DITA: quattro lettere da conoscere

Introduzione

DITA è una delle più importanti innovazioni nella pubblicazione XML degli ultimi tempi. Se già utilizzate o pensate di adottare XML per la pubblicazione di documentazione tecnica, presto o tardi troverete il termine DITA.

DITA è l'acronimo di "Darwin Information Typing Architecture" ed è stata inventata da IBM come contributo alla comunità sotto gli auspici di OASIS (Organization for the Advancement of Structured Information Standards), http://www.oasis-open.org. All'indirizzo http://xml.coverpages.org/DITA-OASIS-CFP.html sono reperibili ulteriori informazioni sul comitato tecnico formato da OASIS per DITA.

Il presente documento non è da intendersi come guida pratica per implementatori né come trattato tecnico. IBM ha pubblicato molte informazioni tecniche eccellenti e approfondite riguardo DITA, reperibili all'indirizzo http://www-106.ibm.com/developerworks/xml/library/x-dita1/.

L'autore ringrazia il personale e le risorse di IBM per ciò che ha appreso di DITA, ma si ritiene il solo responsabile per eventuali errori, omissioni e interpretazioni erronee.

Sommario

Introduzione 1
L'importanza di DITA 3
Da necessità a invenzione: le origini di DITA 3
DITA alla riscossa 4
La specializzazione di DITA: soddisfare le esigenze 5
Riutilizzo: mito o realtà? 5
Progettazione del modello ideale di dati 5
Il funzionamento della specializzazione 6
Creazione semplificata, migliore comprensione 7
Elementi di Topic 7
Classificazione delle informazioni 7
Creazione di migliori informazioni 8
Applicazione di DITA a software, UI e formattazione 9
Conoscenza approfondita del proprio dominio 9
Analisi iniziale 9
Elementi software 10
Elementi di programmazione e UI 10
Utilizzo di DITA 10
Fasi successive alla creazione dei contenuti DITA 10
Visione globale: il supporto di PTC per DITA 11
Riepilogo 12

L'importanza di DITA

DITA, come XML, incorpora alcuni semplici concetti che hanno profonde implicazioni. La definizione di DITA è relativamente semplice; ciò che risulta difficile è descriverne le implicazioni. Cominceremo quindi dalla definizione.

DITA è un'architettura basata su XML per la pubblicazione di informazioni tecniche. Sotto diversi aspetti è come DocBook, creato alcuni anni or sono come base per la pubblicazione basata su XML di manuali tecnici, in particolare per la documentazione di hardware e software in campo informatico. Durante le ricerche intraprese per creare questo white paper, è stato trovato un interessante insieme di risorse all'indirizzo http://www.namahn.com/resources/notes.htm. Tra i vari white paper, è disponibile un documento relativo a DITA, più tecnico rispetto a questo documento anche se un poco datato, ma che include un ottimo confronto tra DITA e DocBook.

Due sono gli aspetti che rendono DITA speciale:

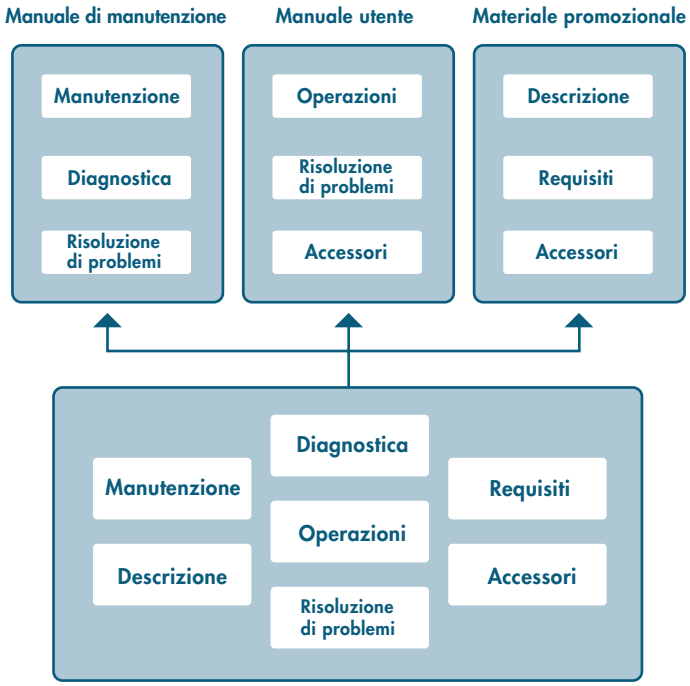
- Modularità: DITA definisce una DTD Topic che supporta un approccio modulare alla creazione di informazioni. Un argomento (topic) è un componente di informazioni e non un documento completo. Un argomento copre un aspetto di una specifica area di interesse. Ad esempio, questo documento può essere suddiviso in più argomenti: introduzione, panoramica, origini e così via.

  Inoltre DITA definisce un meccanismo per associare gli argomenti in documenti. Questo meccanismo, definito "mappa", consente di definire una gerarchia per gli argomenti. Ad esempio, la mappa di un libro produce una gerarchia che consiste di capitoli, sezioni e sottosezioni.

- Adattabilità: una DTD Topic è simile all'HTML in quanto specifica un insieme di elementi generici che posseggono una formattazione diversa quale titoli, paragrafi ed elenchi. Per adattare una DTD Topic alle specifiche necessità, DITA definisce un meccanismo chiamato "specializzazione" che consente di definire nuovi tag che ereditano comportamento e proprietà dai tag della DTD Topic.

  La specializzazione consente alle applicazioni a valle in grado di rilevare DITA, di gestire i tag sconosciuti trattandoli come il tag da cui ereditano le proprietà. Ad esempio, si può creare un tag "Procedura" che eredita da "Elenco ordinato" e un tag "Passaggio" che eredita da "Elemento elenco". Anche se si desidera aggiungere alla propria applicazione un'elaborazione specifica per Procedura e Passaggio, un'applicazione che rileva DITA e non ha alcuna informazione specifica relativa a questi tag, gestirà Procedura e Passaggio come se fossero Elenco ordinato ed Elemento elenco. Ad esempio, un'applicazione che rileva DITA e non riconosce il tag Passaggio, lo formatterà come se fosse Elemento elenco.

La conoscenza della definizione di DITA non fornisce informazioni sufficienti per comprenderne le implicazioni. Di seguito, l'argomento verrà quindi approfondito.



