

# Technical implementation of multi channel content management

Authors:

- Natasja Paulssen, Ordina NV (Partner and Content Management Consultant)
- Juan Carlos Martínez Gil, SAP AG (MDM Expert at SAP NetWeaver RIG)
- John Wenmakers, Philips International BV, Philips Business Applications Services (Manager Center of Expertise SAP MDM)

**Abstract:**

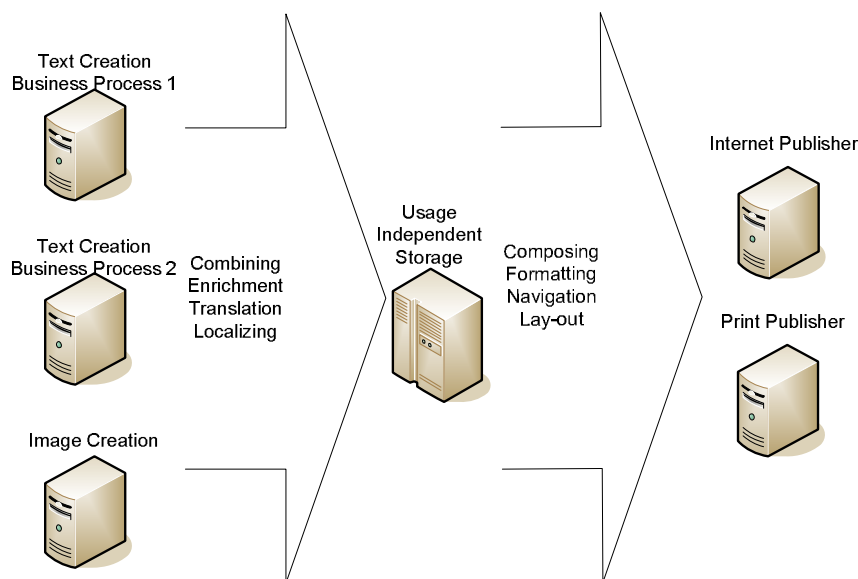
**Content Management is not about the management of static documents or information. Content is alive and will change during its lifecycle. Content is being reused, not by accident, but by design. And content is meant to be read by people, but before it is ready it needs to be processed from one format into another to bridge the gap between author and audience.**

**An IT landscape for this kind of content management contains multiple systems with many interfaces. In this article we discuss an efficient architecture that supports extensive communication of content.**

## Content creation process

Content creation, capture and collection ideally should be transparent to the authors [Paulssen04]: their profession is that of product manager, marketing manager, financial expert etc. They were hired for that specific expertise, not because they are especially good at copy writing. Today's content management systems require specific IT competences. Content creation at the information source in essence is nothing more than a by-product of a specific business process and should not require a lot of training from your highly paid experts.

Enhancing and enriching content should be done automatically whenever possible, since manual intervention is labor intensive and therefore expensive and above all error-prone. Rule-based enrichment can ensure that the correct metadata is in place for searching and retrieving the content. Enhancement of content often is combining multiple inputs into one specialized output or filtering content so only what is needed is communicated. If executed in an automated process all these activities are repeatable with lesser effort and higher predictability of content quality. Figure 1 gives an overview of the content management process from creation to publication.



**Figure 1**

In order to be able to publish to multiple channels storage should be kept generic and independent of usage [Manning04]. For example: in some cases the number of product features is restricted by how

many fit on an A4 leaflet. This is of course totally irrelevant when you want to present that product on a product comparison page on the internet. The real challenge lies in defining the appropriate level of granularity of control (Manning04): what information is placed in one document. Small content components increase the flexibility in recombining the data into documents, but it increases the management overhead. If you store larger components you lose some flexibility but in the case of XML that can be retrieved by filtering. Filtering in Content Management might mean either one of 2 things: either a specialized XML processing system filters away information that is not needed for a certain distribution channel right before sending the information, or the receiving system receives a file that contains also information that it doesn't need and that it has to filter away. In either case however, these files are not stored separately, so there is no duplication of this information. In general it is best to choose filtering if possible.

Publication should be tailored to the user's information needs by providing the right set of information, in the right language and jargon, with appropriate navigational structures and lay-out.

## IT implications

Content Management on a large scale will require multiple specialized IT systems: systems that seamlessly integrate into the business processes where content is created, so content is captured at the source with a minimum effort to the business. We need systems that automatically enrich content with more content or specialized metadata, again minimizing business effort and optimizing information quality. The system landscape has to support a number of processes that are necessary for building content. In particular we must mention translation and localization based on specialization of content to specific regional requirements, and formatting according to the different channels. All those systems will need to communicate, sharing the content.

