaining technical and through-life support documentation. In other words, the documentation cost for a project costing the taxpayer a billion dollars may be as much as $50 to $100 million, a fact that makes defense documentation a big business in its own right.

To win bids in what is often a cost-competitive environment, major defense contractors need to minimize documentation costs while still delivering acceptable documentation products. Basically, this approach requires efficient authoring, management, and production of information through all stages in the documentation life cycle.

Figure 1 also indicates that the information content and document structures are often replicated many times across similar documents in one phase of the project cycle, and several to many times in different phases of the full life cycle. Several key issues relate to this picture:

- **Much of the text is redundant and could be reused several to many times during the history of a project.** Content may often be rewritten, even though it already exists somewhere else in corporate documentation, because it would take even longer to track down in a library or electronic filing system than it would take to rewrite it from scratch. Authors need help to discover and safely reuse information that already exists.

- **Many documents go through multiple cycles of rewriting and review, and thus require secure version and release controls.** Which of many copies held on the network and in the e-mail system is the valid and authoritative copy of a document at any particular point in time? Manual versioning methods backed up by controlling network access privileges are labor intensive, are often fallible, and may lead to crises in the delivery of critical documents against fixed deadlines.
- **Many kinds of documents have to be completed within very tight project deadlines, where failure to deliver on time can invalidate a bid or cause a project to be adversely terminated.** Tenders and bid responses are particularly punishing. Failure to complete critical documents by a tender deadline may invalidate a bid that costs many millions of dollars to develop. Even if delivered on time, if review time is cut, tender documents may contain incorrect information that could also have unfortunate cost or legal consequences. Even after winning a contract, failing to deliver critical support documents on time may entail major liquidated damages or even termination of the contract. (Even where physical products have been delivered, they ordinarily cannot be operated without operation and maintenance documentation).
- **Support documents must be accurate and intelligible.** Aerospace and defense systems are powerful, complicated, expensive, and potentially highly dangerous if improperly operated. Poor or erroneous documents can cause major accidents resulting in death of crew or innocent third parties. Aside from damage to reputation, the potential liability to a supplier may be catastrophic where it can be shown that poor documentation caused a lethal accident.
- **Word processing systems are inherently too fallible to meet the combined requirements of defense documentation.** Any graphically-based word processing system tends to be fallible, perhaps because of the great complexity. Microsoft Word is notoriously so—particularly where multiple authors work on the same documents over several months to years. Where documents are complex and time is critical, word processing failures can be particularly costly.
- **Many defense documents live a lot longer than proprietary word processing formats.** It may take two to three or more years to bring one project to contract, yet many of the bid documents from one project may still be useful for bids five and 10 years into the future. Contractual documents remain live for amendment and tracking for the life of a contract, which may be active for a decade or more. Product support documents remain live from the time they are authored until the last product goes out of service (multiple decades). Based on recent history, no proprietary word processing environment is likely to remain viable for more than about 5 years, let alone for the life of the product.

CHALLENGES SPECIFIC TO TENIX'S ANZAC SHIP PROJECT

Late in 1989, Tenix Defence Systems' corporate predecessor signed the largest defense contract in Australia's history (then worth $5 billion AUD) to design and build 10 ANZAC frigates: 8 for the Royal Australian Navy (RAN) plus another 2 for the Royal New Zealand Navy (RNZN). These 3,500-tonne ships are powered by twin diesel engines, plus a gas turbine and four diesel generator systems. Electronic systems include several kinds of radars and electronic countermeasures, as well as a variety of communications systems. Weapons systems include point defense missiles, a five-inch gun, torpedo tubes, chaff launchers, and an armed helicopter. Weapons and electronic systems are integrated and controlled through a sophisticated fire control system. Platform systems (for example, propulsion, steering, and basic service systems) are integrated and controlled by a machinery control system.

Several unusual aspects of this contract kept us more focused on the costs of producing and managing documentation than is the case with many defense projects funded by traditional cost-plus contracts.

- **Eighty percent of the work by value was to be completed in Australia or New Zealand.** This provision gave a strong incentive for new documentation to be produced in Australia rather than by overseas system or component suppliers.
- **Tenix was to train crews and provide all technical and operating documents for the ships and their weapons systems.** This provision gave the incentive for Tenix to provide central management for all training and related support docu-

mentation activities.

