What is a Content Management System

Table of contents

Abstract

This document explains what content management is and outlines several criteria to use to pick a Content Management (CM) solution which is appropriate for your usage.

They make a list of potential requirements for a Content Management System (CMS) in order to ensure that the solution (bought, developed, or a combination of the two) :

- will meet the goals of the organization and
- is within budget and has the appropriate amount of backup and support.

The core goal of a CM tool is to provide a way to manage, organize and automate the entire content management process, from content creation to content delivery on multiple channels and devices to meet the needs for high-quality, fresh and relevant content for all intended target groups.

A CM tool is required when this process becomes too complex to manage manually.

Approach

The report is structured as follows:

- A first part defining what content is.
- Another part defining what content management is. This allows to have a better picture of what a CM tool should technically be able to deliver.
- A third part, describing the functional and non-functional requirements for a content management system.
- The way TikiWiki is helping you meeting the above elements.

Content

What is content ?

Raw information becomes content when it is given a usable form intended for one or more purposes.

Increasingly, the value of content is based upon the combination of its primary usable form along with its application, accessibility, usage, usefulness, brand recognition and uniqueness.

Content is information put to use.

Information is put to use when it is packaged and presented (published) for a **specific purpose**.

Content is not a single unit of information but a conglomeration of pieces of information put together to form a **cohesive whole**.

There is no single definition of content since content can refer to anything from text articles to streaming video, reviews and descriptions. **All content has one thing in common**: it engages the attention of the visitors, encourages them to spend as much time as possible on your website and gets them to return as often as possible.

We propose the following tentative definition :

Elements of Managed Content

=

integrated and streamlined content entities

+ workflows
+ workflows states
+ relationships

Content entities have various forms.

Contents entities represent the valuable elements produced by the organisation and meant to form its knowledge base's singular elements.

The possibility to reuse content entities is a central point to content management.

The main concern here is the format in which the content entities are presented.

This format should be usable, meaning that reuse of its conveyed semantics is possible.

If content parts and presentation of the content are mixed, it will be problematic to perform any reuse.

Leveraging of content entities can lead to significant work to integrate all of the content in a single database.

In some cases, it is a better idea to use gateways to link the existing content entities to a core structure.

This can be seen as using a reference to the entity in the database and a retrieval mechanism based on the gateway when access to the content is required.

Of course, this approach will lead to scalability problems in some cases. Indeed, you may want to leverage the content entities located in a database by web-enabling it. This may require using special software to allow for high volumes of requests to be supported. This point is of special importance to an organization getting lots of load.

What is content management ?

What is content management ? Content Management as such is difficult to handle due to the genericity of both terms leading to very diverses implementations and concepts. It intermixes with the implementations of solutions offering CM-supporting features.

Those implementations are called Content Management Systems (CMS). We use the words CM and CMS interchangeably, CMS meaning 'the ideal CMS' if such a thing exists (because everybody's set of requirements and constraints is different).

The following sections will define the parts making up a content management system in order to have a clear and modular picture of the system.

This will allow for common understanding of the subject for the involved parties.

We attempted to define exactly what content management is and what a content management system should do.

The part that makes CMS definition tricky, is our differing conceptions of the term "management." For some, it's all about deployment (i.e. getting content onto as many desktops as possible); for others, it's all about control (i.e. restricting the flow of content in accordance with certain rules about who can publish what to whom).

We can come with this tentative definition:

Content Management System
=
A solution to the problems faced by an enteprise that must channel information over the web, from any number contributors to any number of consumers, in accordance with its own business rules.

CMS solutions range from the very simple to the very complex, depending largely upon:

- the numbers of contributors and/or consumers, and the extent to which these basic roles must be classified and distinguished between;
- the volume of content flow, and the extent to which the enterprise ust assume responsibility for its delivery, with maximum reliability and/or minimal latency and
- the number of business rules , and the extent to which they are unique and/or dynamic.

