realtimepublishers.com®

The Definitive Guide™ To

Enterprise Change Management

MERANT

Dan Sullivan

Copyright Statement

Chapter 5: Managing Change in Enterprise Content

Organizations are under new pressures to manage enterprise content, including text, image, audio, video, and other unstructured digital assets. Today's enterprises face internal pressures to better manage operations and improve business intelligence, and external pressures from regulatory agencies and new legislation: HIPAA regulations dictate mandatory protections for private health care information; third-party creators of digital assets—such as research documents, online journals, and other references—demand digital rights management; and consumers expect their private information protected, and governments are ensuring that organizations meet those expectations by applying increasing limits on the use of personal data. We are no longer simply dealing with document management—the need to control the lifecycle of digital assets demands adaptive change-management practices and tools (see Figure 5.1).

In this chapter, we'll explore the lifecycle of content and the internal and external organizational factors that influence this cycle. In addition, we'll develop a model of content change management using the framework we discussed in Chapter 2. We will examine how to build policies that support change management tailored to the most important pressures facing an organization. Finally, the chapter concludes with a look into integrating content, system configuration, and software configuration management.
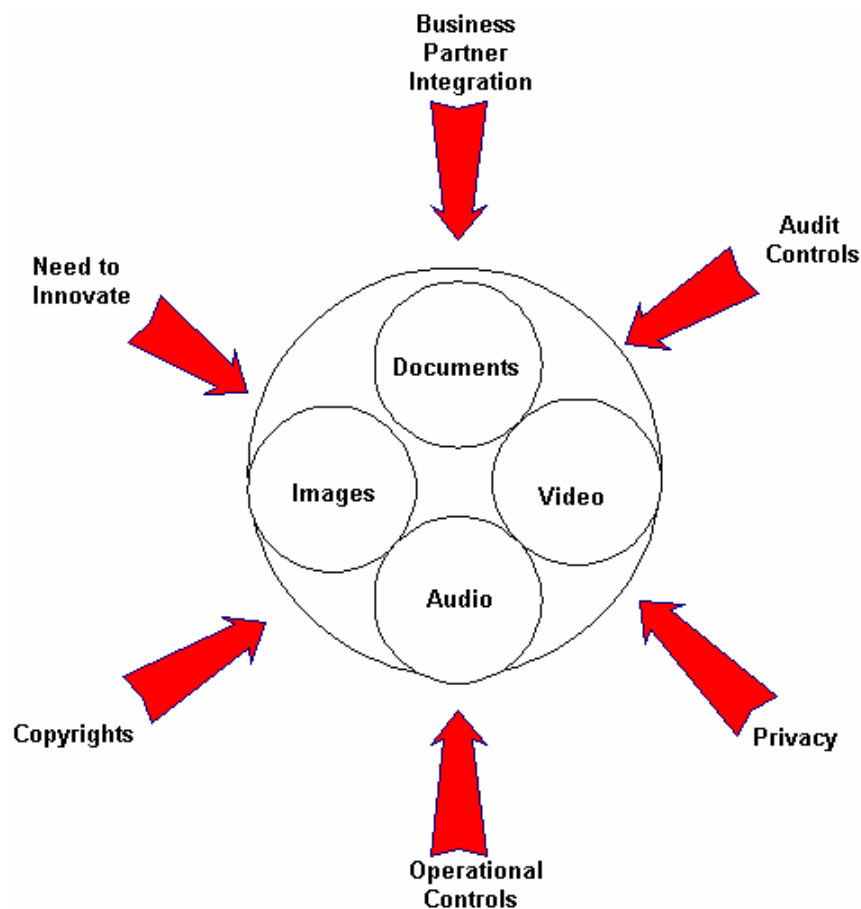
Figure 5.1: Multiple internal and external pressures increase organizations' need for content management and change control.

Understanding the Nature of Content in Today's Enterprise

A new term has emerged in the English language to describe the state of content management in enterprises and across the Web—*infoglut*, which is the condition of being overwhelmed with data and information. Even organizations that practice document and content management are susceptible to infoglut when they step outside their corporate or agency boundaries. Infoglut is the product of the content lifecycle and technologies that surround it. Fortunately, infoglut can be managed. To understand how, we turn to the content lifecycle.

The Origin and Proliferation of Content

Some content is created to serve an end, such as a novel, or as a part of another process, such as a Human Resources memo describing how to conduct performance appraisals. The latter is our primary concern. Organizations create content as a normal part of their operations. The reason to create content is to preserve organizational memory. There are too many details to track, decisions to communicate, and ideas to convey without written discourse.

Although the basic drivers behind content creation have not changed over thousands of years, the methods have. With changing methods, a number of problems have developed, including:

- Growing volumes of content

- Multiple content repositories

- Difficulties in managing unstructured content

- The need to comply with government regulations regarding content

Managing Growing Volumes of Content and Multiple Content Systems

We generate content at a staggering pace. Most organizations have the infrastructure in place to support large numbers of knowledge workers who create and revise content seemingly continuously. Email and instant messaging are supplementing traditional documents and adding to the volume of content that must be managed. The impact of this enabling technology is clear:

- Data on PC hard drives, departmental servers, and enterprise servers is growing at 100 percent per year.

- More than two-thirds of published information is in the form of office documents.

- The deep Web, the online databases that are not accessible to conventional search engines, is 400 to 500 times larger than the surface Web.