- **The contract price, including the cost of the full logistic support documentation and training package, was fixed at the time of signing (except for escalation clauses to cover inflation).** This provision meant that all documentation work had to be performed under very tight budgets. Any technological changes over the history of the project had to be carefully cost justified.
- **Well into the initial production phase of the project, the client decided to deploy a computer-based maintenance management system called AMPS, implemented by an independent supplier.** For planned maintenance documentation, this decision required a shift from a paper-based document delivery paradigm to delivery of relational data that could be managed in an electronic delivery environment that Tenix did not control.

THE ANZAC DOCUMENTATION PROJECT, AUTHORING ENVIRONMENT, AND GROWING PROBLEMS

The ANZAC planned maintenance documentation

By 1992–1993 the ANZAC ship design was mostly complete, and we had to start writing logistic support information. In addition to a variety of logistic support data sets, the contract initially required us to deliver four types of planned maintenance documents, all which had to be completed by the delivery of the first ship in March 1996.

- **Technical Maintenance Plans (TMPs)** TMPs are initially drafted early in the development of support documents to describe an overall maintenance philosophy for each system, the particular components in the system requiring maintenance, and the types and schedules of maintenance against each component. These maintenance plans form the basis for the other three types of paper documents required to cover the maintenance actions performed on the ships.
- **Equipment-level Maintenance Requirement Cards (MRCs)** An MRC details maintenance tasks to be completed against a specific item of equipment under particular circumstances (for example, every month, after 1,000 running hours, and so forth). The document is headed by a hierarchically structured table that lists reference materials, triggering conditions, labor and scheduling requirements, necessary tools, repair parts, materials, and assorted planning information required by the routine. The body consists of one or more numbered tasks to be performed at one time. Task sections detail the maintenance to be performed in hierarchical step paragraphs as modified by warnings, cautions, and notes. Steps may include tests, inspections, and required results, as well as tables and graphics as appropriate. Depending on the particular equipment being maintained and the complexity of the routine, an MRC could run from a single page to more than 50. Approximately 2,000 separate MRCs are required for each ship. Although the ANZAC ships are closely similar, they are built for two different countries, and no two ships are identical. In our original authoring environment, we would have to maintain more than 20,000 separate MRCs for the full class of 10 ships, and for four shore-based training facilities.
- **System-level Maintenance Index Pages (MIPs)** The MIP for each system being managed indexes MRCs for the equipment included by the frequency with which the MRCs are performed and summarizes scheduling requirements for each MRC. There are approximately 160 MIPs for each ship; some MIPs cover only a single MRC, while those for complex mechanical systems, such as the gas turbine, may index more than 40 separate routines.
- **Class level Technical Repair Specifications (TRSs)** While ships are in service, external repair contractors overhaul many system components (pumps, motors, and so forth). The technical repair specification is a scope-of-work document describing to a competent contractor how to remove and replace the component and what the contractor is required to do to return the component to a serviceable condition. There are approximately 600 TRSs applicable to the ANZAC Class as a whole, most 16 to 20 pages long. Because TRSs were not required for use until ships began to return for intermediate- and depot-level maintenance after delivery, authoring did not begin until the first ship-set of MRCs and MIPs was essentially complete.

Structured authoring in a word processing environment

In approaching the documentation requirements, we determined that our first major task was to complete a set of MRCs and MIPs for delivery with the first ship. Because of the volume of work involved and limited time available, a team of 10 logistic analysts authored initial drafts of the documents. Many of the authors were contracted through an organization specializing in logistic documentation, with precedence given to technical knowledge. Most were experienced maintenance supervisors for the kinds of systems included in the ships. None were trained technical writers. The authors were supported by illustrators and quality assurance specialists.

An early assignment was to develop methods and standards for authoring the MRCs and associated MIPs. Authoring had to begin before the delivery formats for the documents were finalized with the client. Because of this fact and because each of the two types of documents followed a consistent structure, it seemed prudent to start the work in structured WordPerfect merge tables. These contain discrete chunks of information in each table cell, with the information for each item on a separate row. Using merge tables allowed content development to start immediately, without having to produce finished documents in formats we would only have to change later.

To this end, we initially developed merge tables using DOS-based WordPerfect 5.1, later switching to Windows-based WordPerfect 5.2, which offered a nearly identical merge table feature. We collected records for all MRCs relating to particular MIPs into separate merge table files for each MIP. Merge/macro scripts could then be written to extract and assemble selected information from the records to form the MIP. As long as the required information was included in the individual MRC records, production of the MIP could be totally automated.