Con DITA è possibile creare moduli riutilizzabili di informazioni che possono essere associati e assemblati in documenti diversi a molteplici scopi.

Da necessità a invenzione: le origini di DITA

Vengono riassunti brevemente gli obiettivi principali di un sistema di pubblicazione XML:

- Riutilizzo: per eliminare le ridondanze, migliorare l'accuratezza e ridurre gli sforzi necessari all'aggiornamento delle informazioni, XML consente di riutilizzare e riciclare le informazioni in modo tale da creare una singola fonte di informazioni in cui una singola modifica nel materiale sorgente aggiorna automaticamente tutti i documenti in cui tali informazioni sono contenute.

- Condivisione: con XML è possibile costruire informazioni in modo tale da consentire ad altri gruppi, sia interni che esterni all'organizzazione, di incorporare automaticamente tali informazioni nei propri processi; ciò aggiunge un ulteriore valore alle informazioni create.

- Utilità: è possibile utilizzare XML per facilitare la creazione di informazioni in moduli che possono essere assemblati automaticamente secondo le necessità di ciascun individuo in modo tale che le informazioni necessarie, e solo quelle, siano disponibili.

- Automazione: per raggiungere questi obiettivi con costi contenuti, l'automazione è essenziale. XML rende possibile l'automazione consentendo di applicare la struttura assolutamente consistente richiesta per automatizzare i processi.

  La struttura assolutamente consistente richiesta dall'automazione è definita dal "modello di dati", la DTD o lo schema che prescrive i tag consentiti nei documenti e le modalità di utilizzo di tali tag.

Immaginate quindi di far parte di una grande azienda con una linea di prodotti estremamente diversificata e di essere responsabile della pubblicazione di ogni prodotto. Avete scelto XML per le possibilità offerte nell'ottenere eccezionali efficienze nella creazione, traduzione e pubblicazione consentendo, allo stesso tempo, di distribuire in modo più accurato e tempestivo le informazioni ai clienti.

Il cuore del sistema che avete progettato per pubblicare la documentazione tecnica di ogni gruppo dell'azienda è il modello di dati. L'approccio aziendale alla pubblicazione non solo ridurrà i costi di acquisto degli strumenti di pubblicazione e di sviluppo delle applicazioni di pubblicazione, ma consentirà anche all'azienda di presentarsi con un'immagine consistente ai clienti, associando i contenuti di vari prodotti nella documentazione che è personalizzata con precisione secondo i requisiti di ciascun cliente.

Iniziando a utilizzare il modello di dati tuttavia, ciascun gruppo trova dei problemi nel modello creato. I gruppi hanno necessità che non avete anticipato e richiedono delle modifiche. Ogni modifica non influisce solo sul modello di dati, ma anche sulle applicazioni a valle, tra cui le principali sono quelle di assemblaggio e pubblicazione, che dipendono dal modello di dati. Ciascuna modifica può influire sia su diversi fogli di stile di pubblicazione che sulla logica di elaborazione e richiede quindi aggiornamenti, testing, documentazione e ridistribuzione dell'applicazione.

Nel frattempo sempre più gruppi iniziano a utilizzare l'applicazione e a richiedere nuove modifiche e i gruppi esistenti trovano nuove opportunità che richiedono ulteriori modifiche. Comprendete infine che non solo le richieste di modifiche non cesseranno mai, ma che siete in ritardo e non riuscirete mai a recuperare.

DITA alla riscossa

Date le problematiche, decidete che è il momento di utilizzare un nuovo approccio. Dovete trovare un modo per soddisfare diversi requisiti in competizione:

- Soddisfare i diversi requisiti di molte aree dell'organizzazione
- Adattare velocemente e facilmente le modifiche imposte dalle necessità aziendali
- Supportare informazioni altamente modulari per ottimizzare il riutilizzo e la personalizzazione
- Mantenere il tutto abbastanza semplice in modo che gli autori possano diventare produttivi in breve tempo

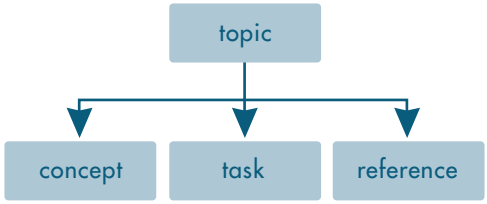
Iniziate quindi con la creazione di una DTD che contiene tutti i costrutti comuni di formattazione richiesti dalle pubblicazioni tecniche. E quale miglior punto di partenza se non l'HTML con la sua comprovata flessibilità in milioni di pagine Web? HTML può rappresentare titoli, paragrafi, elenchi puntati, elenchi numerati, parole in corsivo e molto altro ancora.

La nuova DTD non contiene tag specifici all'azienda, solo tag "generici" simili a quelli dell'HTML. Questi tag generici controllano l'aspetto delle informazioni, ma non rappresentano quasi niente della "semantica", il significato, delle informazioni. La nuova DTD inoltre non contiene una gerarchia di documento in quanto rappresenta solo un modulo di informazioni. In seguito, troverete un modo per associare i moduli in documenti completi per la distribuzione ai clienti.

Assegnate alla DTD un nome, "Topic", quindi progettate un meccanismo che consente di adattare Topic secondo le necessità. Ispirato dai lavori degli anni 90 sulle "forme architettoniche", create un metodo per definire nuovi tag ereditando le proprietà da tag esistenti. Qualsiasi applicazione che comprende la sintassi creata per definire nuovi tag sulla base dei tag esistenti, può elaborare i nuovi tag.

Dato che create documentazione tecnica, ponete attenzione ai tipi di informazioni specifiche richieste da tali documenti. Sulla base dei principi dell'architettura di informazioni, riguardo ai quali esiste un corpo consistente di ricerca, comprensione ed esperienza che definisce il modo più efficiente per comunicare informazioni al pubblico, definite tre specializzazioni di Topic che diventeranno i componenti di base dei documenti:

- Concept
- Task
- Reference



Concept, Task e Reference sono specializzazioni di Topic, dove ognuna rappresenta un modo diverso di fornire informazioni su un argomento. Anche se la maggior parte dei sistemi che rilevano DITA dovrebbero comprendere specificatamente come elaborare Concept, Task e Reference, qualsiasi sistema che rileva DITA può gestire qualsiasi specializzazione.

