Content Manager Version 8 Overview

by Cataldo Mega
Ken Nelson

IBM Content Manager for Universities and Libraries

IBM Forum Stuttgart, Pascalstrasse 100  70569 Stuttgart, Germany
Agenda

- Objectives
- Architecture review
- Data Modelling
- Features / Functions
- Migration
CM V8 Design Objectives

- Application-neutral, extensible, and scaleable
  - Beyond production imaging or document management
  - Enable Enterprise Content Management
- A physical data model describing the "building blocks"
  - Abstractions for modelling content, relationships
- Add high-level APIs as needed
  - To support different application models and standards
  - Direct mapping to the physical data model
- Push function down the stack, including into DB2, as required for performance, data integrity, or to leverage strength of other technologies
- Focus on performance
  - End user response time
  - Scalability
DB2 Content Manager Architecture

AIX, Windows, Solaris, z-Series, Linux, iSeries¹

- WebSphere Business Integration Server
- WBI WS Adapter
- WebSphere MQ Workflow²
- WebSphere Application Server
- CM API (Java, C++, Web Services³)

- Tivoli Privacy Manager
- Tivoli Access Manager
- Lotus Workplace
- WebSphere Portal

- SQL

- Open, enhanced API
- Multiple Platforms
- Expanded Data Model

Library Server

- DB2 UBD/Oracle³
- Stored Procedures

Resource Managers

- WebSphere Application Server
- Documents Images
- Rich Media
- Web Content
- Tivoli Storage Manager
- OAM
- DR 450

Website or Portal App

eClient

HTTP, FTP, ...

Website or Portal App

Other Resource Managers Possible

1 Development under way
2 Packaged as II for Content Advanced Workflow Service
3 Currently under beta
CM Data Model Entities

CM V8 Attribute
- Holds persistent state of runtime entities
- Attribute have properties: length, min / max value, searchable...

CM V8 ItemType
- A basic entity or schema – set of attributes and properties

CM V8 Item
- An instance of an ItemType
- A record in a "component" table representing a document, folder, video, or other business object
Data Model Example

Entity: Article
An Article is represented by Attributes such as:
Title, Subject, Date, etc.

<table>
<thead>
<tr>
<th>Article</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
The state of entities is stored in single – and multi-valued attributes

Articles have Authors
- Last name
- First name
- Company
- Address -- how many?

Articles have Keywords
Attributes are organized in a hierarchical structure:
CM Item Type

Any Depth

Root
Child
Child
Child

Any Width

1 Row, system/user attributes

0-n Rows
system/user attributes
CM Items: an instance of an entity
One row in possibly many tables

Authors have 1 or more addresses
Modeling relationships among entities:
 Associations using Reference Attributes

- What if Authors have Photographs or one Author is associated with multiple Articles?
- A Reference attribute creates a relationship between records
  - Points to version of an item from root or child component
- Often a better choice than hierarchical ItemType
  - Eliminates duplication of data
- "Delete Restrict" ensures referential integrity
Modeling relationships among entities:
Aggregation and composition using links

- Used to represent a relationship between two Items
  - Such as Document in Folder
- Items may be linked to/from 0-N Items, nested to any level
  - Cycles or loops allowed
- User defined Link Types supported for other semantics
  - Maps and Graphs in Info Mining
Modeling relationships among entities: Associations using foreign keys

DB2 foreign keys for data validation

If value being stored in CM table doesn't exist in target table, an error is returned

CM table to CM or customer table

Authors

<table>
<thead>
<tr>
<th>UserID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ICMSTUSERS (CM User table)

| UserID | DomainID | PrivSet | .......
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Advantages and restrictions of linking mechanisms

<table>
<thead>
<tr>
<th>Linking mechanism</th>
<th>Used at component level</th>
<th>Linked elements can be deleted</th>
<th>Limited by version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Root to root</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reference</td>
<td>Root or child to root</td>
<td>Specify when you create the reference</td>
<td>Specify when you create the reference</td>
</tr>
<tr>
<td>Foreign key</td>
<td>Root to a different item type or external table</td>
<td>Specify when you create the foreign key</td>
<td>Specify when you create the foreign key</td>
</tr>
</tbody>
</table>
CM V8 built in entity types: Resource