That's a pretty academic definition, of course, but it is not possible to be much more specific without branching off into distinctly different problems.

And until you do that, it is difficult to define the essential elements of a CMS.

Business rules = A set of entities, interactions and constraints explicitly recognised as being important to the success of an enterprise or organization. Interactions among entities are governed in terms of prescriptive policies defining what should or must occur, and proscriptive policies defining what should not or must not occur, under any number of condition-sets which might be defined.

We will now lay out some questions which will cast some light on the subject with regard to possible or highly touted features:

Does a CMS really NEEDS to provide:

Templating ?

It depends whether you want to distinguish between two different sorts of contribution - design and editorial - and keep these people out of each others faces (a pretty common requirement, but not a given in all cases).

Caching ?

It depends on the challenges in terms of volume of content. If there is a need to serve a very high volume of similar requests for data that isn't too dynamic, then you probably want the pages stored in static form, until deleted for lack of use.
If there is a need for content deployment, with low-latency delivery right to the desktop, then you need big-league infrastructure.

Scripting ?

If the business rules are changing as you go along, in ways that don't fit the parameterised user-preference screens, then you need scripting.
For anyone with complex requirements in terms of "workflow" (i.e. sequencing human interventions on a data flow), this is probably a given. And probably not enough (i.e. think tools with integrated workflow).

Change control ?

It depends whether you want to have multiple programmers working simultaneously on the system (often a function of the business policy).

Database interfaces ?

It depends how far you can go on the one the system ships with, and whether you have got integration requirements (e.g. Oracle is mandatory).
One thing to be clear about is the a need for database partitioning, which might come into play as the system scales up to a certain point.

This list of CMS elements may be considered minimal by some - but you can solve a significant range of content management problems, without site owners having to even think about change control and database interfaces above - or, for that matter caching or scripting (unless they really want to script).

In a more complex enterprise/organization calling for division of labor, you need to have the most common roles defined, linked to some typical business rules that apply to work associated with each role, which can be easily turned on or off - plus the ability to create new roles & rules without undue difficulty.

How to qualify a system as a CMS system ?

The primary requirement of a CMS would be to make the task of managing any content which is to appear in a computer media publication (such as a Web site) a non-technical task; i.e. one that does not require a programmer to perform it, but a person such as an editor. We would define "managing" content as the task of choosing what content appears where, and when; this would be separate from actually creating the content.
In this context, application servers and other middleware systems do not qualify, nor do authoring systems. Systems which can be validly considered components of a CMS include presentation tools (such as an XSL engine or a caching engine), workflow tools, aggregation or syndication tools, and versioning tools.

However, to qualify as a CMS a system should combine several of these components such that it conforms to the above criterion - if you are an editor of a Web site that uses such a system, you use the system to publish content submitted to you by some author, and do not have to call on, say, a programmer or a support person to do so.

Some words on CM differentiating factors

CM systems are all different. This does not eases the choice !

The identity of a CM system is linked to the beliefs that its creators and architects put into it.

From those beliefs, which we could see in the products as decisions in the large (like deploy content as static pages or use a database in real-time), system features are derived.

This reasoning can be used to keep the criteria in perspective when evaluating the solutions.

Further, different environmental constraints (need for very high security for example) will give a significant edge to some criteria which would have a totally different weight in another setting. Having a clear understanding of the envisioned environment will then help in selecting the appropriate product and understanding the choices made.

How do the CMS systems differ ?

