KONNECT ECMS - TECHNOLOGIES AND XML GUIDE/DOCUMENTATION

1 KONNECT ECMS DATA MODEL DOCUMENTATION

1.1 Overview

Konnect Soft's Konnect eCMS Electronic Content Management System uses a data repository to hold all the data used by the system. The data repository consists primarily of a database or databases and can also consist of external files (stored in the file system and referenced from the database), and additionally it can contain links to external information sources (a URL is an example of this type of link).

The data repository is designed to be all encompassing and to be completely managed by the Konnect eCMS system itself, so that users do not need to worry about the technical details of how the data is stored. However where more technical users and system administrators may require these details, documentation on the system is available.

Basically the core of the system is the eCMS database which consists of a highly normalised collection of tables or database objects. Everything needed by the system to run can be stored within the database including the following:

- Configuration information:
  - ⇒ System configuration
  - ⇒ Per user configuration
- Import templates
- Delivery templates
- Logs and statistics
- Help
- Etc.

The configuration of the system for a client involves modelling their data in the system using a number of data objects and configuring the system to recognise their data model.

1.1.1 Note on XML/SGML

SGML and XML are treated as the same for the purposes of this document. When SGML is imported into the eCMS system it is converted into XML on import, and the mapping of elements to database objects works in the same way for SGML as for XML.

1.2 Comparison of relational database technology with a file system

Continuous or structured textual data is often stored in one of two very different ways:

- In a database of some kind
- As files within the file system

Data is traditionally often stored as files where technologies such as SGML, XML, Microsoft Office, etc. are utilised.

1.2.1 File based content storage

Storing data as files means that the data is essentially managed in the file system, often with directories, which is limited by the features of the file system. Often where publishers are storing content in files, the files can become very large which can be caused by the fact that it is awkward to split content across several different files (especially where there are cross-references, etc. within the content).

Large files can be very cumbersome to manage, and some applications can have problems opening them (sometimes users need extra memory in their workstations to be able to work with the files efficiently). Large XML or SGML files can take a long time to parse and to convert to different formats – when often the user will only want to take a small part of the file.

External files also create a greater risk of inconsistency, since they can be moved or deleted invalidating references (cross-references, etc.) to the file contained within the content repository.

The file system does not contain good features for re-use or navigation of content and often publishers end up with multiple copies of files, sometimes without knowing which version is correct. If the user wants to store multiple versions of a file, then essentially they need to make copies (or use deltas, etc.).

If a user wants to re-use part of a file in another file or publication, then this can be extremely difficult to do. They can reference the paragraph by an identifier (if it has one), but this often does not provide the type of re-use that users really need.

Sometimes the file management software improves the situation by hiding some of these limitations from the user, but essentially the software still needs to work with the restrictions of the file system, and so it cannot completely overcome these issues.

1.2.2 Storing data in the database

In relational databases, data is divided into records in different tables, and each record is usually identified by an identifier (often a number). In a database, data is managed in units of records rather than files, and records usually contain smaller units of data than are stored in files.

In the past relational databases where thought to be inappropriate for storing textual content because of technological limitations with the way databases stored large text records and the lack of maturity of usable systems and interfaces for such databases. The technology has rapidly advanced in this area and is now superior to other means of storage in many respects. Konnect eCMS not only overcomes the traditionally perceived problems with databases storing textual content, but offers overwhelming productivity advantages due to its advanced storage and retrieval model and extremely powerful backend system and interface for managing content.

A database based system can quickly and easily open, edit, copy and delete records and only the record (and possibly related records) needs to be read into memory. Relational database based systems are designed to optimally manage the memory and disk space which are accessible to them in order to maximise performance.

Whereas the file system has one main unit of storage which is the file (and directories), the database stores data in all the available tables which can each hold completely different sets of data. This provides a much more powerful and flexible mechanism for storing large amounts of content. Database management systems also have built in backup and disk management facilities to manage data backup and growth.

It is possible to store entire files in records either as textual data (for XML, SGML, etc.) or as binary data (for non-textual files, images, etc.), however Konnect eCMS does not use this approach. In the eCMS system content is separated into a highly normalised set of tables (called database objects) and different data types are stored in different sets of tables. The data that might have been stored in one big XML or SGML file under a file based system is split amongst many records in several tables in the eCMS system.

The eCMS system utilises database technology to support extremely large amounts of data growth. No matter how large a publisher’s data set becomes, users are always only working with a selection of records and the total amount of data in the system does not affect its performance. In a file based system the performance can deteriorate as a file grows, but in the eCMS the performance remains constant as more records are added because users are still working with the same size units (records).

1.2.3 Server based system

The more powerful relational database based management systems are server based, which means they run on a server and all the processing to load, save, create and delete content is performed on the server. A server based architecture is preferable for managing content because powerful server hardware can be used and the individual users of the system do not need such powerful hardware (as they run as clients of the system).

The server based system can provide better performance and can be centrally managed, upgraded and extended. This allows the publisher to deal with large amounts of content effectively (i.e. it would not be effective or manageable to have to upgrade all client machines needed to handle growth in content or improve performance).

1.2.4 Importing XML/SGML data

Generally in the eCMS system where XML data is imported into the system, XML elements are imported into database fields in different tables. In some cases an XML element will correspond directly to a field, in other cases it may be split amongst several fields. For continuous text content, often an XML element and all its contents (including child items) will be imported into the database record used for continuous text content.

1.2.4.1 Fusion between database records and XML content

In this way the eCMS stores valid XML fragments within its continuous text content records, providing a fusion between database records and XML content.

This provides the best of both worlds together in one system, XML based content and relational database based management. The data can be combined into files by the system on delivery, for example, to also support delivering XML files to other systems/software.

1.2.5 Dynamic XML structures

One advantage of XML is that it is hierarchical and contextual, and supports dynamic data structures. So, for example, a heading element could be used in a chapter element, or a section element, or a paragraph element and its meaning can be different depending on the context. This often does **not** map precisely to a fielded database based structure.

