Document-based Knowledge Management in Global Engineering and Manufacturing Projects[1]

William P. Hall, Martin Jones
Tenix Defence Systems, Australia
(email: bill.hall@tenix.com)

Mingwei Zhou, Julia Anticev, Jeffrey Zheng, John Mo, Laszlo Nemes
CSIRO Manufacturing & Infrastructure Technology, Australia

Abstract

The core competence of a global engineering and manufacturing enterprise increasingly depends on the quality of its intellectual resources and how these resources are used. This paper presents an approach to document-based knowledge management in a typical global engineering and manufacturing application, the ANZAC Ship Project.

1. Document-based Knowledge Management

1.1 Documents in the Project Life Cycle

With the globalisation of business activities, the core competence of a manufacturing enterprise increasingly depends on the quality of its intellectual capital and how this capital is used [1]. Intellectual capital is usually presented in the form of various kinds of documents.

This paper discusses an approach towards the document-based knowledge management in the global engineering and manufacturing projects [2]. As shown in Figure 1, knowledge can be captured into documents at many stages in the project lifecycle and the nature and value of the reusable content and contextual knowledge varies with the project stage [3,4].

[1] To be published in **GLOBEMEN, 2002,** "Global engineering and manufacturing in enterprise networks". Proceedings of the GLOBEMEN conference, Helsinki and Espoo, Finland, 10-12 December 2002.

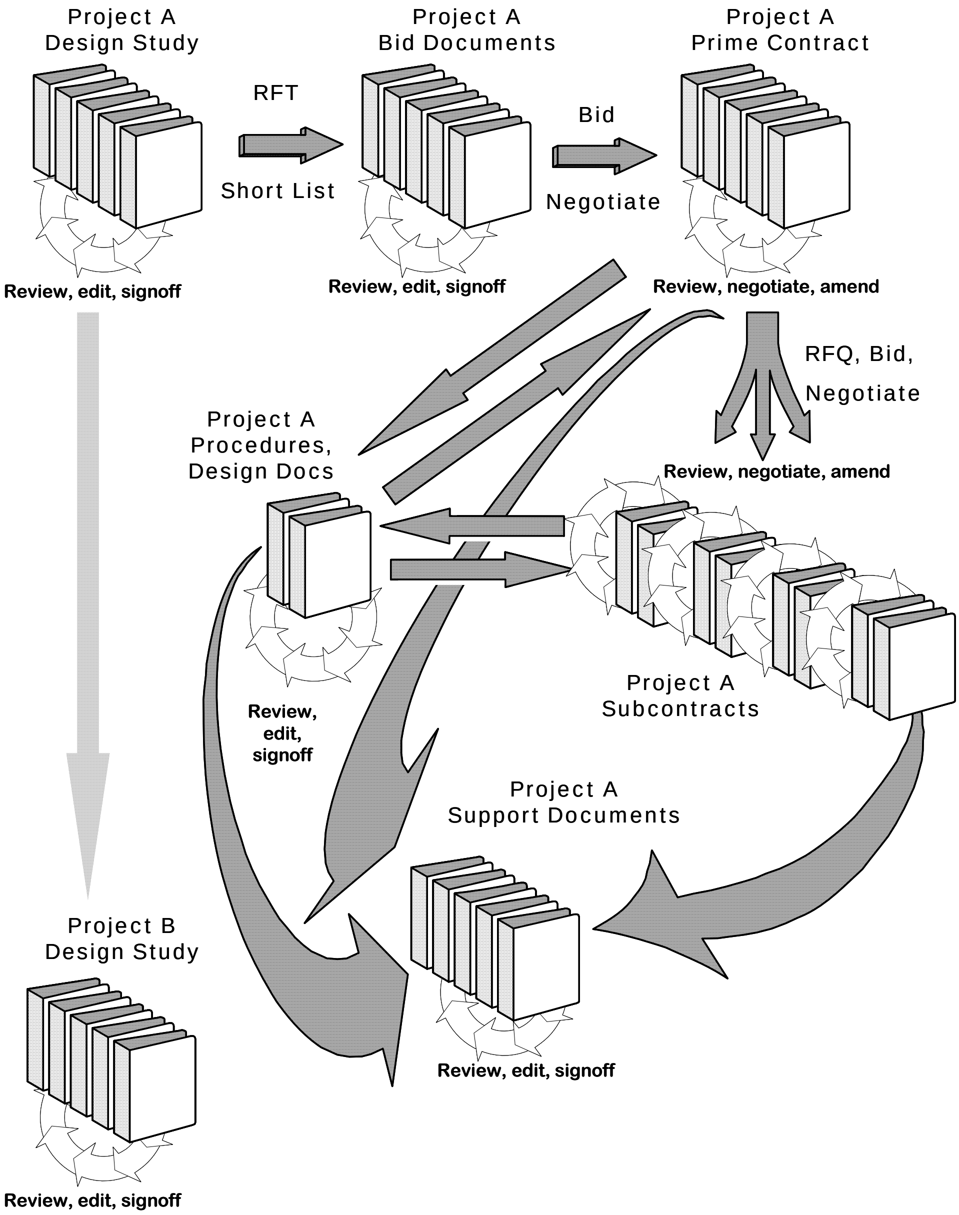

Figure 1. A simplified view of documentation stages and knowledge flow in a typical global engineering and manufacturing project

In a global engineering and manufacturing project involving many different systems, document content is highly redundant within each project stage, in that the same kind of information is required for each of the systems, and the same components may be used in many different systems. Also, document content is often reused as it flows from one stage of the project to the next, and documents developed for one project serve as the basis for later projects having similar requirements. Management of this redundancy through re-use is a major goal for document based knowledge management architecture. On the other hand, different kinds of contextual knowledge need to be captured at each stage.