Although we chose to author maintenance routines in the merge table environment for relatively trivial reasons (primarily so that we could start writing without knowing what output format was required), this approach provided *major* benefits. In 1993–1994, the client added a contract requirement to deliver maintenance routines electronically to their new relational database-based shipboard maintenance managements system (called AMPS) and eliminated the requirement to deliver a planned major database of maintenance-related information. This change in delivery requirements replaced the need to provide a completely new database application with electronic delivery of MRC content in a format compatible with AMPS. Because we were already using the merge tables to structure the information, we were able to use that foundation as a stepping stone to meet the new requirements. The data structure previously developed to support automated production of the MIP documents was compatible with relational database structures. Also, most of the required data already existed in or could easily be added to our structures. It was much less costly to add one or two new data elements and a new extract process to our WordPerfect environment than it would have been to build, populate, and deliver the new database application.

These new requirements were easily met because our documents were already structured. In simple terms, XML or SGML tags, or in this case—Word Perfect merge field delimiters, identify equivalent blocks of text across all the MRC documents. Computers can parse such structured documents to select, extract, and rearrange particular elements of information to produce new documents. Selected elements of information can also be processed against equivalent kinds of information in relational databases. In our case, we both validated information held in the MRC records against master data held in a Tenix relational database, and output comma delimited tables from our standard MRC structure that mapped directly into the relational tables of our client's AMPS system.

By the time Ship 1 was delivered, more than 20 different document types, reports, and electronic deliverables had been "single-sourced" from the same set of merge table files. Merge-macro routines validated critical data items against external master files and databases. These automatic processes greatly reduced the amount of author time that would have otherwise been required to validate text and assemble reports.

In the end, our ability to parse and reassemble text electronically to deliver the range of formats required by the AMPS system eliminated the need to deliver MRC and MIP documents to the client on paper. Around the time Ship 3 was delivered, the client dropped the requirement to deliver MRCs and MIPs on paper.

Although the original labor budgets were far from precise, structured authoring in the WordPerfect environment probably reduced our labor requirement to author and deliver MRC and MIP documents for Ship 1 by 25 to 50%. Had we followed a paper-based paradigm of maintaining each MRC and MIP as a separate document, these deliveries would have required additional staff over at least two years of work to complete the various extracts and validations. Moreover, we probably could not have coped with the client's desire to have data delivered in electronic formats that could be imported directly into their AMPS system.

Growing problems using the word processor-based authoring environment

Although we used WordPerfect in a very structured way, our implementation remained at its core a format-oriented word processing system. This fact caused increasingly severe problems as the volume of documents grew. What was an excellent interim solution for the first ship-set of documents showed its limitations in several areas as we started preparing documents for the second ship and navy.

"Flat" tables were highly redundant data structures, labor intensive, and fallible to author
Although our WordPerfect system could output a set of comma-delimited tables able to be imported directly into AMPS's relational database, the merge table structure remained totally "flat" (that is, nonrelational). As a result, all data required to be included in an output document had to be represented in each record, with no capability to normalize (that is, eliminate redundancy by reusing information in multiple places) the data structure, as can be done in a relational database environment. This limitation resulted in many errors caused by having to enter what could be several fields' worth of totally redundant information across many records and files whenever commonly used text elements were changed.

Configuration management difficulties were increased by the large volumes of redundant data and large number of files The ships (and associated maintenance documents) are identical in most respects; however, in the flat file environment, a complete set of maintenance documents was needed for each new ship to manage minor configuration differences between ships and the frequent but trivial "language" differences between the different navies. Each new ship added another 2,000 MRCs that had to be copied, amended, and managed.

Change management difficulties were multiplied by the volume of redundant data across MRC records, MIP files, and ship-sets of documents Changes impacting MRCs (such as manufacturer-mandated changes to part numbers, new test equipment, and so on) had to be propagated across all ship-sets of documents. WordPerfect provided no help. An Excel table structure was devised to track the changes, but the process of managing and implementing even simple changes was totally manual and unavoidably fallible for the often large number of individual documents that had to be opened, updated, and versioned.

The word processing environment provided limited facilities to control author entries Word processing systems provide no innate capabilities to control what authors enter at any particular point in the document. In our outputs, merge/macro applications performed complex parsing operations, and authors often entered inappropriate or incorrect formatting codes or content that crashed the merge process. To make things even worse, there was no cost-effective way to detect and fix such problems without actually trying to produce the output, which greatly increased the time and labor requirements needed to produce the deliverables. Consequently, delivery processing required several weeks, in turn creating further difficulties maintaining synchronicity between MRC deliverables and other logistics and engineering data deliveries.