Where there is XML content that does not map precisely to the database objects, this is handled in Konnect eCMS by storing the XML content within the text records as well as splitting it amongst different records – so dynamic markup is preserved within the text. For example if there is a specific element called <journalreference> which is applied to the textual content in many different locations in the XML, this is imported into the continuous text record along with all the textual content (in this case it is transformed into an eCMS special reference element within the text). See the diagram in section 1.7.4 for more details.

1.3 Konnect eCMS database objects

The system consists of a number of core database objects composed of application tables. These tables are used by the application to hold configuration settings, etc. which are needed by the system to function. The system also contains content objects/tables which contain the actual content managed by the system.

The content objects can vary depending on the configuration for different users, however some content objects are always present. All objects in the system are identified by an ID which allows them to be re-used or referred to in different contexts.

The taxonomy and continuous text content objects are always present and hold the structure and textual content for the system.

1.3.1 Taxonomy/hierarchical tree

The taxonomy objects consist of tables which contain the hierarchical representation of the data within the system. This data is used by the eTOC/taxonomy component to display the table of contents view of the publisher's/user's data. For example it will show the division of a book into chapters, sections, sub sections, etc. (see Editorial System - Editorial table of contents / taxonomy editor in the Konnect eCMS Product Description).

1.3.2 Re-use of data

The taxonomy objects contain references to the content in the tree/table of contents – the data is actually stored separately. In this way a record can be used in multiple locations in the tree but is only stored once, its ID being referenced in each location in which it is used.

This method importantly separates the content structure from the content itself and allows users to build up different publications or taxonomies containing any number of different combinations of their content.

1.3.2.1 Power of the Konnect eCMS model for content re-use

The re-use functionality within the system is very powerful because data can be re-used down to extremely small units if desired, which gives the most flexibility for re-use. However the publisher/editor can control the size/granularity level of records, so they do not need to work with small records if this is not desirable.

Users can very easily create products or items (chapters, sections, etc.) which consist of the best possible mix of re-used and new content. Content can be re-used in as many different items/locations as is needed, and the same content can be re-used at different levels and presented in a different way in different publications.

The system is flexible, easy-to-use and powerful, allowing the publisher to actually achieve the previously difficult goal of easily repurposing content for multiple products and output formats.

1.3.3 Meta data

Meta data is stored in a collection of database objects which are associated with the content/records that it applies to. This means that rather than recording meta data such as:

- Description
- Author
- Publisher
- Etc.

…. in the text itself, it is separated out from the textual content and linked to the content records to which it applies. This allows the same meta data to be linked to a collection of items and allows meta data to be processed and queried to generate reports, or delivered to different formats, etc.

1.3.4 Content items

The system contains different types of content items for different types of data, some examples are:

- Headings/classifications
- Continuous text
- Images
- External files
- Directories model content:
  - ⇒ Organisations
  - ⇒ Contacts
  - ⇒ Products
- Users/accounts
- Groups
- Delivery templates
- Import templates
- Etc.

Each different record type consists of a different combination of fields and different relationships. For example images can have associated information such as resolution, description, type, format, creator, purchased from, web link, caption type, caption text, etc. All of the content items can appear in the eTOC/taxonomy for the system

Content items can be created and configured to meet the specific needs of the publisher or data model.

1.4 Multiple user support

The way that the system splits the content into multiple records provides very powerful support for multiple concurrent users working on the content. When a user edits a record, it is checked out and other users are locked out of the record and unable to edit it until the user finishes editing it and it is checked back in.

Different users can work on different records within the same section, or sub section at the same time. In this way different users could even be working on different consecutive paragraphs under the same heading if the paragraphs where in separate text records. Also the system allows a user to check out a heading and all its contents if they wish to prevent other users from working on any of the content at the same time as themselves.

This is a far superior scenario to that of many file based systems, where one user works on a large file and no one else can work on the file until they close it.

1.5 Continuous text and XML

The Konnect eCMS system stores continuous text data in the continuous text objects/tables in the data repository. The textual data is divided into records (it is often split on headings or to the paragraph level) and each record contains XML markup. The XML represents the structural divisions within the content and is often used to attach styles and formatting to the content.

1.5.1 XML within Konnect eCMS

The type of XML used for markup within continuous text records by the Konnect eCMS system is modular XHTML[1] (which follows the World Wide Web Consortium - W3C - standard) with Konnect Soft extension modules for special elements and system data, such as system tags. System tags are used for cross-references, index terms, footnotes and special references.

In this way the XML used employs some XHTML modules but not the entire XHTML DTD. Where appropriate the system can be configured to handle extra XML elements from the publishers content by including additional XML DTD/schema modules for additional elements which should be included.

Accordingly the core syntax is XHTML with XHTML elements for:

- Structure:
  - ⇒ Paragraphs
  - ⇒ Grouping elements: the DIV and SPAN elements
  - ⇒ Line breaks
- Formatting:
  - ⇒ Bold
  - ⇒ Italics
  - ⇒ Superscript
  - ⇒ Subscript
  - ⇒ Etc.
- Hyperlinks
- Tables
- Lists:
  - ⇒ Ordered
  - ⇒ Unordered
- Images
- Styles (the class attribute is used to indicate that a style applies to an item)
- Etc.

Each continuous text record in the system contains XHTML formatting but does not contain the structural tags such as <HTML>, <HEAD>, <BODY>; or special items like <FORM>, <SCRIPT>, etc.

The eCMS system requires valid XML and is currently set up to support the display and editing of XHTML with specific extensions. The content editing area is configured to display an easy-to-use WYSIWYG view of the data in which users can easily edit content without having to worry about the structure of the data. The system ensures the validity of the data and inserts the correct markup.

