

DITA – ein Begriff, den Sie kennen sollten

Einführung

DITA ist im Bereich der XML-Veröffentlichung eine der wichtigsten Neuerungen der jüngsten Vergangenheit. Wenn Sie XML bereits für die Veröffentlichung technischer Dokumentation nutzen oder den Einsatz planen, werden Sie früher oder später auf DITA aufmerksam werden.

DITA steht für „Darwin Information Typing Architecture“ und wurde von IBM im Rahmen des OASIS-Projekts (<http://www.oasis-open.org>) der Organization for the Advancement of Structured Information Standards (OASIS) beigesteuert. (Weitere Informationen über das technische Komitee für DITA, das von OASIS ins Leben gerufen wurde, finden Sie unter <http://xml.coverpages.org/DITA-OASIS-CFP.html>.)

Dieses Dokument soll keine Anleitung für die Implementierung sein und geht auch nicht allzu sehr auf die technischen Hintergründe ein. IBM hat unter <http://www-106.ibm.com/developerworks/xml/library/x-dita1/> einige hervorragende und detaillierte technische Informationen über DITA bereitgestellt.

Der Autor möchte sich bei den Mitarbeitern und Ressourcen von IBM für die Erläuterungen zu DITA bedanken und übernimmt hiermit für etwaige Fehler, Auslassungen oder falsche Einschätzungen die alleinige Verantwortung.

Inhaltsverzeichnis

Einführung	1
Warum DITA für Sie wichtig ist	3
Von der Notwendigkeit zur Entwicklung: die Ursprünge von DITA	3
DITA ist da	4
DITA-Spezialisierung – Anpassung an Ihre Anforderungen	5
Wiederverwendbarkeit – Mythos oder Wirklichkeit? ..	5
Entwurf des idealen Datenmodells	5
Die Funktionsweise der Spezialisierung	6
Einfachere Erstellung, bessere Verständlichkeit	7
Der Inhalt eines Topics (Themas)	7
Die Kategorisierung von Informationen	7
Erstellen besserer Informationen	8
DITA für Software, Benutzeroberflächen und Formatierung	9
Die Aufgabe in den Griff bekommen	9
Klein anfangen	9
Software-Elemente	10
Benutzeroberflächen- und Programmierelemente ...	10
Die Umsetzung von DITA	10
Der DITA-Inhalt ist bereit! Wie geht es nun weiter? ..	10
Alles unter einem Hut – DITA-Unterstützung durch PTC	11
Zusammenfassung	12

Warum DITA für Sie wichtig ist

Wie auch XML selbst umfasst DITA einige einfache Konstrukte, die tief greifende Folgen haben. Die Definition von DITA ist relativ einfach. Schwierig ist vor allem die Beschreibung ihrer Auswirkungen. Beginnen wir also mit der Definition.

DITA ist eine auf XML basierende Architektur für die Veröffentlichung technischer Informationen. DITA ähnelt in einigen Bereichen DocBook, das vor einigen Jahren als Grundlage der XML-basierten Veröffentlichung technischer Handbücher entwickelt wurde, insbesondere für Computer-Hardware- und Software-Dokumentation. (Bei den Recherchen für dieses White Paper fanden wir unter <http://www.namahn.com/resources/notes.htm> einige interessante Ressourcen. Neben einigen anderen lesenswerten White Papers findet sich dort auch ein (leider mittlerweile etwas veraltetes) White Paper über DITA, das die technischen Hintergründe besser beschreibt als das vorliegende und eine Gegenüberstellung von DITA und DocBook enthält.)

Es sind zwei Aspekte, die DITA besonders auszeichnen:

- **Modular** – In DITA ist eine Topic-DTD definiert, die eine modulare Vorgehensweise beim Erstellen von Informationen unterstützt. Ein Topic (Thema) ist eine Informationskomponente, kein vollständiges Dokument. In einem Topic wird ein Aspekt eines bestimmten Themenbereichs behandelt. (Beispielsweise könnte man in diesem White Paper mehrere Topics unterscheiden: Einführung, Übersicht, Ursprünge usw.).