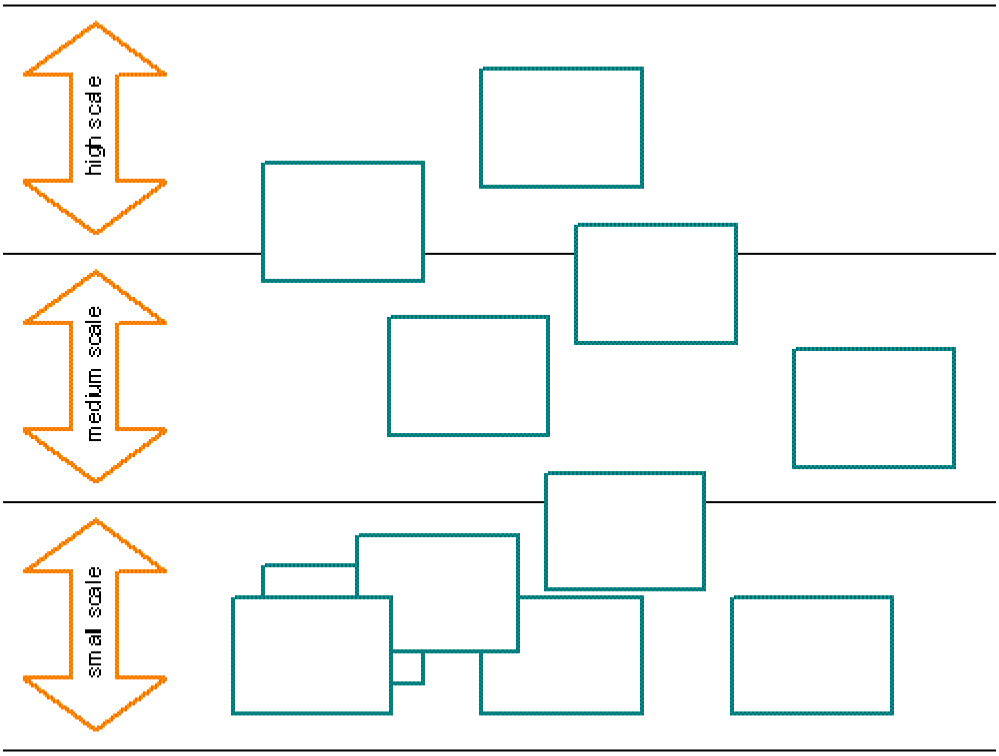
While most users, whether individuals, small institutions or enormous corporations have similar CMS feature wish-lists, the differentiator is found in their varying scalability requirements. These differing requirements will dictate the implementation and support strategies of real products.

We must partition the enormous potential CMS space (since "everything" is content on the Web) into meaningful domains based on scalability.

In theory, a single CMS can provide a variety of scalability layers that map nicely to these differing domains.We are not sure that ideals and reality converge when it comes to delivering (and, more, supporting) real software.

Scalability addresses a number of obvious but no less critical dimensions for being obvious: workflow, security, replication, performance, customizability, etc.