[1] XHTML is an XML version of the HTML DTD which is good because it is a standard, well know and easy to understand. Modular XML breaks the XHML DTD into different modules for different features, such as tables, formatting, etc and can be extended with additional custom modules.

This type of XML is only used for specified markup with records, a lot of information being stored in other related records as fielded database data rather than being stored using inline markup. Examples of this are style definitions, meta data, headings, general notes, footnotes, and much more which are stored in separate tables rather than being stored in the text in the record.

1.5.2 Tables

The eCMS system stores tabular data in XHTML tables.

The XHTML table format is a powerful and a relatively easy to understand representation of tables which handles advanced table features.

1.5.2.1 Comparison of CALS and XHTML tables

One of the major competing models for table markup from the XML/SGML world is the CALS table model (http://www.oasis-open.org/specs/tm9502.html). The CALS and XHTML table models are actually not so different, and for XHTML tables utilising CSS directly supports 95% of CALS table markup - the major differences between the two models are the tag names and the way spanning is handled. The remaining 5% of the markup is handled by using alternative mechanisms such as nested tables.

XHTML tables support nested tables which allow very complex table structures to be built.

XHTML tables support some features not present in CALS table such the ability to specifically associate cells with their headers, whether those headers are column headers, row headers, or even header cells at arbitrary locations in the table (http://www.w3.org/TR/html4/struct/tables.html#h-11.4).

CALS has some elements such as **tgroup** and **title** which are not directly present in the XHTML model, but which can still be fully handled using other equivalent items - see the table below.

Description	XHTML Element	CALS Element	Comments
Container for table elements	table	table	
Caption	caption	title	XHTML tables do not have a title element, but the caption element is similar (“When present, the CAPTION element's text should describe the nature of the table”)
Wrapper for column specs and content	Not available	tgroup	XHTML tables do not use sub sections of a table with different column specifications, on import/migration - this is supported using nested tables
Column specifications	col, colgroup	colspec, spanspec	The spanspec element specifies horizontal spanning (joining) of cells
Wrapper for header rows	thead	thead	
Wrapper for body rows	tbody	tbody	
Wrapper for footer rows	tfoot	tfoot	In both XHTML and CALS, tfoot must appear before tbody
Row	tr	row	
Cell	td, th	entry	

Note that the XHTML table DTD is the same as HTML 4 tables, but following the rules for XML compliance. (Also see the XHTML table module for a good summary of the elements for a table.)

1.5.2.2 Will I suffer data loss when my data is migrated?

No data whatsoever is lost when migrating CALS tables to XHTML tables using the mappings defined by Konnect Soft.

1.5.3 System tags

System tags in the eCMS system handle:

- Cross-references
- Index terms
- Footnotes
- Special references
- Etc.

The tags refer to other database objects which store the information about the item (also see section 1.5.4) and support advanced functionality. Storing the system tag information in separate database records supports advanced functionality such as validation, re-use and reporting. It allows the system tag information to be processed without needing to process all the textual content which has much better performance and scalability.

This is a big advantage of the database model over the file system model where cross-references and other types of elements, which need to be validated across files, are not handled well.

1.5.3.1 Cross-references

When a cross-reference is inserted it refers to a special cross-reference record which holds the source and destination information for the cross-reference. When the links are checked the system processes the cross-references

1.5.3.2 Index terms

Index terms tags refer to index term records that have their own index taxonomy or classification/table of contents. Users can easily build up their own index term hierarchy and adjust and classify index terms with the index. The index is easily previewed and validated by processing the index term records.

1.5.3.3 Footnotes

Footnotes are stored as separate special continuous text records which can contain normal XML markup including cross-references. When a footnote is inserted into the content, the XML tag insert links to the ID of the footnote record. When delivered, footnotes can be deployed as:

- End notes
- Popup windows/layers
- Linked items
- Etc.

1.5.3.4 Special references

Special references are references to items such as legislation, legal cases, scientific journals, literature, documents or other items that have conventions which govern how they are referenced or cited. Special reference tags refer to special reference records for the reference items.

1.5.4 References from the content to database records

From within continuous text records in the eCMS system, users can insert links/references to other records/objects within the system. This is inserted in the text of the record as a pointer or reference to the database item being referred to. This allows the user to reference existing items in the database in the middle of a text record without needing to repeat the data.

1.5.4.1 Inserting a reference to an external file

Inserting a reference to an external file or image is one example of this. When a user inserts an image into the text, they can select an image from the available image records and insert it into a point within the text. When doing this users are actually inserting an image element which has a pointer to the ID of the image content item/record in the system. In this way the image can be re-used in multiple records and all instances refer back to the same image item.

This can also be done with external files such as PDFs or an Excel spreadsheet (containing financial calculations or charts for example), or any database item in the system (for which the system is configured to allow insertion).

When the content is delivered, these referenced items can be delivered/copied to the location in the text in which they appear. If one is delivering to XML, images or external files are delivered/copied into sub-directories and the file names are referenced (with a relative path) from the delivered XML file.

1.5.5 Granularity

[Extract from the Konnect eCMS Product Description]

Granularity is the relative size, scale, level of detail or depth of penetration that characterises an object or activity. In the content of database records, the granularity specifies how much data a record contains. It does not determine the exact amount of text, but rather whether a record corresponds to a line of text, paragraph of text, page of text, etc.

The data repository is based on relational database technology where data is stored as records. Headings and data that were previously part of a document are stored as separate records in the repository. This gives the publisher the maximum flexibility to categorise, re-use and structure their data.

The granularity of text records determines how much and what type of data is stored in a record. Records are the smallest assignable unit, which means one can re-use an entire record in multiple publications or areas (but one cannot re-use part of a record). The granularity of records in the content management system is defined as the text between headings (i.e. text records never contain a heading).

A record can contain a line of text or a page of text - the size is defined by the user. Whenever a heading needs to be inserted, a new heading record is added and a text record is inserted below the new heading record. Multiple text records can be attached to each heading record to allow editors to divide content in the exact way they desire to facilitate re-use of content.