The design stage of a project normally begins with a client's Request for Proposals (RFP) outlining the capabilities sought from the project. The essential contexts are knowledge relating to the client's statement of capabilities, engineering decisions, notes, standards, correspondence, and a variety of documentation from lower tier suppliers. Suppliers distil this kind contextual information into design proposals and bids. Based on information provided by design studies, the client normally issues a Request for Tender (RFT) to a short list of suppliers. Suppliers then rework and extend the documentation developed in their proposals and include a detailed commercial response to the RFT. Design proposals and RFT responses are required to be completed within very limited time periods, such that if documentation is not delivered by the specified due date the proposal or bid will not be considered. Development of proposals and bids is highly competitive and places major pressure on bidders to minimise document production cycle times and to maximise the quality of information/knowledge the documents contain [5].

After assessing bids, the client negotiates a contract with one or more of the tenders. Contract negotiations with preferred bidder(s) can be protracted as the parties work to maximise their respective commercial advantages - but this is at the expense of delaying start of production. Contexts (e.g., notes, memos, correspondence, etc.) relating to the contract negotiations can be extremely important to explain details of the agreement to minimise scope slippage, but can be easily lost as personnel changes in the client and supplier organisations.

Subcontract and partner negotiations are normally completed after the prime contract is finalised and agreed. These represent all of the problems encountered in the negotiation between the client and supplier.

Internal design and procedural documentation (which may also involve client reviews) must then be developed to give effect to the contractual agreements. Here, the contextual knowledge represents an understanding of the linkages between contractual requirements and conditions and what is being done to give effect to them.

The final stage of the project documentation cycle involves the production of product support and maintenance documentation. The process is driven by contractual requirements, and information is assimilated from supplier document, and a variety of internal and external resources. This information can then be assimilated into maintenance documentation.

1.2 Knowledge Models

Knowledge is the internal state of an agent following the acquisition and processing of information, here the agent could be a human being or a computer system. To categorize human knowledge, many knowledge models have been proposed. Polanyi [6,7]] identified that human knowledge has two major components: the tacit and explicit knowledge. Following Polanyi's concept, Nonaka [8,9] further proposed his theory that tacit knowledge consists of personal relationships, practical experience, shared values and explicit knowledge consists of formal policies and procedures.

Nickols [10] further clarified the intrinsic meanings of various knowledge terms by proposing a testable knowledge model that includes: explicit, tacit, implicit, declaration, and procedural knowledge. As shown in Figure 2, Nickols' model also depicted a testable procedure to distinguish the relationship among different classes of knowledge.

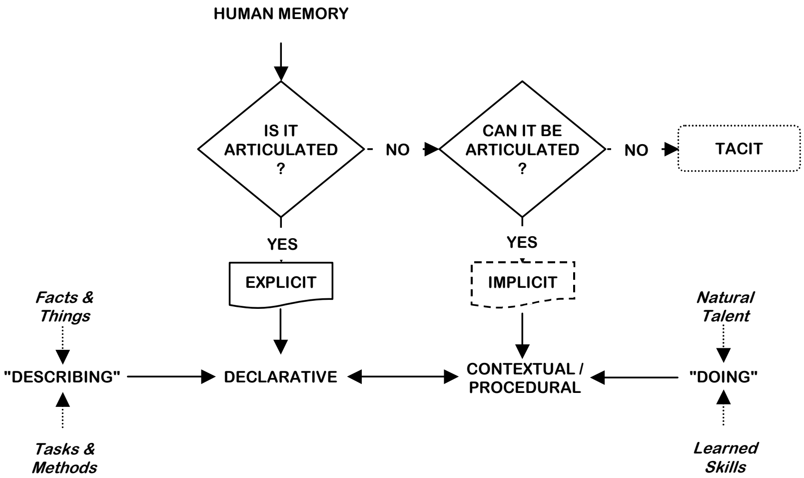

Figure 2. Nickols' Knowledge Model

In our approach, we classified the knowledge involved in the engineering and manufacturing project into three categories based on Nickols' framework.

The first category is the "direct knowledge" or facts, which are the explicit or declarative knowledge. This category of knowledge is visible, written, transferable, sharable and reusable. It is usually documented and stored and transmitted externally to a human brain. In the bidding process, engineers assimilate their information and turn it into bidding documents conveying distilled knowledge. A knowledge management system should help the engineers capture, validate, and preserve knowledge; and assist discovery, reuse, retrieval and transmission of that knowledge.

The second category of implicit knowledge includes "procedural knowledge" or the best practices, which are usually implicit, and context sensitive. This category of knowledge is related to processes, methods, practices in groups and professions. This type of knowledge needs to be identified, captured and made explicit in the way that can be shared. However, it is not always well documented. A second form of implicit knowledge includes "contextual knowledge", which relates to sources of information, background knowledge and the like that was available to the author at the time of writing, but that has no place in the formal, explicit document. This knowledge is easily lost with time as authors forget or change positions in the organisation. Benefits can be achieved if this contextual knowledge can be captured and made explicit.

The third category is the "tacit knowledge", which is the most difficult to understand and represent. It is indirect, embedded in experience, owned by individuals and cannot be articulated in words.

Our research presented in this paper focuses on the first two categories of knowledge, aiming to capture and make available for preservation, management, discovery and reuse as much direct and procedural knowledge as possible, and attempt to interpret and convert the contextual tacit knowledge surrounding the documents into direct and procedural knowledge and contextual information that can be preserved, managed, discovered and reused.

1.3 Project Information Architecture

Essentially all activities in a large engineering organisation are related either to winning contracts or fulfilling the requirements of existing contracts. Aside from the contractual documentation that states the requirements that drive the project, there are three major bodies of information and knowledge that must be managed: design, production and documentation.