There are numerous challenges to managing this much content. Multiple systems are used to store, index, and retrieve content. The Internet is a distributed Web of servers with common protocols for exchanging data. Organizations often use multiple systems for storing content, including:

- Local PC hard drives

- Shared network drives

- Intranet sites

- Document management systems

- Relational databases

- Portals

At one end of this extreme is the local PC hard drive that is generally accessible only to the machine's users. Local drives can be shared across small groups, but ad-hoc sharing of these drives creates more management problems than it solves. At the other end of the spectrum are portals that have the promise of providing a single point of access to multiple systems. For this configuration to be useful, application integration and enterprise search issues must first be addressed.

Managing Unstructured Content

As the volume of content grows, the difficulty of finding content also increases. Structured data is fairly easy to manage using well-developed and proven database design techniques. These techniques allow program designers to create applications that can rapidly find and retrieve specific records and particular pieces of information (Figure 5.2 shows a simple database table).

Relational Database Table					
Primary Key	Code1	Code 2	Last Name	Initial	Phone
ID01	CD1	10	Jones	K	555-1234
ID02	RL1	20	Smith	B	555-3458
ID03	DF9	15	Anderson	R	555-9090
ID04	CD1	39	Williams	L	555-9972
ID05	CD1	19	Johnson	M	555-9811
ID06	RL1	18	Smith	J	555-8081

Figure 5.2: Structured data is organized in fixed structural elements.

MERANT

For example, a customer inquiry application can quickly retrieve M. Johnson's record if the application has the unique identifier for that record: primary key ID05. Without the primary key, the application can still search for "Johnson" in the last name column and "M" in the Initial column. These techniques do not work with unstructured data.

Content such as images and text is considered unstructured data. It does not neatly fit into relational databases, spreadsheets, or other well-defined structures, and unstructured data does not have clearly delimited attributes such as first name and phone number. Instead, information is spread across text in a free-form manner. Consider, for example, a note in a CRM application:

> *Received call today from M. Johnson checking on status of order #45876 that should have shipped on Oct. 8. Need to verify status of order and return call (555-9811) by 10/15.*

In this example, the name of the customer is not clearly identified. Although we can easily find it, a typical database application cannot. There is no predefined location to put attributes (for example, start name at character position 25 and phone number at position 128). Extracting attributes from free-form text requires specialized applications that can identify the many forms of names, dates, currency amounts, and other entities, as in the following example:

> *Received call today from <Name>M. Johnson </Name>checking on status of <Order>order #45876 </Order> which should have shipped on <Ship Date October 10, 2003</Date> Need to verify status of order and return call (<PhoneNumber>555-9811</Phone Number>) by <Date> October 10, 2003</Date>.*

In this example, XML tags are used to identify features and entities that can then be extracted and loaded into structured databases. These processes are complex and subject to error, and applications cannot always distinguish attributes of the same type (for example, the order date and the ship date).

At best, managing large volumes of unstructured content requires additional processing to extract key attributes. In most cases, management requires developing custom extraction rules and robust data-quality checks to ensure high-quality extraction.

Extracted data serves two purposes. First, it maps elements of unstructured data into a structured representation, making it accessible to existing information-processing applications. Second, it is used as metadata to describe content. The additional metadata enables better content management.

📖 We'll discuss how you can manage assets using metadata later in this chapter.

The complexity of content management is increasing because of three factors within organizations:

- The growth in the volume of content within and outside of organizations
- The proliferation of content storage and management systems within enterprises
- The need to manage unstructured as well as structured data

External factors are contributing to the growing complexity as well.

Managing Distributed Documents

Documents range from simple one-page memos and short emails to large-scale, multi-segment content with multiple authors distributed over several repositories. Complex and distributed documents present additional challenges:

- Coordinating revisions

- Setting up access control and edit rules

- Repurposing content

- Handling multiple renderings

Complex documents have multiple authors, editors, and users who must approve the content. Their activities must be coordinated to ensure that editors read the most recent version of a document segment, authors maintain an auditable trail of revisions, and those who must approve the content work from final versions.

Access controls typically prevent unauthorized use of documents, but they can also enforce workflow rules. For example, once a document segment has been submitted for review, an editor has to review and revise before the author can make further changes. Similarly, approvers cannot approve a document until a final author and editor version has been released.

XML, which creates semi-structured data, creates numerous opportunities for repurposing content. For example, a standard legal statement can be incorporated into all financial reports created for public review with a single reference. If the boilerplate text is changed, any document that references that text will have the latest version when the document is regenerated. Similarly, a standard product description can be used in sales brochures, Web site content, and other documents through an XML reference. The text of the content remains the same, and the appearance can change as needed.

XML documents are rendered through four methods:

- Cascading Style Sheets (CSS) in Web browsers

- XSL Transformations (XSLT), a transformation language, and FO, a format object specification

- Document Object Model (DOM) and Java (or other programming language)

- Proprietary systems

Although these methods vary in their approaches, they all use basic XML as input and present the content of the XML document in a precisely defined format.

> You can find more information about each method in the XML.com Resource Guide at http://www.xml.com/resourceguide/.

When Document Management Is About More than Documents

Document management requires more than a repository with version control to store files. The process now encompasses:

- Multiple repositories logically linked into a single repository through enterprise search and retrieval services

- Support for multiple authors, access controls, and workflows

- Metadata management and, in some cases, metadata extraction

- Semi-structured data for formatting and rendering

A truly enterprise-scale content-management system must address each of these requirements. As we are about to see, external factors are increasing the need for such organization-wide applications.