Queste specializzazioni sono non solo una grande idea, ma rappresentano un'altra importante innovazione di DITA: l'utilizzo di modelli di dati per guidare gli autori verso la creazione di informazioni che siano al contempo facili da scrivere e da comprendere.

La specializzazione di DITA: soddisfare le esigenze

La "specializzazione" è il meccanismo interno di DITA che consente di personalizzare le proprie necessità mantenendo al contempo l'interoperabilità e la possibilità di riutilizzo sia dei contenuti esistenti che del sistema esistente di pubblicazione. La specializzazione riduce inoltre i costi di manutenzione delle applicazioni di pubblicazione con il variare dei requisiti aziendali, in quanto, in molti casi, è possibile modificare i modelli di dati senza modificare le applicazioni a valle.

Questa sezione illustra l'impatto e l'importanza della specializzazione ma per comprenderla a fondo, è necessario comprendere prima i principi di riutilizzo delle informazioni, che rappresentano uno dei punti principali del sistema di pubblicazione XML.

Riutilizzo: mito o realtà?

Tra le molte promesse della pubblicazione XML, una delle più significative è la possibilità di riutilizzare le informazioni in diversi modi. Se si utilizza XML, è possibile inserire, senza modifiche, le stesse informazioni in diversi documenti e pubblicarle su diversi tipi di supporto, mantenendo al contempo una singola origine in modo che una sola modifica possa aggiornare tutti i documenti in cui è presente.

Questo non è un mito, molte organizzazioni hanno raggiunto questo tipo di riutilizzo. I problemi si verificano quando si tenta di riutilizzare le informazioni in diversi tipi di documenti, specialmente quando le informazioni provengono da origini diverse, sia interne che esterne all'organizzazione.

Di seguito viene descritto il problema; per soddisfare i propri requisiti, ciascun reparto crea un modello di dati diverso (DTD o schema) che contiene tag specifici ai propri scopi. Ad esempio, le informazioni di riparazione potrebbero contenere il tag <procedura> o <passaggio>, mentre un catalogo di parti potrebbe contenere i tag <numeroparte>, <descrizione> e <prezzo>.

Se si desidera includere le informazioni di riparazione nel catalogo di parti o includere informazioni sulle parti in un manuale di riparazione, è necessario modificare il modello di dati per consentire l'utilizzo dei nuovi tag. Oltre a ciò è anche necessario modificare tutte le applicazioni che elaborano le informazioni. Ad esempio, sarà necessario modificare i fogli di stile per adattare i nuovi tag in modo da pubblicare le informazioni che contengono i nuovi tag.

Questo problema di interoperabilità peggiora quando si cerca di riutilizzare le informazioni tra più tipi di documenti, ulteriori reparti e più aziende. A questo punto la promessa di riutilizzo diventa un mito, in quanto il costo di riconciliazione dei diversi modelli di dati supera i benefici derivanti dall'incorporazione degli stessi.

Se ciascun reparto si limitasse a un semplice e generico insieme di tag, quale HTML, allora la condivisione di informazioni sarebbe semplice. Ma l'utilizzo di un semplice e non intelligente modello annullerebbe molti dei benefici dell'automazione e della possibilità di elaborare i contenuti.

Che dilemma. Il valore delle informazioni risulta tanto maggiore quanto più un'organizzazione mette a punto il modello di dati per soddisfare i requisiti. Purtroppo maggiore è la specializzazione del modello di dati, maggiori saranno le difficoltà nel riutilizzare e condividere tali informazioni.

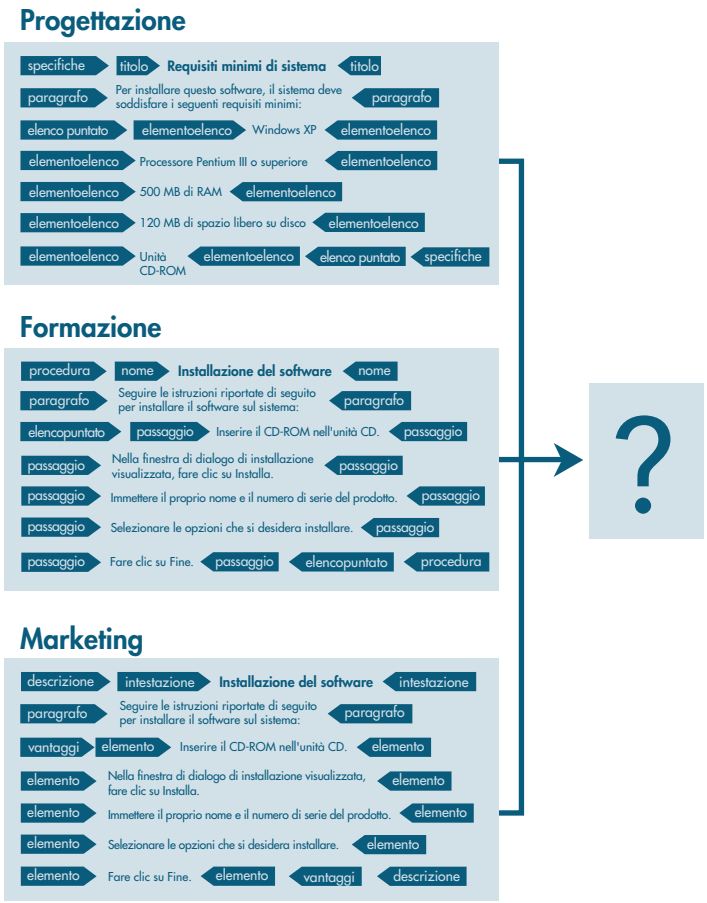
Nonostante questo dilemma, molte organizzazioni hanno ottenuto significativi benefici dal riutilizzo e dall'automazione di contenuti tra gruppi di lavoro o reparti. Immaginate i benefici che si potrebbero trarre se queste funzionalità potessero essere estese tra reparti e anche tra aziende.

Progettazione del modello ideale di dati

Il modello di dati ideale soddisferebbe i requisiti di chiunque nel mondo e sarebbe al contempo molto semplice da imparare. Come potrebbe essere tale modello di dati?

Per soddisfare i requisiti di chiunque, il modello di dati dovrebbe contenere centinaia di migliaia di tag con decine di migliaia di possibili combinazioni. Tuttavia per rimanere di semplice comprensione, dovrebbe essere limitato a meno di un centinaio di tag.

Quindi questa soluzione non è fattibile.

