Xaraya
Request for Comments: 0001
Category: Informational

F. Besler
Xaraya Development Group
January 2002

Content Management System

Status of this Memo

This memo provides information for the Xaraya community. It does not specify an Xaraya standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The contents of this RFC contain the literal content of the old plain text version of RFC-0001

When time is a less scarcer good, someone might convert the plain text into structured XML so we can benefit from it.

Table of Contents

1. Introduction

I read through the articles on http://www.postnuke.com, its forum threads, feature request on sourceforge, the xaraya developpers mailing list and reviewed some of the other open source web content management systems on the market. The following RFC is a summary and contains some solution proposals either compiled from the basic documents or from general definitions concerning content management systems. This is not a static document but work in progress

2. General statements about Content Management

"Web content can be articles, pictures, products, email archives, Flash presentations, streaming audio, whatever. This content needs a lot of things done to it. You might need systems for creating the content (authoring), describing it (metadata tagging), changing and updating it (editing), letting several people edit it together (collaboration), letting the right people do the right things to it (workflow), stopping the wrong people from manipulating it (security), keeping track of how it has changed (versioning), deciding when to display it (scheduling), displaying it in the right standard format (templating), allowing it to be displayed by others (syndication), allowing it be displayed differently to different visitors (personalisation) and more." [12]

"... The content of an online learning community always includes questions and answers in a discussion forum. A programmer might start by building a table for discussion forum postings. ... Most online learning communities offer published articles that are distinguished from user-contributed questions. A programmer would therefore create a separate table to hold articles. Any wellcrafted site that publishes articles provides a facility for users to contribute comments on those articles. This will be another separate table.

Is a pattern emerging here? We distinguish a question in the discussion forum table because it is an item of content that is not a response to any other discussion forum posting. We distinguish articles from comments because an article is an item of content that is not a response to any other content item. Perhaps the representation of articles, comments on articles, questions, answers, etc. should be unified to the maximum extent possible. Each is a content item. Each has one or more authors. Each may optionally be a response to another content item. Here are some services that would be nice to centralize in a single content repository within the content database: ..." [17]

We now have to step back and think about our goal: Do we want to develop a true Web Content Management System? Or do we want to stay a Web Portal / Community Management System? For the following RFC I assume the first one because it is stated everywhere (e.g. the Xaraya logo, the Xaraya admin message, the description on HotScripts). If the project managers of Xaraya dont share my basic assumption, then I will adjust this RFC.

3. List of requirements for our Content Management Engine

1. incorporate the content management in the core, and just have the display of the content / publications handled by different modules & blocks.[3]

    1. this is more template system related: be able to combine different content items with a template, that can be predefined online. (extending some phpwebsite feature)

    2. defining publications as composition of content components + templates, we have to create separate tools for creating/editing content components (online editors, upload), templates (online editors, upload) and publications + tools for archiving, recovering and deleting content components, templates and publications

    3. have different editing tools in place, like wiki,bbcode, wysiwyg (that uses the CSS of the selected template for display), ... [5][7]

    4. a secure way to upload binary content (images, pdf, worddoc, ...) into the file system and manage them. (using the official MIME types and having special security checks!) [11]

2. basically we want to move away from articles system towards a more generic content system [3][6]. in order to accomplish this we need to unify the content management of current articles, sections, reviews and news system. if possible, the content of some additional modules such as admin_messages, downloads, weblinks, quotes ... it needs to be discussed whether the comments system should stay separated (my favorite) or whether it shall be part of the unified content (both options have advantages and disadvantages).

    if a separate comment system is our choice: assign comments to every content fragment (-> contentid as foreign key in the comments table)

3. fine grained categorization of content with subcategories of unlimited depth (-> lets get rid of topics) [4][6][13]

    have different weights for the categories (for display search or listing results) [8]

4. enable a better structuring of content (-> tree structure) [2]

5. implementation of a content life cycle -> status (draft, reviewed, published, archived), workflow and approval processes (different queues for users with special roles/permissions, e.g. writers, reviewers, publishers)

    record the different editors (table content_editors) [1] [5]

    no content editing without either increasing the version or changing the status. in both cases a new contentid will be generated. this behaviour is the only way to enable true versioning and content life cycle including rollbacks. the downside is that the database will have to hold much more information. to cut down this bloat, the older versions / status of a content component could consist of only diffs. Another way to speed up the database is to have 2 tables of each proposed table in chapter 4, i.e. have one set of tables to hold the live content and publications of the website, and another set of tables to hold all work in progress and outdated versions.

    have an archiving mechanism [10]

6. add a rating system so that ratings can be assigned to each and every content component. (i.e. separate rating from comment system) [1] [9]

