JUST.
SYSTEMS

DITA Maturity Model

By Michael Priestley, IBM and Amber Swope, JustSystems

A JUSTSYSTEMS WHITEPAPER

The Darwin Information Typing Architecture (DITA) is an open standard from OASIS[1] for creating, managing, and publishing modular content. It supports the definition of new content types within a comprehensive content ecosystem, and has been increasingly adopted across a wide range of content disciplines and industries.

DITA overview

You will better understand how DITA can support your organization and how it can scale to meet your enterprise content needs by first understanding the basics of DITA standardization.

DITA topics and maps

DITA is a modular, structured, XML framework based on topic-oriented content. This means that content developers author units of content, called topics, which can then be assembled into deliverables, such as books and Web pages. Typically, each topic covers a specific subject with a singular intent, for example, a conceptual topic that provides a system overview, or a procedural topic that tells readers how to accomplish a task.

In addition to chunking information into small units, DITA structures content by type. By default, DITA provides a base type, *topic*, and several more specialized types: *task, concept, reference,* and *glossary entry.* Each type has a specific structure that defines the valid elements that can exist within that type. For example, DITA does not allow you to create a <step> element in a concept topic because steps are parts of procedures and thus belong in task topics.

You can organize topics into collections using a DITA map, which can then be used to generate a Portable Document Format (PDF) file, Web site, or other information application. Maps can reference topics and other maps. Because the same topics and maps can be reused in many different collections and deliverables, DITA enables a powerful reuse architecture that can scale from simple Web pages or newsletters up to complex inter-related libraries or information centers.

DITA specialization

No one set of content types could meet the needs of every organization or even of a given organization as it grows. For this reason, DITA supports the creation of new content types and collection types as required. Specialized types can exist at many different levels: for example, a relatively generic type such as reference might be specialized for a particular subject area, such as semiconductor design, or for a particular company's needs, or for a product area within a company.

Specialized types can inherit associated behaviors from their more generic ancestors, so even new DITA content types can be included in standard publishing streams, although it is common to extend processing to take advantage of some of the new markup. For example, a new content type for policy analysis might introduce sections for risks versus rewards, and processing could be extended to automatically create subheadings for the new section types.

DITA support

DITA is an open-source standard approved and supported by OASIS. Participating OASIS members, drawn from active vendor and broad user communities, work together on the DITA Technical Committee to evolve the DITA specification.

In addition to vendor-specific implementations of the standard, the DITA Open Toolkit[2] provides open-source processing support for the specification. As DITA matures, more companies and organizations are participating on the Technical Committee and its subcommittees, contributing functionality to the Open Toolkit, and contributing specializations to the community.

[1] Organization for the Advancement of Structured Information Standards www.oasis-open.org
[2] http://dita-ot.sourceforge.net/

Because of the benefits of XML in general, such as the separation of content from format, and of DITA in particular, DITA is becoming a popular information model in today's global, multi-channel environment.

Because of the benefits of XML in general, such as the separation of content from format, and of DITA in particular, DITA is becoming a popular information model in today's global, multi-channel environment.

Maturity Model investment/return summary

One of DITA's most attractive features is its support for incremental adoption: you can adopt DITA quickly and easily using a subset of its capabilities, and then add investment over time as your content strategy evolves and expands. However, this incremental continuum has also resulted in confusion, as communities at different stages of adoption claim radically different numbers for cost of migration and return on investment.

The DITA Maturity Model addresses this confusion by dividing DITA adoption into six levels, each with its own required investment and associated return on investment. You can assess your own capabilities and goals relative to the model and choose the appropriate initial adoption level for your needs and schedule.

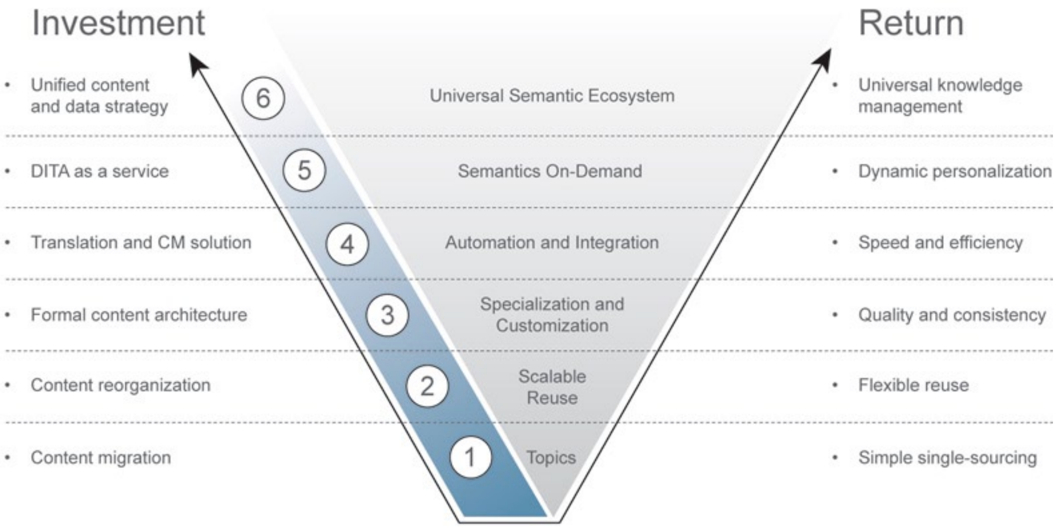

Figure 1. DITA Maturity Model

The DITA Maturity Model: investment/return summary

- **Level 1: Topics** – achieve simple *single-sourcing* by migrating current XML content sources.
- **Level 2: Scalable reuse** – achieve *flexible reuse* by architecting content using DITA topics and maps.
- **Level 3: Specialization and customization** – achieve *quality and consistency* by expanding DITA architecture to a full content model, which explicitly defines the content types required to meet different author and audience needs and specifies how to meet those needs using structured, typed content.
- **Level 4: Automation and integration** – achieve *speed and efficiency* by leveraging investments in semantics with automation of key processes, and unify the semantics across different specializations or authoring disciplines.
- **Level 5: Semantics on demand** – achieve *dynamic personalization* as DITA is adopted as a cross-application, cross-silo solution that shares a common semantic currency for content authoring and management needs.
- **Level 6: Universal semantic ecosystem** – achieve *universal knowledge management* with a new kind of semantic ecosystem that can move with content across old boundaries, wrap unstructured content, and provide validated integration with semi-structured content and managed data sources.