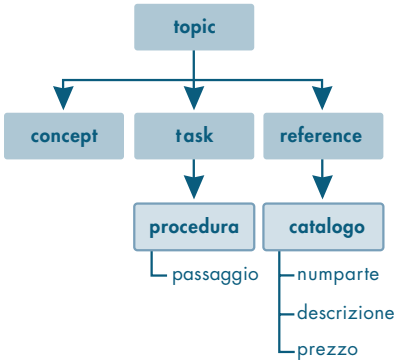


Ciascun reparto ha requisiti unici per i contenuti creati. Tuttavia, le informazioni devono essere riutilizzabili all'interno dell'azienda. Come è possibile creare un modello di dati che supporti i requisiti di un'intera organizzazione e allo stesso tempo rimanga di facile utilizzo per i singoli autori?

I progettisti di DITA hanno optato per una ingegnosa alternativa. Invece di creare un modello di dati che tenti di soddisfare i requisiti di chiunque, hanno creato un modello di dati che è adattabile ai requisiti di chiunque. Questa è l'ispirazione alla base della "specializzazione", una funzionalità di DITA che consente di modificare il modello di dati DITA per soddisfare i propri requisiti abilitando al contempo le applicazioni che rilevano DITA a integrare i propri contenuti anche se non hanno nessuna conoscenza dei tag utilizzati.

Quando si utilizza DITA, si parte dall'insieme di tag di base che DITA specifica. Questo insieme di tag di base assomiglia all'HTML, con la sua comprovata flessibilità in milioni di pagine Web e diverse generazioni di migliorie, e fornisce la maggior parte dei comuni tag strutturali e di formattazione desiderabili.

Per modificare DITA in modo che soddisfi i propri requisiti, è necessario aggiungere i propri tag. Un'operazione che se operata su qualsiasi altro modello di dati, inibirebbe immediatamente o eliminerebbe l'interoperabilità dei propri contenuti. Ma uno degli assi nella manica di DITA consiste nel fatto che quando si aggiunge un tag, oltre al nome, viene specificato anche a quale dei tag di base assomiglia di più. In altre parole, si crea un tag che eredita le proprietà e i comportamenti di un tag esistente. La "D" di DITA infatti sta a significare Darwin per il lavoro svolto sull'ereditarietà biologica.



È possibile creare specializzazioni da specializzazioni come illustrato in questo esempio, dove procedura è una specializzazione di task e catalogo è una specializzazione di reference. All'interno di ciascuna specializzazione, è possibile creare nuovi tag per soddisfare i requisiti specifici, dove ciascun tag è una specializzazione di un tag in topic o una specializzazione di topic.

Un esempio è la creazione di un tag <numeroparte>. Dato che i numeri di parte appaiono in linea, il tag è configurato in modo da ereditare dal tag <emph> ("emphasis", enfasi) che fa parte del modello di dati DITA di base. È sempre possibile configurare l'applicazione in modo che elabori i tag <numeroparte> in modo particolare. Ad esempio, è possibile configurare diverse formattazioni e personalizzare la convalida con un database.

Il beneficio è evidente in quanto un altro utente che non ha nessuna conoscenza in merito ai tag <numeroparte> può sempre integrare e riutilizzare i contenuti senza modificarne il funzionamento di base. Come? Dato che il proprio modello di dati contiene le informazioni di ereditarietà, se un'altra applicazione che rileva DITA non riesce a interpretare il tag <numeroparte>, lo interpreterà come <emph>.

Di seguito viene descritto un altro esempio in cui si desidera impostare una procedura che consiste in una serie di passaggi. In questo caso si può configurare il tag <procedura> in modo che erediti dal tag <ol> ("ordered list", che è un elenco numerato) e il tag <passaggio> in modo che erediti dal tag <li> ("list item", elemento elenco).

Il funzionamento della specializzazione

La specializzazione è conforme con gli standard XML e XSL; i documenti DITA non contengono informazioni che li rendono incompatibili con questi standard.

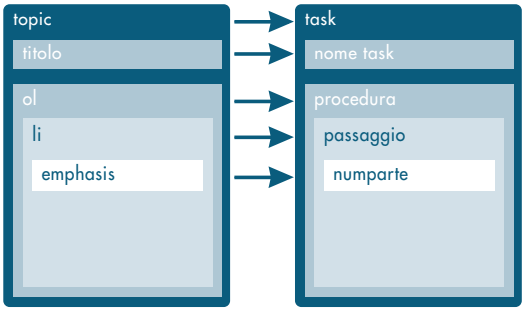
Per adattare DITA ai propri requisiti, è necessario iniziare con Topic, il modello di dati DITA di base da cui sono originate tutte le specializzazioni, e assegnarvi un nome appropriato. A titolo di esempio, si supponga di creare un modello di dati che verrà chiamato "Operazione". Per creare il modello di dati Operazione, si inizia con i tag in Topic, quindi si aggiungono i propri tag. Continuando con l'esempio precedente, saranno aggiunti i tag <procedura> e <passaggio> al proprio modello di dati.

Seguendo il modello DITA per definire un nuovo tag, si specifica il tag da cui il nuovo tag eredita le sue proprietà. Queste informazioni sono memorizzate in uno speciale attributo che i progettisti di DITA hanno nominato "class". L'attributo class contiene le informazioni di ereditarietà. In una DTD, la sintassi della definizione dell'attributo class utilizza la parola chiave ATTLIST. Nel seguente esempio sono definite le informazioni di ereditarietà del tag <passaggio>:

<!ATTLIST passaggio class CDATA "- topic/li operazione/passaggio">

La dichiarazione ATTLIST dichiara non solo l'attributo "class" per il tag <passaggio>, ma specifica anche il valore di default per tale attributo. Per mantenere la conformità con la metodologia DITA, è necessario inibire la possibilità di modifica di questo attributo durante le modifiche effettuate dall'autore.

Nell'esempio precedente, l'attributo class per <passaggio> è impostato su "- topic/li operazione/passaggio" (le virgolette non fanno parte del valore dell'attributo). Per le applicazioni che rilevano DITA, "- topic/li operazione/passaggio" significa che se l'applicazione riconosce il modello di dati Operazione, gestirà il tag <passaggio> come tale. Ma se l'applicazione non riconosce il modello Operazione, gestirà il tag <passaggio> come <li> dal modello di dati Topic.