Growing Demands of Regulation

The second factor affecting the need for content management is the growing trend of regulations governing what content is produced, how it is managed, and who has access to it:

- The Sarbanes-Oxley Act requires a new standard for financial reporting processes

- HIPAA regulations cover protected health care information, including measures health care providers must take to ensure the privacy of patient information

- 21 Code of Federal Regulation (CFR) Part 11 requires auditable processes for electronic submission of regulated content in the pharmaceutical industry

- The Gramm Leach Billey Act (GLBA) dictates how banks can use customer information

- The Securities and Exchange Commission (SEC) 17 CFR Parts 240 and 242 have extended book and records regulations of security brokers and dealers to cover business-related instant messages

The result of these regulations is that organizations are required to maintain more control over content—content that is proliferating at a breakneck pace. The solution is to develop ECM practices that encompass change management and support dependencies across enterprise assets.

Managing External Content and Publishers Rights

Digital rights management adds another dimension to the complexity of managing enterprise content. Content and information licensed from third parties often carry restrictions on use and distribution. For example, access to online databases and journals may be restricted to a set number of concurrent users; users may be limited to a department within a company; the content itself may be accessible only for a set time.

Two commonly used techniques for digital rights management are *containment* and *marking*. Containment uses encryption to restrict access through proprietary programs. Marking inserts metadata tags describing acceptable use of the content. The Extensible Rights Markup Language (XrML) is one markup language used to specify rights and conditions.

> 🖉 For more information about digital rights management and XrML, see http://www.xrml.org.

Although the complexities of content management are growing, its fundamental nature does not change. This is reflected in the content lifecycle.

Managing the Content Lifecycle

Enterprise content follows a complex lifecycle (see Figure 5.3). Some stages are common to all documents, like create and publish; some stages are limited to special cases, such as repurposing; others should be common to all documents but in many cases are not, such as archiving and destroying.
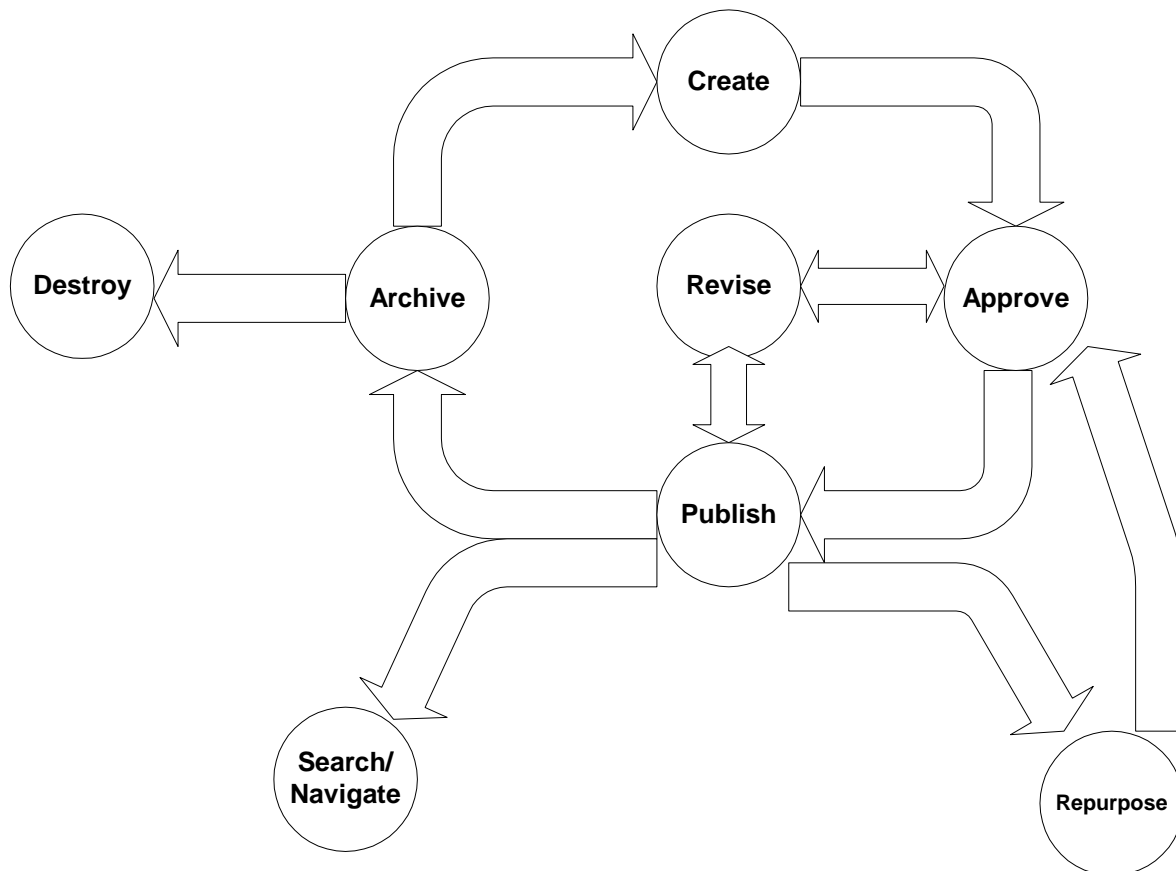

Figure 5.3: The stages of the content lifecycle.