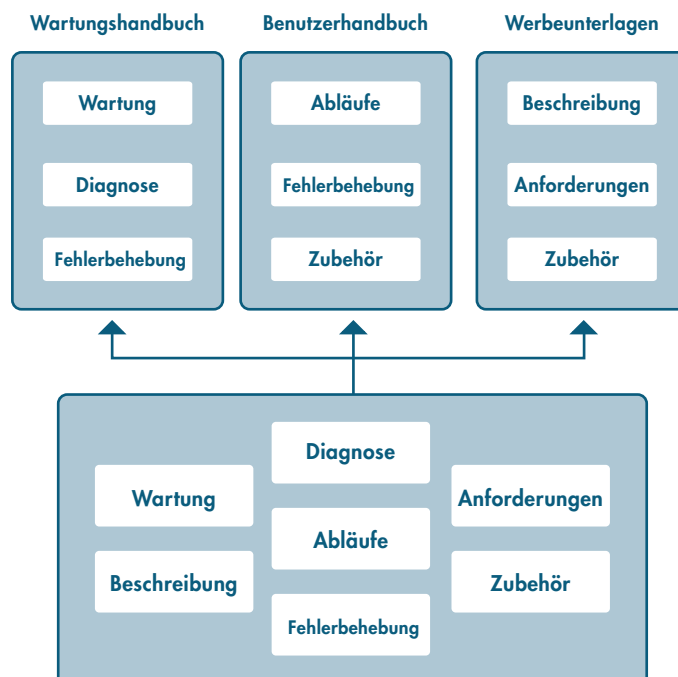
DITA definiert außerdem einen Mechanismus für das Zusammenfügen von Topics zu Dokumenten. Dieser als „Map“ bezeichnete Mechanismus ermöglicht die Definition einer Hierarchie für die Topics. Beispielsweise kann mit der Map für ein Buch eine Hierarchie geschaffen werden, die aus Kapiteln, Abschnitten und Unterabschnitten besteht.

- **Adaptierbar** – Die Topic-DTD ähnelt insofern HTML, als darin eine Reihe von generischen Elementen angegeben wird, die jeweils eine eigene Formatierung aufweisen, z.B. Titel, Absätze und Listen. Für die Anpassung der Topic-DTD an Ihre spezifischen Anforderungen definiert DITA einen Mechanismus, der „Spezialisierung“ genannt wird. Diese ermöglicht die Definition neuer Tags, die ihr Verhalten und ihre Eigenschaften von Tags in der Topic-DTD erben.

Dank der Spezialisierung können nachgeordnete Anwendungen, die DITA unterstützen, ein unbekanntes Tag verarbeiten, indem sie es behandeln wie das Tag, dessen Eigenschaften es erbt.

Beispielsweise könnten Sie ein Tag namens „Procedure“ erstellen, das von „Ordered List“ erbt, und ein Tag namens „Step“, das von „List Item“ erbt. Auch wenn Sie Ihrer Anwendung natürlich spezifische Verarbeitungsanweisungen für „Procedure“ und „Step“ hinzufügen würden – eine Anwendung, die DITA unterstützt und der diese Tags nicht bekannt sind, würde „Procedure“ und „Step“ behandeln, als wären sie „Ordered List“ und „List Item“. Folglich würde eine Veröffentlichungsanwendung, die DITA unterstützt, ohne weitere Informationen über „Step“ dieses Tag genauso formatieren wie „List Item“.

Allein das Wissen über die Definition von DITA gibt Ihnen nicht genügend Hintergrundwissen, um die Auswirkungen zu verstehen. Kommen wir also nun zum Kern der Sache.



DITA ermöglicht die Erstellung wiederverwendbarer Informationsmodule, die zu beliebigen Zwecken zu unterschiedlichen Dokumenten kombiniert und zusammengestellt werden können.

Von der Notwendigkeit zur Entwicklung: die Ursprünge von DITA

Lassen Sie uns zunächst die wichtigsten Ziele eines XML-Veröffentlichungssystems rekapitulieren:

- **Wiederverwendung** – Zur Vermeidung von Redundanzen, Verbesserung der Genauigkeit und Senkung des Aufwands für die Informationsaktualisierung ermöglicht XML die Wiederverwendung von Informationen und deren Veröffentlichung in anderen Formaten. So können Sie eine zentrale Informationsquelle schaffen, mit der durch eine einzelne Änderung an der Quelle automatisch alle Dokumente aktualisiert werden, in denen diese Informationen enthalten sind.
- **Gemeinsame Nutzung** – XML ermöglicht den Aufbau der Informationen auf eine Weise, die es anderen Gruppen innerhalb und außerhalb der Organisation erlaubt, Ihre Informationen automatisch in ihre eigenen Prozesse einzubinden und so einen Mehrwert zu schaffen.
- **Relevanz** – Mit XML können Sie modulare Informationen erstellen, die Sie automatisch entsprechend den Anforderungen einzelner Personen zusammenstellen können, so dass diese alle benötigten Informationen erhalten – nicht mehr und nicht weniger.