Componenti base di DITA **Specializzazione di Task**

Le specializzazioni sono basate sui componenti base forniti con DITA. Se un'applicazione o un foglio di stile non riconosce i tag nella specializzazione, li gestirà come se fossero tag DITA da cui sono stati originati.

Per pubblicare contenuti basati su DITA, è necessario configurare il foglio di stile in modo che corrisponda all'attributo class invece che al tag. Il foglio di stile per il modello di dati di base, Topic, può gestire qualsiasi tag specializzato analizzando semplicemente l'attributo class.

Creazione semplificata, migliore comprensione

Di seguito sono esaminate in maggior dettaglio le tre specializzazioni definite da IBM, concept, task e reference (concetto, attività e riferimento), e il loro impatto sulla creazione e sulla comprensione.

Elementi di Topic

Il componente base di DITA è "Topic" e consiste di un breve resoconto con uno scopo specifico.

Topic consiste di un insieme fondamentale di tag per organizzare e presentare informazioni. Alcuni dei tag inclusi in Topic sono:

Tag di organizzazione	
Nome tag	**Descrizione**
<title>	Intestazione o etichetta; può essere un titolo principale (sotto <topic>) o un titolo subordinato, ad esempio sotto <section>
<shortdesc>	Breve descrizione per l'anteprima di collegamenti e ricerche
<section>	Divisione di un argomento (topic); un argomento può avere più sezioni, ma una sezione non può contenere altre sezioni (section)
<body>	Contiene i contenuti principali dell'argomento
<p>	Paragrafo
	Elenco non ordinato (elenco puntato)
	Elenco ordinato (elenco numerato)
<table>	Tabella

Tag di presentazione	
Nome tag	**Descrizione**
<b>	Grassetto
<i>	Corsivo
<u>	Sottolineatura
<tt>	Telescrivente (spaziatura singola)
	Elenco non ordinato (elenco puntato)

Topic non contiene tag specifici di un particolare tipo di settore, applicazione o documento. Dato che DITA può essere adattato per soddisfare i propri requisiti, è necessario "specializzare" Topic aggiungendo i propri tag.

Quando si aggiunge un nuovo tag, è necessario specificare anche un tag esistente che sia simile al proprio tag creato (vedere la colonna "Eredita da" nella seguente tabella). In questo modo le altre applicazioni che rilevano DITA possono elaborare i contenuti, anche se non riconoscono i nuovi tag. In tal caso utilizzeranno i nuovi tag come se fossero i tag da cui hanno ereditato. Ad esempio, nella tabella seguente un'applicazione che rileva DITA e non riconosce il tag <prereq>, lo gestirà come <section>.

Classificazione delle informazioni

Specializzando Topic, si crea un nuovo "tipo di informazione", chiamato anche "infotype". Ad esempio, IBM ha creato una specializzazione di Topic chiamata Task (attività) che contiene i passaggi da eseguire per completare una procedura. In altre parole, Task è un tipo di informazione che contiene tag specializzati quali:

Tag di Task		
Nome tag	**Descrizione**	**Eredita da**
<prereq>	Prerequisito: ciò che l'utente deve sapere o fare prima di iniziare l'attività	<section>
<steps>	Serie di passaggi che l'utente deve seguire per completare l'attività	
<result>	Il risultato atteso dall'esecuzione dell'attività	<section>

DITA consente di creare i propri tipi di informazioni, che possono essere adattati da Topic o da un altro tipo di informazioni. A titolo di esempio, si supponga di creare un nuovo tipo di informazioni chiamato "Procedura di riparazione". È possibile partire da Topic e adattarlo alle proprie necessità, ma potrebbe essere più semplice utilizzare Task in quanto contiene già alcuni dei tag necessari.

Per creare un documento, si assemblano uno o più Topic o altri tipi di informazioni. I documenti possono essere composti da qualsiasi combinazione di tipi di informazioni. Ad esempio, è possibile utilizzare un singolo documento che contiene diversi Topic e Task oppure i propri tipi di informazioni.

Creazione di migliori informazioni

Sulla base dei principi dell'architettura di informazioni, riguardo ai quali esiste un corpo consistente di ricerca, comprensione ed esperienza, IBM ha definito tre tipi di informazioni specifici alla documentazione tecnica, ossia Concept, Task e Reference.

Esistono anche altre architetture di informazioni per la documentazione tecnica. Un esempio è rappresentato da Information Mapping, una società che ha sviluppato una metodologia e ha pubblicato una ricerca che dimostra miglioramenti sorprendenti sia nella scrittura della documentazione tecnica, misurata in numero di parole e in tempi di completamento, che nella lettura della documentazione tecnica, misurata in tempo e comprensione. Per ulteriori informazioni su Information Mapping, visitate il sito Web all'indirizzo http://www.informationmapping.com/. Per consultare le ricerche pubblicate sul miglioramento della documentazione tecnica, visitate il sito Web all'indirizzo http://www.informationmapping.com/documents/IMI%20history%20and%20results.pdf.

Indipendentemente dall'approccio specifico, ciascuna metodologia condivide caratteristiche comuni.

- Prescrive il tipo di informazioni (ad esempio: concept, task, reference) che gli autori possono creare
- Consente la modularizzazione delle informazioni in componenti piccoli e riutilizzabili
- Focalizza l'autore sulla creazione di un componente alla volta
- Automatizza la pubblicazione per liberare gli autori dal peso della progettazione e della formattazione
- Predispone rapidamente i lettori a ricevere informazioni classificate in tipi specifici e presentate in modo consistente
- Consente un facile riutilizzo dei componenti di informazioni in più documenti e una semplice condivisione tra diverse organizzazioni

Con solo tre tipi di informazioni di base per la documentazione tecnica, DITA mantiene le cose semplici. È possibile specializzare uno qualsiasi di questi tipi per settori o documenti specifici, ma in ogni singolo manuale devono essere presenti solo tre tipi differenti di informazioni.

Di seguito viene presentata una descrizione di ciascun tipo di informazioni DITA.

- Concept: fornisce una panoramica di Task o Reference. Il testo in Concept tende a essere breve e semplice, in parte perché l'autore può aggiungere solo altre sezioni o esempi dopo una sezione o un esempio.
- Task: contiene i passaggi che consentono al lettore di eseguire la procedura. Task contiene molti tag specializzati che supportano un'enorme varietà di modi per presentare le istruzioni e le informazioni che stanno a monte.
- Reference: descrive le caratteristiche di un insieme di elementi, quali un elenco di prodotti, definizioni o comandi di programmazione.