7. the ability to have promoted publications, i.e. e.g. stick a certain article to the top of a list for a while. [4]

4. Solution proposals - database tables

```
// Lets think of the tables contenttype and publicationtype to
// contain the classes (structures) and content and publications to
// contain the instances.
// there are relation tables for the classes and instances, and there
// are relation tables that combine instances with users, comments,
// ratings, ...
// In the following text whenever "timestamp" or "date" is mentioned,
// this will mean an int(10) field holding the UTC unix timestamp.
```

4.1 Table content

```
// as the content repository
// you may note that there is no field that indicates the
// formating of textual content. this is intentional and my
// thought is that we should store these content components using
// html and to convert wiki and bbcode on creation / editing time.
// i was considering xml as storage format but since our main
// target medium is the web, storing in html gives cheapest
// rendering while still able to separate content and layout using
// only allowed html tags with css that the template system can
// override.
contentid   // unique long integer value
conttypeid  // foreign key from table "Contenttype"
status      // (draft, review phase 1, publishable)
version     // revision number of the content
lastedit    // timestamp of when the current version was saved
preid       // id of the contents predecessor in the content life
// cycle enables to rollback to an earlier version or
// delete the oldest version of a certain content
// (purge)
language    // self-explanatory
editorsnote // message from the last editor for the next editor
// in the workflow to give a quick note what has to
// be done
caption     // depends on the content type. (title, link text,
// alt text of images, name of a binary content)
content     // the text itself, i.e. a chapter of a document, an
// article a sections content, an abstract, a title.
// for binary content like worddocs, pdfs and images
// the entry could either be a link to the document
// in the file system or the document itself (with
// the former being my favorite method.
```

4.2 Table Publications

```
Table "Publications"
// this is the point of contact to the templating system
// every publication uses a unique template from the template
// repository. every publication has a status similar to the
// status of a content component.
// publication types define howmany and what content components
// are allowed or required.
pubid       // publication id, primary key
templateid  // unique identifier for a template, requirement 1a
pubtypeid   // publication types are defined in a separate table.
// they are equal to a type field.
parentid    // id of the parent content to be able to structure
// the content into chapters, subchapters, pages...
// the structure can be different for each publication
// see requirement 4
status      // (draft, review phase, published, archived)
version     // version of the current publication
preid       // id of the publications predessesor id in terms of
// status (workflow) or version (-> versioning)
```

```
language    // self-explanatory. have the publishing tool making
// sure automatically, that only content components
// with corresponding languages are able to be tied
// together
editorsnote // message from the last editor for the next editor
// in the workflow to give a quick note what has to
// be done
pubdate     // date (timestamp) when the publication is
// auto-published / was manually published
expdate     // timestamp when the publication will be expired /
// auto-archived
```

4.3 Table Content_Publication

```
Table "Content_Publication"
// this table provides the ability to join 1 or more content
// components to one or more publications. you can only choose
// a content component whose status is "publishable"
pubid       // publication id, foreign key to table publications
contentid   // foreign key to content table
has_child   // boolean value to determine whether a content
// component has a child component in the present
// publication: for faster content structure parsing
```

4.4 Table Contenttype (ct)

```
Table "Contenttype" (ct)
// most content types are predefined in a standard installation
conttypeid  // primary key
conttypname // name of the content type, e.g. [abstract, fulltext,
// link, picture, downloads, pdf, worddoc]
MIMEtype    // e.g. application/pdf or text/html
```

4.5 Table Publicationtype (pt)

```
Table "Publicationtype" (pt)
// there are predefined ones like article, news, ... and custom
// pts. while templates are for display and formatting the
// output, pts are for the structure of publications.
pubtypeid   // primary key for the pts
pubtypename // name for the pt, e.g. article, document
```

4.6 Table PT_CT

```
Table "PT_CT"
// this table defines the structure of a pt: which and howmany
// content components are required / allowed.
pubtypeid   // compound key (1of2), foreign key to pt table
conttypeid  // ---- " ---- (2of2), ----- " ----- ct table
itemsmin    // minimum amount of the specific content component
itemsmax    // maximum ----------------- ------------------
```

4.7 Table Content_Editors

```
Table "Content_Editors"
// see requirement 5a, the version the user has edited is
// reflected by the unique contentid
contentid   // compound key (1of2), foreign key to content table
editorid    // ---- " ---- (2of2), ----- " ----- users table
```

4.8 Table Publications_Editors

```
Table "Publications_Editors"
// see requirement 5a, the version the user has edited is
// reflected by the unique pubid
editorid    // compound key (1of2), foreign key to users table
pubid       // ---- " ---- (2of2), ----- " ----- the
// publications table
```