- **Automatisierung** – Die Automatisierung ist der Schlüssel zu einer kostengünstigen Umsetzung dieser Ziele. XML ermöglicht die Automatisierung, da Sie damit die für automatisierte Prozesse nötige konsistente Struktur vorschreiben können.

Diese konsistente Struktur für die Automatisierung wird vom „Datenmodell“ definiert, der DTD oder dem Schema, in der bzw. dem die in den Dokumenten zulässigen Tags und deren erlaubte Verwendung beschrieben sind.

Stellen Sie sich einmal Folgendes vor: Sie arbeiten für eine großes Unternehmen mit einem sehr vielfältigen Produktspektrum und sind für die Überwachung der technischen Dokumente für jedes einzelne Produkt zuständig. Ihre Entscheidung fiel auf XML, weil Sie damit enorme Effizienzen beim Verfassen, Übersetzen und Veröffentlichlichen realisieren können. Gleichzeitig kann Ihre Organisation ihren Kunden damit präzisere, zeitnähere und relevantere Informationen bereitstellen.

Im Kern des Systems befindet sich ein Datenmodell, das Sie für die Veröffentlichung technischer Dokumentation für jede Gruppe im Unternehmen entworfen haben. Mit diesem unternehmensweiten Veröffentlichungskonzept können Sie nicht nur die Kosten für den Kauf von Veröffentlichungswerkzeugen und die Entwicklung von Veröffentlichungsanwendungen senken, sondern Ihr Unternehmen kann gegenüber seinen Kunden auch einen einheitlichen Auftritt pflegen und den Inhalt für verschiedenste Produkte zu Dokumentationen kombinieren, die genau auf die Anforderungen jedes einzelnen Kunden zugeschnitten sind.

Als die Gruppen allerdings nach und nach die Arbeit aufnehmen, kommen die Mängel Ihres Datenmodells zum Vorschein. Jede Gruppe hat Anforderungen, die Sie nicht vorausgesehen haben, und fordert nun Änderungen. Jede Änderung betrifft nicht nur das Datenmodell selbst, sondern auch alle nachgeordneten Anwendungen, die von diesem Datenmodell abhängig sind – insbesondere die Zusammenstellung und die Veröffentlichung. Jede Änderung kann zahlreiche Veröffentlichungs-Formatvorlagen und auch andere Verarbeitungslogik betreffen. Die Anwendung muss daher aktualisiert, getestet, dokumentiert und erneut implementiert werden.

Je mehr neue Gruppen weitere Änderungen von Ihnen fordern oder bereits bestehende Gruppen neue Gelegenheiten erkennen, die weitere Änderungen notwendig machen, umso mehr wird Ihnen klar, dass die Änderungswünsche nicht nur kein Ende finden werden, sondern dass Sie auch noch den Anschluss verlieren.

DITA ist da

Angesichts dieser Probleme kommen Sie zu dem Schluss, dass ein neues Konzept her muss. Sie müssen eine Möglichkeit finden, mehrere widersprüchliche Anforderungen zu erfüllen:

- Den vielfältigen Bedürfnissen zahlreichen Bereichen der Organisation gerecht werden
- Schnell und einfach auf geänderte Geschäftsanforderungen reagieren
- In hohem Maß modulare Informationen für optimale Wiederverwendung und Personalisierung unterstützen
- Die Einfachheit wahren, damit die Autoren schnell produktiv arbeiten können

Als Erstes erstellen Sie eine DTD, die alle allgemeinen Formatierungskonstrukte enthält, die für technische Dokumente

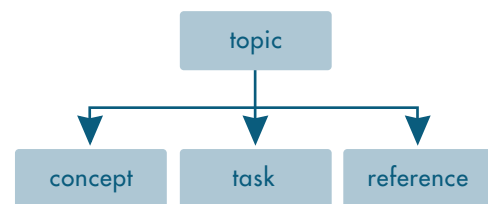
notwendig sind. Und was wäre ein besserer Ausgangspunkt als HTML, das seine Flexibilität bereits bei Millionen von Webseiten bewiesen hat? Titel, Absätze, Aufzählungen und nummerierte Listen, kursive Wörter – in HTML lässt sich all dies und noch viel mehr darstellen.