Design information encompasses the engineering knowledge and structural model required to describe and build the physical product from its components. This knowledge is hierarchically structured, with the product logically broken down into smaller components. Components of the design that must be managed include catalogue items that are purchased or manufactured and drawings that provide 2D or 3D models for what must be assembled. A product data management (PDM) environment provides tools for relating all kinds of data, drawings and documentation to particular components in the product structure. Careful management of Engineering Changes (ECs) to components and related data, drawings and documents is crucial to building a successful project.

The design process culminates in the production of an Engineering Bill of Materials (EBOM) describing all of the components that must be assembled into the product. The engineering bill of materials is changed into a Manufacturing Bill of Materials that describes how to build a particular instance of the product. Production information is managed in an Enterprise Resource Management environment to ensure that the knowledge embodied in the MBOM is reflected in the actual production of the products. This involves resource management, production planning and scheduling, procurement and warehousing of materials and components, and the management of work orders. Production also interfaces with a number of subsidiary systems including human resources management (HRM), Accounting and Cost and Schedule Control System (CS^2).

Ideally, the knowledge in documents should be managed in the same way that the engineering knowledge is managed in a PDM environment. Documentation products include a number of document types, where each type generally has a well-defined content model. When document content is encoded in SGML or XML, rather than defining formatting styles, the document type description (DTD) defines elements that are allowed/required to occur in a document validly conforming to the type, and rules for the allowed/required sequence and hierarchy of in the document. Tags defined by the DTD can be used to label the kind of content enclosed by the tags, which can in turn be used to infer semantic relationships or contexts.

Requirements that are common to both the Product Data and Document Content management environments are configuration control and change management. Changes to elements of content in the documentation must be managed coherently with engineering changes to the product as designed and as built. The documentation must manage applicability and effectivity of elements that document changes are released in synchrony with engineering changes. Management of the engineering and documentation change processes can be most effectively managed and tracked using an electronic workflow system.

The following sections discuss the systems and information architecture adopted by Tenix for managing support knowledge and documentation for the ANZAC Ships.

2. Documentation for the ANZAC Ships Project

2.1 The ANZAC Ship Project

ANZAC Ship Project is one of the largest defence contracts in Australian history. It aims to design, build, and support 8 ANZAC Class frigates for the Australian Navy and 2 for the New Zealand Navy. The A$ 6 billion ANZAC Ship Project contract was signed with Tenix Defence in November 1999 after several years' project development, bidding and contract negotiation activities, and will run through 2006 when the last ship is delivered. Following completion of the production contract, ships will have to be supported in service for at least another 27 years (the design life) or longer if work is done to extend their lives. In-service support is provided by the ANZAC Ship Alliance (ASA) comprised of Tenix, Saab, and the Royal Australian Navy's ANZAC Systems Program Office.

Defence projects are knowledge intensive, capturing much of the required knowledge in various kinds of documentation, both within the project and as deliverables to the client. Many document types are required to be maintained for several decades. For example, the authoring of ANZAC Ship maintenance procedures began more than three years before the first ship was delivered. Documents will have to be maintained up to date with ship configuration changes until the last ship goes out of service sometime around 2033, representing a total document lifespan of some 40 years. Based on Tenix's experience, the cost to produce, manage and deliver project-related data and documentation is several percent of total project acquisition costs.

2.2 Fleet Support Documentation

An area that caused considerable difficulty has been the collecting knowledge for, authoring and maintaining support documents for ANZAC ship systems and equipment. Figure 3 summarises the process by which information is assembled from a variety of different sources to produce this documentation. Requirements for the support documentation are given in the contract and supporting documents. These are consolidated and extended in the Integrated Logistic Support Plan. System and equipment technical manuals and other supplier documents provide factual information about systems and equipment. Details of the systems hierarchy, materials, parts, tools, fluids and lubricants and other miscellaneous and test equipment are assembled and managed in the Integrated Logistic Support (ILS) Database.

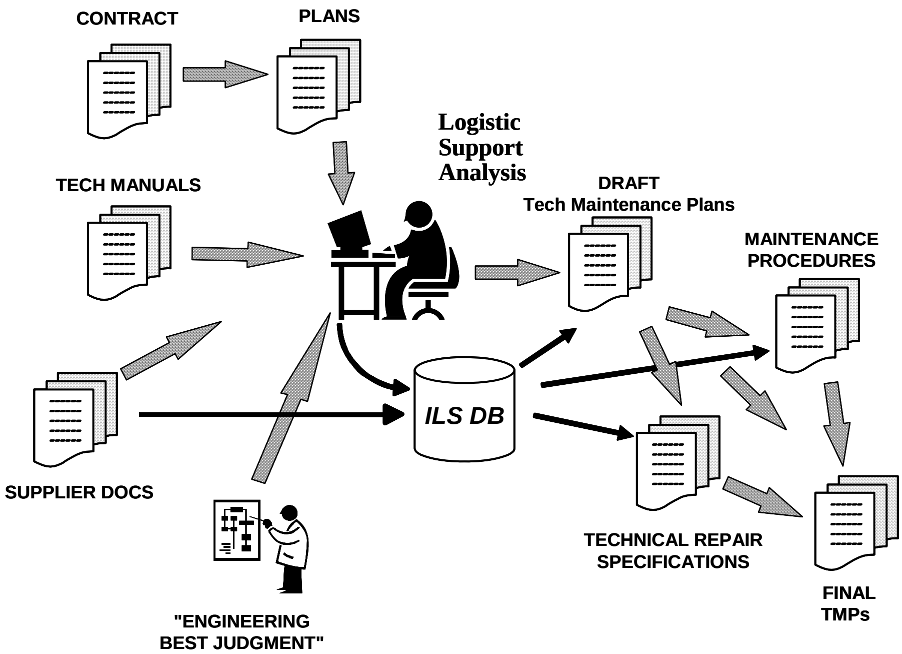

Figure 3. Information flow in the production of maintenance documentation

Logistics and maintenance engineers digest the input documentation and produce a draft Technical Maintenance Plans (TMP) for ship systems, which establish basic maintenance philosophies for each system. TMPs describe two kinds of maintenance, Technical Repair Specifications (TRSs) - describing equipment overhaul type tasks normally done by external repair agents, and maintenance procedures (Maintenance Requirement Cards or MRCs) - detailing periodic or conditional maintenance tasks to be done by naval personnel on-board or using shore-based facilities. On completion of the detailed analysis required to draft the TRSs and MRCs, TMPs are finalised to reflect the TRSs and MRCs as written.

