

Content Modeling

The content model is the technical representation of the logical domain of content, consisting of all content types, properties, data types and relationships between content objects.

Structured Content

The ability to subdivide a single content item into separate attributes or properties in order to aid with editing, searching, or templating.

Variations

- The selection of built-in property types available to construct content types.

Type Inheritance

The ability to base a type on another type, inherited the base types properties, while adding additional properties. Changes to the base type will be reflected in the inherited type.

Variations

- The ability for the base type to be "abstract," or not available to be created. The base type can serve as a base only.

Type Composition

The ability to build a type from pre-defined "sets" of properties.

Variations

- The ability for the sets to be "abstract," or not available to be created. The sets can only be injected in other types.

Singleton Types

The ability to enforce that only a single object of type may exist in the repository.

Variations

- The ability to administratively prevent creation of objects from a type, thus allowing enforcement of the singleton pattern at the UI level.

Validation

The ability to validate text properties for data values.

Variations

- The variety of built-in validation options provided: required, numeric, numeric range, regular expression, etc.
- The ability to delay validation until publication, to enable editors to "rough-in" or "stub-in" content that remains in draft status.

Referential Properties

The ability to refer or "connect" to another content object from a single property.

Variations

- Type limitations; allowing only links to specific types, or disallowing specific types
- The existence of all other types of property features: validation, repeating, etc.
- UI enhancements, allowing for the creation of the object inline, within the context of the linking object
- The ability to specify the uni- or bi-directionality of links, and to control the discovery of inbound links to a specific object.

Embedded Objects

The ability to embed or encapsulate an object as the value of a property. This object operates in the same lifecycle space as the owning object.

Variations

- Type limitations; allowing only specific types, or disallowing specific types
- The existence of all other types of property features: validation, repeating, etc.

Repeating Properties

The ability to repeat a single property a specified number of times.

Variations

- The ability to control the range of repetition, both minimum and maximum.

Content Modeling

File Handling

The ability to manage file assets as content.

Variations

- The ability to link to files as the value of a property.
- The ability to "wrap" file assets in a content object and thereby exposing them to all content services.

Content Modeling

Localized Properties

The ability to specify if a property is universal to all languages or a value should be supplied for each enabled language.

Content Modeling

Property Grouping

The ability to aggregate properties into groups.

Property-Level Permissions

The ability to specify read/write permissions on specific properties.

Variations

- Tri-state permissions: hidden (not viewable without permission), read-only (viewable with permission, but not editable), and full read and write
- The ability to specify these permission on groups of properties.

Content Aggregation

A basic task in content management is to aggregate individual content objects into larger structures to provide value and meet requirements.

Parent-Child Content

The ability for a content object to be a "parent" of one or more child objects, thus forming a "content tree." This sets up multiple familial relationships: parent, child, sibling, ancestor, descendant, root, branch, etc.

Variations

- The ability to strongly type the content tree, so that each type has an allow/disallow list of types which are allowed or prohibited from being created as child objects.
- The ability for content objects to appear in more than one location in the tree -- to be the child of more than one parent
- The ability for content objects to inherit values from parents

Taxonomy or Categorization

The ability to assign content to conceptual nodes in a taxonomy or categorization structure.

Variations

- The ability to assign content to more than one category or taxonomy node.
- The ability for "implicit ancestral categorization," where any assignment implies assignment to all ancestors.
- The ability to model categories or taxonomy nodes as content items, thereby exposing them to all content services.

Content Aggregation

Tagging

The ability to create ad-hoc, flat categorization structures from a content object's editorial UI.

Collections, Lists, or Menus

The ability to create arbitrary, ordered collections or lists of content.

Variations

- The ability to model collections or lists as content items, thereby exposing them to all content services.
- The ability to set structure-specific data on the assignment of content.
- The exposure of content services to lists, collections, or menus. Example: permissions, versioning, etc.

Content Aggregation

Relational Repeating Properties

A combination of (1) relational properties, and (2) repeating properties, which together provide the ability to form collections as a property on a content object.