Level 1: Topics

At its most basic level, DITA is an XML document markup language; but even at its simplest level, DITA enforces a topic structure and reuse architecture that allows DITA documents to reuse content from other, more structured projects. This standardization also sets the stage for topic-level reuse by others as an initial migration of document-oriented content evolves to incorporate better management and authoring practices around topics and maps.

Scenario

An author for a government agency may need to produce audience-specific versions of a government policy. The author can write all the content in one file and apply conditional processing values to produce different versions of the policy for permanent and contract employees.

Investment

The minimum DITA adoption requires that you migrate the current sources of content in XML. You do, however, have the flexibility to decide which sources to migrate when, and how much structure to apply to the migrated content. Many teams have a large amount of legacy information that was authored in a variety of sources. Some teams may choose to migrate only the content that will require updates in the future. Other teams migrate everything, but do not move the content into typed topics; instead they move the content en masse into generic topics, which are the least restrictive topic type and hence require the least amount of content restructuring. However, the generic topic type also provides the least amount of semantic value.

Figure 2. Topics

Another way that teams save time at this level is to defer splitting the content into discrete topics and simply recreate their existing document-focused structure by nesting multiple topics within a single file. For example, recreating chapters as DITA files allows you to continue to store all the chapter content in a single file. While this strategy takes less time than restructuring the content into units based on subject, it does not provide small enough units of information to enable easy reorganization of the content into multiple deliverables.

Because XML separates the formatting from the content, the transform for each deliverable type applies the styles and formatting defined in the cascading style sheets (CSS) when you generate or publish the deliverable. Although the DITA Open Toolkit provides default processing for multiple deliverable types, you must customize the transforms to generate deliverables that meet the style, standard, and branding requirements for your organization.

Return

Even with minimal investment, you can realize returns from adopting DITA. Many teams make the move to DITA to gain greater reuse of their content. Working with their current source, they use conditional processing to generate multiple versions of the same document. Even with non-typed topics or multiple topics in the same file, you can easily specify conditions and generate conditional output with DITA. This remains the primary means for reusing content at the first level of adoption.

However, to make progress toward the goal of additional reuse, you can use DITA to meet the challenge of publishing new or multiple deliverables that contain the same information by single-sourcing the content. The DITA Open Toolkit provides default output processing for a wide variety of popular formats, including HTML files, Eclipse[3] plug-ins, PDFs, and CHM (Microsoft® Compiled HTML Help) files. You can easily generate the same information in multiple formats by specifying a different output type when you publish.

When you publish content, the publishing transform applies the specified formatting to each element, which allows you to easily update format styles for large quantities of information. For example, if the style for highlighting the first instance of a term is italics, but later is changed to bold, you simply update the CSS and regenerate the deliverables. This is much more efficient than searching for and updating each instance of a term or style element across the information set.

For links between content, most teams use hard-coded cross-references in their current source. At the basic DITA-adoption level, you can continue this practice and link between DITA topics, as well as to external documents or locations, such as Web sites.

Lastly, at this level, most teams utilize minimal or unmanaged metadata and primarily focus on terms, such as index terms.

By migrating the content source to XML and chunking it according to the appropriate topic type, the first level of adoption supports conditionally generating output and positions you for greater reuse and output flexibility at the next level.

DITA features used

This adoption level uses the following DITA features:

Nested DITA topics

DITA provides the ability to nest topics hierarchically within a single XML file. You can chunk the content by topic type, but you don't have to create separate files.

Cross-reference elements

You can create cross-references to elements, such as linking to other topics, to non-DITA files, to Web pages or to specific sections referenced from within a bulleted list. These references are hard-coded into the content, which may have implications when you reuse the content.

Conditional processing

You must define the processing attribute and its valid values in the .ditaval file in order to conditionally process content. By default, DITA provides three processing values, but you can create additional values by specializing the props attribute.

[3] http://www.eclipse.org

Level 2: Scalable reuse

Topic-oriented authoring creates reusable content organized around an audience's primary unit of use, the safest and most scalable reuse strategy for most modular content. The same topics can be reused, reassembled, and reorganized for different media and for different variations on a subject, such as documentation for product variants, by using DITA maps to encode higher-level structure, such as chapters or even Web pages, outside the topics that make up a deliverable.

Scenario

A small technical publications team for a mobile phone vendor can organize the same content differently to optimize the user experience for a book versus a Web site. They can use the bookmap specialization to provide book-specific items, such as a cover page, notices page, and appendices, and another DITA map for the HTML output that does not require these items. They can also generate embedded online help from the same content for display directly on the phone.

The following figure shows the same topics appearing in multiple maps.

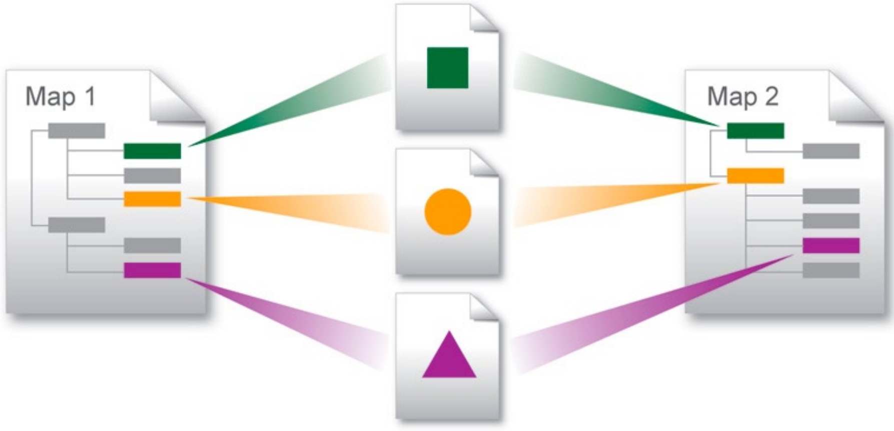

Figure 3. Multiple maps using same topics

Investment

The major activities at this level are to break the content down into topics that are stored as individual files, and then to use DITA maps to collect and organize the content for output as specific deliverables. This effort requires that you create an information architecture that includes the following information:

- Lowest level of reuse, which may be at the phrase, element, topic, or map level.
- Strategy for content reuse, which identifies the mechanisms for reuse at each level.
- Metadata, which can be used for both build-time and run-time filtering if it is standardized and properly managed.
- User access paths, which specify how users access content and then navigate through deliverables using links between topics.