2.3 Requirements

Obviously, authors apply "engineering best judgement" based on their own experience, and a large amount of implicit contextual understanding is developed regarding the many different sources of information condensed into the deliverable documents. However, the documents themselves only offer limited capabilities for explicitly capturing the contexts. Much of the knowledge about the sources of information behind the documents is lost when the authors go on to other things.

There is only a single TMP for each of the ~160 systems comprising a ship, and a single TRS for each kind of equipment to be overhauled. However, due to configuration and language differences between the ships, MRCs need to be ship-specific. Also, there may be many MRCs for each equipment requiring maintenance.

The requirements for authoring and managing the MRCs are onerous:

- ~ 2000 different MRCs per ship, times 10 ships,
- coherent management of commonly used information affected by engineering changes across the fleet documentation,
- language differences between Australian vs New Zealand ships,
- major health and safety implications if the wrong information is provided or essential safety warnings omitted,
- requirement to include scheduling and resource requirement information in procedures (maintenance "metadata") and deliver this data electronically into the client's relationally-based computerised maintenance management system (AMPS), and
- Ship-specific information in the MRCs needs to be maintained concurrently with changes to ship configurations throughout the ships' anticipated in-service life of 27 years.

Innovative solutions are needed to meet these requirements. Documents were initially authored in WordPerfect merge table formats. More than 20 different deliverables could be sourced from a database comprising a single "record" corresponding to each maintenance procedure and its associated metadata. Merge/macro processes were also developed to validate critical data items in the records against master data held in the ILS DB. This solution sufficed for the first ship, but as soon as work began on MRCs for the second ship, it was clear our authors would face substantial difficulties maintaining complete consistency of key data items across many thousands of independently maintained documents. These had to work without error when loaded into the client's relationally based AMPS maintenance management system. By the time MRCs were completed in 1999 for Ship 04, client complaints about inconsistencies in common information used throughout the ~8,000 MRCs for the first four ships had escalated to the point that Tenix faced a threat that Ship 05 would not be accepted.

Implementing the system provided us with a platform for an architecture that by October 2000 more than successfully resolved all of the outstanding issues with MRCs for our Ship 05 delivery, both internally and for the client.

3. Knowledge Management Systems Architecture for In-Service Support

3.1 System Architecture

The stringently fixed-price contract for the ANZAC Ship Project has a number of relatively unique features that have forced the company to focus not only on managing the flow of knowledge into maintenance routines, but on supporting and maintaining this knowledge after delivery of the ships into service. The fixed price contract terms included delivering logistic support requirements for the ships' in-service operational requirements.

Figure 4 illustrates the information flow for the maintenance of logistic support knowledge in the in-service support portion of the project lifecycle [11]. The shaded arrows indicate the feedback loop between the operational knowledge about performance of the logistic support package (including documentation) in service, and making adaptive modifications to the various forms of support and maintenance documentation. This knowledge that has been refined with in-service experience is also available to be flowed down into other projects using similar systems. However, as is the case for other kinds of documentation, in a word processing environment much of the author knowledge used to assemble the deliverable documents remains implicit, and is readily lost through staff movements.