- An Item with additional System Defined attributes defining the location, type, size, etc of an Object
  - User attributes, ACL, expiration date, etc
- Ideal for representing single-part documents
- Only a "root" component can point to an object
CM V8 built in entity types: Document

- Document Model -- using the building blocks
- Control of Part Types per Document Type
  - For example: Image and Notes, but no Annotations
- Versioning rules per Part Type
- ACL per Part Type or Part
- Enablement for advanced applications
  - Attributes on parts
  - Compound Documents
Auto Foldering

- When an Item is created, add it to a folder
- Implemented in the server vs. client
  - multi level and multi attribute
Views

DB2 Views are created for each table to support read-only access with SQL:
- Access control logic built into view definition
- Intended for use by Host or Workstation applications
- Can be joined with non-XML tables in queries
CM Access Control

Access control is an integral part of Content Manager

Access Control List (ACL)
- One or more Users or Groups and a Privilege Set
  - Privilege Set is a list of privileges
    - Create, Retrieve, Update, Delete, etc. for Items
    - Add User, Modify ACL, etc. for SysAdmin operations

Access control may be at the Item Type level
- Each type of document could have a different ACL

Or at the Part Type level
- Scanned image, Annotation, Notes can have different ACL for each type of document

At the Item level
- Each document could have a different ACL

Exits provided in to enable authorization through UDF call
Administrative Domains

Users may be defined as "domain administrator"

- Add, modify, delete users or manage groups within the domain
- Only groups, RMs, Privilege Sets, etc. in the domain may be assigned by a domain administrator
Transactions

- BeginTransaction, EndTransaction
  - Uses DB2 for Library Server
  - Resource Manager is included in the transaction scope
- Eliminate orphan or missing objects
- Ensure consistency between LS and RM
- "Explicit" transactions can include any number of actions
  - But locking of rows can impact performance or
- "Implicit" transactions ensure LS/RM integrity without work by the application writer
Versioning

Some applications require that the original document be preserved and modifications result in new versions.
Administrator will specify per ItemType:
  - Always, Never, Application control
  - Maximum number of versions to maintain
Meta-data and parts (objects) can both be versioned:
  - Document can be versioned without copying objects
Retention Management

- Each Item Type can have a retention period
  - Expiration date calculated when Item is created
  - Can be updated by applications
- Removal performed by custom application or client
  - Complete solution is provided by IBM Records Manager
History/Event Logging

- Who did what to which CM entity and when?
- Store history/events/audit records in a table
  - Enables queries using SQL
  - Workflow events will be in CM tables

- Server logging will be controlled by administrator
  - Per ItemType, records for Create, Retrieve, Update, Delete
- APIs provided for client applications to write to or query the Item event log
  - Reports will be generated via SQL queries
- Records are also written for events such as Resource Manager state change (available/unavailable)
- If configured, can include count of “concurrent” users
Document Workflow

- Small footprint, High performance
- Single definition of users, privileges
- ACL checking on routed Items
  - Work Lists contain only documents/folders to which user has access
  - Count of work is accurate based on ACL

- No "out of synch" conditions
- Document Routing operations can be included in a CM Transaction
- Single audit trail for Item create, retrieve, update, delete, and routing
Document Workflow

Process
- One or more work nodes
- "Target" for routing Items
- Route definition includes menu action

Work Node
- Collection point, Work Basket

Work list
- "Source" for obtaining work
Document Workflow

Collection point
- A Work Node at which a Folder waits for arrival of other Items
- 0 or more Items of 1 or more types

Notification
- Work which has remained at a Work Node longer than expected
- Used for filtering in Work Lists

Suspend for time and/or Items
- Manual or automatic resume
- Suspended work stays at the Work Node, is filtered by Work List

Work List
- 1 or more Work Nodes
  - A Work Node may be in multiple Work Lists
- Return up to "n" work packages with ordering, filtering
  - 1 work package = "system assigned"
  - Order by priority, date
  - Filtering -- Notify state, Suspend state, Assigned user

Server exits on entering/leaving work nodes, Overload
Document Workflow new in V8.3
Event based routing
Value based routing
IBM Content Manager Version 8

- Rich function to meet the requirements of many types of applications
- Extensible architecture to meet future requirements
- Exploiting the strengths of strategic IBM technologies
- Achieving both performance and scalability

The foundation for Enterprise Content Management