The ability to reuse content in a scalable manner depends upon knowing what you have, how it fits together, and what you need to do with it.

Return

At this second level of adoption, you realize the value of flexible reuse by using DITA maps to assemble each deliverable. Because each map is specific to a deliverable, you can optimize the content to include the organization of the content and the links between the topics for each deliverable type.

DITA maps provide a way to abstract the relationships between topics that result in links from the topics and to specify the relationships within the map. This ability is crucial for reuse. You cannot reuse a topic in multiple components if it has a hard-coded link to another topic that might not be included in every component. When you specify component-specific relationships in the maps rather than including links in topics, you are free to use the topic in any component where the content applies without fear of broken links.

In the following figure, Map 1 and Map 2 reference specific topics in a repository to generate multiple deliverable outputs.

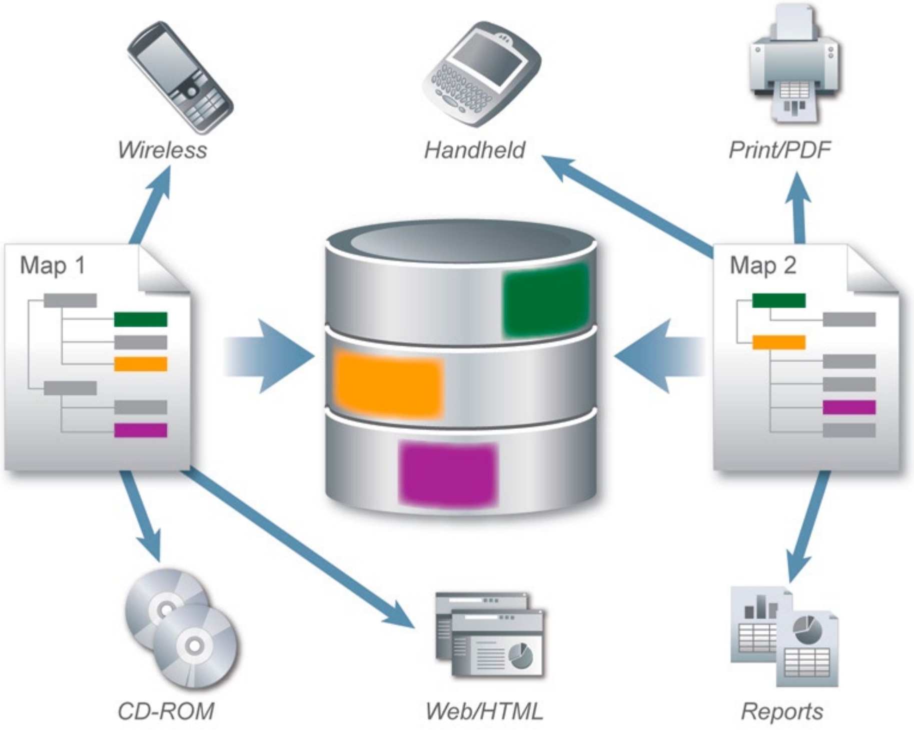

Figure 4. Multiple maps using the same topics

Another way that DITA maps help you reuse information is by grouping sets of topics into units that can be further organized into components and easily included in multiple outputs. Consider the mobile phone technical publication team tasked with creating documentation for various phone models, each with different combinations of features. If the corresponding content for each feature set is organized by DITA maps, the technical team can quickly generate the appropriate documentation for each phone by assembling the feature set maps using a phone-specific map. In this way, the organization can tailor its documentation to specific end-user groups and thus increase customer satisfaction across those groups.

In addition to organizing topics into maps and applying conditional processing at the element, topic, and map levels, DITA provides a mechanism to reference content from one topic to another. With the **conref** attribute, you can reference content with a unique ID into another topic. The benefit is that you can maintain "a single source of the truth" in a topic and display that content in multiple places. Additionally, content updates performed at this single source will be automatically reflected throughout all information outputs the next time you generate the topic. This mechanism is particularly useful for managing common content, such as legally approved notes or acronym lists, and for maintaining variable content, such as product names that require global updates across the content set.

In the case of the mobile phone technical publications team, instead of hard-coding a phone model name into the content, they can create a reference to the model name and conditionally process the reference to automatically include the appropriate name for each phone model.

However, you must have a strategy for tracking and communicating when information is referenced, updated, and generated. Without a strategy to handle this communication, there is a great risk of negatively impacting content accuracy and quality by inadvertently changing content.

Although you can reuse content with DITA in many ways while storing the content on a file system or in a source control system, you can only reuse content that you can find. This means that you must provide a way, through process or technology, for content authors to find and reuse information. Like a source control system, a content management system (CMS) maintains content integrity and supports content versioning; however, it also optimizes content retrieval through managed metadata and provides workflow management. In addition, a CMS can quickly identify where content is reused and help you avoid unintentional propagation of changes throughout the content set.

When you organize topics into deliverables by using maps, you can easily control the content for deliverables and generate custom output without impacting the content in the topics. In addition, you can reduce redundant authoring by reusing content at the element, topic, or map level.

DITA features used

This adoption level uses the following DITA features:

DITA maps

You create DITA maps to generate various deliverables. The DITA map serves three purposes:

- Manifest for the deliverable: all topics that contain content to appear in the deliverable must be listed in the map.
- Hierarchy for the deliverable: the TOC or hierarchy for the deliverable is determined by the nesting of the topics in the map.
- Defined relationships between topics: the non-hierarchical links between topics can be specified in the map, which allows the topics to be used for different deliverables.

Content references

DITA provides a reuse mechanism through the conref attribute, which allows you to reuse elements with a unique ID in various locations either within the same topic or another topic, as long as the source and target are the same element type. One consideration is that many content management systems only support references to entire files, so the content source must be saved as a separate file.

Level 3: Specialization and customization

With specialization, DITA can provide structural support for information typing strategies, improving authoring consistency and guiding quality improvements. Specialization can also model content more closely for particular subjects or types of deliverable, which can be leveraged by semantic search and customized processes.

Scenario

An insurance company team wants to author all their content in XML to take advantage of the conditional processing and multi-channel output. They create a domain specialization, as well as structural specializations for claims, and policies and procedures in order to handle the insurance-specific concepts. With all the content sourced in XML, they can automate their system to combine policy and procedure information with actual claim information to create just-in-time compound documents.