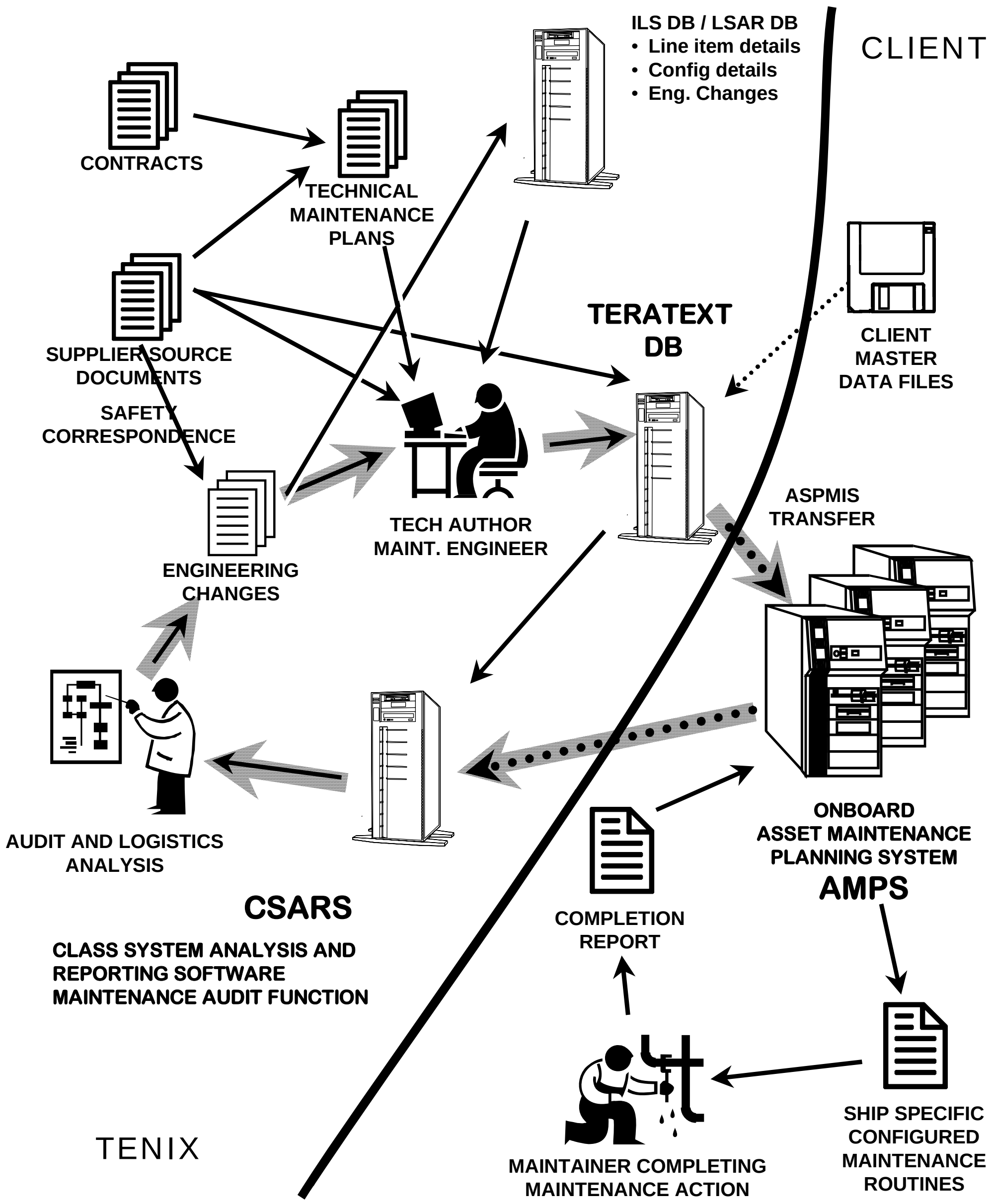

Figure 4. Closing the circle with operational knowledge to optimise in-service support

Figure 5 illustrates a generic systems architecture developed for managing the information and knowledge flows for the in-service phase for major projects. Supplier-based systems are shown in the black boxes, client-based systems are shown in the white boxes. Heavy arrows represent the flow of operational knowledge through the support optimisation loop.

The central system on the supplier side is the engineering/product data management environment, which ideally includes workflow responsibilities to co-ordinate all engineering and all engineering-related documentation changes. Document content is either managed at the file level within the PDM system and for distribution to the client, or within a document and content management system (DCMS) for management of the authoring process.

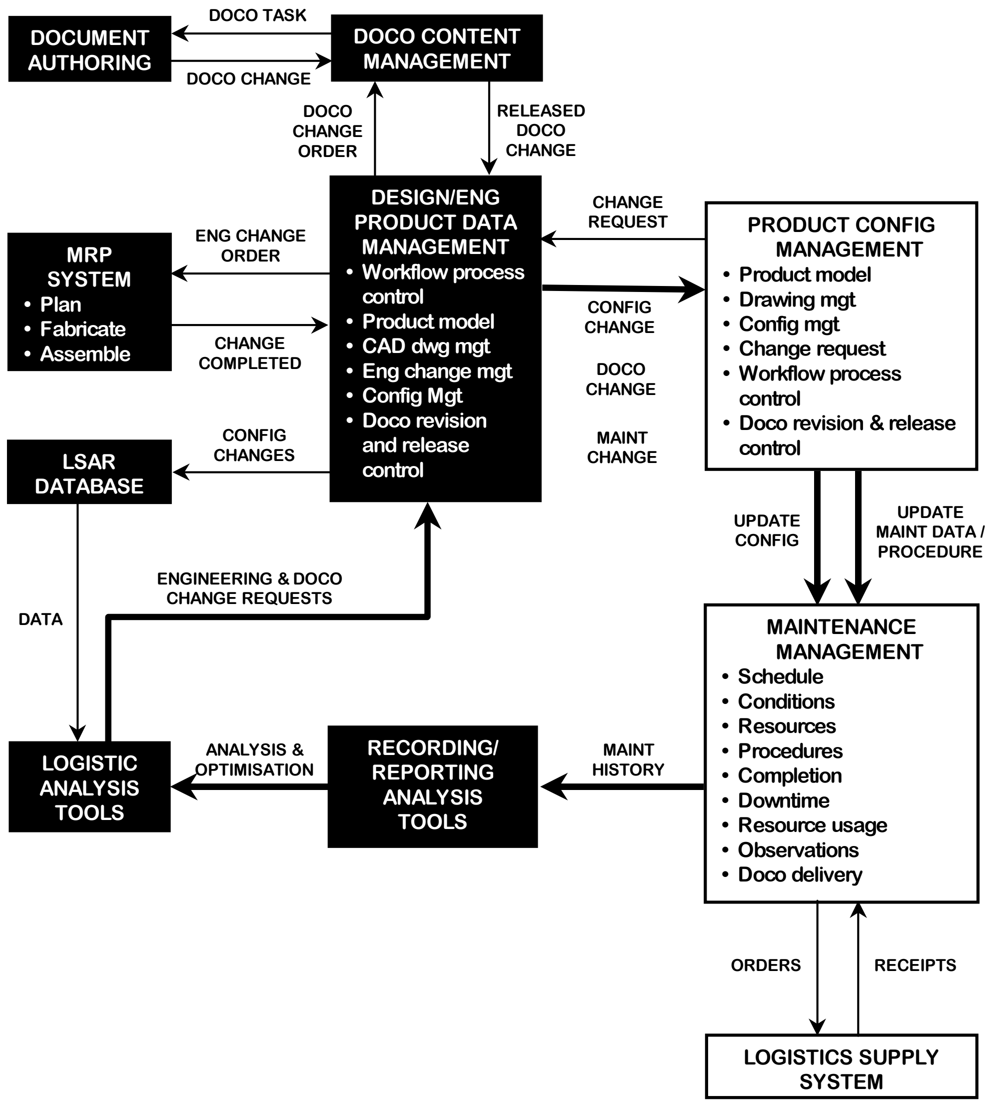

Figure 5. Systems architecture for project in-service maintenance management and support