1.6 Consistent format, single interface and integrated functionality

The data model and XML DTD used by the Konnect eCMS system support a consistent user interface to all content and allow the system to provide all the content management tools needed in one environment. One of the goals of the eCMS system is to eliminate the need for users to employ a wide variety of tools to manage their core content. Accordingly, the system provides integrated facilities for functionality such as:

- Search
- Delivery
- Copy and re-use
- Printing
- Cross-references
- Etc.

…. due to the consistent format and data model used. All data can be managed in through a single powerful consistent interface.

1.7 Importing XML/SGML data and delivering XML/SGML data

1.7.1 Overview

Import and delivery of data works in similar ways, they are simply the reverse of each other.

An import maps external content to objects in the eCMS system, and delivery maps objects in the system to external content. Import and delivery templates essentially control this mapping between eCMS objects and external content.

XSLT can be used for both import and delivery.

1.7.2 Mapping XML/SGML to the eCMS data repository

When importing XML or SGML, each element and attribute in the imported content or in the DTD for the content is mapped either to an eCMS database object or to elements in the extended eCMS XHTML DTD. Under certain circumstances, some new tables or fields may need to be configured within the eCMS to handle some of the client’s data - for continuous text data the textual content maps into the core continuous text tables and the hierarchical tree tables.

By mapping every element and attribute to database tables, fields or elements in the eCMS DTD, one ensures that no data is lost on import. The import template contains the mapping between the content being imported and items in the eCMS.

Facilities are available to check a file or DTD against an import template to ensure there are no unmapped elements.

1.7.2.1 Importing the heading structure

Where the imported content contains a heading hierarchy, the headings are mapped to the hierarchical tree tables in the eCMS system. The heading hierarchy is imported by examining the element hierarchy and attributes for headings in the XML, so that once imported, the heading hierarchy appears in the eTOC/taxonomy.

It is also possible to import headings into a continuous text record in the eCMS. Headings imported in this way do not appear as headings in the eTOC/taxonomy but are part of the textual content in a record. This option is usually used when a publisher does not wish to split the content into records right down to the lowest level of heading.

1.7.2.2 Importing the textual data

The textual data in the XML file is imported into continuous text records based on the level of granularity defined within the import template. Often records correspond to paragraphs in the content being imported, but they can correspond to large or smaller units. In this way within the import template users specify on which element or elements (and in which context) the content is divided into records, and when the import processes encounters the specified element in the content being imported, it creates a record in the eCMS system and inserts the element’s contents into the record.

Mark-up inside the elements designated as records needs to be mapped to the appropriate eCMS content elements, meta data, or to a system tag such as a cross-reference or footnote.

1.7.2.3 Importing cross-references

Cross-references in the content being imported are imported as eCMS cross-reference system tags and records. The import functionality supports the import of a file with cross-references to another file which has **not** yet been imported. When a cross-reference with a destination not currently present in content is imported, the system remembers its destination, and if the file containing the destination is later imported then the cross-reference is linked to the imported destination.

1.7.2.4 Importing footnotes

Footnotes or end notes are imported as eCMS system tags and footnote records. Any elements in the footnote content are mapped to items in the eCMS DTD.

1.7.2.5 Importing images

Images are imported by mapping any elements which refer to external images to the eCMS images object. The import component searches for images using any path information specified in the image element, and then can search in a list of alternative directories if an image is not in the default/specified directory. Image meta data (such as caption, format, etc.) is mapped to fields in the image content item.

1.7.3 Handling attributes

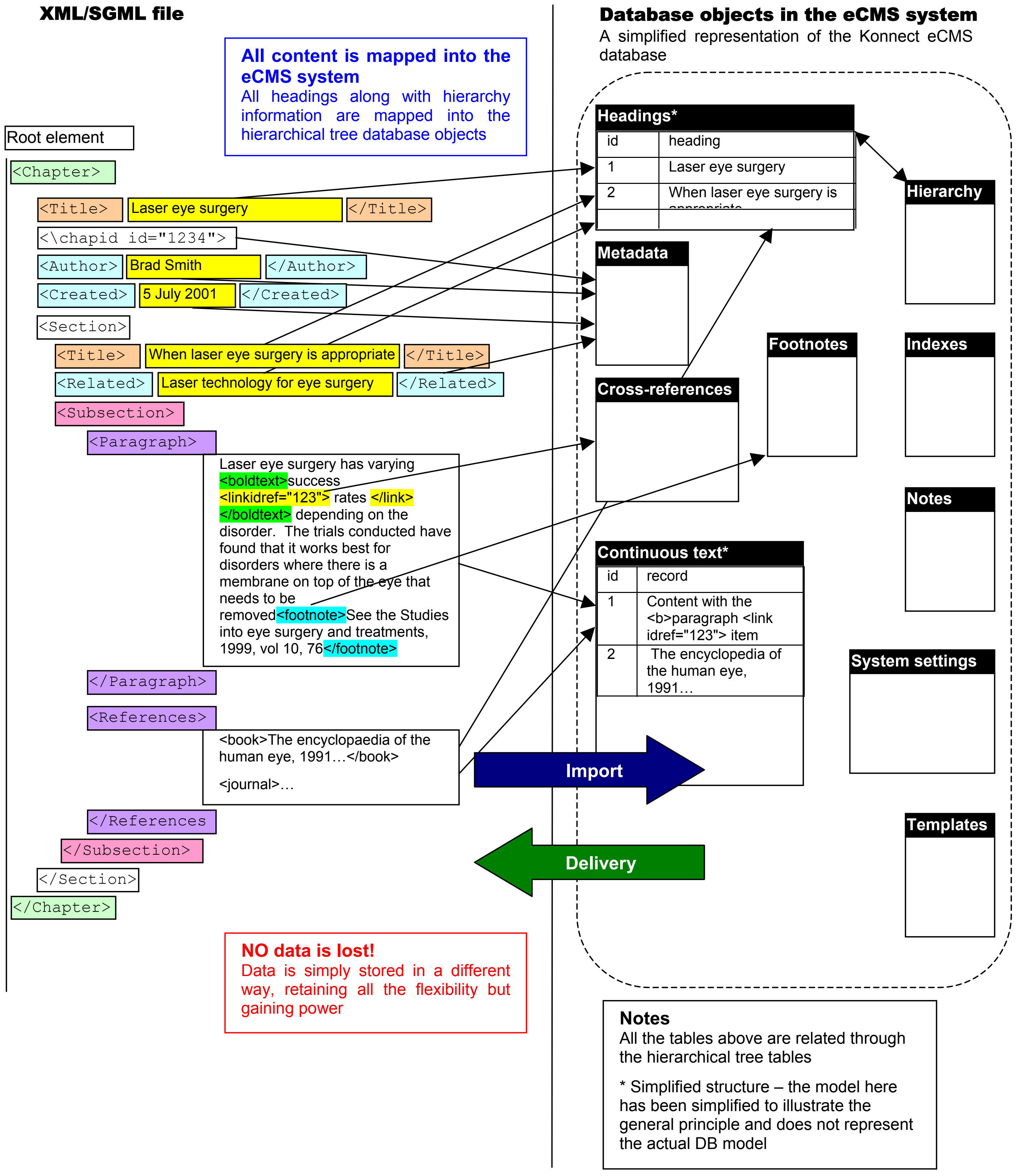
Attributes in the source XML/SGML can either be mapped to fields in the eCMS system, such as for record meta data (in this case each attribute is mapped into fields within meta data database objects/tables). Alternatively if the markup is dynamic and tied to a specific point in the text, then it is preserved in the text (usually in the body of a continuous text record) and the user manages the data through the eCMS interface by clicking on the text/element and viewing and editing its properties.