Communication of content in this landscape is not a one time process [Rockley02]. The objects the content describes change over time. E.g. a common misconception is that product information, once published, should not change, or that if change is needed this is an exception rather than business as it should be. Corrections and adding missing information are a major reason why content needs to be updated, but over time new insights in an existing product may arise, legislation may change or the marketing communication strategy adjusted. All of these events will lead to extension of existing content or even new content for a product during its lifecycle. This is only possible if all of content management systems are consolidated in a single managed environment.

Real multi-channel content management is based on an IT architectural landscape that is constituted by many systems that intensively communicate between each other. This is not an exception; it is content management's core business.

## Actual situation: chained systems

A usual strategy to communicate content is to implement point-to-point interfaces between the systems that need to exchange information. Since content is progressively enhanced from the author's perspective to the audience's needs, the combination of systems is often referred to as a content management *chain*. Each interface in the chain is tailored to the alignment of sending and receiving system: interface model, technical format, data model, content scope and frequency of communication are agreed in this unique combination.

Over time more and more interfaces are needed. Changing interfaces becomes a challenge because at least 2 parties have to align on content, process and technology. Due to cost and time restrictions ad-hoc interfaces that do not follow corporate standards are built. In the long term that leads to a situation where any change in the system landscape or enterprise information policies will be associated with excessive implementation costs.

A typical Content Management Architecture consists of so many systems that it is not uncommon to see almost a full communication model: each system has interfaces to all other systems. This especially holds when content is extensively reused.

## The Content Broker concept

A more efficient communication model corresponds to a star architecture where one system in the middle receives all information and combines it from all sources and then sends out only the information needed by the target systems. We will call that system a content broker, see Figure 2.

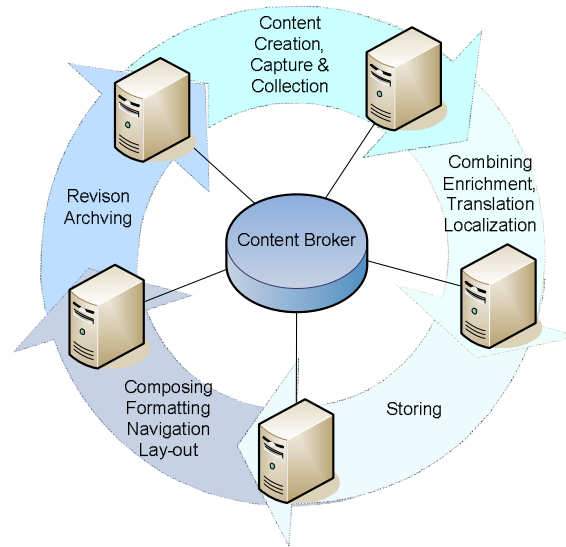


Figure 2

The architecture with a single system as content broker has an increased risk of communication failure as drawback. That leads to strong requirements on availability and stability. However that can be achieved by a large spectrum of technical solutions like the usage of clusters or Virtual Machines.

Of course the costs of maintaining the extra system should outweigh the costs of maintaining all separate point-to-point interfaces. A non-representative study showed that over 50% of all the work on maintenance could be accredited to interfaces. Introducing the content broker is estimated to reduce interface effort with 50% by reducing the total amount of interfaces, standardizing technology and decoupling business logic. In most large organizations that manage content in complex system landscapes the introduction of a content broker leads to cost reduction, and flexibility: the architecture itself ensures a reduction in the amount of interfaces to be maintained, and fixes a central point for implementing changes.

Additionally, flexibility can be considerably enhanced by limiting the content broker to contain only specific business processes that are associated with cross-system and/or corporate relevant content: all specific business logic and data structures should be kept in the target systems. This can be enforced by establishing standard XML interfaces that are as generic as possible and that describe content which is independent of the business processes supported by the target systems, more over, the content broker should only know fields that are used for managing the different syndication channels. All other content should be encapsulated in specific documents. If we have this, additional interfaces will cost less than the initial interfaces. Technically using XML is the preferred choice since we reside in the content management area where XML is already the standard for coding the content.

Any content broker should obviously have functionality to import and export (or syndicate) content. The import and the export do not match one on one, since multiple imports may be combined or vice versa for an export only a subset of the import is needed. Thus the content broker also needs to be able to recombine and filter information before sending it to the recipients. Besides sending out the correct content scope the content broker will send out only the content that actually has been changed.

The content broker will need a persistent store for content: i.e. subscribers that are added after go-live should be able to download the full content without the source systems having to resend all. A persistent store is also a business benefit: the store contains only published information, i.e. content of a certain predefined maturity. This ensures that the source systems can store work in progress without other systems immediately copying the content and business people immediately basing decisions on non-approved content.