As will be discussed below, the DCMS system provides a key for capturing authors' implicit contextual knowledge. Tenix currently uses the Web-based TeraText platform. Authoring and editing can be done in any SGML or XML compliant editing tool. Bespoke tools were developed in the TeraText platform to automatically validate all key MRC metadata items against master copies of the information held in the ILS DB or as supplied by the client.

On the supplier side, all engineering changes to the manufacturing environment (i.e., MRP) and to the master logistics data should be controlled and tracked by the central PDM system.

In the case of the ANZAC Ship Project, configuration information and maintenance documentation is delivered electronically from the Tenix PDM environment to the client's PDM environment via an agreed interface standard. For the MRCs, the deliverable consists of a single master document containing configuration and language specific information for all the ships, comprising an HTML file describing the procedure to the human maintainer, plus a set of relational table records (delivered as ascii comma delimited files) comprising the MRC metadata the maintenance management system requires in order to properly schedule the maintenance activity and associated human and material resources required for its completion.

The client's PDM/AMPS central maintenance management system then resolves the master MRC for a particular kind of equipment (including information for all ships using the particular routine) into ship-specific MRCs that are delivered electronically to the ship-board AMPS systems. Ship-board AMPS systems then automatically trigger maintenance jobs based on calendar periodicity or the existence of certain system conditions (e.g., engine running hours) as captured into the AMPS system. When a human scheduler selects a job for action, AMPS prints a job card (on flimsy paper) for the human maintainers that includes the procedural document, portions of the metadata (e.g., the job's requirements for tools, spares, and other supplies), and forms as required for reporting any observational information back into the AMPS environment.

Maintainers complete the job and enter required observations and other relevant information onto the job card. These returned details are then entered into the AMPS system together with a number of job related parameters that AMPS collects automatically.

As part of the requirement to measure and prove the effectiveness of the logistic support and documentation package for the ANZAC Ships, Tenix developed a Class Systems Analysis and Reporting Software (CSARS). Tenix periodically downloads selected data items relating to the performance of corrective and planned maintenance tasks from AMPS into the CSARS application for calculation of measures of effectiveness for each ship system and analysis of factors contributing to the poor performance of systems falling below their effectiveness targets. Logistics engineers try to identify and find engineering solutions for components that prove to be inherently faulty and/or any issues in maintenance or operating procedures that may have caused abnormal failures.

Engineering change requests to correct system components or documents are then fed back into the engineering PDM environment for management and execution - to close the circle with corrective feedback to the project engineering and maintenance knowledge base [11].

3.2 Managing Content within Documents

If the only concern was to manage and distribute document files, any of many different document management systems could have met their requirements. However, Tenix decided that only an SGML (or XML)-based content management system had the appropriate capabilities to provide the full functionality required for the ANZAC Ship maintenance documentation.

Authoring in SGML or XML offers many advantages where it is useful to automate or intelligently index identifiable elements of content within the documents [12], [13]. Appropriately named element tags identify to the computer system the location of particular kinds of content. Also, SGML or XML authoring tools and SGML/XML content management systems validate documents against document type DTDs to ensure that content is structured according to the rules of the DTD. Where the content management system is able to parse the DTD, content can be indexed in semantically useful ways to assist human or automated retrieval of particular kinds of content. Also, the tagging systems make it easy to identify particular elements of content within documents that require validation against master data repositories.

Some of the immediate benefits that were derived from this system are significant. For example, by implementing "single source" content management after completing the documentation for Ship 04, four ship sets of WordPerfect merge data files were collapsed to one class-set of SGML instances for the delivery of Ship 05. This immediately reduced text under management by 80% and once the class set was delivered, further delivery requirements were reduced by more than 95%. The documentation change cycle time was reduced from a year to delivering only net changes within a matter of days.

Since the initial implementation of the DCMS, the ANZAC Ship Project has added functionality to annotate documents and include the client in the pre-release document review and signoff workflow.

3.3 Integrating Documentation with Engineering/Product Data Management

A major goal for integrating document content management with engineering/product data management is to provide hard links between configuration related information in the documents and master data as held in the engineering tools. Two kinds of connections ensure this linkage.

- Integration of documentation changes and engineering changes into a single Engineering Change (EC) workflow process, where both engineering tools and authoring processes are under command of the one workflow engine, to ensure that changes are made in concert.

- Automatic validation of relevant document content against master engineering data. This is facilitated in a structured authoring environment where the element tag and associated attribute data can uniquely identify an element of document content with a particular configuration item in the engineering data.

In the ANZAC Ship Project's DCMS implementation, the EC process still involves a paper-based workflow - but one that is strictly controlled by the Engineering/PDM system. To minimise response times in the documentation system, the DCMS system automatically downloads a configuration master file every night, and then automatically validates all configuration specific information in a document against this master file, whenever the document is checked in or out. Any changes or mismatches in the configuration-related information are immediately alerted to the author/user for attention in a formal document change process.

3.4 Capturing Implicit Knowledge with Annotations

From a knowledge management point of view, the ability to apply annotations to elements of content within documents is one of the most powerful tools that can be applied in a content management architecture.