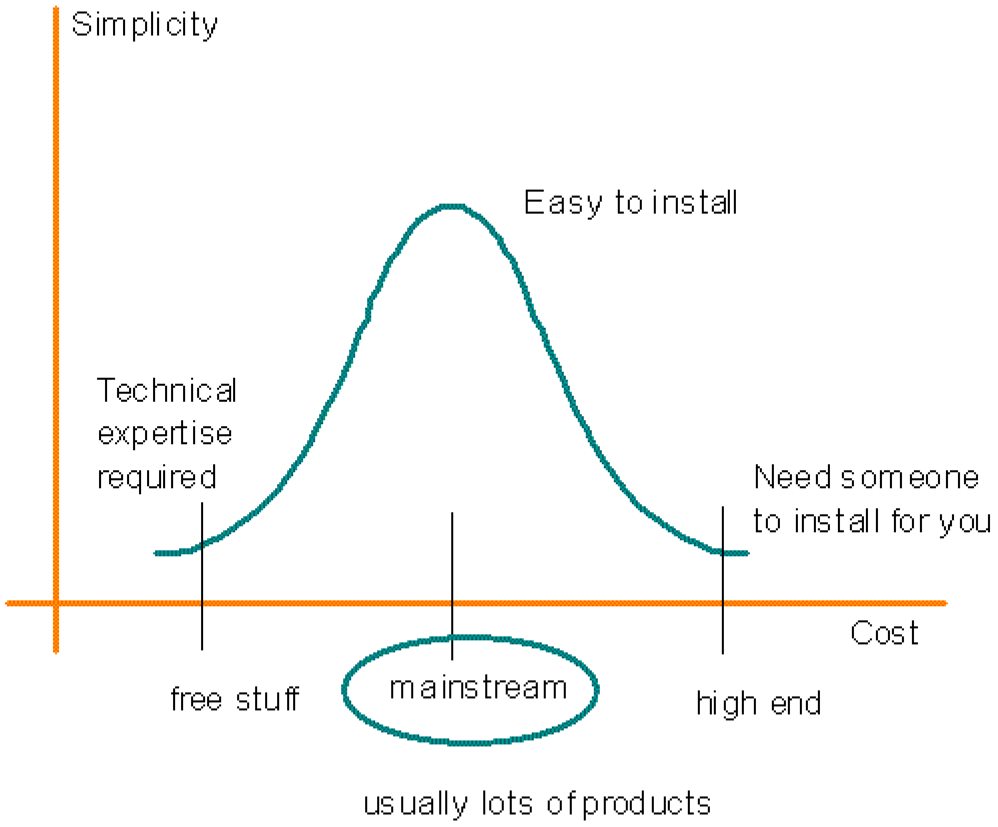
Once we can get a rough handle on where the scalability layers partition, knowing how to deliver products within each layer that are as simple as possible for users will then define the best applications.



These systems follow the Bell curve of simplicity as depicted hereafter.

Simplicity cover the following elements:

- Installation
- Use
- Maintenance
- Administration



The capacity to pass documents and data between layers and, equally, between applications, is the other area besides scalability that is absolutely basic to defining a CMS. These two efforts - scalability and common protocols - are basic to developing such a definition.

Differences between tools

Content Management Systems are a separate line of products from Document Management Systems, Web Scripting Environments, and Newsprint Publishing Systems.

Within CMSes, further distinctions could be made between template engines, editorial workflow management, and dynamic/personalized content delivery systems, etc. but we could consider that all fall under CMS.

Key : The CMS products are turn-key solutions.

- Scripting/templating systems like Microsoft ASPX, ColdFusion, JSP, and PHP are not CMSes, though they may be used to build a CMS
- Source Control Systems like SourceSafe, CVS, etc. are mainly used by software programmers and similar teams. Such systems can sometimes be used in conjunction with certain CMSes in the production of a news media web site.
- Newspaper publishing systems designed for print would not be CMSes even though they do manage content (and do so quite well), since they aren't yet suitable for the web and new media on their own. They are currently too tied into the daily newsprint model, but we are sure future versions will be fully web enabled.

In summary, we could define a CMS as a fairly turn-key product used by an organization that authors a lot of textual content.

First classification of content entities

This classification is based on a physical type of separation.
Some elements are handled in the same way and at the same time represent really different organizational entities or functional parts of managed information (e.g. a press release and a job offer).

- Specifications, benefits of services, prices
- Databases
- Multimedia assets (Flash, Applets, Sound, Videos)
- HTML templates, scripts and stylesheets
- Logos, photographs and diagrams
- Contact information
- Job opportunities
- Press releases
- Help and support information
- E-mail newsletters
- Legal information
- Mailing lists
- News items, features and articles
- Expert subject matters
- Other