The WordPerfect applications were becoming dangerously obsolete Our authoring and delivery applications were built on WordPerfect versions 5.1 (DOS) and 5.2 available in 1993. For several reasons, no attempt had been made to carry out the extensive analysis and redevelopment process that would be needed to maintain compatibility with WordPerfect's newer versions.

WordPerfect knowledge was a potential single point of failure for ensuring delivery of contractually required data Since developing them, I was the only person within Tenix who understood WordPerfect's merge/macro applications well enough to maintain or extend them. Budget limitations dictated that my knowledge was never duplicated. Because the WordPerfect languages were poorly documented and buggy, my hard-earned practical experience remained a critical single point of failure. The company risked being unable to deliver maintenance routines for new ships if my trouble-shooting skills became unavailable for any reason.

Because of the range and complexity of the problems with the maintenance routines, we believed that any generic application able to solve these problems would probably also have the capacity to manage the whole range of documentation requirements.

THE ROAD TO A STRUCTURED CONTENT-MANAGEMENT SOLUTION

In exploring solutions to these growing problems, we initially attempted to solve these and other problems with "cheap" technology and inhouse skills, but then eventually developed requirements for and selected an SGML/XML-based content management system.

Before exploring purely SGML/XML-based content management systems, we experimented with a variety of technologies (some of which used SGML as well). Ultimately, these other approaches did not meet our authoring needs.

SGML and file management

We implemented an inexpensive file management system to control the TRS documents and gained benefits from working in a trouble-free SGML environment that authors found easy to use; however, documents were still managed as flat files without any ability to reuse elements or manage content effectively (as in a relational database), which would be required for any practical solution for the maintenance routines. (An experimental database constructed with an Omnimark script after most of the TRS authoring was complete showed that even in these stand-alone class-level equipment-specific documents, over 50% of the text was redundant across the suite of documents.)

Relational databases

Relational databases eliminated the problems associated with flat files and allowed data to be "normalized" across a number of different tables to eliminate multiple entries and the associated problems of that redundancy. In a normalized relational database, a given data element may be used many times in a variety of documents, but entered and maintained at a single point in a database.

As documentation provider for new systems being introduced to another class of RAN ships, Tenix had the opportunity to develop a new authoring environment for maintenance requirement type documents. SGML-based database solutions were still seen as too costly, so using our default applications, I designed a relationally structured system using Microsoft Access. The document structure was first modeled in an SGML Document Type Definition (DTD) and then mapped to a relational model. Data was normalized down to the level of reusable paragraphs in the procedure text. Authoring more than 100 MRCs in Access proved the concept; however, the experiment was ultimately unsuccessful from the authors' points of view because of the time required to process the large number of tables and the fragility of Microsoft applications when pushed to their limits.

Other solutions

We also priced several other alternatives for moving the ANZAC MRC data and associated applications into a newer environment, including rehosting into Microsoft Word, developing a Visual Basic application using Access tables, and developing an Oracle-based solution. Even though we had already completed all the necessary analytical work to define exactly what was required, none of the solutions could be implemented for less than several hundred thousand dollars (including data conversion costs).

The move toward an SGML-based solution

By 1998, we had eliminated all reasonable alternatives to an SGML/XML-based content management system. Tenix management provided funding under a research and development budget to study SGML technology, to price a content management solution, and to develop a detailed business case for implementation. To ensure that our selection process was of a high quality, two suppliers helped us to develop our requirements specifications, and we hired content management experts from the Text Information Management Group from CSIRO's Mathematical and Information Sciences Division to review our Request for Quotation processes and technical evaluations.

Our major requirements for a content management solution included the following:

- Convert data to application independent standard format
- Resolve the following issues relating to inconsistent data:
  - Validate specified elements against source masters
  - Share and re-use elements (write once, use many times)
- Produce required deliverables (documents, extracts, electronic transfer files)
- Manage document configuration and release against ship configuration and engineering change orders
- Provide release and version control
- Manage and track reasons for document changes
- Manage source references and determine change impacts
- Manage and track workflows
- Manage and reuse elements of content
- Manage dual languages at the element level
- Provide content indexing and query facilities
- Provide for minimal modifications to third-party software
- Provide user training and develop content administration skills
- Provide low cost maintenance and extension

A major underlying issue in our weighting of the various comparisons was that we sought a core technology that could eventually be extended to cover the full range of long-lived corporate documentation.

With these requirements in place, we searched the Web exhaustively to identify commercial systems that might meet these requirements. Two rounds of quotation were completed before we finalized our selection. RMIT University's Structured Information Manager (SIM) emerged from the second round as a leading contender, albeit with a couple of question marks.