Ihre neue DTD enthält keine Tags, die für Ihr Unternehmen spezifisch sind, sondern nur „generische“ Tags wie jene in HTML. Diese generischen Tags steuern das Aussehen Ihrer Informationen, stellen aber kaum etwas von der „Semantik“ – ihrer Bedeutung – dar. Die neue DTD enthält auch keine Dokumenthierarchie, sondern stellt lediglich ein Informationsmodul dar. (Später werden Sie noch eine Möglichkeit finden, Module zu vollständigen Dokumenten zusammenzustellen, die Sie Ihren Kunden zur Verfügung stellen können.)

Sie nennen die DTD „Topic“ und entwerfen dann einen Mechanismus, mit dem Sie „Topic“ einfach an Ihre spezifischen Anforderungen anpassen können. Inspiriert von Arbeiten aus den neunziger Jahren des vergangenen Jahrhunderts zum Thema der Architekturformen, entwerfen Sie eine Methode für die Definition neuer Tags durch Vererben der Eigenschaften von vorhandenen Tags. Jede Anwendung, die Ihre Syntax für die Definition neuer Tags auf Grundlage bestehender Tags versteht, kann Ihre neuen Tags verarbeiten.

Da Sie technische Dokumentationen erstellen, wenden Sie sich nun den spezifischen Informationstypen zu, die für derartige Dokumente erforderlich sind. Basierend auf den Grundlagen der Informationsarchitektur, zu der umfassende Forschungsergebnisse, Grundlagenwissen und Erfahrungen vorliegen, welche die effizienteste und effektivste Methode definieren, um einer Zielgruppe Informationen zu vermitteln, definieren Sie drei Spezialisierungen von „Topic“, die als Baublöcke Ihrer Dokumente dienen sollen:

- Concept
- Task
- Reference



„Concept“, „Task“ und „Reference“ sind Spezialisierungen von „Topic“ und stellen jeweils eine andere Möglichkeit dar, Informationen über ein Thema bereitzustellen. Zwar sollten die meisten DITA-fähigen Systeme die spezifischen Verarbeitungsanforderungen für „Concept“, „Task“ und „Reference“ verstehen, jedes DITA-fähige System kann aber jede beliebige Spezialisierung verarbeiten.

Diese Spezialisierungen sind nicht nur eine hervorragende Idee, sondern stellen eine weitere wichtige Neuerung in DITA dar: die Verwendung von Datenmodellen, um Autoren dahin zu führen, dass die von ihnen erstellten Informationen einfacher zu schreiben und zu verstehen sind.

DITA-Spezialisierung – Anpassung an Ihre Anforderungen

„Spezialisierung“ ist ein Mechanismus in DITA, der Ihnen die Anpassung von DITA an Ihre eigenen Anforderungen unter Wahrung der Interoperabilität und Wiederverwendbarkeit des bestehenden Inhalts und Ihres bisherigen Veröffentlichungssystems ermöglicht. Die Spezialisierung senkt außerdem die Kosten für die Pflege Ihrer Veröffentlichungsanwendungen bei geänderten Geschäftsanforderungen, da Sie in vielen Fällen lediglich Ihre Datenmodelle ändern können, ohne die nachgeordneten Anwendungen zu verändern.

In diesem Abschnitt werden die Auswirkungen und die Bedeutung der Spezialisierung beschrieben. Damit Sie sich die Wichtigkeit der Spezialisierung umfassend vor Augen führen können, müssen wir zunächst die Prinzipien der Wiederverwendung von Informationen erläutern, einem der Eckpfeiler eines XML-Veröffentlichungssystems.

Wiederverwendbarkeit – Mythos oder Wirklichkeit?

Von den vielen Versprechungen der Veröffentlichung mit XML ist die Möglichkeit zur vielfältigen Wiederverwendung von Informationen eine der bedeutendsten. Bei Verwendung von XML soll es möglich sein, dieselben Informationen ohne jegliche Veränderung in unterschiedliche Dokumente einzufügen und diese Informationen in völlig unterschiedlichen Medien zu veröffentlichen – unter Wahrung einer einzelnen Quelle, die es ermöglicht, mit einer einzigen Änderung automatisch alle Dokumente zu aktualisieren, in der die betroffenen Informationen erscheinen.