Mentre ciascuno di questi tipi di informazioni presenta alcune speciali caratteristiche, essi condividono molti tag. Oltre ai tag relativi a organizzazione e presentazione, illustrati precedentemente, sono disponibili altri tag per i collegamenti (riferimenti incrociati e collegamenti correlati), ricerca, navigazione, parole chiave e così via.

Ciò che è interessante è che DITA può portare a calcolare alcuni criteri di misurazione dai propri documenti, quali:

- Numero medio di parole per componente, che consente di identificare componenti che possono avere un numero di parole eccessivo o insufficiente (potreste decidere che diversi tipi di informazioni devono rispettare determinate dimensioni)
- Proporzione dei tipi di informazioni presente in un manuale, che consente, ad esempio, di identificare le istanze in cui i manuali enfatizzano oltre misura i materiali di riferimento o non enfatizzano abbastanza i concetti

In breve, applicando la metodologia DITA alla documentazione tecnica, è disponibile una struttura pronta all'uso per i propri manuali. È possibile adattare rapidamente ai propri requisiti i tre tipi di informazioni DITA, concept, task e reference. Gli autori devono conoscere a fondo un semplice insieme di tag per diventare esperti nella creazione in XML, e saranno maggiormente produttivi non solo perché sono liberati dalla responsabilità di formattare ma anche perché lavorano con modelli ben definiti e provati sul campo. E, ancora più importante, i lettori troveranno le informazioni necessarie più rapidamente dato che, per arrivarvi, dovranno leggere una quantità di testo inferiore con una struttura organizzativa non arbitraria.

Applicazione di DITA a software, UI e formattazione

I Domain Element (elementi di dominio) sono tag specializzati che IBM ha definito per argomenti specifici. Non forniscono solo un punto di inizio ben concepito per la specializzazione di soggetti quali programmazione, software, interfaccia utente e formattazione, ma dimostrano anche come procedere nell'implementazione in qualsiasi applicazione richiesta del supporto DITA per la specializzazione.

Conoscenza approfondita del proprio dominio

DITA crea un eccellente punto di inizio per un approccio orientato all'argomento per creare informazioni, ma la chiave per utilizzare con successo la metodologia DITA per un proprio scopo consiste nel specializzarla aggiungendo tag per soddisfare i propri requisiti.

Ad esempio, per pubblicare manuali di manutenzione, potrebbe essere necessario creare un tag chiamato "procedura" che contiene un elenco di operazioni sequenziali. In un'applicazione DITA, il tag <procedura> è creato come specializzazione di un tag esistente. Dato che le procedure assomigliano molto agli elenchi numerati, <procedura> sarà una specializzazione di <ol> (ordered list), generalmente presentato come elenco numerato.

Mentre il tag <procedura> può essere utilizzato per diversi tipi di manuali di manutenzione, si potrebbero creare ulteriori tag specifici a una particolare organizzazione o settore. Nel caso di una azienda di software, ad esempio, potrebbe essere necessario documentare l'interfaccia di programmazione dell'applicazione (API) e potrebbero essere necessari diversi tag specializzati a questo scopo.

I tag specializzati come questi discussi, appartengono a un "dominio", che non è altro che una parola ricercata per descrivere un argomento. Nella progettazione, IBM ha definito diversi insiemi di tag specializzati per trattare diversi argomenti specifici, e sono stati chiamati "Domain Element". Questi argomenti includono programmazione, software, interfaccia utente e formattazione. Le sezioni successive illustrano lo scopo di ciascun insieme di Domain Element.

L'utilizzazione dei Domain Element esistenti consente di ridurre i tempi di sviluppo della propria applicazione, incrementando al contempo le probabilità di condividere i contenuti con altri nello stesso dominio. Dopotutto, che senso avrebbe creare un tag <elencoparametri> quando altri colleghi utilizzando il tag <parml> che ha lo stesso significato?

Analisi iniziale

Sono quindi esaminati in prima istanza i Domain Element per la formattazione. Questi sono chiamati "elementi tipografici" ed esistono per applicazioni in cui si desidera lasciare il controllo della formattazione del testo all'autore.

È tuttavia necessario procedere con cautela. La presenza di tag che controllano solamente l'aspetto di un elemento non è in violazione della separazione dei contenuti dalla formattazione che XML incoraggia? Effettivamente ciò è corretto. Per cui, nel caso in cui non si voglia violare tale separazione, è possibile specializzare ulteriormente gli elementi tipografici e assegnare loro un significato maggiore. Ad esempio, è possibile specializzare il tag <i>, che specifica il testo in corsivo, creando tag quali <titololibro>, <enfasi>, <parolastraniera> e così via, fino a quando non sono state specificate tutte le istanze in cui il testo può essere in corsivo. Il valore associato alla differenziazione di questi diversi significati di corsivo dipende dai propri requisiti e non dalla "purezza" dell'XML.

I nuovi tag, che possono essere chiamati tag "semantici" dato che riguardano il significato del testo e non il loro aspetto, forniscono la flessibilità necessaria per adattare in seguito la loro formattazione senza dover reinserire tag. Questa è certamente una delle virtù dell'XML.

Ma ci sono anche situazioni in cui il valore associato allo sforzo richiesto per creare tag speciali per comunicare il significato del testo non è giustificabile. In tal caso gli elementi tipografici sono sicuramente un'opzione utilizzabile. Perciò, anche se questa opzione non rientra nel puro stile XML, è sicuramente possibile rendere disponibili gli elementi tipografici ai propri autori.

Di seguito viene presentato l'elenco completo.

Elementi tipografici	
Nome tag	**Descrizione**
<b>	Grassetto
<i>	Corsivo
<u>	Sottolineato
<tt>	"Teletype": seleziona un tipo di carattere monospazio, quale Courier, in cui ogni carattere è della stessa larghezza. Ciò è utile, quando si desidera che le lettere siano allineate verticalmente.
<sup>	Soprascritto: rende il testo più piccolo e lo pone in posizione elevata rispetto al testo circostante. Utile per note a piè di pagina ed espressioni matematiche quali $2^2 = 4$.
<sub>	Sottoscritto: rende il testo più piccolo e lo pone in posizione inferiore rispetto al testo circostante. Utile per formule chimiche quali H_2O.