In the next paragraphs we use the above requirements to discuss a possible implementation of the content broker based on a master data management system (SAP NetWeaver MDM).

## SAP NetWeaver MDM as Content Broker

To find and handle the mismanagement of master data, SAP introduced Master Data Management (MDM), which enables companies to store, augment and consolidate master data, while ensuring the consistent distribution of master data to all applications and systems within an IT system landscape [David06]. The last part of the prior sentence 'ensuring... landscape' would imply a possible fit to our requirements for the content broker. David goes on defining 3 different IT architectures, functionality groupings relevant to a particular implementation need:

- Master Data Consolidation: import data from multiple systems, cleanse and eliminate redundancy.
- Master Data Harmonization: import data from multiple systems, cleanse and distribute back to the connected client systems;
- Central Master Data Management: create master data within MDM and distribute to the client systems.

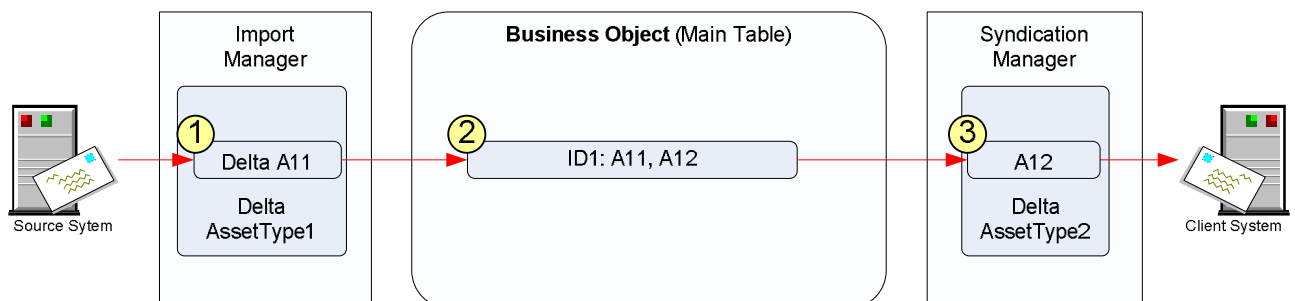
For our content broker we need a variant of Master Data Harmonization, i.e. we need import & syndication, but primarily we do not need the cleansing. The idea of distributing back to the originating systems is therefore not appropriate; we are communicating 'forward'. The concept of planned creation and subsequent enrichment in multiple systems and then sharing of master data over the entire Content Management Chain is not acknowledged in this business rationale of SAP NetWeaver MDM, yet it does not seem an illogical extension.

We will now look into the requirements that were defined in the paragraph on the content broker architecture and see how they are met by SAP NetWeaver MDM.

According to Loren Heilig et al. [Heilig07] master data is (relatively) stable data [...], which does not change with each order. The examples given are that of billing addresses or bank details. These are correct examples, but they do not cover the entire content scope that is managed within the multi channel content management environment as described in this article. From a content management perspective it is e.g. entirely plausible that an object has a couple of hundred assets associated with it, whereas the likelihood of a single customer having that many billing addresses or bank details is small. High performance and flexibility was achieved by storing the different sorts of assets in a specialized Asset Management system, and managing only the meta-data in MDM.

In MDM, import and syndication are not bound to one single technical format; several xml document formats are available. Defining a generic communication model can be done by defining import interfaces on the level of creation, update and delete of main object fields or assets.

Since business logic in the client systems is triggered by the arrival of new messages from the Content Broker, client systems require only real deltas, i.e. they require information to be sent only when data relevant to them has actually changed. The following example shows the challenge in this (see Figure 3).



**Figure 3**

In this situation a Client System receives updates of all assets of AssetType2:

1. Source System 1 sends out assets of AssetType1.
2. The update of A11 flags ID1 as changed.

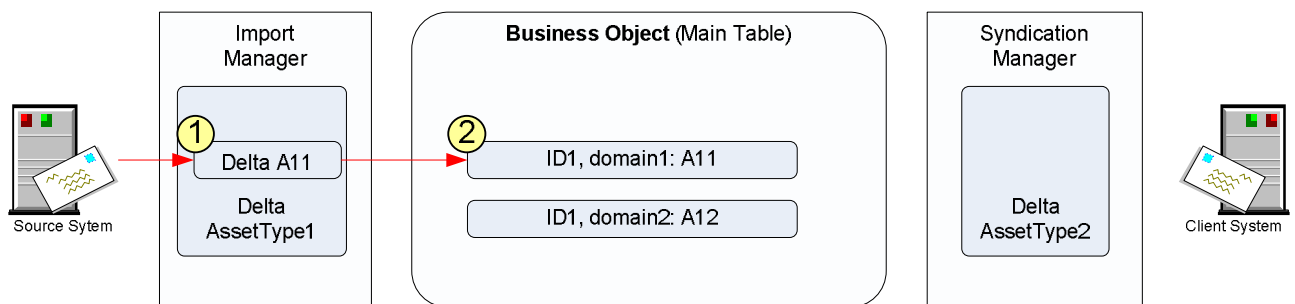
3. The syndication manager triggers syndication of all ports that involve ID1 and sends A12 to Client System 1 while A12 has not changed.