Investment

In this third level of adoption, you expand the information architecture to be a full content model, which explicitly defines the different types of content required to meet different author and audience needs, and specifies how to meet these needs using structured, typed content.

Organizations that use DITA benefit from the ability to specialize or evolve the standard to provide the structure and semantic control needed for their content model. They can create their own specialization or participate on the DITA Technical Committee and work with others to create industry or content-specific specializations. DITA specializations require resources, time, and expertise, but provide content structure standardization.

In addition to creating new structural standards, organizations may choose to customize transforms to provide customized output deliverables, such as training materials or data sheets.

In an industry where several companies work together and exchange content, it makes more sense to develop a common specialization that structures the content to meet industry-specific requirements than for a single organization to develop a specialization that applies only to their content. The benefits of working on a common specialization are that you can easily incorporate and re-brand content as well as share the resource burden for specialization development.

Return

By investing in a content model that differentiates between the needs of the content authors and deliverable consumers, you can truly customize the output deliverables to meet the needs of various audiences. The first step is to adopt specializations supported by the DITA Technical Committee (TC) to provide more structure for authors when creating common content types. By utilizing these specializations, you make it easier for authors to create consistent information and maintain a standards-based architecture that supports interchange with other teams or organizations.

The next step is to create specializations to meet the specific needs of your organization, industry, or users. There are different types of specializations:

- Topic information type specializations, such as glossary or API reference, which provide a standard structure for authoring specific types of information.
- Deliverable specializations, such as bookmap, which provide a consistent structure and metadata optimized for a particular deliverable type.
- Domain-specific semantic and structural specialization, such as semiconductor design documents, learning materials, policy and procedure documents, and financial documents, which have standard structures within the domain or industry.

As more industries embrace standards for increased quality and reliability, specialization can provide structure for meeting the standards as well as provide a mechanism for thought leadership.

The following figure shows how task, concept, and reference topics are specialized from the main topic type and how you can specialize directly from the main topic type or from any of the other specializations.

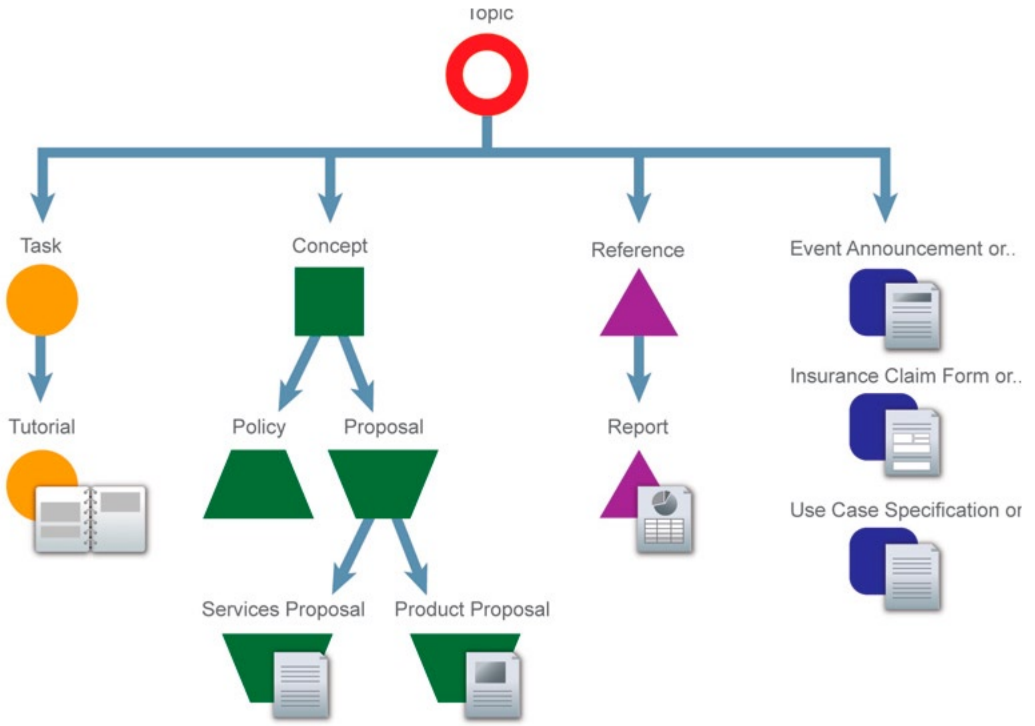

Figure 5. Specializations

Once you specialize to specify semantic values, you can customize the content processing to leverage additional semantics. For example, once an insurance company team has created specialized markup for the provider of a policy, they can quickly create summary tables of policy claims, arranged according to provider.

In addition to providing consistency and control for content authoring and publishing, you can initiate discipline-specific quality initiatives, such as task analysis for technical documents, or training or use case development for engineering. These types of process maturity activities also include identifying all the stakeholders in the content creation and generation processes and providing appropriate, customized authoring and editing experiences for each stakeholder role. For example, if the team has a mix of professional content developers and subject matter experts that collaboratively author content, you can tailor the authoring environments to meet the team's various needs. For example, the subject matter experts may need a subset of the functionality required by the professional content creators.

Creating more standard, well-formed information at this third level of adoption provides a basis for improving quality and consistency across the content set.

DITA features used

This adoption level uses the following DITA features:

Specialization for different authoring needs/audiences

Specialization allows you to extend or evolve the DITA specification to create domain-specific or structure-specific content types. You can apply specialization at both the topic and map levels.

Modular processing architecture for shared infrastructure

Even as specializations create new markup, they can continue sharing processing logic and applications with unspecialized or differently specialized content. DITA's processing architecture allows for easy extension and customization by adding, removing, or overriding specific modules in a processing chain.

Level 4: Automation and integration

Once content is specialized, you can leverage your investment in semantics with automation of key processes, and begin tying content together—even across different specializations or authoring disciplines. For example, you can share common content across marketing and training, or share common processes and infrastructure throughout your content life cycle.

Scenario

The software division of a large technology company stores their content in a CMS, which allows all the teams in the division to reuse the content. At this level, they have moved beyond single-sourcing of content and achieved multi-way reuse. Product descriptions created by the marketing team can be reused by the technical publications group to create product overviews, and by the training group to create product tours. At the same time, product architectural specifications created by technical publications can be reused by training, technical support groups, and the marketing team.