4.9 Table Publications_Promote

```
Table "Publications_Promote"
// see requirement 7
// the modules to display the publications will check this table
// first, not all display modules will use this table.
contentid  // ck (1of2), foreign key to the publications table
weight     // ck (2of2), order in which the content will appear on
// display
promotexp  // expiration date (timestamp) when the weight looses
// its effect and the content drops into normal order
```

5. User-authored content

Everything in 4 relates to publisher-authored content [14]. But the strength of the nuke-alikes is that users can also contribute to the sites content. This is called user-authored content. So how can we handle this valuable source of website content? My proposal is to make a module that provides a combination of content editors (online editors, uploads) for a predefined set of content components and a predefined form of publication (-> publication templates).

So the user will find a "Submit content" that gives them the choice of publication form, i.e. article, faq, weblink, and maybe custom publication templates. This must hide the complexity of splitting up the submission into content components using simple forms that are composed by reusable predefined form components. The delivered system should also hide complexity to the admin by providing completely predefined forms.

Then the site admin can either let those publications automatically be published (shortcut in the workflow) or he can have them dropped into a workflow so that anyone responsible for publishing can review the user-authored content and decide to either publish it in the current form, or edit the content, or rearrange the content, or change the publication form from, say, an article submission, to, say, a forum entry if we can regard a top level forum entry to be equal to an article and the forum replies as equal to comments. Again, content can be published in more than one form and in more than one structure -> you can have an article and a forum entry that share one or more content components, while only one copy of each version of the component is stored in the database. [15] I hope i could show the flexibility of the proposed system

6. Relationship to other areas

6.1 Categorization system

Categorization system // to be able to assign the content to more than 1 category // see requirement 3 [1] [6] [8] [13] The details on this issue will be part of the categorization RFC whose content is currently discussed on the xaraya-data list. We might want to be able to categorize the content components in the content repository so that it is easier to search in. We definately want to be able to assign the publications to one or more categories. We should be able to have an hierarchical structure of categories of unlimited depth. To cut down the size of a centralized relation table, every module should provide its own category_content relation table using the centralized categories and API.

6.2 Permissions system

Permissions system // this is essential for security, workflow and publishing. The details on this issue should be part of an RFC on the permissions system. It is obvious that the permissions system is strongly linked with access control to different versions and status of content components and publications. Permissions are essential to the realization of a content life cycle, workflow and publication and therefore should be automatically adjusted by the system. Permissions should be on a group and single user basis. To implement a workflow ability each content component and publication needs to have at least 2-3 permissions: "admin" for the user/user-group that has to edit / review the item in the current state of the workflow, "read" for all others (or those intended to view the content) and none for those who shouldnt be allowed to view the publication (even when the status of it is "published", the publication date is reached and the expiration date has not yet been reached. In order to not having one really big permissions table the core system should only provide the most basic permissions while each module that uses content from the content repository should provide its own permission table using the core APIs for handling the permissions for the content it displays and edits. -> decentralizing the permissions table like e.g. the lang files. credits for the modularization idea to: BlackV_Li from PostTEP

6.3 Comments system

Comments system // this will allow a unified comment system to attach a set of // comments to each publication. see requirement 2a The details on this issue will be part of the comment system RFC which is currently written by Carl P. Corliss (rabbitt) and Gregor J. Rothfuss.

6.4 Rating / voting system

Rating / voting system // see requirement 6 The details on this issue should be part of a rating system RFC. Here are some thoughts for someone who wants to write it. We should be able to have a per user rating for each publication. Maybe we want only authorized users being able to vote, maybe we want also anonymous user votes. We might want to allow multiple votes or only 1 vote per user. A scale has to be defined: from 1 to 5, from 1 to 10, ... One interesting side effect of a separated rating system comes up: What about the ability to rate registered users by registered users ? A big community plus!

6.5 Multisites system

Multisites system The details on this issue should be discussed. We definately want to be able to assign the publications to one or more subsites using the built-in multisites module. We might want to redesign the implementation of the current multisites system and think of it as just another categorization.