SIM's potential

As we explored SIM in detail, we discovered that RMIT had a U.S. vendor (currently SAIC) and that one of the largest SIM users was (and still is) a U.S. defense security and intelligence agency that would have had a very strong preference for U.S. products.

SIM is actually a very powerful native XML database developed specifically to index, manage, and retrieve structured texts, combined with associated Web and supporting servers, and a powerful object-oriented programming and script-processing language called Ace. The whole SIM application was developed from the Ace library and then optimized to be very fast and scalable to multi-hundred gigabyte databases that could index more than 50 MB of text per hour (by now it may have been tested to terabyte databases and hundreds of megabytes of text per hour). SIM also proved to have an advantage not offered by any of the other products assessed: it contains no third-party code. Because of its architecture and built-in development environment, it is relatively inexpensive (for a high-end system) to implement and maintain.

In the end, we felt that the relatively great initial developmental risk with SIM (because we were asking for a lot of functionality that had not yet been developed) was sufficiently mitigated by the fixed-price contract and the fact that, for us, SIM was literally an entirely home-town product (development, integration, and support are all located in Melbourne). Also, SIM appeared to be inherently relatively easy to extend to new document types.

Although the implementation would take place late in the documentation cycle for the ANZAC ships, after most authoring was complete, we could still find enough savings in the ongoing activities against the maintenance routines to at least break even. Additional savings should be achieved by applying the already installed technology to new classes of documents and especially to new projects where authoring could be started from scratch in a SIM/SGML/XML environment.

We finalized our decision to move to SIM in March 1999, with implementation beginning in May 1999. The project has been a resounding success.

PROJECT SUCCESSES

Immediate successes

Data conversion proved to be easier than anticipated and provided us with the time and opportunity to add substantial value to the content. A WordPerfect macro converted MIP files containing MRC records to RTF format, with field and record delimiters replaced by ASCII markup. Ace's native functionality to understand RTF was used to convert the MRC records from RTF into SGML instances.

Ninety-seven percent of the records were automatically converted to SGML. Of those converted, 70% were compliant with the strict authoring DTD and the remaining 30% were converted to a "loose" (more tolerant of irregularities) DTD and were able to be corrected in the SGML editor. Problems in the 3% that failed to convert had to be resolved in the WordPerfect environment. We anticipated having to retype most tables after conversion. Actually, all tables created using WordPerfect table functions converted perfectly to RTF, and Ace converted the RTF tables perfectly to the required table format. Only minor edits for writing style were required in the Framemaker+SGML environment.

All converted SGML files were reviewed and processed through a complete editing workflow by experienced technical writers to validate the conversion, to improve structure and readability, to create dual language texts where required, and to add new warnings and cautions to most files.

SIM's user and administration environment proved unexpectedly easy to use. Except for authoring procedure texts, all user and administrator interactions with SIM (including with the workflow environment) are performed using the default Web browser (Internet Explorer version 5). This Web browser functions as a very light-weight client. SIM populates Web pages and forms through HTML templates that return user feedback to SIM using standard Internet protocols. Elements of information in the MRC headers (that is, metadata) are edited and validated in the browser environment. Framemaker+SGML is normally only used for drafting and editing procedure text in the routines.

Text can be authored in any SGML compliant editor. SIM launches SGML texts for editing in the user's default SGML editor. If the editor is set up for the DTD used, SIM requires no changes of any kind to the authors' default editing environments. Although FrameMaker+SGML is our preferred editor, we also tested XMetaL, Epic Editor, and even NotePad against the same documents with no problems.

Naive authors (technical writers with no prior FrameMaker or SGML experience) learned to use SIM's workflow and FrameMaker's SGML environments with less than a day of training, and they achieved full productivity (significantly better than in the WordPerfect environment) after less than a week of occasional handholding. Here it should be emphasized that authors have no control over formatting when they work in an SGML environment. The only training required is to explain the concepts of structured authoring and how to use FrameMaker's structural views, element selection palettes, dialogs for tables and attributes, and related issues.

SIM inherently provides for remote authors who can potentially be located anywhere on the Internet. With no special requirements for the authors' workstations or networks, and only minor modifications to the way SIM issues files, SIM's management and authoring environments can be accessed over the Internet by remote authors located anywhere in the world. For new projects, this capability will allow system and equipment suppliers to author documentation related to their products directly in our controlled environment according our predefined structural rules—something that is virtually impossible to control in a word processing environment. We can also easily include client staff directly in our document review and release workflows if that approach proves to be desirable.