The following figure illustrates how content created by different teams can be reused in multiple deliverables by multiple teams across the division.

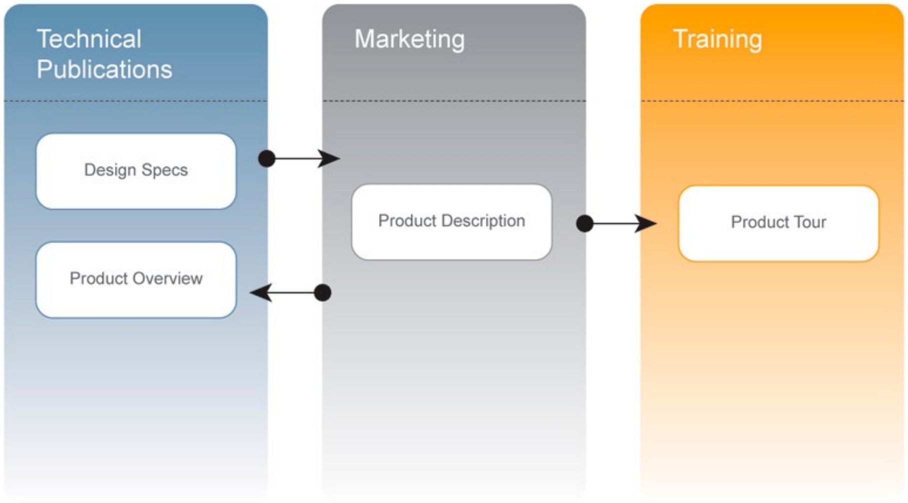

Figure 6: Content reuse across teams

Reusing its content across the teams in the division, the company can save a significant amount of money by translating the content source rather than each deliverable that instantiates the content.

Investment

Organizations need a CMS to effectively control and automate the content development life cycle. In addition to storing content and providing versioning control, the CMS provides workflow automation support that assists authors in creating, reusing, and publishing. However, the investment in implementing a CMS is non-trivial in terms of preparation and cost.

In preparation for a CMS implementation, you must understand the structure of the content and where it is appropriate for reuse. This requires a significant amount of research, planning, and coordination to identify the reuse possibilities, requirements, and standards across disciplines. In addition, you need to define a robust metadata model to support the content model and apply it to all topics. Lastly, you must have agreed-upon content development processes in order to automate them with workflow control. This requires consensus and support from all stakeholders in the content life cycle. The cost for implementing the CMS includes the following items:

- Price of the CMS software
- Hardware to run it and store the content
- Resource time to prepare and plan for implementation
- Resources to customize and maintain the CMS
- Resource time for training stakeholders to use it

Although such an undertaking may seem daunting, the initial implementation is a one-time cost but the improvements in speed and efficiency will allow you to recoup the investment in a minimal amount of time.

A translation management system is another key automation and integration investment to manage and automate content localization. If you are translating content into more than one language, you must have processes in place to handle this additional work. A translation management system provides automated process management for translating content and integrates into the CMS workflow support.

To implement a translation management system, you must have a defined translation process that can scale to meet your localization needs as they increase, and you must understand the requirements for a scalable system. In addition, you must build your translation memory, which is the library of localized content.

Return

The return on investment in a CMS is the ability to reuse content across disciplines and automate the content development workflow. If content is not stored in a repository that provides easy retrieval through metadata, it will be impossible to reuse content across teams. In addition to obvious characteristics such as automated status change notification and reporting, workflow support enables you to see quickly what information is reused in which topics. This crucial feature of this fourth level of adoption enables true reuse and mitigates the risk of inadvertently propagating change throughout the content set.

The following figure shows how users can share content stored in multiple repositories.

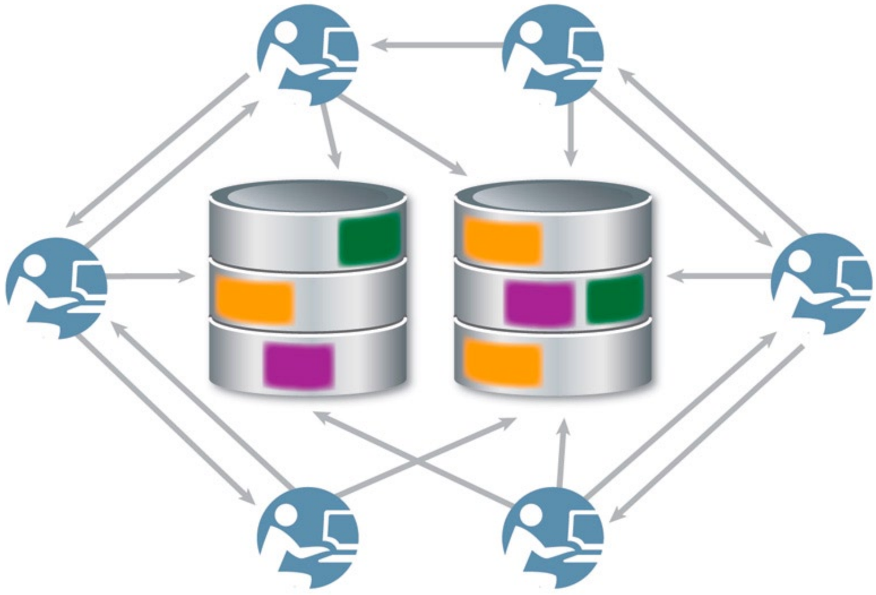

Figure 7: Multiple users sharing content from multiple repositories

Traditional publishing and translation processes involve sending each deliverable out for translation. Although you can leverage the translation memory for the content in each deliverable, the translation vendor must compare each deliverable to the translation memory to determine what content is new and what needs to be translated. If you have multiple deliverables with the same content, you pay for each analysis pass. If you have multiple deliverables with similar but non-identical information, you pay for the analysis pass, as well as the cost to translate each "version" of the information. Organizations that produce multi-language documentation can incur large, unnecessary costs if they have to multiply the number of languages by the number of versions of the content for each release.