Textual records in the eCMS system can contain unstructured markup or references to other records (using the system tag mechanism) containing further markup - this allows many records to refer to the same entity without content duplication.

See also section 1.2.5.

1.7.4 Diagram: XML file – Konnect eCMS

(Microsoft Word drawing must be viewed in **Print Layout View** in Word)



All content is mapped into the eCMS system
All headings along with hierarchy information are mapped into the hierarchical tree database objects

Laser eye surgery has varying <boldtext>success <linkidref="123"> rates </link> </boldtext> depending on the disorder. The trials conducted have found that it works best for disorders where there is a membrane on top of the eye that needs to be removed<footnote>See the Studies into eye surgery and treatments, 1999, vol 10, 76</footnote>

<book>The encyclopaedia of the human eye, 1991…</book>

<journal>…

A simplified representation of the Konnect eCMS database

id	heading
1	Laser eye surgery
2	When laser eye surgery is appropriate

id	record
1	Content with the <b>paragraph <link idref="123"> item
2	The encyclopedia of the human eye, 1991…

NO data is lost!
Data is simply stored in a different way, retaining all the flexibility but gaining power

Notes
All the tables above are related through the hierarchical tree tables

* Simplified structure – the model here has been simplified to illustrate the general principle and does not represent the actual DB model

1.7.5 Validating/checking the import

The import component validates data on import and generates reports showing a summary of all information about the import. The report includes:

- Number of files imported
- Number of XML elements imported
- Amount of bytes imported
- Number of records created in the eCMS
- Size of content created in the eCMS
- Special items imported such as:
  - ⇒ Number of images imported
  - ⇒ Cross-references imported
  - ⇒ Broken links in the content/cross-references with no destination
- External items (images, etc.) not found

The report is created in HTML and users can click on individual items to see more details and drill down to check specific items. Users can view the import statistics for an individual file or for the total content, and can click on a link in the report to jump to the content in the eCMS. A more detailed technical import log is also accessible for use by system administrators.

Imports can be cancelled at any time and the file being currently imported is deleted, however the previously imported content remains (there is a mode where the entire import can be rolled back but this only supports smaller import volumes). If the import fails or there is a crash during an import (due, for example, to a power cut), files which have been fully imported before the crash will remain imported.

1.7.6 Delivery from the system

Delivery uses a mechanism similar to import where database objects and eCMS XML content are mapped to items in the required output format. On delivery, the data within the system can be transformed into any output format desired.

The data can be delivered to different types of XML, SGML or other text-based content by configuring the delivery template for the desired type of output. During the delivery, XSLT can be used to easily change the names of elements, remove or combine elements and perform more sophisticated actions.

1.8 If you already have your own DTD

1.8.1 Overview

Publishers often have a substantial investment in their own DTD to which they can have import, delivery and integration processes attached.

There can be a reluctance to leave behind their DTD and switch to using the eCMS system. However moving to the eCMS system is designed to improve efficiency and functionality for a publisher while **still supporting everything available under the existing XML/SGML based system**.

Additional XML elements can also be configured for the publishers data where required.

The eCMS system imports every element and attribute in your data set and provides you with much more powerful functionality to manage the content. **No content is lost on import** – the content is simply stored in a structure different from that of an XML file based system. The eCMS data structure allows just as much flexibility as the XML file based system, and is in many, many respects more powerful.

1.8.1.1 Provides better functionality than XML file based systems

Using the eCMS system to manage content provides better functionality than XML file based systems, for example:

- Re-using content in multiple publications/items
- Working with large amounts of content
- Multi-user access
- Classifying content
- Delivering content
- Importing content
- Navigating content
- Building and managing heading and publication structures
- Editing content generally
- Managing cross-references
- Copying and moving content between publications and items
- Searching content
- Validating cross-references, etc.
- Applying meta data to content
- Managing content workflow
- Etc.