SIM, in conjunction with the ship-based AMPS maintenance management system, solved many of our configuration management issues. In theory, all changes to ship structures and documentation are required to be performed in association with unique engineering change orders. Our MRC DTD and SIM's workflow environment allowed us to securely link particular versions of a document to specific engineering change orders and the related applicability and effectivity information:

- *Applicability* relates the document version to a particular item of ship configuration (which can only be changed by an engineering change order).
- *Effectivity* relates the document version to the circumstances under which the engineering change takes effect (which may be a specific date determined by when a ship's configuration is actually changed, or which may be immediate on delivery of the document to the ship).

The AMPS system is aware of the configuration status of each ship and configuration changes relating to engineering change orders, and it will act on and display only MRC's relating to the current configuration of the particular ship. This automated system is substantially less fallible than the manually driven configuration management processes used previously.

Single-sourcing in the SIM environment reduced our document management requirements by more than 80% and our data delivery requirements by more than 95%. We successfully delivered a full class set of documents to the client in October 2000, tied to the delivery schedule for our fifth ship, due in March 2001. With appropriate DTDs, we were able to collapse ship-specific documents into documents applicable to the whole class of 10 ships for two navies.

SGML inclusions gave any element the option of containing multiple "languages." Automatic delivery processing scripts split the deliverables into separate sets of Australian and New Zealand language files for separate delivery to the two navies. Also, by making maintenance routines applicable to the configuration IDs for specific equipment items rather than to a particular ship, we are able to rely on the shipboard maintenance management systems to determine which routines relate to equipment installed on the specific ships. This fact also gave us the "relational" capability to apply one MRC text to several different configuration IDs when appropriate. In several cases of this type, we collapsed 54 WordPerfect documents into one SGML instance.

To reduce the number of documents managed we converted and loaded the most recent ship sets of Australian (Ship 3) and New Zealand (Ship 4) ship-specific documents into SIM. Ship 4 files were cloned and then edited in the SIM environment to form Class documents by including Australian language texts where these differed from the New Zealand version. Ship sets 1 and 2 were reviewed to identify any routines that were unique to one or both of these ships, and these were added to the SIM environment (generally by cutting and pasting elements of text from the WordPerfect version into a blank SGML document).

In essence, this process replaced what would have been five ship-specific sets of documents with one set of document generally applicable to the whole class of 10 ships. This approach immediately reduced text under management by more than 80% (2,000 routines $\times$ 5 ships were collapsed into approximately 1,800 routines applicable to all ships).

Also, once the baseline class-set of MRCs was delivered, only new routines or those that actually change (for instance, as the result of an Engineering Change) need to be delivered. With the WordPerfect system, for Ship 5 specifically, we would have had to deliver 2,000 new routines for this ship in addition to reissuing 2,000 annual updates for each of the previous four ships (the only way we and the two navy fleets could be reasonably sure that all ships had reasonably current documents). With SIM, we needed to deliver only about 400 new documents relating to new systems or configuration changes applying to Ship 5 onward—reducing our data delivery requirement by more than 95%. Long term, we will be able to deliver documents on a net change basis in complete synchrony with engineering changes—a far better solution for the client than once-a-year updates covering all changes that may have taken place over the year.

The SIM environment has proven to be remarkably robust for editing and management. One of the failings of our implementation strategy was that we were unable to define a good stress test for our acceptance testing.

However, in a 4.5-month period, a team that never included more than five people including myself converted and added substantial value to more than 2,000 MRC records selected to form the class set of documents. All us were experienced technical writers, but two had no prior SGML expertise. Each record was extensively reworked to add language elements and new warnings and cautions to many paragraphs, and to adjust structural aspects that had never been properly established in the WordPerfect environment. Some of the converted documents were assessed to be so badly written in the first place that they were completely redrafted. Most were commented for further rework at a later date. All reworked documents were progressed through a formal peer review and quality control signoff workflow process conducted by other people on the same team.

These activities stressed our SIM system far beyond any normal authoring requirement—even for a new project. At the peak, the repository held over 6,000 active documents ("as converted" RAN and RNZN versions for each MRC, plus the clone being edited to convert it to a dual language, configuration ID-linked version), with more than 2,000 active items in the workflow environment.

Despite its being an early SIM implementation, having several kinds of requirements not previously implemented in any SIM system, and running the system on an underpowered server, SIM was crash-free (but not bug-less) for 3.5 months through the period of maximum stress. We delivered (and the client has accepted), converted, validated, and extensively reworked maintenance routines on schedule for Ship 5 delivery.