In contrast, because DITA is an XML topic-based architecture, you send only the source topics that contain changed content to the translation vendor. This means that you can control the content in smaller units, and thus the amount of content the vendor analyzes for each language is significantly reduced. In addition, if you are reusing content rather than rewriting multiple versions of it, you simply pay to translate the original source instead of multiple versions of the same information. Content that is translated at the source rather than at the level of each deliverable, radically changes the translation cost structure. The ability to translate content at the source, combined with the ability to identify changed content and thereby reduce the actual amount of content by reuse, gives you greater control over the translation process and your overall localization costs.

By automating workflow support with a CMS and integrating the translation process, you can reuse content with confidence across teams and realize significant savings when localizing to multiple languages.

DITA features used

This adoption level uses the following DITA features:

Metadata

DITA provides some basic metadata attributes for all topics, including author, audience, resource ID, keywords, and index markers. Maps also have default metadata, including copyright information and critical dates. However, specializations provide additional, deliverable-specific attributes. For example, the bookmap specialization includes book-specific metadata including book identification numbers and publication data.

Translation and language attributes

DITA provides the translate and xml:lang attributes to support localization. The translate attribute "indicates whether the content of the element should be translated or not."[4] The xml:lang attribute identifies the language into which the content should be translated. You can specify these attributes at the element, topic, or map level.

Generalization for cross-specialization reuse

When reuse happens across different content types, issues of cross-type validation can quickly result: some of the semantics in the source may not be valid in the context of reuse. For example, a <step> is allowed in a task topic but not in a concept topic. But since a <step> is just a specialized type of list item (<li>), you can reuse a <step> any place where a <li> is allowed by stripping away the extra semantics that do not apply in the new context. In this way, you can reuse the content of a <step> between tasks and concepts, even if the specialized semantics and structure only apply in the source type.

Level 5: Semantics on demand

As DITA diversifies to occupy more roles within an organization, single-application solutions can no longer provide the specialized support each author or product may require. Instead, a cross-application, cross-silo strategy that shares DITA as a common semantic currency lets groups use the toolset most appropriate for their content authoring and management needs, while sharing content and even moving authoring responsibility between groups throughout the content life cycle. Beyond automation of known processes, we now have the flexibility to combine new applications and sources of content as needed, providing processing flexibility and an adaptable, evolutionary content strategy.

Scenario

A financial services company can integrate financial data from a trusted source with quarterly report text and product marketing overviews written in DITA to create different combinations of year-in-review content for employees versus investors. They can also use DITA to create subscribable feeds for news and updates about specific products and investment tips or news items that match an individual investor's portfolio or profile.

Investment

There are several major investments needed to reach this level. First, content applications need to be enabled to integrate not just with particular peer applications, but with any peer application that can provide and consume DITA topics and maps. This goes beyond existing DITA content applications and becomes a strategy that covers every source of semantic data or content: DITA becomes the common currency between semantic applications. Data can be exposed as DITA maps; structured or semi-structured content can be exposed as DITA topics at various levels of specialization; and unstructured content such as PDFs, images, or multimedia files can be wrapped using DITA maps to provide a common interface for associating and storing titles, descriptions, and metadata.

Every application that authors, manages, relates, consumes, or publishes content becomes a service that provides DITA content as subscribable feeds. Unlike traditional RSS feeds, DITA feeds have scalable semantic bandwidth: they allow applications with different levels of semantic understanding to continue sharing content. This is accomplished through common agreement on a content currency or language that itself maintains multiple levels of semantics.

Second, an organization needs ways to organize and retrieve these newly consumable sources of DITA content, which means, at a minimum, some basic taxonomies for subject area or product, and potentially a full suite of taxonomies to serve both internal and external audiences, including values for audience, platform, activity, required skills, and so on. These values allow the rapid retrieval and organization of discovered content into task-specific or role-specific assets: for example, a cross-product installation and orientation guide for a new customer, or a customized set of learning materials for a new employee given the role of administrator for three unfamiliar products.

[4] DITA Version 1.1 Architectural Specification

DITA content services and taxonomies work together to provide a standardized level of semantic interchange across any enabled application, moving the focus from integrating proprietary application APIs to providing general content exchange services based on the DITA schemas and other standards such as Atom[5] and Representational State Transfer (REST)[6]. The result is an application ecosystem as diverse and interoperable as the content ecosystem it supports, which can be quickly extended or adapted by adding or replacing components to meet evolving content management or delivery needs.

Return

One of the most immediate and visible returns on investment at this fifth level of adoption is the ability to dynamically personalize content: putting the power of DITA metadata, topics, and map-based publishing into the hands of the audience. When we make data and content available as DITA, it can be integrated and republished using DITA pipelines and services: for example, creating custom PDFs that include indexes and tables of key figures. When we wrap unstructured content using DITA maps, we create a single interface for finding, retrieving, and mashing together any online resource in the organization: for example, tying company financial data into quarterly reports, along with comparative stock performance and industry news feeds.

The following figure shows content wrapped in DITA maps being used in multiple outputs.

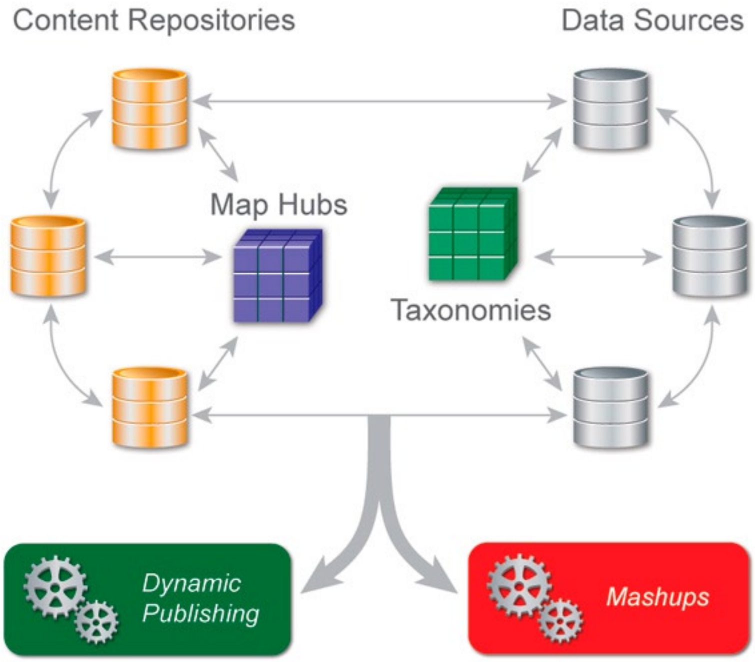

Figure 8: Dynamically instantiating and publishing

All of this is possible without DITA, of course: dynamic personalization, mashups, and data and content integration have many examples outside the realm of DITA. What DITA offers, however, is a way to gain dividends from your investment by making content and services shareable not just within a repository, or between a specific repository and its consuming applications, but across any and all repositories and services that use or can provide a common unit of content and metadata: DITA topics and maps.