This increased functionality is provided by splitting the file based data into multiple database based objects and the use of the eCMS system's internal DTD. So by using the eCMS system data structure for content rather than their own DTD, publishers gain more power and flexibility to manage their content.

Users are still able to import from and deliver data into their own DTD, however editorial users no longer need to use cumbersome SGML/XML editing tools to edit the data. Once import and delivery templates have been set up and validated, content can be imported and exported at the press of a button and users do not need to worry about the process.

1.9 XML support in databases

1.9.1 SQL Server XML support

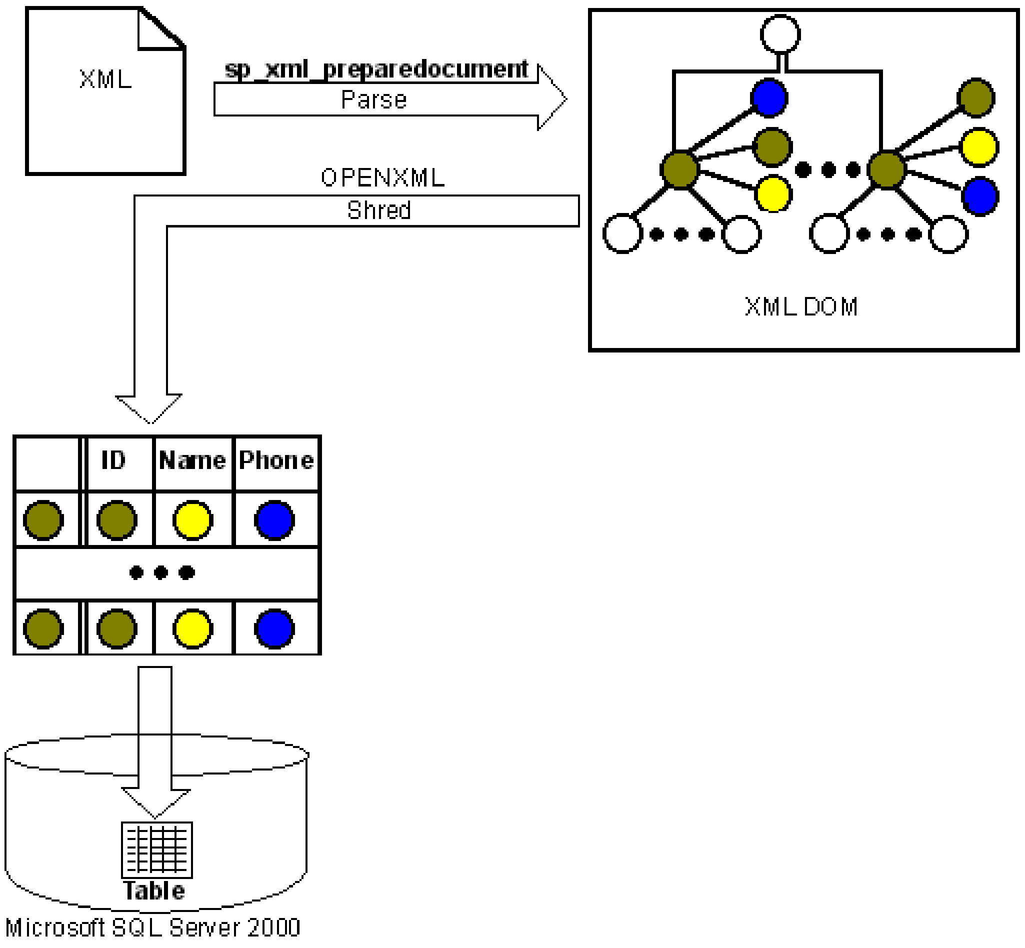
SQL Server 2000 supports a range of XML features including the following:

- Returning the results of a query against the database as XML
- Reading XML data and updating the database
- Creating XML database views with XDR schemas
- Using XPath queries to access XML data

It provides an XML application program interface (API) for accessing the data in the database and supports advanced XML functionality for querying, importing and delivering XML data. This also allows the eCMS system to work with data treating it as XML and ignoring the database structure, and also supports the long term extensible development of the system.

All of the functions available in SQL Server can be utilised in the Konnect eCMS system - see section 1.9.3.

1.9.1.1 SQL Server



1.9.2 Oracle XML support

The more recent versions of the Oracle database (9i onwards) have an application program interface (API) and features that allow users to store, query and update data as XML. Oracle possesses the features that are available in SQL Server 2000, such as XPath access and a number of others.

Oracle is also frequently used to store XML content within text type fields in the database - which is how Konnect eCMS operates, but utilising instead Microsoft SQL Server. With a system such as Konnect eCMS, the actual database used at the backend of the product does not affect users, and the difference between SQL Server and Oracle for the type of storage Konnect eCMS uses are minimal.

1.9.3 Konnect eCMS and XML databases

The Konnect eCMS system is essentially an integrated XML database and user interface which stores XML content and allows users to import and deliver XML content. The user does not need to worry about the access and management of the XML data which is handled by the eCMS. This is one of the great benefits of the eCMS system - that it provides an interface which abstracts away the complexity of the XML, but still allows users to obtain the features and benefits of using XML (users have the comfort of utilising a Microsoft Word-like interface).

Both fielded structured XML and unstructured XML content can be stored in the eCMS.

Unstructured XML in this context means XML where elements can occur within different locations in different order and with different hierarchies in the text. XHTML is an example of this, where style/formatting mark-up can be applied in any order to any parts of the text. Most types of XML which represent article or publication content (especially book content) are unstructured XML in this sense.

Fielded structured XML is XML that has a regular structure of elements or attributes appearing in the same sequence – such as with name and address data for individuals or organisations.

Fielded structured XML content is mapped into tables and fields within Konnect eCMS where each XML element may be mapped into a distinct database field. A unit of unstructured content (dependent on the chosen level of granularity, a unit may be a paragraph to a page of content) is stored within a single Konnect eCMS record - this applies primarily to continuous text records, but also to other records such as footnotes, meta data and general textual fields.