Diese Geschichte ist kein Mythos: Ein derartiges Maß an Wiederverwendung wurde tatsächlich in vielen Organisationen erreicht. Problematisch wird es dann, wenn Sie versuchen, Informationen in unterschiedlichen Dokumenttypen wiederzuverwenden, insbesondere, wenn die Informationen aus unterschiedlichen Quellen innerhalb und außerhalb der Organisation stammen.

Das Problem entsteht folgendermaßen: Jede Abteilung erstellt ein eigenes Datenmodell (DTD oder Schema) für ihre Anforderungen, das spezifische Tags für ihre Zwecke enthält. Beispielsweise können Reparaturanweisungen ein <procedure>- oder ein <step>-Tag enthalten, während ein Teilekatalog die Tags <partnumber>, <description> und <price> aufweist.

Wenn Sie nun Reparaturanweisungen in einen Teilekatalog oder Teileinformationen in ein Reparaturhandbuch einschließen möchten, müssen Sie das Datenmodell ändern und die neuen Tags ergänzen. Darüber hinaus müssen Sie auch alle Anwendungen bearbeiten, die diese Informationen verarbeiten. Beispielsweise müssten Sie in Ihren Formatvorlagen die neuen Tags hinzufügen, um Informationen veröffentlichen zu können, die die neuen Tags enthalten.

Dieses Interoperabilitätsproblem wird umso schlimmer, in je mehr Dokumenttypen, Abteilungen und Unternehmen Sie Informationen wiederverwenden möchten. An dieser Stelle wird das Versprechen der Wiederverwendbarkeit zum Mythos, da die Kosten für die Anpassung unterschiedlicher Datenmodelle die Vorteile ihrer Einbindung überschreiten.

Wenn sich jede Abteilung auf eine einfache, generische Gruppe von Tags wie in HTML beschränken würde, wäre der Informationsaustausch ganz einfach. Andererseits würde der Einsatz eines einfachen, starren Datenmodells vielen Vorteilen der Automatisierung und Inhaltsverarbeitung entgegenstehen.

Was für ein Dilemma! Je genauer eine Organisation ihr Datenmodell an ihre Anforderungen anpasst, umso wertvoller werden die

Informationen. Doch je spezialisierter das Datenmodell, umso schwieriger werden die Wiederverwendung und gemeinsame Nutzung der Informationen.

Trotz dieses Widerspruchs ist es vielen Organisationen gelungen, aus der Inhaltswiederverwendung und der Automatisierung innerhalb einer Arbeitsgruppe oder Abteilung beträchtliche Vorteile zu ziehen. Stellen Sie sich nur vor, welche Vorteile realisiert werden könnten, wenn man diese Fähigkeiten über Abteilungs- und sogar Unternehmensgrenzen hinweg ausdehnen könnte!

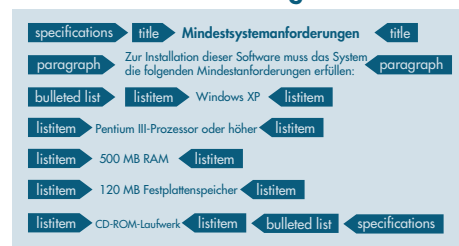
Entwurf des idealen Datenmodells

Das ideale Datenmodell sollte den Anforderungen jedes Einzelnen gerecht werden und dennoch so einfach sein, dass es schnell erlernt werden kann. Wie würde dieses Datenmodell aussehen?

Damit es den Anforderungen jedes Einzelnen gerecht werden kann, müsste das Datenmodell Hunderttausende von Tags in Zehntausenden möglicher Kombinationen enthalten. Damit es jedoch von jedem einfach erlernt werden kann, sollte es maximal 100 Tags aufweisen!

Hier kommen wir also nicht zusammen.

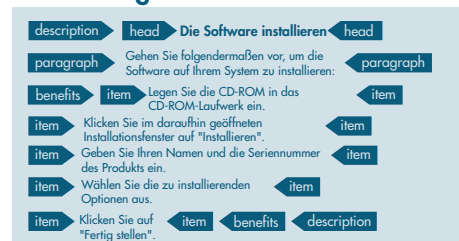
Technische Entwicklung



Training



Marketing

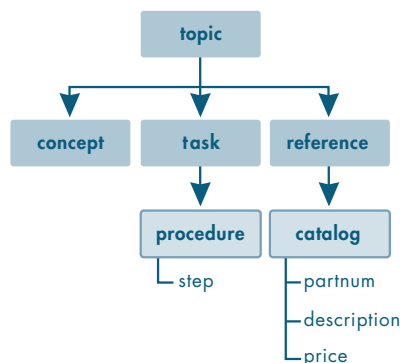


Jede Abteilung stellt individuelle Anforderungen an die von ihr erstellten Inhalte. Dennoch sollten diese Informationen in der gesamten Organisation wiederverwendet werden können. Wie können Sie ein Datenmodell erstellen, das einerseits den Bedürfnissen der ganzen Organisation gerecht wird und andererseits einzelnen Autoren ausreichende Anwenderfreundlichkeit bietet?