As implemented in Tenix (Figure 6), the content management architecture contains two logically separate repositories; one for managed content and one for "source" documents that are preserved for reference purposes as binary objects. Each annotation is managed as a separate structured document. Content managed documents and annotations are managed within the content management repository. Any kind of object (physical or electronic) can be referenced as a "source". The critical thing is to enter metadata describing the source object and its revision status in the Source Registry. If the object is available in an electronic format it can be preserved for reference and archival purposes in the Source Repository linked to the Registry.

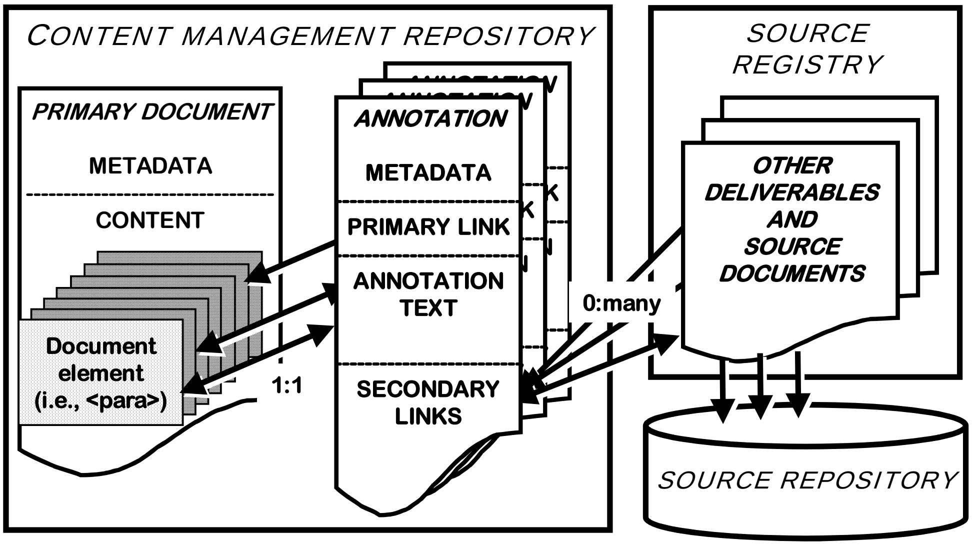

Figure 6. Annotation architecture.

Within this architectural framework, an annotation consists of:

- metadata describing the annotation itself (author, date, version),
- a primary link associating the annotation with a particular element of content in a managed document version - where zero to many annotations can be linked to the one element of content,
- free text entered by the annotation's author that provides contextual information relating to the link that cannot appropriately be entered into the primary document's text, and
- zero to many secondary link associating the particular annotation with the source registry entries for particular source(s) relevant to the annotation.

The annotation function as implemented in SIM/TeraText provides an easily used capability for capturing contextual information relating to the primary link.

Three types of annotation functions are provided:

- *Author annotations.* These persist for the lifetime of the linked element of content in the managed primary document, and may include any number of secondary references. Author annotations are visible to other internal staff. They are used primarily to capture authors' implicit knowledge of the circumstances and contextual information surrounding the authoring process.

- *Internal review annotations.* These persist only for the life of a major version, and are dropped from the document when it is released for publication. Internal review annotations are only visible to authors and internal reviewers. Review annotations are primarily used to communicate reviewer comments and requests for corrections to authors and supervisors. Because all versions of a document are archived, the reviewer comments remain available if required for change tracking or audit purposes.

- *External (client) review annotations.* Except that the client can only see their own annotations, external review annotations are managed the same way that internal review annotations are managed.

In the ANZAC Ship Project, author annotations explicitly significant sources of information authors use in creating or changing each paragraph of text, e.g., references to supplier and client documentation including correspondence, standards, third party procedures, change requests and the like. For each reference, the author is required not only to provide the link to the referenced object, but details of why the source was referenced and the location(s) of the referenced information within the source. This information is of great value to Tenix in managing document changes.

In the past, when a lower-tier supplier changed a part in an assembly, or a specification or determined that some aspect of maintenance processes should be changed, this is normally notified by correspondence - or perhaps only a new version of an existing document is received. To determine the potential impact of the change, when the original authors of the documents were no longer available, experts first had to compare the new document with the prior version to identify source changes, and then had to guess which systems and documents would be affected by the change, and finally had laboriously study Tenix's deliverable documents to see if these had to be changed. The process often took person days of work over several weeks to complete a single impact analysis.

With the implementation of annotation, impact analyses can be completed in a few minutes. When a change is received to a registered source (as determined by its metadata), the system will print a "Where Used" report, identifying all deliverable documents linked to the source by annotations. The text of the annotation identifies particular locations in the source that have been used. If these are unchanged between the original and new version of the source, there is no impact. If there are relevant changes in the source, the linked deliverables can be opened to the paragraph(s) referencing the source. A quick read of the deliverable text, the text of the annotation, and the text referenced in the source will reveal any need to change the deliverable to reflect the change. In other words, impact analyses that often took an author familiar with a the particular system weeks to complete - leaving a significant risk that some impacts may have been missed, can be completed by any author in a few minutes with more confidence that no impacts had been missed.

3.5 Minimisation of Redundancy

As noted in the introductory material, a major issue with project documentation can be the requirement to separately maintain common information that is used in many different places. There are three strategies akin to "normalisation" in a relational database (see Kent [14]) that can be used in the architecture described here to minimise the requirement to manage redundant content.

- *Variant documents from a single source master file*. All variant information is held in a single master file, with the variant information held in parallel elements within the overall document structure. Attributes on the variant elements determine which element at any point in the document structure should be included in a particular output variant. This requires a significant processing task to resolve the content for the particular output type, but once the process is developed it allows one document to satisfy a number of different outputs. In this case, the single master document contains components for all the documents that can be delivered from the single version managed container. The strategy is particularly appropriate for a limited number of language or configuration-specific texts within an overall consistent document structure. Tenix gained all of the benefits it required to resolve the MRC documentation issues for the ANZAC Ship Project using this approach. However, single sourcing is much less appropriate where documents share texts but have significantly different structures where it is not possible to maintain a 1:1 co-ordination between corresponding elements in the different output documents.