SIM substantially reduced our document administration requirements. SIM's workflow automation has reduced our growing document administration and management needs in the WordPerfect environment from two full-time administrators to one part-time administrator, even through the period of maximum stress during the document conversion period.

As of January 2001, the SIM implementation project is still working to the budget established in late 1998—a truly phenomenal result for any IT project.

Remaining functionality

Risk analysis performed early in the implementation process suggested that we defer implementing some of the more complex SIM functionality our contract required until the baseline set of class documents was converted and delivered to the client and until these functions were incorporated in a commercial-off-the-shelf (COTS) SIM product. Implementation of the remaining functions required by our contract will take place early in 2001. These include

- **Content re-use (element sharing)** Our analysis shows this capability will reduce the volume of text requiring management by another 50% or more. The COTS SIM product will also provide automatic detection of similar texts wherever they occur in the repository, which will help authors to standardize usage across all documents. Versioning and release control will be applicable to individual elements as well as at the document level.
- **Two-way content linking and annotation at the element level** This capability will allow us to link elements of content to reference documents (including dates and versions) identified in a source registry. Once content links have been added to the MRC records, these will allow us to quickly determine the impact of changes to source documents on the documents we deliver to our client. Annotations may be attached/linked to the reference links or to any other element to capture author knowledge, reasons for change, and so on for posterity.

SUCCESS FACTORS

A number of factors seem to have contributed to the success of the project.

A good commercial business case and an appropriate corporate structure were essential. Content management is revolutionary. You probably can't convince executives who learned business when "secretaries" typed documents on paper using typewriters. You definitely can't convince them when responsibility, authority, and accountability are held by different executives; however, if the manager responsible for the problems has budget authority to fix them, the business case will be evaluated on sensible commercial grounds. In early 1999 an ANZAC Project management reorganization coupled with an increasing profile and effectiveness of the Logistics Group provided an appropriate structure, and the project was approved within 4 months.

End-users (documentation specialists) managed the project. Like most technology companies, Tenix has experienced some IT implementation failures, where projects were managed by IT people or consultants who did not have an intimate understanding of the business requirements and problems. Establishing the SIM implementation as an user-driven research and development project enabled those of us most concerned with the result to maintain control over the project to ensure that we actually got what we needed.

Appropriate outside consultants were hired to increase credibility and check work. We understood what we were doing, but the employment of consultants added substantial credibility when the business case was presented for the final funding decision.

We specified what we needed, not how it must be achieved. Suppliers know what their technology can do better than any user. They will probably come up with better solutions than you can dictate.

We checked supplier financials. One of the overseas companies (Texcel, our initially preferred supplier) closed at the end of 1998. Another was forced to reorganize—hence our concern about reliance on commercially weak overseas suppliers. The vendor we selected had very good financials.

Because we didn't know more about the technology than suppliers, we issued a request for quotation not a request for tender. You may learn a lot from a supplier's initial quotations against your statement of requirements, and what you learn may lead to substantial scope changes in what you actually want. You may wish to go through this process with more than one supplier at the same time. Tender environments are too restrictive.

We used a reliable local integrator as supplier, not the original system developer. At least during the implementation stage of a major IT project, it is important to have close access to technical knowledge of the product. It is time consuming and costly in terms of travel costs to access this knowledge from another state or overseas. Your local integrator will also be more concerned to get a good recommendation from you when the work is complete.

Aspect Computing, one of Australia's larger system integration and support organizations, was our supplier. Aspect solved with no fuss several problems that might have caused major difficulties for less experienced or professional organizations. By comparison to our exeriences with some other IT projects and suppliers, Aspect delivered what we asked for—on time and on budget.

We should have negotiated from a commercially realistic draft contract. One definite mistake we made in our acquisition process was to try to follow a standard Defence Acquisition Organization contract that was simply not commercially realistic or acceptable from IT suppliers' points of view. Substantial time was lost negotiating intellectual property ownership and related issues that no rational IT supplier would find acceptable for anything less than a multi-million dollar government contract.

We realized that the biggest risk may be delaying or not starting the project. We first recognized our need for content management applications in 1994. If we had implemented a very costly early-generation system at that time, we would have saved many millions of dollars in documentation authoring and rework costs that are now beyond recovery, no matter how much better and cheaper SIM technology is today.

We began our first implementation with well understood and stable documents. Learn to use the technology with old and stable documents where you can focus your attention on the technology rather than actually writing documents, but recognize that the greatest savings on a large documentation project can be made by implementing content management before the major authoring requirements begin.