[5] http://www.mnot.net/drafts/draft-nottingham-atom-format-02.html
[6] http://en.wikipedia.org/wiki/REST

DITA features used

This adoption level uses the following DITA features:

DITA feeds

Feeds of DITA content allow reusers to subscribe to particular topics or maps, receiving automatic notifications when the source is updated. The feed can provide standard abstracts and metadata to simplify search and retrieval across multiple repositories.

Generated DITA for semantic interchange

Content from other structured sources such as databases or other XML standards, or even semi-structured content such as wiki pages, can be exposed as DITA topics of varying types. The generated DITA view of these different information sources provides a common integration point for different types of content, allowing the exchange of structured or semi-structured content across format boundaries without loss of underlying semantics. For example, a wiki how-to article type could provide a DITA task form that allows the wiki page to be reused in support portals or policy manuals.

DITA wrappers for non-DITA content

Even unstructured or binary content, such as word-processing files or multimedia, can be described in DITA terms using a map that assigns a DITA title, short description, and metadata, making unstructured content in a common repository as searchable and retrievable as the DITA content it sits alongside.

Specialized elements for data integration

DITA provides special elements such as <data> and <simpletable> that are designed to integrate data from structured sources such as databases or other XML sources. By specializing these elements, a DITA architect can create integration points between DITA and data sources that provide semantics and validation around where and how the data is included. For example, a specialization for financial reports could include standard elements for selecting and formatting earnings information.

Dynamic content

As DITA content becomes subscribable and mashable with other types of content, new dynamic capabilities become available: Web sites can be personalized, users can construct their own manuals, and information delivery becomes dynamic, automatically incorporating new topics or changes to topics. For example, a user could create a custom travel guide that includes a mix of Web articles, restaurant recommendations, and subscribed travel alerts that could be downloaded to their phone and automatically updated as the original DITA sources change.

Level 6: Universal semantic ecosystem

As DITA provides for scalable semantic bandwidth across content silos and applications, a new kind of semantic ecosystem emerges: semantics that can move with content across old boundaries, wrap unstructured content, and provide validated integration with semi-structured content and managed data sources. DITA becomes the semantic interchange standard for cross-organization, cross-standard, universal content use.

Scenario

Companies that can share all their information across company boundaries allow new partnerships. For example, a publishing company can incorporate data from real product specifications into articles about the product; governments can combine information from any level of government that's relevant to a particular citizen's problem; applicable legal precedents can be attached to contentious insurance claims...the list is endless because all the information can be used where it is needed.

The following figure indicates how organizations that make the move to DITA become part of a semantic ecosystem that enables information sharing and collaboration where and when it's required, without expensive infrastructure negotiations.

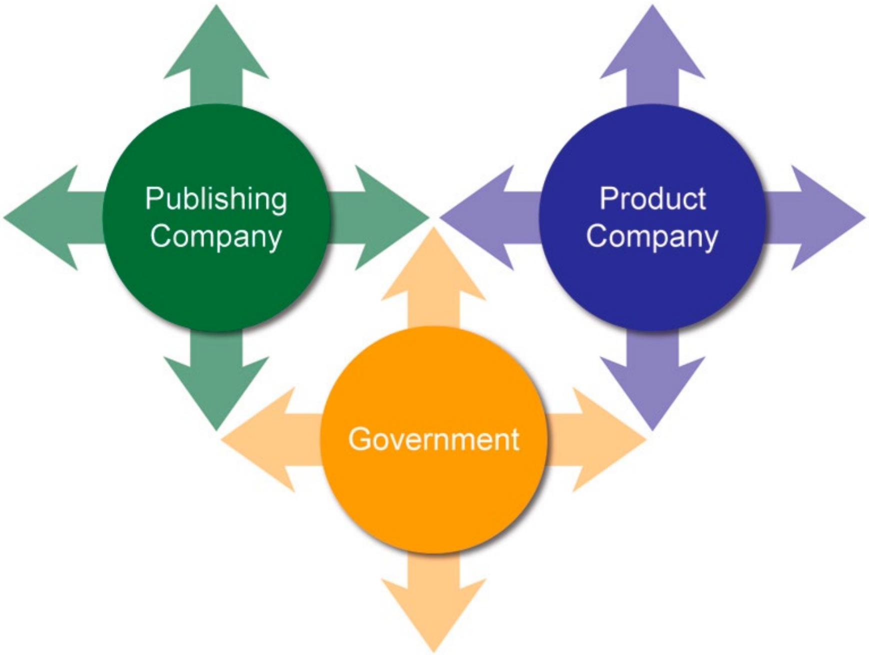

Figure 9: Unified Semantic Ecosystem

Investment

The greatest investment at this level is in the following efforts:

- Propagating a pervasive change of mind set across an organization and its partners.
- Recognizing the value of standardization and collaboration.
- Establishing common goals and processes, not only for current sharing of infrastructure and content, but as a foundation for the ongoing evolution of people, processes, and technologies that can take advantage of new partnership opportunities as they emerge, bypassing many of the traditional barriers to organizational change and process improvement.

Return

Traditional knowledge management depends on the consolidation of knowledge resources and processes into a few tightly integrated applications and repositories. In other words, the challenge of cross-silo knowledge flows is typically managed by creating bigger silos. This approach is problematic even within an enterprise where differing knowledge needs can drive differences in tool choice and content architecture. The approach is almost impossible to scale across multiple enterprises: even if you could convince everyone within an organization to converge on a single repository and tool platform, it would be nearly impossible to convince business partners and collaborators to discard their existing investments for the sake of returns on a minority of content shared across organizations.

A universal semantic ecosystem replaces this notion of monolithic, proprietary silos with an adaptable network of applications that can share content and integrate processes wherever organizational agreements allow or require. Because the connections are based on open content standards rather than on proprietary APIs, different parts of the network can provide radically different kinds of services without breaking agreements on shared content and metadata. The network as a whole can evolve asynchronously at this sixth level of adoption. Each part meets local needs without compromising global interoperability, and enables a radical change in how change itself happens: from crisis-driven revolutionary upheavals to evolutionary, incremental, adaptive growth in which the best ideas and applications are shared and propagated freely without requiring wholesale replacement of systems and processes.