Creating Content

Content originates in a number of ways:

- Produced from scratch, such as a simple email or memo

- Created based on a template, such as a standard project-management document or government-specified report

- Generated as the compilation of multiple pieces of existing content (for example, a product catalog generated by combining structured data from an ERP database with semi-structured product descriptions from an XML database-management system)

Often content is introduced to an organization from outside the enterprise. Email is an obvious example. Documents come from business partners, customers, vendors, regulatory agencies, and other outside organizations. External content that is intentionally acquired—such as business publications, journals, and analyst reports—should be managed like other valuable assets. Unwanted external content, such as spam, requires its own management regime. As we shall see when discussing the model for ECM, policies governing content will vary widely to accommodate these differences.

Approving Content

Fairly small amounts of corporate content require formal approval. Word processing documents, spreadsheets, and presentations are common artifacts circulating in today's organizations. Most of the time, documents are formally reviewed and approved when:

- The content is intended for external use (for example, a sales brochure)

- The content is a part of a legal process, such as contract negotiations

- The content is an official corporate document, such as a regulatory filing

- The content is published as part of a broad or sensitive business activity, such as a corporate reorganization

Implicit in the approval process is a mini-lifecycle of submissions, commentary, and revisions. The final stage of the limited process is approval or rejection. In the former case, the content moves on to the publishing phase of the broader lifecycle.

Publishing Content

Publishing takes on many forms. In simple cases, a file is saved to a shared network drive or added to a document-management system. Descriptive metadata—such as keywords, summary, version indicator, and author information—is added at this point to facilitate search and retrieval. Figure 5.4 shows a typical metadata entry form.

Figure 5.4: Descriptive metadata such as keywords and archive date should be included with published documents.

Publishing may require XML tags to add structure to the document before publishing through a content-management system. It may also require the definition of style sheets or other rendering document to control the format of the final presentation.

Revising Content

Revising content sounds straightforward but is one of the most problematic stages in the content lifecycle. The problems stem from the need to control versions, prevent concurrent editing, and audit changes.

Controlling Versions

Versions are easily controlled within content- or document-management systems. The repository tracks versions, maintains access controls over document versions so that only authorized users are able to create new versions, and provides a single point of reference for the "official" copy of a document. However, the real world does not always fit into this model. Copies of documents are distributed through email, shared in an ad-hoc fashion, and are often edited outside of formal change-control mechanisms. The result is that multiple editions of a document can exist at the same time. Discipline on the part of users is required to ensure that documents revised outside of the defined workflow are not added back to the repository as "official" versions.

MERANT

Concurrent Editing

Users making concurrent changes are a problem in many applications. In databases, for instance, two users can update a record at the same time and save their changes one after another. Which change is saved depends on who saved the record last. To prevent one user from overwriting the changes of another, database vendors use several options for dealing with concurrent change; some that require complex mechanisms. We do not have that luxury when dealing with documents.

As documents are unstructured, they do not lend themselves to the same types of concurrency controls that work in structured databases. Users are generally left with the option of losing one set of changes, as Figure 5.5 illustrates.

1. User A checks out document from the respository and edits the file.

User A

2. User B checks out the document from the respository and edits the file.

User B

3. User B checks in the document to the repository.

Document Management Server

4. User A tries to save the document to the repository and receives an error message: either overwrite User B's changes or lose User A's changes.

User A

Figure 5.5: Concurrent editing can lead to lost changes.

To avoid this problem, users should:

- Check in documents frequently

- Segment logical documents into multiple small documents

- Use XML to create semi-structured segments that are compiled as needed to generate complete documents

> ✎ There are technology options to apply annotations to documents when reviewing documents concurrently. These annotations can then be audited and versioned as well.

Auditing Changes

Tracking changes is especially important in regulated industries in which companies must demonstrate the integrity of processes and workflows. Applications maintain change history by either keeping complete copies of each version of a document or by tracking only the differences between the versions. The latter is more efficient because it requires less storage. It does, however, add to the complexity of the document-management system by introducing transactional data that must be managed along with the document itself (see Figure 5.6).
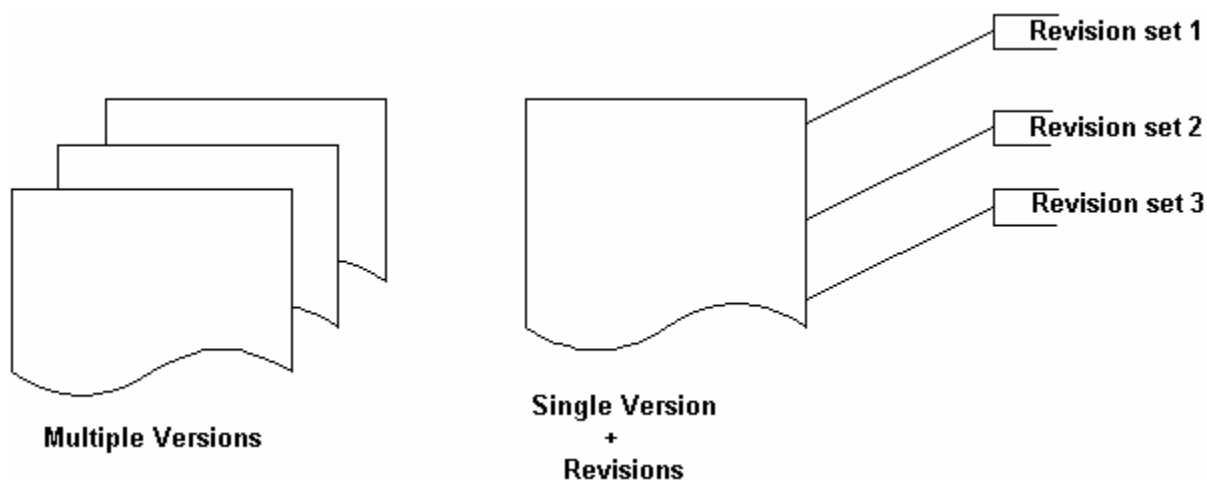
Figure 5.6: Auditing document changes is more efficient when tracking revisions instead of maintaining full copies of documents.