One of the great strengths of the eCMS system is its eTOC/taxonomy component and the way it stores data in this hierarchical tree representation fully supporting re-use and content structuring. Oracle for example provides features to allow XML data to be accessed/viewed in a type of file folder structure while still being stored in the database. The Konnect eCMS eTOC/taxonomy functionality is far superior to this as it supports a hierarchical representation fully integrated into the system, providing much more power than the simple folder/directory based paradigm.

Whatever system is used to store the content, it should have an interface that hides the complexity of the storage format, while providing users with the power and flexibility contained in the underlying system.

Konnect eCMS possess these features and is built on a solid foundation of XML and database objects.

2 MIGRATING CONTENT FROM SGML/XML TO KONNECT ECMS

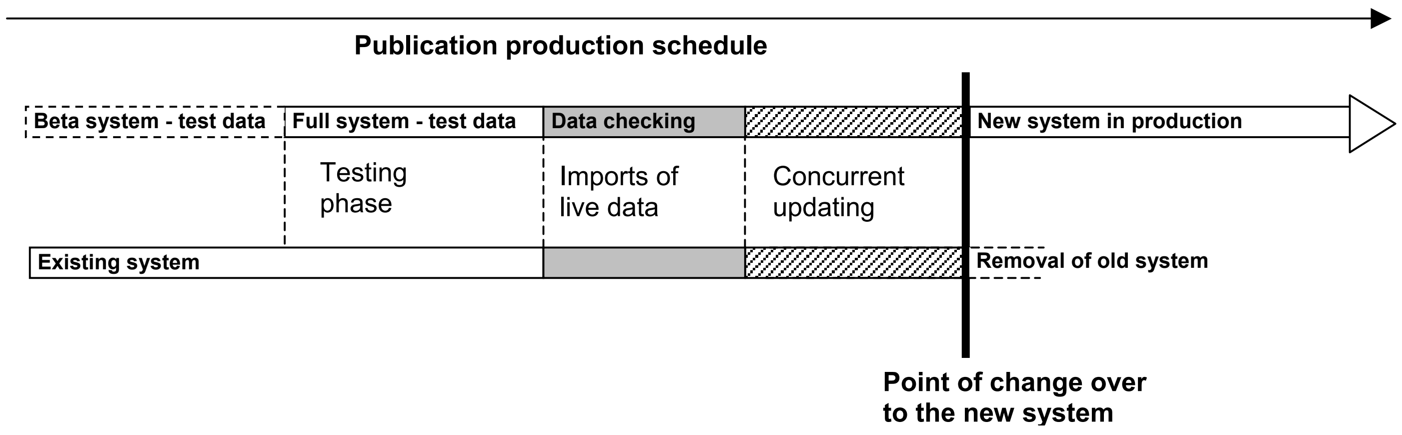
2.1 Overview of data migration

[Extract from the Konnect eCMS Product Description]

Konnect Soft provide systems that are designed to minimise the impact of migration, for example for transferring data from existing systems/legacy systems to the new content management system. This is achieved by allowing partial import of data for testing purposes and running the systems concurrently. Client staff are then able to use the system with test data to learn, experiment with and test the functionality before the change over to the new system.

Once the testing is complete and the client is satisfied with the new system, then a full import of the data from the old system into the new system is performed. The client verifies that all data has been imported correctly and, as circumstances dictate, some or all data can be re-imported. Often it helps the client to import some data before deciding on the precise import configuration. Once it has been verified that all data is present in the new system, then a period can follow where the old and new systems are concurrently updated until the client confirms that full transfer to the new system.

This process ensures that no data is lost and gives several levels of fault tolerance. Konnect Soft aim for a single live import of the data set, but fail-safe mechanisms exist to ensure data integrity and that the client is fully satisfied before they start using the new system.



2.2 Migrating delivery

To remove any risk to the client in terms of them being at all times able to deliver their content, Konnect Soft migrate the delivery scripts for the client at the same time that they import the content. Konnect Soft then deliver a system to the client that has all data imported and that also delivers all required outputs.

This means that the publisher can switch to the new system and know that they can go on producing their publications with minimal disruption - providing the safest form of migration. The publisher is able to preview the migrated content with delivery in a staging environment and test the delivery before they perform the migration in their live environment.

2.3 Importing and delivering XML/SGML data

See section 1.7.2.

3 Delivery Templates

3.1 Overview

The delivery component uses a template and a selection of content to produce delivered output. The structure and format of the delivered output is determined by the delivery template, and the actual content that is delivered is determined by the selection of content passed to the delivery component.

Within the eCMS system users click on the item or publication they want to deliver in the eTOC/taxonomy and then select the template to use to deliver the content using the selected template. Alternatively a user can run a search and select a template with which to deliver the search results. Other types of delivery, such as web delivery, often run on a scheduled basis using templates configured as the web delivery templates.

3.2 Delivery template structure

Delivery templates consist of the following items:

- Optional SQL statement
- Core eCMS template section:
    - ⇒ Headers
    - ⇒ Footers
    - ⇒ Structure per record
    - ⇒ Replacements/filters
    - ⇒ Conditional statements
    - ⇒ Etc.
- XSLT
- Links to other external components
- Post processors

3.2.1 Optional SQL statement

Delivery can contain an optional section holding a SQL statement or stored procedure, which determines which content is delivered. If this section is not specified, then the selection of content is passed to the delivery component dynamically by the system when the delivery is run.

3.2.2 Core eCMS template section

This is the core template section that specifies how the database content is mapped to text/markup within a file. It specifies which database fields are included in the output and what markup appears around each field or group of fields. It can also contain a lot of powerful functionality such as replacements for specified text, grouping for delivered items, headers and footers, etc.