Elementi software

Per tentare di descrivere il funzionamento di un programma software, i seguenti Domain Element sono estremamente utili:

Elementi software	
Nome tag	**Descrizione**
<msgph>	Message Phrase (frase messaggio): contiene il messaggio visualizzato dal programma
<msgblock>	Message Block (blocco messaggio): contiene messaggi su più righe o più messaggi
<msgnum>	Message Number (numero messaggio): se disponibile
<cmdname>	Command Name (nome comando)
<varname>	Variable (variabile): contiene una variabile immessa in un programma
<filepath>	File Path (percorso file): contiene il nome e, facoltativamente, la posizione di un file
<userinput>	User Input (input utente): contiene i valori immessi dall'utente nel programma
<systemoutput>	System Output (output sistema): contiene qualsiasi forma di output di un programma (<msgph> è più specifico di <systemoutput>)

Per utilizzare questi elementi ci sono delle regole. Ad esempio, nella definizione di questi Domain Element, il tag varname può essere contenuto nei tag msgph, msgblock, filepath, userinput e systemoutput, ma non può essere contenuto nel tag filepath.

Queste regole di struttura sono importanti tanto quanto, e forse di più, dei nomi dei tag. La conversione di un tag in un altro e viceversa è relativamente semplice, ma è molto più difficile convertire una struttura in un'altra e viceversa.

Elementi di programmazione e UI

A completare i Domain Element sono disponibili gli elementi per interfaccia utente e programmazione.

Gli elementi per l'interfaccia utente vengono utilizzati per descrivere in dettaglio i valori immessi dall'utente in un programma. Questi elementi includono menu, tasti di scelta rapida, schermo del computer e controlli dell'interfaccia utente quali pulsanti e voci di menu.

A differenza degli altri domini dove sono disponibili solo pochi elementi, il dominio di programmazione dispone di 25 elementi dato che sono presenti molte e diverse parti per documentare come scrivere codice in un linguaggio di programmazione. Con elementi per voci quali parole chiave, codice, variabili e parametri, è possibile descrivere nei minimi dettagli un linguaggio di programmazione.

Utilizzo di DITA

I contenuti basati su DITA consistono, per progettazione, in pezzi di informazioni, non libri. Come è possibile convertire questi pezzi in un insieme coerente di informazioni, quali file di guide in linea o una pubblicazione cartacea? La risposta è da ricercare nelle mappe.

E infine, dove è possibile trovare strumenti che supportano la creazione di applicazioni DITA, la creazione di contenuti DITA, l'assemblaggio dei contenuti in collezioni coerenti e la pubblicazione di tali contenuti su supporti diversi? La risposta è da ricercare in PTC.

Fasi successive alla creazione dei contenuti DITA

Una delle maggiori virtù dei contenuti orientati all'argomento promossi da DITA, consiste nel fatto che gli autori possono scrivere in pezzi relativamente piccoli. Invece di dover affrontare un incarico per scrivere 1000 pagine, gli autori lavorano su componenti di informazioni che consistono di poche pagine.

Per assemblare questi pezzi in pubblicazioni, DITA definisce una mappa che consente di specificare quali pezzi includere, la loro sequenza e relazione gerarchica.

Si presupponga, ad esempio, di aver scritto i seguenti pezzi:

- Topic: Carburatori
- Concept: Funzionamento dei carburatori
- Task: Regolazione del carburatore
- Task: Sostituzione del carburatore
- Reference: Relazione tra carburatori e altri componenti del motore

Per assemblare questi pezzi in una pubblicazione, è necessario elencarli in una mappa. La mappa consente non solo di elencare quale pezzo includere ma anche di specificare le relazioni gerarchiche.

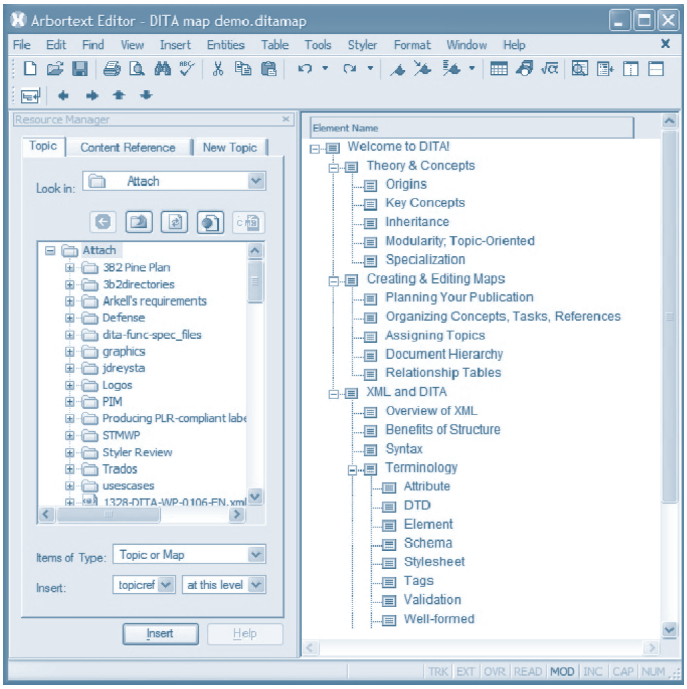
In questo caso, ad esempio, gli ultimi quattro pezzi sono subordinati al primo, come mostrato dalla seguente gerarchia:

- Topic: Carburatori
  - Concept: Funzionamento dei carburatori
  - Task: Regolazione del carburatore
  - Task: Sostituzione del carburatore
  - Reference: Relazione tra carburatori e altri componenti del motore

L'elemento primario in una mappa è <topicref>, che agisce come meccanismo di inclusione (in modo simile alle entità di file o XInclude) per specificare un pezzo da includere nel documento corrente.

La seguente immagine illustra la mappa per la pubblicazione relativa ai carburatori. Per stabilire la gerarchia, è possibile "nidificare" l'elemento <topicref> su sé stesso. Ciò significa che il <topicref> interno è subordinato al primo.

Nel seguente esempio, il <topicref> per Carburatore contiene gli elementi <topicref> dei rimanenti pezzi e ciò rende questi pezzi interni subordinati al pezzo esterno.



Incluso in Arbortext 5.3, Arbortext Editor fornisce diversi strumenti sia per semplificare la creazione di topic (o specializzazioni di topic quali Concept, Task e Reference) che per creare "mappe" che elencano i topic da includere in un insieme di informazioni, quali un libro o un file di guida in linea, e che indicano come navigare tra questi topic.