Repurposing Content

Repurposing content is possible when content developers follow a few basic rules:

- Write content in fairly small, logical units

- Use standardized XML tags to identify content structure

- Store content in accessible repositories

Writing Logical Units

In the case of the first rule, "fairly" is the operative word. Obviously, segmenting content into phrases or single sentences is too fine-grained for practical use. Logical units should be based on the semantics of the domain one is writing about.

Using Standardized XML Tags

Repurposing content is difficult without standardized tags. Aggregating content requires finding and retrieving logical units of text (or other unstructured data) much like we do with structured database applications. We need to identify attributes, such as "customer id" or "segment name," and values, such as "CBA1789" or "short product description."

Standard tags are defined in schemas, which also describe the structure of content. Listing 5.1 shows a simple example of a product with four attributes: name, id, short description, and long description.

```
<xs:element name="product">
  <xs:complexType>
   <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="id" type="xs:string"/>
      <xs:element name="short description" type="xs:string"/>
      <xs:element name="long description" type="xs:string"/>
   </xs:sequence>
  </xs:complexType>
</xs:element>
```

Listing 5.1: A simple example schema.

This schema defines an element called "product" that has a simple list structure (indicated by the sequence tags). XML query languages allow users to extract target pieces of information, such as the short product description, without retrieving other elements associated with the entity.

📖 For more information about and a tutorial on XML schemas, see
http://www.w3schools.com/schema/default.asp.

Storing Content in Accessible Repositories

The final guideline when repurposing and aggregating content is to store content in accessible repositories. These could be document-management systems, XML databases, relational databases with XML interfaces, intranets, or even the Internet. If content is accessible through linking, it is not restricted to a single repository.

Search and Navigate

The longest stage of the content lifecycle is searching and navigating. At this point, content has been developed and refined to the point at which it is suitable for general use. From a content creator's point of view, the document is done. From a user's perspective, the work is just beginning. How are users to find documents in repositories of tens of thousands of documents? How are they to know which documents even exist?

Enterprise search systems index content in multiple formats on a variety of source systems. This approach provides benefits similar to Web search engines: users do not need to know about the physical implementation or location of content. This approach also offers similar drawbacks to Web search engines: it can overwhelm the user with hits by including irrelevant content while simultaneously missing relevant content. As Figure 5.7 shows, a single enterprise search system can index content from across the organization.
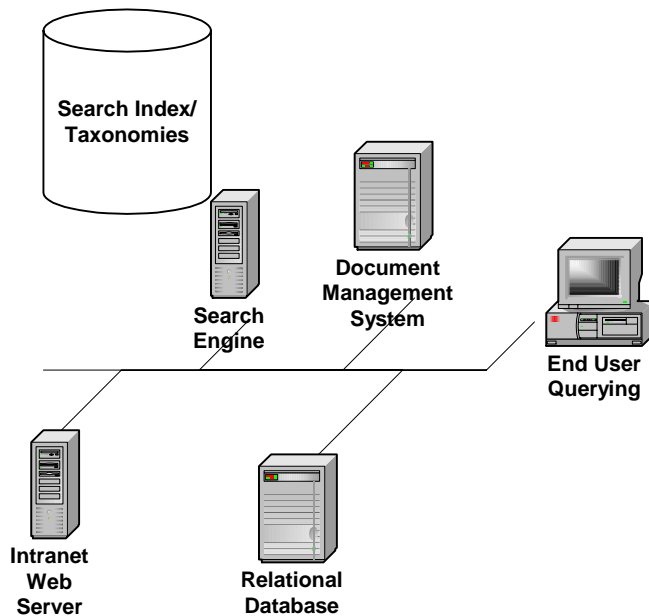
Figure 5.7: Enterprise search applications index content in multiple formats from a variety of source systems.

To improve the quality of keyword searches and to provide an alternative method for finding content, many organizations are now using navigational taxonomies. Taxonomies are hierarchical or linear classifications of content along a dimension. Common dimensions are:

- Geography
- Organizational structure
- Time
- Product categorization

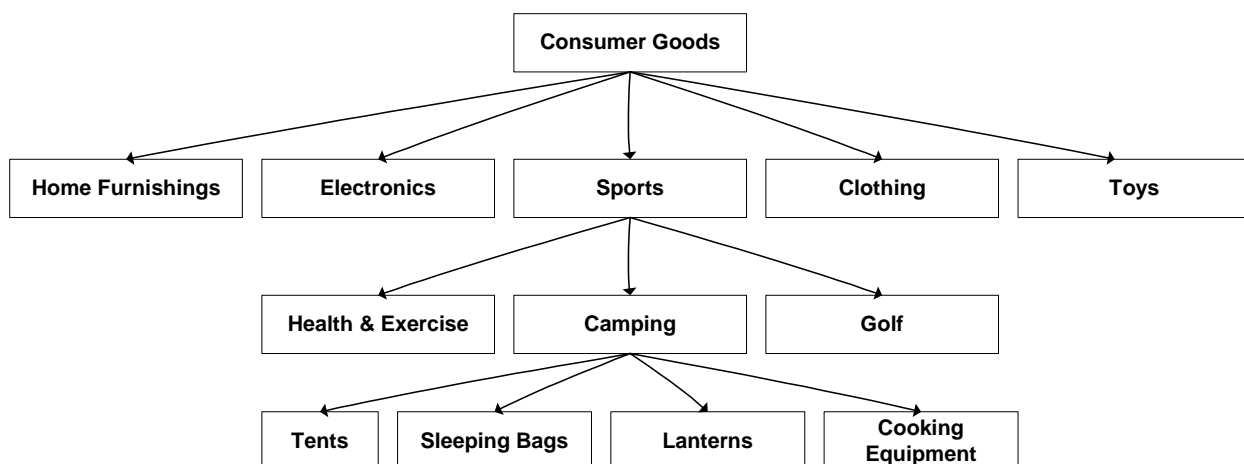Figure 5.8 shows an example that a retailer might use:

Figure 5.8: A product taxonomy allows users to move from broad, general areas to specific topics.

Combining keyword searching and navigation is a powerful alternative to using either alone. This approach allows users to search within a sub-region of the taxonomy (for example, searching only content in the Camping region and below). How long content remains useful will vary, but eventually its active use will wane and it will move to the final stages of the content lifecycle.

Archive and Destroy

The end of the active life of content is the optional archiving stage. Information that must be retained for legal and historical reasons is archived to long-term storage media (typically, CD-ROMs and tapes). Unlike active content, archived content is not readily accessible through online content repositories.

Much of what we write, edit, and revise has a short useful life and should be eventually destroyed. Of course, many records are covered by statutory retention requirements; other documents may be covered by corporate record retention policies.

📖 We'll explore record retention policies later in this chapter.

The content lifecycle is complex and so is content change management. Like software change management and system configuration management, the change-management model developed in Chapter 2 provides a framework for managing that complexity.

Elements of Change Management for Content

Modeling change in enterprise content requires:

- Assets
- Dependencies
- Policies
- Roles
- Workflows

Assets

The major types of assets that change in ECM are:

- Documents
- Document metadata
- XML schemes
- Formatting and rendering specifications

There are secondary assets as well, such as single sign-on services, databases, and portals. Their roles extend beyond ECM, and so we will not discuss them in this chapter.

Managing Changes to Documents

Documents are rather generic entities. In the current context, a document can be a:

- Word processing file
- Spreadsheet
- Presentation
- Email
- HTML file
- Dynamically generated portal page
- Collection of aggregated XML-based content
- Image file
- Audio file
- Streaming video

Documents are also dynamic. Authors, reviewers, and approvers can all change documents. A copy that is sent as an email attachment can quickly become outdated. To maintain control over documents and to minimize the impact of obsolete versions, a definitive version of a document should be used.

For example, consider a sales team that is developing a proposal for a client. The original request for proposal (RFP) should be kept as a read-only file in a document-management system, a portal content area of some other shared repository. Team members should receive a link to the RFP rather than an email attachment. This link saves disk space by not duplicating multiple copies of the file, and ensures that any revisions to the document are accessible to everyone on the team. Related documents, such as background material provided by the client, should be saved in the same folder, giving the team a single location for client material.
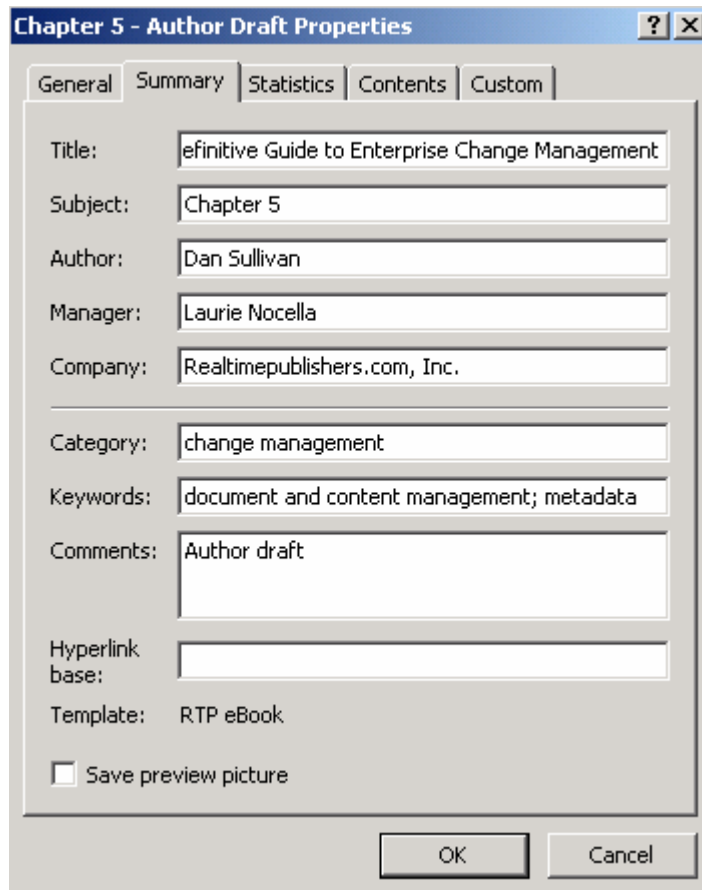
To effectively control change with documents, follow these general guidelines:

- Use a single content repository for project documents. Primary documents—such as RFPs, draft responses, and background material—should be stored here. Secondary materials, such as news articles about the client, do not require the same level of control. Linking to those documents is sufficient.