Die Entwickler von DITA kamen auf eine geniale Alternative: anstelle eines Datenmodells, das versucht, allen Anforderungen zu entsprechen, entwickelten Sie ein Datenmodell, das an alle Anforderungen angepasst werden kann. Diese Idee steht hinter der „Spezialisierung“, die es Ihnen erlaubt, das DITA-Datenmodell an Ihre eigenen Bedürfnisse anzupassen, während andere DITA-fähige Anwendungen Ihren Inhalt einbeziehen können, selbst wenn Ihre spezifischen Tags unbekannt sind.

Wenn Sie DITA verwenden, beginnen Sie zunächst mit den Basis-Tags von DITA. Diese Basis-Tags sind mit HTML vergleichbar, das seine Flexibilität bereits bei Millionen von Webseiten und in mehreren verbesserten Generationen bewiesen hat und die meisten gängigen Formatierungs- und Struktur-Tags bereitstellt, die Sie benötigen.

Sie passen DITA an Ihre Anforderungen an, indem Sie eigene Tags hinzufügen. Bei einem anderen Datenmodell würde durch das Hinzufügen eigener Tags die Interoperabilität Ihres Inhalts sofort eingeschränkt oder verhindert werden. Genau hier kommt einer der Tricks von DITA ins Spiel: Wenn Sie ein Tag hinzufügen, geben Sie nicht nur dessen Namen an, sondern auch, welchem der Basis-Tags es am meisten ähnelt. In anderen Worten: Sie erstellen ein neues Tag, das die Eigenschaften und Verhaltensweisen eines bestehenden Tags erbt. (Genau hierfür steht das „D“ in DITA: für Darwin und dessen Arbeit in der biologischen Vererbungslehre.)



Wie das Beispiel zeigt, können Sie Spezialisierungen auch von anderen Spezialisierungen ableiten – so ist „procedure“ eine Spezialisierung von „task“ und „catalog“ eine Spezialisierung von „reference“. Innerhalb jeder Spezialisierung können Sie neue Tags für Ihre spezifischen Anforderungen erstellen, wobei jedes der Tags eine Spezialisierung eines Tags in „Topic“ oder einer Spezialisierung von „Topic“ ist.

Nehmen wir an, Sie möchten das <partnumber>-Tag erstellen. Da Teilenummern in der Regel im laufenden Text (inline) erscheinen, konfigurieren Sie es so, dass es vom <emph>-Tag („emphasis“ – Hervorhebung) erbt, das im DITA-Basisdatenmodell enthalten ist. Sie haben dennoch die Möglichkeit, in Ihrer Anwendung eine spezielle Verarbeitung für <partnumber>-Tags zu definieren. Beispielsweise könnten Sie eine andere Formatierung und eine angepasste Validierung anhand einer Datenbank einrichten.

Die Vorteile werden dann deutlich, wenn eine Person, die nichts über <partnumber>-Tags weiß, Ihren Inhalt dennoch problemlos einbinden und wiederverwenden kann, ohne Änderungen an ihrer eigenen zugrunde liegenden Anwendung vorzunehmen. Wie ist das möglich? Nun, Ihr Datenmodell enthält die Vererbungsinformationen. Eine andere DITA-fähige Anwendung kann anhand dieser Informationen feststellen, dass das unbekannte <partnumber>-Tag stattdessen wie <emph> zu behandeln ist.

Ein weiteres Beispiel: Angenommen, Sie möchten eine Vorgehensweise definieren, die aus mehreren Schritten besteht. Sie könnten nun das <procedure>-Tag so konfigurieren, dass es vom -Tag („ordered list“ – nummerierte Liste) erbt, oder Sie könnten das <step>-Tag so konfigurieren, dass es vom -Tag („list item“ – Listenelement) erbt.