- *Virtual documents from shared content.* Each deliverable document has its unique version managed container, and includes all content that is unique to that particular container. However, where content to be included in the deliverable already exists somewhere else in the content management system, the container for the specific deliverable will simply point to the location where the shared content already exists. For this architecture to work, content elements need to be maintained within the content management repository as individually indexed items, and the output process must be able to assemble the referenced elements together with unique elements to produce a deliverable document. Tenix is currently working to implement this methodology in a generic version of its DCMS based on functionality provided in later versions of the TeraText application. The virtual document approach is particularly suited to reducing redundancy in the less rigidly structured documents such as technical manuals and the range of document types found in the earlier stages of the project documentation cycle.

- *Managed entities.* SGML and XML include the concept of named entities in their DTD languages, where an entity is an object defined in the DTD that can be included in a document instance simply by invoking its name. Entity references are used to invoke graphics (where a single identified graphic may be used in many different documents) or sufficiently standardised and commonly used texts that it pays to define them separately from ordinary elements of text that may be subject to casual reuse. For example, in some systems standard health and safety warnings and cautions are treated as entities, which can be invoked wherever required in deliverable documents by entering the entity number.

The major benefits from virtual documents and managed entities that a block of content needs to be written only once but can be used many times. In highly redundant bodies of documentation such as are often encountered in engineering project documentation, this can reduce authoring and document maintenance costs by 50% or more by comparison to standard file management systems where each document has to be separately maintained. To safely manage content for reuse requires the ability to track and manage versioning at the level of reusable elements as well as at the file level. It also requires the development of change management workflow to identify all documents using the changed element, so users can determine whether the effected documents should include the new version of the element or should preserve the original version.

4. Conclusions

This paper presented an approach towards a document-based knowledge management system for the ANZAC Ship Project. Their benefits have been proven in a real-world engineering environment.

Although in the ANZAC Ship environment the direct exchange of deliverable documentation is only between Tenix as the prime contractor and its client, the TeraText-based content management architecture and workflow environment is fully Web based and could readily be deployed to include all members of the supply chain contributing directly to the documentation package. Also, the annotation function was expressly developed to provide a means for quickly and accurately determining the impacts of changing supplier documentation or client requirements on the content of deliverable documents by comparison to a documentation environment based on paper formats where much of the necessary information required for impact analysis remains implicit, and was lost through time when the original authors of the deliverable documents were no longer available.

This system and architecture described in this paper serves as a prototype and proof of concept for what will in time become a generic capability. Tenix is currently working with TeraText's developers and implementors to "package" the product to make it easily deployed for arbitrary document types across its divisions and supply chain. The first implementation of the generic solution is likely to be in the recently formed ANZAC Ship Alliance, a virtual organisation comprising Tenix (supporting hull and mechanical systems), Saab Systems (supporting communications and combat systems), and the ANZAC Systems Program Office (representing the ships' operators).

References

[1] Mo J., Zhou M. (2002). Tools and Methods for Managing Intangible Assets of Virtual Enterprise, to appear in Computers in Industry

[2] Zheng J., Zhou M., Mo J.P.T., Tharumarajah A. (2000). Background and Foreground Knowledge in Knowledge Management, Global Engineering, Manufacturing and Enterprise Networks (Proceedings of 4th International Conference on Design of Information Infrastructure Systems for Manufacturing), 15-17 November, Melbourne, Australia, pp.332-339

[3] Hall, W.P. (2000). Structured Management of Maintenance Documentation, XML Asia-Pacific 2000, Sydney, Australia

[4] Hall, W.P. (2001). Maintenance Procedures for a Class of Warships: Structured Authoring and Content Management. Technical Communication 38:3 pp. 235-247. http://www.tenix.com/PDFLibrary/91.pdf

[5] Zhou M., Mo J., Nemes L., Hall W., (2002) Knowledge management in bid preparation for global engineering and manufacturing projects, DIISM2002, Japan

[6] Polanyi, M. (1958). Personal Knowledge: Towards a Post-Critical Philosophy. [Corrected Ed., 1962]. University of Chicago Press, 428 pp.

[7] Polanyi, M. (1966). The Tacit Dimension. Routledge & Kegan Paul.

[8] Nonaka, I., Takeuchi, H. (1995). The Knowledge Creating Company: How Japanese companies create the dynasties of innovation. New York: Oxford University Press.

[9] Nonaka, I. (1996) Knowledge Has to Do with Truth, Goodness, and Beauty. in Dialog on Leadership: Conversation with Professor Ikujiro Nonaka, Tokyo, Japan, February 23, 1996 and with Claus Otto Scharmer

[10] Nickols, F. (2000) The Knowledge in Knowledge Management (KM). in J.W. Cortada and J.A. Woods, eds. The Knowledge Management Yearbook 2001-2002. Butterworth-Heinemann

[11] Hall, W.P., McFie, K. & Beer, J. (2002). Managing Maintenance to Reduce Life-Cycle Costs for a Multi-National Fleet of Warships. Proceedings. 2002 International Maintenance Management Conference, Gold Coast, Australia

[12] Berners-Lee, T; Handler, J. & LaSalle, O. (2001). The semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American. May 2001

[13] W3C, (2002) Extensible Markup Language (XML). W3C Architecture Domain. World Wide Web Consortium

[12] Kent, W. (1982) A Simple Guide to Five Normal Forms in Relational Database Theory. Communications of the ACM 26(2), Feb. 1983, 120-125