- Distribute links to, rather copies of, documents.

- Use version control; include descriptive metadata about each version of a document.

- Define policies dictating who can create, edit, and delete documents within a project area, and enforce these policies with access controls.

- Use distinct folders for different stages of workflow (for example, Draft Folder and Final Revisions Folder).

Other specific guidelines should be developed with regards to dependencies within a content-management environment.

Using Appropriate Document Metadata

Two types of metadata are required for effective document management: content and administrative. Content metadata describes documents with keywords, summaries, author names, creation and revision dates, and categorization information (see Figure 5.9). This metadata is especially useful for search and navigation. Search engines, for example, will rank documents higher if a search term appears in keyword metadata than if it appears only within the body of text. Advanced search forms allow users to specify values for other metadata elements, such as creation dates and author name.

Figure 5.9: Metadata added to the properties sheet of a Microsoft Word document can be used by enterprise-scale search engines to improve the quality of search results.

Administrative metadata is used for maintenance and compliance operations. Archive or expiration dates identify the end of the useful life of a document. Rights metadata describes copyrights and limits on the use of content. Last change dates are used to implement incremental backups. Administrative metadata will vary according to specific operational requirements.

Tracking XML Schemas

As I mentioned earlier, XML schemas describe the framework of semi-structured documents. The first entry of an XML document typically indicates the XML schema used. For example, the product information that Listing 5.2 shows is based on a schema called catalog-schema.xml.

```
<?xml version='1.0'?>
   <consumer-good xmlns="x-schema:catalog-schema.xml">
      <item category="camping">
        <product-name>Acme Outdoors 6-person Tent</product-name>
        <manufacturer>
           <manf-name>Acme Outdoors</manf-name>
           <manf-id>ACME0133</manf-id>
        </manufacturer>
        <price>198.99</price>
        <short-description>This 6-person tent is a versatile …</short-
description>
        <long-description>Designed for the camping family, this tent
offers large
           doors, 3 windows, steel stakes and a polyester mesh
ceiling. ...
        </long-description>
      </item>
     ….
    </consumer-goods>
```

Listing 5.2: An example schema.

Schemas are documents and are therefore subject to the same change-management issues as other content. Their close relationship to XML documents introduces dependencies that I'll discuss shortly.

Formatting and Rendering Specifications

Formatting and rendering specifications are like XML schemas in that they are documents that function closely with XML and other semi-structured documents. In addition to managing them as documents, we must manage their referring documents as well.

Content assets are no longer restricted to traditional, self-contained documents. Structuring frameworks, such as XML, and rendering schemas, such as XSLT and CSS, are now important elements in ECM. These new elements greatly enhance our ability to reuse content and manipulate text with greater precision. The cost of these benefits is more dependencies between assets.

Identifying Dependencies in Enterprise Content

Dependencies in ECM emerge from several places, including infrastructures that support content management, the relationship of content to other enterprise assets, and the use of externally controlled assets, such as XML schemas and protocols.

MERANT

Infrastructure Dependencies

Infrastructure dependencies can be subtle. For example, how should documents be archived so that they can be opened in 5, 10, or 20 years? File formats change and commonly used programs may not be available in a decade. Storage media also changes. Although magnetic tapes continue their long life as a staple of IT infrastructure, media formats change along with their hardware. Any layer of IT infrastructures (OS, file formats, storage media, and encoding schemes) that can change will have an impact on how archived material is used.

Related Asset Dependencies

Content is sometimes closely linked to other content, as in the case of XML schemas and formatting specifications. Documents that use those schemas or specifications are limited to using tags and functions defined in the corresponding document. The more complex schemas and formatting specifications become, the more difficult it is to change them while maintaining backward compatibility. Changing a schema without backward compatibility can force authors to update documents, which, in turn, can initiate workflow and approval processes, which can introduce more changes. These changes ripple through to other dependent assets (see Figure 5.10).
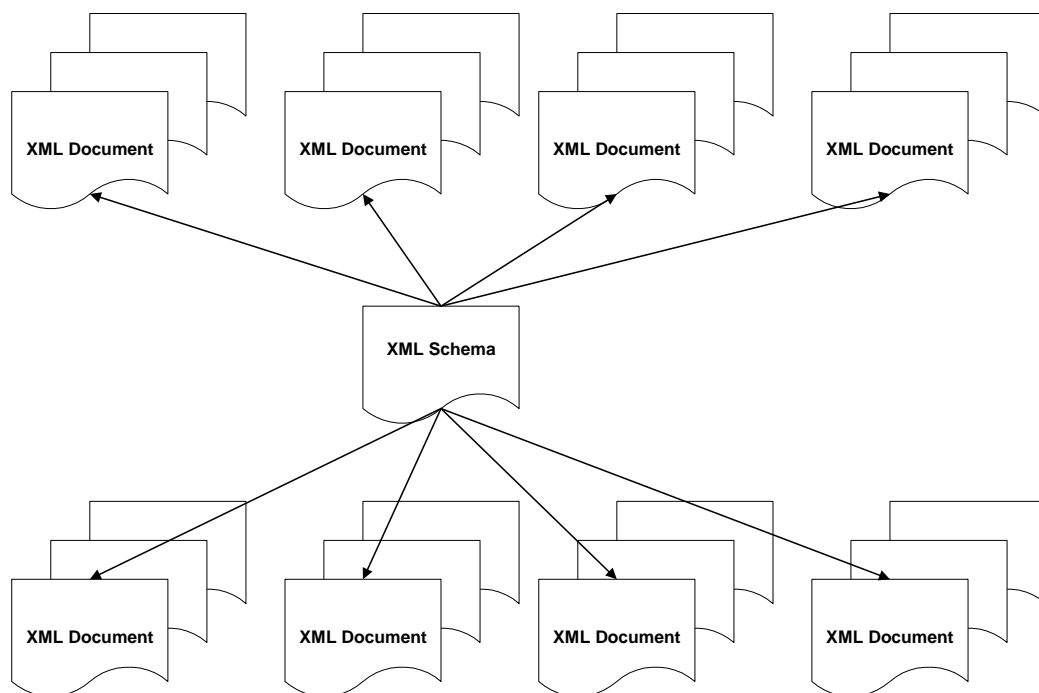