Editorial Tooling

These are the tools that enable editors to effectively work with content.

Versioning

The ability to save content changes alongside the existing content, rather than replacing the content.

Variations

- The ability to easily "roll back" to a prior version, by either moving a publication label, or duplicating a prior version and publishing it.
- The ability to compare versions, both on a property basis, and visually.
- The ability to show Word-style "Track Changes" annotations in visual compare.
- The ability to branch the version tree into multiple derivative versions.

Editorial Tooling

Content Scheduling and Expiration

The ability to schedule when new content is published or unpublished.

Variations

- The ability to schedule version advancement, not only new content items.
- The ability to group content scheduling or expiration to publish or unpublish groups of content simultaneously.

Editorial Tooling

Archiving

The ability to change the storage status or disposition of content at a specified point in time.

Approvals

The ability for a publication action to trigger proactive approval requests before content can be published. Approvals normally operate serially, advancing as approvals are received, or ending when a rejection is received.

Variations

- The ability for the approval request to operate in parallel, rather than serially.
- The ability for a parallel approval request to auto-approve after a specified number of approvals.
- The ability for comments to be provided on approval or rejection.
- The ability for an approval step to be managed by a code process.
- The ability for an approver to edit the content without restarting the approval process.

Preview

The ability to view what content will look like when rendered for a specific channel (normally the web).

Variations

- The ability to select the channel for which to render.
- The ability to "spoof" a specific user, personalization group, or set of request parameters.

Dependency Management

The ability to reveal the inbound and outbound links to and from a content object to inform editors of where changes will be reflected.

Variations

- The ability to prevent deletion of content objects with inbound links.

Content Permissions

The ability to specify the depth to which user accounts can interact with content.

Variations

- The granularity of permissions, commonly: Read, write, create children, delete
- The ability to specify creation permissions on specific content types. (Once created, permissions revert to the object.)

Localization

The ability to store multiple translations of content in different languages.

Variations

- The existence of a side-by-side translation UI.
- The existence of add-ons or plug-ins for translation providers.
- The ability to view the repository through the "filter" of a single language, showing only content in that language, or prompting to create content in that language if it doesn't exist for a specific content object.

Editorial Tooling

Editorial Commenting

The ability to provide editorial notes in the context of a content object.

Variations

- The ability to provide property-level notes.
- The ability to provide text-level notes in rich text.
- The ability to tag specific users in notes.

Notifications

The ability to trigger notifications to specific users or user groups based on system events. By default, these notifications occur inside the UI itself.

Variations

- The ability for notifications to be extended into other channels.

Output and Publishing Management

This is the set of tools that turns content into consumable artifacts that can be delivered to your customers.

Templating

The ability to combine managed content with static content to form consolidated output. Most systems use a known/commodity templating language to accomplish this. At a minimum, this language must support token replacement.

Variations

- The addition of programming concepts such as looping, conditionals, output filters, etc.
- The usage of a standardized templating language: Liquid, Mustache, Handlebars, etc.
- Whether or not the templating language executes in the same process as the system itself, or is safely "sandboxed" to its own execution space.
- The variety of template storage and provider systems.

Web Serving

The ability to respond to an HTTP request with the templated output of a content object or other artifact.

Variations

- The ability to directly return content files to an HTTP request.

Localization Detection

The ability to detect the incoming preference and select the correct language version.

Variations

- The methods available for language detection: sub-domain, URL segment, browser setting, etc.
- The ability to specify a default language for use in the absence of a detected language.
- The ability to specify "fallback" logic for each available language. For example: (French Canadian; if that doesn't exist, French; if that doesn't exist, English)

CDN Integration

The ability to integrate with a content delivery network (CDN) to cache responses outside the system itself. CDNs are usually always separate business services from separate vendors.

Variations

- The ability to automatically purge/refresh the CDN cache, either globally or individually.
- The ability to administer the technical parameters of the CDN service.