Visione globale: il supporto di PTC per DITA

Anche se le intenzioni originali per DITA erano di mantenere la conformità con standard esistenti, sono state comunque introdotte delle funzionalità che richiedono elaborazioni addizionali per supportarle. In previsione di vedere DITA acquistare un ruolo importante in un'ampia gamma di applicazioni, PTC ha deciso di investire in modo significativo nell'aggiunta di funzionalità specifiche a DITA.

PTC ha aggiunto alla linea di prodotti Arbortext diverse funzionalità che forniscono o migliorano il supporto per DITA.

- Creazione: sono state aggiunte diverse funzionalità a Arbortext Editor per distribuire un ambiente semplice da apprendere e altamente produttivo per creare e modificare topic e mappe. Le funzionalità offerte includono: aggiunta di componenti tramite trascinamento della selezione a un topic o per aggiungere topic a una mappa, controllo ortografico e ricerca/sostituzione in tutti i topic di una mappa, nuovo Gestione risorse che rende disponibili i componenti riutilizzabili e la nuova Visualizzazione colonna che visualizza la mappa come un sommario.
- Assemblaggio e pubblicazione: Arbortext Publishing Engine utilizza la mappa DITA per assemblare i documenti, quindi li pubblica su diversi tipi di supporto, sia cartacei che elettronici.
- Specializzazione: la linea di prodotti Arbortext supporta le funzionalità di specializzazione di DITA consentendo di utilizzare un singolo foglio di stile per controllare lo stile di Topic, Task, Concept, Reference e altre specializzazioni. Anche se i fogli di stile XSL forniscono già un meccanismo di supporto per le specializzazioni, era necessario aggiungere il supporto per le specializzazioni al formato del foglio di stile per la modifica.
- Tabelle semantiche: la DTD DITA contiene alcuni elementi che dovrebbero essere in forma di tabella invece che narrativa, per cui i prodotti Arbortext consentono di selezionare i tag che sono visualizzati e stampati come una tabella.

  Ad esempio, potrebbe essere desiderabile creare i tag <numeroparte>, <descrizione> e <prezzo> per un catalogo e visualizzarli come una tabella. Senza il supporto per le tabelle semantiche, questi tag sarebbero visualizzati come contenuti narrativi.

numparte 123-456 numparte

descrizione Carburatore descrizione

prezzo 45,99 € prezzo

In alcuni casi le informazioni sono più chiare se sono visualizzate in tabelle (come mostrato di seguito) invece che in elenchi o frasi.

Per semplificare le modifiche è possibile configurare Arbortext Editor in modo da visualizzare i tag come celle di tabelle come nel seguente esempio:

Numero parte	Descrizione	Prezzo
123-456	Carburatore	45,99 €

- Supporto per conref: DITA utilizza “conref” per includere un componente di informazioni in un documento. Questo è un metodo sostitutivo per il meccanismo entità di file e XInclude e richiede funzionalità specifiche che PTC ha aggiunto nella linea di prodotti Arbortext.
- Supporto per topicref: DITA utilizza “topicref” per includere un topic (o specializzazioni di topic quali Concept, Task e Reference) in una mappa. Questo è un metodo sostitutivo per il meccanismo entità di file e XInclude e richiede funzionalità specifiche che PTC ha aggiunto nella linea di prodotti Arbortext.
- Doctype: nel software Arbortext è incluso un insieme di doctype per Topic, Concept, Task e Reference. Un doctype include la DTD, un foglio di stile e altre personalizzazioni che sono appropriate per tale doctype.

Riepilogo

Uno dei maggiori benefici di un sistema di pubblicazione basato su XML è la possibilità di riutilizzare e condividere informazioni nell’azienda. Tuttavia, in molte organizzazioni, ciascun reparto utilizza le proprie regole e requisiti. La creazione di modelli di dati che siano semplici da utilizzare per gli autori e che supportino i vari requisiti di ciascun reparto è un compito molto impegnativo. La gestione di tali modelli di dati con l’evoluzione e i cambiamenti, sia aziendali che degli autori, è quasi impossibile.

DITA affronta queste problematiche di interoperabilità fornendo una metodologia nella quale è possibile creare modelli di dati univoci per soddisfare i requisiti individuali di ciascun reparto dell’azienda. Ciò limita la dimensione di ogni modello di dati semplificandone l’utilizzo da parte degli autori. Tuttavia, dato che tutti i modelli di dati sono basati su un singolo modello di dati globale, gli autori possono sempre condividere e riutilizzare le informazioni all’interno dell’organizzazione. Questo perché i fogli di stile e le applicazioni possono elaborare i tag non riconosciuti in modelli di dati univoci sulla base di quelli globali.

Gli autori sono maggiormente produttivi in quanto possono riutilizzare le informazioni esistenti di qualsiasi reparto aziendale. Gli aggiornamenti al materiale di origine aggiornano automaticamente qualsiasi istanza riutilizzata. Di conseguenza, gli autori spendono meno tempo a ricreare informazioni esistenti e ad aggiornare informazioni presenti in più istanze. Grazie al riutilizzo, gli aggiornamenti sono più veloci, semplici e globali e, di conseguenza, le informazioni sono consistenti e più aggiornate. Infine, la certezza che i fogli di stile e le applicazioni XML siano in grado di gestire i tag sconosciuti in base ai tag DITA globali da cui sono stati derivati, assicura l’automazione della pubblicazione senza gli errori e gli aggiornamenti che potrebbero rallentare il processo.

In conclusione, DITA può aiutare le organizzazioni a realizzare le promesse di XML: autori maggiormente produttivi, migliore qualità delle informazioni e processi di pubblicazione che non sono solo più veloci ma anche meno costosi.

Copyright © 2006, Parametric Technology Corporation (PTC). Tutti i diritti riservati. Le informazioni contenute nel presente documento sono esclusivamente per scopi informativi, sono soggette a modifiche senza preavviso e non devono essere interpretate come garanzia, impegno, condizione o offerta da parte di PTC. PTC, il logo PTC, The Product Development Company, Arbortext, il logo Arbortext, Pro/ENGINEER, Windchill e tutti i nomi di prodotti e i logo di PTC sono marchi o marchi registrati di PTC e/o delle sue consociate negli Stati Uniti e in altri paesi. Tutti gli altri nomi di prodotti o di aziende appartengono ai rispettivi proprietari.

2016_DITA_WP_0901_IT