Figure 5.10: Changes to a single XML schema can force changes within many dependent documents.

External Dependencies

External dependencies emerge from two sources: business partners and publicly available standards. Customers, suppliers, and business partners need to collaborate through documents. In most organizations, communication is ad hoc and done through email. Through more efficient methods such as shared document repositories and semi-structured documents, the dependencies become more pronounced.

The trend toward XML-based business communications is clear. There are more than 60 technical committees within the Organization for the Advancement of Structured Information Standards (OASIS), a technical consortium dedicated to developing structured information standards that cover a range of business areas:

- Auto repair
- Biometrics
- Business transactions
- Conformance
- Customer information
- Digital signatures
- E-Government
- Emergency management
- Law (court filings, contracts, e-notary)
- Product lifecycle
- Provisioning
- Tax

These standards will be adopted as they evolve because of the efficiencies they bring. Organizations need to take care in selecting standards and versions of standards to ensure that the standards they choose are used by the business contacts with whom they communicate. These communication partners need to stay in synch as the standards evolve and new versions are released.

Content dependencies cross organizational and technical boundaries. Identifying and modeling them is the first step to successfully managing change in this domain.

MERANT

Developing Content-Management Policies

Policies for ECM address several areas:

- Roles, audit controls, and record retention

- Tagging rules

- Access controls and digital rights management

Staying in Compliance with Policies

Policies in content management are similar to policies in software development. They identify who (or which roles) can create, modify, and delete content; define templates and schemas; and define new shared folders and other management functions.

Audit controls and record retention ensure compliance with legal requirements and management practices. At a minimum, effective change management requires a system to track which user has made a change and at what time the change occurs. More effective is including a description of the changes as metadata associated with the new version. This metadata can be supplemented by a list of the actual changes derived from a utility for identifying differences between two files. These are progressively more complex controls and only the most important processes will warrant the most comprehensive procedures.

Record retention is also a form of compliance. The object of record retention policies is to ensure that legal and corporate records are kept as long as necessary without creating a save-everything-because-you-never-know-when-you-will-need-it culture. Well-defined policies, such as a schedule of how long different types of documents should be kept, are key elements of a well-governed ECM system.

💣 The failure to retain records has legal ramifications. Although it is not as popular a topic as other forms of compliance, it should be treated as a legal matter.

Maintaining Consistent Document Structures and Metadata

To ensure that document schemas and metadata are used consistently, develop polices specifying key features. Policies governing schemas should address which version of XML standards are used for a particular application, when and what controlled vocabularies are used for attributes, and procedures for changing schemas. Similarly, metadata policies should identify which attributes are required and which are optional; when controlled vocabularies are used for attributes; and what dimensions or facets are used for creating taxonomies.

Controlling Access and Protecting Digital Rights

Access controls are common in content- and document-management systems. They limit which operations can be performed on an object, such as creating a file in a folder and reading a document, and who can perform such actions. Less common but of increasing importance is the ability to manage digital rights. Digital rights specify how content can be used:

- How many users can access content concurrently

- How long content is available online

- How content can be redistributed and repurposed

- Which users or organization members can access the content

Specific rights will vary with content and providers but organizations should maintain policies to support the protection of digital rights. Policies should be defined to specify functional requirements in portals and other access systems that preserve digital rights.

> 📖 For a thorough overview of digital rights management and support architectures, see
> http://www.dlib.org/dlib/june01/iannella/06iannella.html.

Specifying Roles and Workflows

Roles and workflows are the final elements in content and document change-management models. Roles typically reflect common tasks such as:

- Creation

- Review

- Approval

- Administration

These roles, in turn, are used to specify who is responsible for steps in workflows. In addition to the lifecycle described in this chapter, workflows are based upon audit and review processes and administrative activities, such as archiving and backup. As with software development and system configuration, change management in the content- and document-management arenas is best understood and managed along the lines of the ECM model.

Summary

ECM takes a broad view of change within an organization. As we have seen in this and the previous two chapters, domains once limited to silo-based change-management approaches can actually be managed with a single model. Content, systems configuration, and software development share common content-management tasks and requirements. Managing assets has more to do with understanding assets, their dependencies, and their workflows than the low-level implementation details.

The next, and final, chapter explores ECM practices, tools, and benefits. We'll conclude with a look at the future of ECM—what will change and, just as important, what will stay the same.