The core template section can be configured to produce all types of output including:

- XML
- SGML
- HTML
- Folio Views - Folio flat file format
- RTF
- QuarkXPress Xtags
- CSV or other delimited format
- And many other formats…

In many cases, the core eCMS template section is all that is needed. However if the user is delivering into XML or SGML formats, it is often useful to define the delivery using an XSLT transformation.

3.2.3 XSLT transformations

The XSLT transformation section of the delivery template is optional, and can only be used if the eCMS template section produces valid XML.

The default configuration for the eCMS template section automatically produces valid XML, so that XSLT can be utilised (unless the user changes the core template section to produce a non-XML format such as CSV).

The XSLT can contain functionality implemented in scripting languages, such as VBScript, to provide even more powerful functionality for performing complex operations on the data as it is delivered.

3.2.4 Links to other external components

The template can contain links to external components which can be called to process the content as it is delivered. For example an external component that changes the colour depth of images linked to the content could be called whenever images are processed. This mechanism allows users to incorporate powerful third party functionality into the delivery process.

3.2.5 Post processors

External applications can be automatically called on the completion of delivery to further process the delivered content.

3.3 Interface for the delivery scripts

An interface can be configured for the delivery component to allow users to adjust the template without the need to view or understand the full details/syntax of the template.

The interface allows users (e.g. non-technical editorial users) to select database fields from the eCMS system and configure the output text generated for the fields (for example the markup or general before and after text for each field).

4 TECHNOLOGIES

4.1 Standards

Konnect Soft follow published standards for development related technologies to achieve best possible compatibility and support from applications. This includes SGML/XML related field of technologies, for example, HTML, CSS, XHTML, XML, XSL, etc.

Konnect Soft develops code that correctly implements the standards for HTML, CSS, XML, etc.

Organisations whose standards we monitor include W3C, ISO, and IEEE.

4.2 Microsoft technologies

The main Microsoft technologies employed by Konnect Soft for the implementation of our products are listed below.

The technologies are selected as the most appropriate to our system architecture and include:

- Windows 2000/2003 Server
- Microsoft .NET
- COM+
- ASP
- Visual C++
- ActiveX
- ADO/OLE DB
- SQL Server 2000
- SOAP
- Web services
- XML
- XSLT
- HTML/CSS
- Internet Explorer

5 USE OF KONNECT ECMS BY THE CLIENT

5.1 Metamorphosis scripts

Konnect Soft have a substantial amount of SGML expertise and experience, including experience with SGML parsing languages such as Omnimark and other tools such as Perl.

Accordingly, we are, for example, able to take Metamorphosis scripts and convert them into delivery templates.

The business intelligence within the templates is re-used with the templates becoming eCMS delivery template format (largely consisting of XSLT).

For some of the other script types such as the HTML and Folio Views flat file scripts the exact output data could be re-used, it is simply copied and pasted into the relevant section in the new templates (and associated with database objects rather than SGML elements).

This means for the client that the eCMS system builds on the investment made in the existing scripts rather than requiring them to be re-constructed from scratch.

5.2 Support for dictionaries

Dictionaries are supported in the Konnect eCMS system by the eTOC/taxonomy control with advanced taxonomy features. Each term in the dictionary appears as an item in the eTOC/taxonomy (which is a special record type) and the definition/description appears as a text record attached to that item (and possibly attached to several items where they are synonyms).

Related taxonomy items/related record features allow users to create links between different taxonomy items/classification – to designate that these items are related. This can be used to handle multi-language taxonomies by setting up relationships between instances of the same word in different languages in the taxonomy. In this way users can view all of the related taxonomy items for a term and click on a link to jump to the selected item in the eTOC/taxonomy.

Managing a dictionary uses the data explorer eTOC/taxonomy functionality where, rather than manually ordering the data in the eTOC/taxonomy by creating headings, etc., data is explored based on different values. So for example when one expands a dictionary in the eTOC/taxonomy, you see all the letters of the alphabet, which can then be expanded to display all the different terms under each letter. Or one can have further categories such as AA, AB, AC for large taxonomies where expanding AA shows all the terms that start with AA, and so on.

This functionality handles: related terms and SEE/SEE ALSO type references which can be managed as hyperlinks in the text. Synonyms can be managed using the re-use/insert reference functionality in the eTOC/taxonomy.

6 APPENDIX DATA MODEL/DATABASE RESTRUCTURING & MIGRATION

Before the Konnect eCMS system is installed for a client, the data model design is extended and configured as required. This is done, for example, if there are particular fields which the client needs in their data set which are not present within one of the standard eCMS data model configurations.

The Konnect eCMS system is purposely designed so it can link to different data models/databases. This allows a data model to be configured which is truly optimised for the client's content, and allows elements of the client's data model to be taken into the system where either necessary or useful.

Konnect Soft also has generic directories and continuous text models which contain a large set of highly normalised and designed database tables with common database elements - usually most client data is mapped or linked into these tables.

This process for creating/updating the new/modified data model from the client's data model may be as simple as configuring a few meta data fields or adding a few fields to a table. On the other hand, it may also involve going through some or all of the following processes:

- All tables from the client data model are merged and combined with the Konnect eCMS data model
- Tables from the three existing databases are mapped to the new database
- The design is normalised and optimised
- The new database is documented
- The Konnect Soft system and application tables are installed
- The database is indexed
- The database server settings are configured

Once the new data model has been created, then an import template is configured to import the data from the existing databases into the new database model which needs to:

- Split records into a normalised structure
- Map data into new tables and fields
- Process all tables and fields
- Check/validate the imported data
- Apply replacements, rules, or transformations to the processed data
- Produce a log and report displaying the results of the import:
  - ⇒ Records imported per table
  - ⇒ Source and destination tables
  - ⇒ Errors or warnings
  - ⇒ Total records processed
  - ⇒ Other quality assurance statistics

After an extensive quality assurance phase, a full import/migration of data from the old system to the new system is performed.

Once completed and signed off by the client the new system is taken live.