Die Funktionsweise der Spezialisierung

Die Spezialisierung entspricht den XML- und XSL-Standards: DITA-Dokumente sind in allen Aspekten vollständig mit diesen Standards kompatibel.

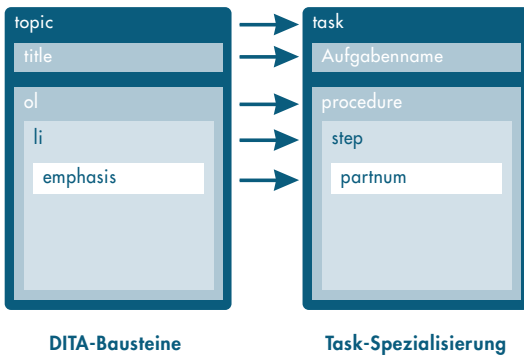
Wenn Sie DITA an Ihre individuellen Bedürfnisse anpassen möchten, beginnen Sie mit „Topic“, dem DITA-Basisdatenmodell, von dem alle Spezialisierungen abgeleitet sind, und geben Ihrer ersten Spezialisierung einen geeigneten Namen. Nehmen wir an, Sie erstellen ein Datenmodell und nennen es „Task“. Zur Erstellung des Datenmodells „Task“ verwenden Sie als Ausgangspunkt die Tags in „Topic“ und fügen dann eigene Tags hinzu. Sie fügen Ihrem Datenmodell nun also das <procedure>- und das <step>-Tag hinzu.

Sie folgen dem DITA-Modell zur Definition neuer Tags und geben das Tag an, von dem die neuen Tags jeweils ihre Eigenschaften erben. Diese Informationen werden in einem speziellen Attribut gespeichert, das die Entwickler von DITA „class“ nannten. Das class-Attribut enthält die Vererbungsinformationen. In einer DTD enthält die Syntax der class-Attributdefinition das Schlüsselwort ATTLIST. Im folgenden Beispiel werden die Vererbungsinformationen des <step>-Tags definiert:

```
<!ATTLIST step class CDATA "- topic/li task/step">
```

Mit dieser ATTLIST-Deklaration wird nicht nur das class-Attribut für das <step>-Tag deklariert, sondern auch der Standardwert des Attributs angegeben. Gemäß der DITA-Methodik würden Sie das Ändern dieses Attributwerts während der Bearbeitung durch die Autoren unterbinden.

Im Beispiel oben wird „- topic/li task/step“ als class-Attribut für <step> festgelegt (die doppelten Anführungszeichen sind nicht Teil des Attributwerts). Für DITA-fähige Anwendungen bedeutet „- topic/li task/step“, dass Anwendungen, denen das Datenmodell „Task“ bekannt ist, <step> als <step> behandeln sollen. Ist „Task“ der Anwendung jedoch nicht bekannt, soll <step> wie im Datenmodell „Topic“ behandelt werden.



Spezialisierungen basieren auf den Bausteinen in DITA. Wenn eine Anwendung oder eine Formatvorlage die Tags in der Spezialisierung nicht erkennt, werden sie wie die DITA-Tags behandelt, von denen sie abgeleitet wurden.

Für die Inhaltsveröffentlichung mit DITA müssen Sie eine Formatvorlage einrichten, die mit dem class-Attribut übereinstimmt, nicht mit dem Tag. Die Formatvorlage für das Basisdatenmodell „Topic“ kann beliebige spezialisierte Tags einfach anhand des class-Attributs behandeln.

Einfachere Erstellung, bessere Verständlichkeit

Untersuchen wir nun einmal näher die drei Spezialisierungen, die von IBM definiert wurden – „Concept“, „Task“ und „Reference“ – und deren Auswirkungen auf Erstellung und Verständlichkeit.

Der Inhalt eines Topics (Themas)

Der grundlegende Baustein von DITA ist ein „Topic“ – ein kurzer beschreibender Text mit einem bestimmten Zweck.