We negotiated fixed-price contracts. Open-ended IT contracts are very risky where business cases are concerned. In most cases, it is probably better to pay more than you think you should to get a fixed price for your business case budget. If you still have a good business case against a fixed cost, executives are much more likely to approve it.

We managed project risk above all else. The first priority for an IT implementation project should be to identify all project risks in terms of cost and schedule blow-out. The next priority should be to develop an action plan to evaluate and mitigate each risk. Solve potential show-stoppers first, and prove your major concepts early. Know your priorities and phase work so you can guarantee some early wins before you have to get the really hard parts to work. All risks should be periodically monitored to be sure that they have actually been dealt with and are not getting out of hand.

THE FUTURE

In summary, even before Release 2 is implemented, structured authoring in SGML/XML and content management in SIM has reduced our authoring and maintenance requirements for maintenance documents for five ships by more than 80% (projected to be a 90% reduction for 10 ships). The move to SGML/XML and SIM has also given us a major improvement in the quality of the text being delivered and has reduced delivery requirements by more than 95%. Note also that savings were measured against a word processing technology that was already doing some very smart things—real value in anyone's book.

Many of the savings we have achieved in the ANZAC Ship Project could also be achieved for other classes of repetitive documents in other organizations.

Other organizations such as the major airframe manufacturers have been using configuration-managed SGML-based environments for several years, and Tenix still has a way to go before our SIM system is the truly generic content management environment we seek.

However, I believe what we are doing for maintenance routines with SIM represents the current worldwide state of the art, and we are working to apply this to other areas of the documentation cycle. I am particularly interested in bringing bidding and contract negotiation under content management control and have managed to raise our concerns in these areas with the Legal XML Contracts Work Group, the Australian Industry Defence Network's eCommerce Committee, and CSIRO Manufacturing Sciences and Technology Division's Manufacturing

Systems and Automation Group. Based on progress to date, it is likely that the standards able to be used by the rest of the world will be developed with a major input from the Australian defense industry.

For example, we are currently working with Australian industry groups and other organizations to create the XML standards we need to minimize chaos in the early stages of the project cycle. All these organizations are working to develop global standards for building virtual enterprises able to address major defense or engineering projects. Tenix already has all the technology required to make such a virtual enterprise work. All we lack are SGML/XML DTDs that we are not in a position to invent unilaterally. TC

FOR FURTHER INVESTIGATION

Debate over the value of structured authoring and content management for technical communicators
http://www.raycomm.com/cgi-bin/lyris.pl?visit=techwr-l

Search for the strings "real value," "single source," or "content management."

History of the ANZAC Ship Project
http://www.tenix.com/anzac_ship_project/anzac_index.html

In addition, searching http://www.google.com/for the string "ANZAC Ship Project" will provide hundreds of links to many details of the project.

SIM
http://www.simdb.com/

Although SIM is XML-based, the benefits of inclusions allowed by SGML for alternative languages led to our decision to develop our maintenance documents to an SGML DTD standard.

FrameMaker+SGML
http://www.adobe.com/products/framemaker/prodinfosgml.html

XMetaL
http://www.softquad.com/top_frame.sq?page=products/xmetal/content_xmetal_intro.html

Epic
http://www.arbortext.com/Products/Product_Overview/Epic_Editor_LE/epic_editor_le.html

LegalXML for contracts worldwide
http://www.legalxml.org/

Legal XML is based in the U.S. but has a significant Australian component and is well on the way towards developing generic XML models for contracts that will flow down through the rest of the project cycle.

XML-based document exchange standards for global enterprises
http://cic.vtt.fi/projects/globemen/

The international GLOBEMEN project is developing technologies and methodologies to assist distributed organizations exchange contractual and technical information to form "virtual enterprises." There is a developing synergism between LegalXML and GLOBEMEN.

WILLIAM P. HALL is currently documentation systems specialist for the Tenix Anzac Ship Project, where he recently specified and coordinated the implementation of a state of the art content management system for ship maintenance documentation, and is currently moving technical manuals into this environment. He earned his PhD in evolutionary biology from Harvard University in 1973 studying species formation and evolution in lizards. He migrated to Australia in 1980. In 1981 he bought his first personal computer and opened a technical word processing bureau to pay for it. He found computers more interesting than lizards, so he soon found himself writing and teaching about computers. In 1990 he began work for the predecessor of Tenix Defence Systems on one of the largest and longest-lived documentation projects in Australian history. Contact information: bill.hall@tenix.com