Another way to classify content entities

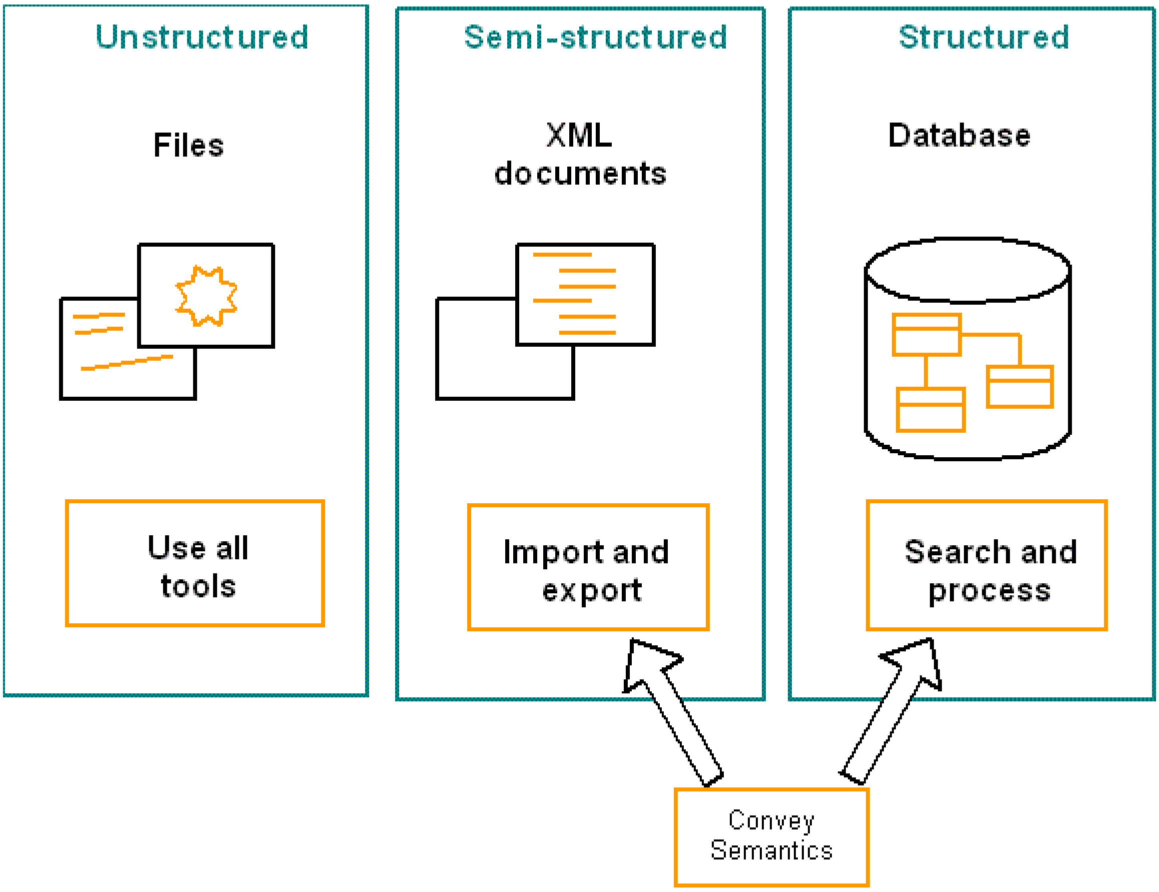
This second categorization gives a more conceptual view on things: We can see that content can be classified in very different ways. What is important is that the classification is used consistently in the system to avoid chaos. In fact, we can see that all of the following elements of content are fundamentally handled in different ways.

So, content can also be seen as:

- Documents including letters, memos, agreements, and proposals. Typically, the "simple" document is the same as the computer file that contains it.
- Relational data - Facts about people (such as subscriber data) and transactions, and tabular data.
- Paper documents - Older or external documents (such as membership applications or order forms); the source is usually paper.
- Images - Including possibly paper-based documents that have been converted to digitized image files.
- Assets - An term for images and graphics of all sorts are used publications and processes today. Will include at least audio and video in the future.