Summary

DITA is a flexible, scalable architecture that can provide process and content improvement at each maturity level. With a minimum of investment, you can start to realize the benefits of authoring in XML topics. As your organization's needs increase in sophistication and complexity, you can more fully implement DITA to support your dynamic content vision. See *Table 1 DITA Maturity Model* reference for an overview.

Feedback

This document is intended to provide information about adopting DITA using a graduated approach. The DITA Maturity Model is the first and only such contribution of its kind and is intended to be a living document that will evolve over time through application of the concepts herein and direct feedback from the community.

Please submit your comments and feedback to: info@justsystems.com

---

Biographies

Amber Swope is a Principal Consultant at JustSystems, where she applies her information architecture and DITA experience to help clients address their business challenges.

Amber's experience as information architect spans almost 20 years in the information development field and encompasses her work with teams throughout the full information development lifecycle, authoring numerous papers and articles, and presentation at leading industry conferences. Amber holds a Masters in Technical and Professional Writing.

Michael Priestley is the lead DITA architect for IBM, and co-editor of the OASIS DITA 1.0 and 1.1 specifications. He is an experienced information architect and XML architect, and has presented and published prolifically on information development processes, information design principles, XML development techniques, structured authoring and Web 2.0, and of course DITA. He is currently supporting new DITA projects and working on DITA 1.2.

DITA Maturity Model reference

The following table provides an overview of the maturity levels and key adoption points.

Table 1: DITA Maturity Model reference

Level	Scope	Key DITA Features	Reuse Strategy	People	Process	Tools
1 DITA topics	Individual	• Nested DITA topics • Cross-reference elements	• Conditional processing across single structure for all outputs/ variants	• Basic editing skills	• Preserve existing processes	• WYSIWYG editor • File or source control system • Editor-based publishing
2 DITA maps	Team	• DITA maps • conref • conditions	• Reuse of topics through maps for managing multiple deliverables/ product variants • Reuse of elements through conref for common/volatile content • Conditional processing for remainder	• Team-level coordination • Map editing skills • Reuse planning skills	• Reuse process • Configuration management • Component content management	• Map editor • Version control system • Scriptable toolkit
3 Specialization and customization	Product family	• Specialization for different authoring needs/ audiences • Modular processing architecture for shared infrastructure	• Repurposing content with multiple processes • Sharing design modules through specialization • Sharing processing modules	• Content analysis skills • DITA specialization skills • XSL/XSL:FO skills • Content authoring skills • Information architecture skills	• Content model guidelines • Content type authoring guidelines • Map authoring guidelines • Style guidelines • Coding guidelines • Specialization policies • Quality initiatives (structural consistency, task analysis...)	• Specialization-aware or specific editors • Differentiated editing environments for different authoring needs • Component content management system • Extensible toolkit • Customized processing
4 Automation and integration	Brand	• Metadata • Translation and language attributes • Generalization for cross-specialization reuse	• Single-sourcing machine-readable and human-readable (like literate programming) • Reuse across specializations for different disciplines/ subjects • Reuse of trusted source content (for example product name lists) • Componentized reuse with metadata-driven integration	• Cross-product and cross-discipline coordination/ councils • Awareness of related discipline's content • Content strategy skills • Globalization skills	• Cross-discipline content strategy • Topic-based translation process • Cross-discipline reuse planning and coordination • Metadata management/ controlled metadata values • Automated review and approval workflows • Globalization policies • Requirements management	• Diversified editor solutions • Translation services • Enterprise content management system with automated workflow • Automated metadata-driven builds • Requirements traceability

Level	Scope	Key DITA Features	Reuse Strategy	People	Process	Tools
5 Semantics on demand	Enterprise	• DITA feeds • DITA wrappers for non-DITA content • Generated DITA for semantic interchange • Specialized elements for data integration • Dynamic content	• Reuse on-demand/dynamic reuse • Personalization on-demand/ dynamic personalization • Semantic content as a service (subscribable DITA feeds for approved content) • Reuse/integration of data sources into document contexts • Aggregation and validation of DITA and non-DITA sources using map wrappers and generated DITA • Reuse across multiple repositories	• Web development skills • Taxonomy development skills • Developer coordination across products/ disciplines	• Collaboration strategy • Mashup strategy • Semantic Web strategy • Taxonomy/ ontology management • Application development strategy (DITA as a common currency) • Portable content strategy (use appropriate repository for activities of content in current phase)	• Web-based embedded DITA editing/wikis • DITA generation/ • wrapping • Multiple integrated repositories • Dynamic content personalization server • DITA servers and feeds • DITA mashups • Standards-based loosely coupled application integration using DITA as a common currency
6 Universal semantic ecosystem	Universe	• Drive evolution of new DITA capabilities	• Cross-silo and cross-application knowledge management • Continual improvement • Sharing best practices	• Standards definition and consortium formation	• Continuous improvement	• Situational applications/ mashups • Universal content management framework

The opinions, solutions, and advice in this article are from the experiences of Michael Priestley and Amber Swope and are not intended to represent official communication from IBM or JustSystems, or an endorsement of any products. Michael Priestley, Amber Swope, IBM or JustSystems are not liable for any of the contents in this article. The accuracy of the information in this article is based on Michael Priestley's and Amber Swope's knowledge at the time of writing.

JUSTSYSTEMS.
http://www.justsystems.com

North America:
+1.866.793.1542
contactsales-na@justsystems.com

Europe:
+44.0.1462.889.082
contactsales-emea@justsystems.com

International: +1.604.697.8705

About JustSystems

JustSystems is a leading global software provider with three decades of successful innovation in office productivity, information management, and consumer and enterprise software. With over 2,500 customers worldwide, the company is continuing a global expansion strategy based on its xfy enterprise software, XMetaL content lifecycle solutions, and its pioneering work in enabling XBRL financial reporting technologies. JustSystems is one of the 2008 KMWorld 100 Companies that Matter in Knowledge Management, a 2008 EContent 100 member, and was recognized on the 2008 KMWorld Trend-Setting Product list for XMetaL. Major strategic partnerships include IBM, Oracle and EMC.