So each and every change on a record in the main table leads to syndication of all information of that record, which in many cases will be the same as sending a message with no actual change.

We identified two possible solutions for addressing this issue.

In the first solution we keep track of the last modified date registered by the source systems. In that way a filtering can be done either at the EAI level, or directly in the client systems. The processes triggered in the client systems will now depend on the real time, independently of the delays introduced by the distribution process itself. We would like to make the remark that the safest solution would be indeed to perform the filtering in the client systems.

A second solution to the problem, this time without involving other applications, requires a different way of modeling the business objects. Each record in the main table is an object for a specific domain. In the example given we now stop A12 being resent to Client System (see Figure 4) since the record that is used for syndication to Client System is not flagged as changed by the arrival of the update of A11.



**Figure 4**

If the domains are matched one on one to the syndication ports, syndication by default delivers real deltas. Drawback is that in a real life situation you will end up with a solution where the Import Manager duplicates incoming information to multiple records in the main table of MDM. If the domains represent different business processes however, unnecessary syndication is reduced considerably without introducing information duplication.

In SAP NetWeaver MDM we only need the fields that define the content scope as far as syndication is concerned. All other content can be encapsulated in assets and where the actual file is stored outside MDM with just metadata needed for retrieval and a referral to the file being stored in MDM. In this way the business knowledge is encapsulated as well. Technically the total number of assets will soon be too much for being managed in a flat lookup table in SAP NetWeaver MDM. The shared part of the metadata for each asset defines an asset type and can be modeled in SAP NetWeaver MDM as the non-qualifiers in the qualified lookup table [Heilig07]. The object specific attributes of the asset, like e.g. the document name, will be modeled as the qualifiers of the qualified lookup table and be stored with object record. The drawback of using qualified lookup tables is that the object defined by the non qualifiers cannot be directly de-duplicated, but it needs to be done through an extra repository in a similar manner as Martinez suggests for consolidating data in qualified tables in [Martinez07].

Using SAP NetWeaver MDM as a content broker provides additional benefits that are not in the essential requirements list, but that are beneficial in the use case of multi channel content management. There exists a certain class of information, which in the content management community is referred to as reference data that should be created centrally and then distributed to all client systems. Some examples are country names, language names and categorization labels. The creation and management of this data often does not justify the costs of a completely separate content management system. SAP NetWeaver MDM Data Manager can be used in the scenario of Central Master Data Management for this functionality.

A second additional benefit of using SAP NetWeaver MDM lies in its matching and cleansing capabilities. The metadata on business objects and assets that are published in MDM (versus the content that is embedded in the assets) can be checked for completeness and consistency. If any

anomalies are encountered a workflow could be started to alert the proper content manager. This is especially interesting when content migration projects are executed.

## Conclusions

SAP NetWeaver MDM meets the requirements for the content broker and is a natural fit although in some cases a little thinking is required. SAP NetWeaver MDM has the modeling capabilities necessary for modeling such a scenario. The tool has as well the necessary functionality for managing the content flow. On the other hand delta handling and its implications in the target processes must be analyzed with care. Additionally SAP NetWeaver MDM delivers out of the box a platform where content consistency can be analyzed and managed.

The success in the implementation of a Content Broker scenario depends greatly in a detailed understanding of the business processes and corporate information flow(s).

## Literature

[David06] Klaus David: **Centralize, harmonize, and distribute your master data with SAP NetWeaver Master Data Management (MDM)**, www.sdn.sap.com 2006

[Heilig07] Loren Heilig, Steffen Karch, **SAP NetWeaver Master Data Management**, SAP PRESS 2007

[Manning04] Steve Manning **Implementation Issues with Granularity**, in The Rockley Report, June 2004

[Martínez07] Juan Carlos Martínez Gil **How to consolidate data in qualified tables**, www.sdn.sap.com 2007

[Paulssen04] Natasja Paulssen (2004) **Het managen van de mierenhoop**, VIP Magazine 5-2004

[Rockley02] Ann Rockley **Managing Enterprise Content: a unified content strategy**, New Riders 2002.