- Components - Another term for journal articles, book chapters, conference programs and other highly formatted and structured documents, as distinct from day-to-day office documents described in “Documents”. While individual components of these documents can be extracted and used independently, the most efficient way to manage this content today is at the document level. This is primarily due to the limitations of the available tools rather than anything intrinsic in the content.
- Links - both to other things that we publish and to content published by others.
- Workflows - computerized states and transitions along with actions for supporting processes of the organization.

All kinds of data have their uses



- Common files are well suited to be used with editing tools like dreamweaver, word, ...
- XML files are well suited to exchanging data with other systems and specifying templates
- Databases are well suited for storing metadata and tabular information, performing searches...

Functional Requirements

The sections hereafter cover most of the functional requirements that a content management system may have.
Following the needs of the users and givens of the problem domain, some requirements may be more appropriate than others.
Anyhow, the items will allow to profile a CMS system and compare it with others.

Inclusive content entry environment

Look for an inclusive entry environment that gives your subject matter experts and casual contributors easy access to your content repository. They should be able to view their additions and changes immediately and in context.

- Will your nontechnical users and content authors have an easy-to-use and intuitive interface?
- Can your external or freelance collaborators use a web browser to enter content or administer the CMS from outside of your internal network?
- Can your site designers use standard HTML editors like Homesite and Dreamweaver to create site templates, or does the CMS lock you into a proprietary editing environment?
- Can the CMS you're looking at convert your content in a simple and efficient manner?
- When you hand creative content off to your web development staff, can the CMS transform word processor documents to formatted HTML text, optimize bitmapped images to load faster on the Web, and change image formats?

All of the above elements are expanded further hereunder.

Import of exisiting data – Data Conversion

Conversion to a highly structured format is difficult because it means adding structure to the data. The scope of such an effort would require a team of workers dedicated to the conversion alone given the size of the data at hand in this project.

The data has to be imported and converted to content elements associated with meaningful metadata.

There is a simple lesson we have learned on data conversion: volume always complicates matters. Most recipes will work if you double the ingredients. But try multiplying by 50 or 100 and all you'll have is a mess in the kitchen and a big room full of hungry people.

So, the solution has to bring support tools to the table to help this conversion.

The tools will have to be integrated into the process which will be used to carry this conversion.

Automated solutions are ideally suited for high volumes of data. The computer is many times faster than a person. All you have to do is find or develop software that will completely and accurately convert your data to a structured format.

Well, it is not that simple! Why? Because this isn't just a conversion. You are adding structure to your documents, which requires inference and subjective decision-making.

Rights tools and process for the job

Surely the computer can do most of the grunt work and then experts can fix it up afterwards.

It is a fact that combining automation with expert review seems to be the best approach. But only if it's done right.

If you do enough damage to your car, the insurance company will give you money to buy another one rather than fix the one you have. Similarly, "fixing" a conversion tool can actually take longer than tagging by hand. It's clear that one key to a successful conversion is to automate as much as you can as cleanly as you can.

Large volumes require standardization to prevent chaos. Otherwise, different interpretations will generate inconsistent results. Problems can be prevented by implementing "conversion specifications," which detail every element in a document and how it should be coded in the new format. These specifications are used as a standards document throughout the project.

So, one key to successfully using conversion software is to customize it. Some CMS vendors have developed suites of conversion filters that can be configured to the specifications of the project.

It is crucial to minimize the amount of cleanup necessary after the conversion is finished. You can view this as a quality control.

The most critical element of quality control is customer feedback. The entire conversion process should be open to the organization, so that a misunderstanding doesn't result in thousands of mistagged pages.

Once the conversion is underway, partial deliveries should be sent to the client as they are completed. This will give understanding of how new data will best implement on the new system.

Perhaps the most pernicious problem of large volumes is that the work involved is impossible to predict. In other words, even if you do budget for all the expert days you think you need, you might very well need more. This could lead to disgruntled workers and even more disgruntled executives. This speaks on the importance of visibility and partial deliveries for early implementation.