7. Code that will need to be rewritten

```
        Code that will need to be rewritten
        /backend.php
        /print.php
        /themes/
        /modules/Avantgo
        /modules/Downloads
        /modules/News
        /modules/NS-Addstory
        /modules/NS-Admin_Messages //if included in the content table
        /modules/NS-Autolink
        /modules/NS-Blocks
        /modules/NS-Comments
        /modules/NS-Ephemerids     //if included in the content table
        /modules/NS-Multisites
        /modules/NS-Quotes         //if included in the content table
        /modules/Reviews
        /modules/Search
        /modules/Sections
        /modules/Submit_News
        /modules/Top_List
        /modules/Topics
        /modules/Web_Links         //if included in the content table
        and some blocks
```

8. Tools that need to be created from scratch

Tools that need to be created from scratch

1. Text editor modules for creating (publisher- and user-authored content) and editing (publisher-authored only) textual content, that provide APIs for categorizing, versioning, access control and workflow. // see requirement 1c
2. A secure file upload (publisher- and user-authored) and managing (publisher-authored only) module for binary content, that provides APIs for categorizing, versioning, access control and workflow. // see requirement 1d
3. A workflow editor module to create specific workflows (default, or per publication type), that provides APIs for access control.
4. A publicationtype editor module to create and edit different publication types, that provides APIs for access control
5. A publication module for assembling publications and putting them on the website, that provides APIs for categorization, versioning, access control and workflow. // please add here

9. Retractions

We list features that were considered but rejected for this system below.

10. Changelog

```
7  (2002-01-06)
Added the idea of predefined input forms for hiding complexity.
moved the parentid from table "Content_Publication" to table
"Publication"
6  (2002-01-05)
Added proposal for modularizing the permissions and categorizing.
5  (2002-01-05)
Added "Author contact"
Added "Retractions"
credits to: Carl P. Corliss (rabbitt) and Gregor J. Rothfuss.
4  (2002-01-04)
Added the quotation [17]
Added the tools that need to be created.
Added the publication type, content type tables.
Added the user-authored content
Added the permissions system in the "Relationship ..." section
3  (2002-01-03)
Reorganized the requirements.
Moved the tables concerning comments, ratings, categories and
multisites to the new "Relationship ..." section
Added the "Relationship to other areas"
credits to: Carl P. Corliss (rabbitt) and Gregor J. Rothfuss.
Dropped the container idea in the content table in favour of the
notion of having "Publications" introduced in version 2
Added the "Changelog"
2  (2002-01-03)
Reorganized the content repository
Introduced the "Publications" table and its implications
1  (2002-01-02)
Initial version
```

11 Reference title

[1] newbienetwork and spannah, "http://www.postnuke.com/modules.php?op=modload&name=News&file=article&sid=1354", unknown.

[2] Yagi, " http://www.postnuke.com/modules.php?op=modload&name=News&file=article&sid=1376", unknown.

[3] alarion, " http://www.postnuke.com/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artid=9", unknown.

[4] jerryj, toph and Landi, " http://www.postnuke.com/modules.php?op=modload&name=News&file=article&sid=935", unknown.

[5] nekvasilt, malexandria, ED and alarion, " http://www.postnuke.com/modules.php?op=modload&name=News&file=article&sid=562", unknown.

[6] Florian Bruckner, " http://groups.yahoo.com/group/pndev/message/295", unknown.

[7] niceguyeddie, alarion, jimbeam, gregor, bradnickel, ranrinc (KR) and spliffster, "http://sourceforge.net/tracker/index.php?func=detail&aid=438855&group_id=27927&atid=39223", unknown.

[8] Asparagirl, " http://sourceforge.net/tracker/index.php?func=detail&aid=441077&group_id=27927&atid=39223", unknown.

[9] bam-bam, " http://sourceforge.net/tracker/index.php?func=detail&aid=459239&group_id=27927&atid=392231", unknown.

[10] simmayor, "http://sourceforge.net/tracker/index.php?func=detail&aid=465973&group_id=27927&atid=392231", unknown.

[11] anonymous, Tobias, Andy, mulpinsf and duup, "http://sourceforge.net/tracker/index.php?func=detail&aid=473841&group_id=27927&atid=392231", unknown.

[12] David Walker, "http://www.shorewalker.com/pages/cms_woes-1.html", unkwown.

[13] ncwbiz, clnelson and jnlewis, "http://sourceforge.net/tracker/index.php?func=detail&aid=439872&group_id=27927&atid=392231", unknown.

[14] Philip Greenspun, " http://philip.greenspun.com/internet-application-workbook/planning", unknown.

[15] nickrazer, JM (Jun), "http://www.postnuke.com/modules.php?op=modload&name=News&file=article&sid=1471", unknown.

Author's Address

Frank Besler
Xaraya Development Group
EMail: besfred@gmx.net
URI: http://www.xaraya.com

Intellectual Property Statement

The DDF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the DDF's procedures with respect to rights in standards-track and standards-related documentation can be found in RFC-0.

The DDF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the DDF Board of Directors.

Acknowledgement

Funding for the RFC Editor function is provided by the DDF