Artifact Transmission

The ability to move the artifacts of a templating operation or other file-based asset to another system.

Variations

- A existence of a target-side process to ensure synchronization.

URL Management

The ability to map a URL to a specific content object for retrieval in a web browser, and organize multiple content objects into a hierarchical URL structure.

Variations

- The ability to control the URL or URL segment assigned to content.
- The ability to map "vanity" or shorter URLs to content to provide easier embedding in non-digital media.

Output and Publishing Management

Redirection Management

The ability to manage URLs and URL patterns that redirect requests to other URLs, or to specific content.

Programming and Extensibility

This represents the ability for the how the software functions to be modified to fit specific requirements.

Code/In-Process API

The ability to write code native to the system's programming language and have that code execute within the system's computational process. The API generally provides access to content in the repository and other system functions.

Remote API

The ability for a remote process to retrieve and modify content from the repository or otherwise affect system functioning. These APIs usually operate over HTTP and are commonly referred to as "headless."

Variations

- The usage of a known protocol/convention, such as SOAP, REST, or GraphQL
- The ability to both read and write to the repository.
- The ability to query or modify content model definitions.
- The ability to manage other system administrative functions.

Event Modeling

The ability to capture content events and insert code execution when they occur.

Variations

- The ability to capture both pre- and post-events ("bookend events"). For pre-events, also allowing for content modification.

Remote Event Notification / Webhooks

The ability to trigger a formatted HTTP request based on a content event. These systems are usually configured based on content type or location.

Variations

- The ability to post the entire triggering content object to the remote system, usually in the form of a POST request.
- The ability to receive and operate on the return result of the HTTP request

Programming and Extensibility

Repository Abstraction

The ability to "delegate" a section of the repository to an external datasource which generates pseudo-content objects on-the-fly which appear to be stored in the repository.

Programming and Extensibility

Repository Synchronization

The ability to synchronize one instance of the repository with another, remote instance.

Import/Export

The ability to export a set of content objects as a file-based "package" which can be imported to another instance.

Variations

- The ability to export content type definitions for the imported content. The target repository should create new definitions as necessary, while retaining existing definitions.
- The ability to export any related/referred content. The target repository should import them if they do not already exist.

Programming and Extensibility

Custom Properties

The ability to define custom type properties, complete with custom storage and editorial UI elements.

Programming and Extensibility

Custom Validation

The ability to define custom validation functions on built-in property types.

Variations

- The ability to define both client- and server-side validation functions.

Scheduled/Batch Jobs

The ability to execute arbitrary code on a scheduled or on-demand basis.

Variations

- The ability to supply input parameters for on-demand execution.
- The ability to execute jobs "on" a single content object or group of objects, rather than globally.

Administrative Tools

These are the tools by which the system is managed outside of content operations.

User Management

The ability to create, edit, and delete user accounts, and perform password management.

Variations

- The ability to aggregate users into groups or roles, to which bulk operations can be performed.
- The ability for user self-service: the changing of names, emails, and password directly by the user without outside help.
- The ability to wrap or connect a user account with a content object, allowing additional content data to be stored with the user.

Authentication Providers

The ability for authentication to be delegated to other providers, such as Active Directory, Okta, etc.

Variations

- The ability for external services to synchronize groups and group assignments.
- The ability for external services to initiate the creation of a local user account.
- The ability for external services to populate or update a local user account with data items (example: email, phone number, department, etc.)

Administrative Tools

Permissions for Functions

The ability of the system to manage permissions for non-content related system functions.

Configuration Management

The ability to set operating parameters of the system.

Variations

- The ability to set these parameters from the UI.
- The ability to set these parameters from files and reload from the UI or a file-system event.
- The ability to maintain a version history of configuration settings to allow for auditing and rollback, if necessary.

Content Type Conversions

The ability to convert existing objects from one type definition to another.

Variations

- The ability to specify a subdivision of the repository on which to execute the conversion.
- The ability to map properties from the source type to the target type.