„Topic“ umfasst einen grundlegenden Satz von Tags für die Organisation und Darstellung von Informationen. Hierzu gehören u.a. die folgenden Tags:

Organisations-Tags	
Tag name	Beschreibung
<title>	Überschrift oder Beschriftung. Kann der Haupttitel (unter <topic>) oder ein Untertitel (z.B. unter <section>) sein.
<shortdesc>	Kurzbeschreibung für die Link-Vorschau und die Suche
<section>	Unterabschnitt eines Topics. Ein Topic kann mehrere Abschnitte enthalten, ein Abschnitt kann jedoch keinen anderen Abschnitt enthalten.
<body>	Enthält den Hauptinhalt des Topics.
<p>	Paragraph
	Ungeordnete Liste (Aufzählung)
	Geordnete Liste (nummerierte Liste)
<table>	Tabelle

Darstellungs-Tags	
Tag name	Beschreibung
	Fett
<i>	Kursiv
<u>	Unterstrichen
<tt>	Fernschreiberschriftart (Festbreitenschriftart)
	Ungeordnete Liste (Aufzählung)

„Topic“ enthält keine branchen-, anwendungs- oder dokumenttypspezifischen Tags. Da DITA die Anpassung an Ihre spezifischen Anforderungen erlaubt, können Sie „Topic“ durch Hinzufügen eigener Tags „spezialisieren“.

Beim Hinzufügen eines neuen Tags müssen Sie ein bestehendes Tag angeben, das dem neuen Tag am ähnlichsten ist (siehe Spalte „Erbt von“ in der folgenden Tabelle). Dies ermöglicht anderen DITA-fähigen Anwendungen die Verarbeitung Ihres Inhalts, selbst wenn die neuen Tags völlig unbekannt sind. Sie behandeln die neuen Tags dann so, als handelte es sich um die Tags, von denen sie erben (z.B. würde eine DITA-fähige Anwendung, die <prereq> nicht kennt, dieses Tag wie <section> behandeln, siehe folgende Tabelle).

Die Kategorisierung von Informationen

Durch die Spezialisierung von „Topic“ schaffen Sie einen neuen „Informationstyp“ (im Englischen als „information type“ oder „infotype“ bezeichnet). Beispielsweise erstellte IBM eine Spezialisierung von „Topic“ mit dem Namen „Task“, welche die Schritte einer Vorgehensweise enthält. „Task“ ist somit ein Informationstyp und enthält spezialisierte Tags wie:

Task Tags		
Tag name	Beschreibung	Erbt von
<prereq>	Voraussetzung: etwas, das der Anwender vor Beginn der Aufgabe wissen oder tun muss.	<section>
<steps>	Eine Reihe von Schritten, die der Anwender zum Abschluss der Aufgabe befolgen muss.	
<result>	Das erwartete Ergebnis der Aufgabe.	<section>

DITA ermöglicht Ihnen die Erstellung eigener Informationstypen, die Sie auf Grundlage von „Topic“ oder eines anderen Informationstyps anpassen können. Nehmen wir beispielsweise an, Sie möchten den neuen Informationstyp „Reparaturverfahren“ definieren. Sie könnten „Topic“ an Ihre Anforderungen anpassen, es wäre aber einfacher, „Task“ als Ausgangspunkt zu verwenden, da darin bereits einige benötigte Tags enthalten sind.

Wenn Sie ein Dokument erstellen möchten, stellen Sie ein oder mehrere Topics oder andere Informationstypen zusammen. Dokumente können eine beliebige Kombination von Informationstypen enthalten. Beispielsweise kann ein einzelnes Dokument mehrere „Topics“ und „Tasks“ sowie Ihre eigenen Informationstypen enthalten.

Erstellen besserer Informationen

Basierend auf den Grundlagen der Informationsarchitektur, zu der umfassende Forschungsergebnisse, Grundlagenwissen und Erfahrungen vorliegen, hat IBM mit „Concept“, „Task“ und „Reference“ drei Informationstypen spezifisch für die technische Dokumentation definiert.

Für die technische Dokumentation gibt es noch andere Informationsarchitekturen. Beispielsweise hat das Unternehmen Information Mapping, Inc. eine Methode entwickelt und entsprechende Forschungsergebnisse veröffentlicht, die bemerkenswerte Verbesserungen beim Schreiben technischer Dokumentation (gemessen in Anzahl der Wörter und Dauer bis zum Abschluss) und auch beim Lesen technischer Dokumentation (gemessen in Dauer und Verständnis) demonstrieren. Weitere Informationen über Information Mapping erhalten Sie unter <http://www.informationmapping.com/>. Die Veröffentlichung der Forschungsergebnisse zur Verbesserung der technischen Dokumentation finden Sie unter <http://www.informationmapping.com/documents/IMI%20history%20and%20results.pdf> (über diesen Link öffnen Sie ein Adobe PDF-Dokument).

Unabhängig von der spezifischen Vorgehensweise haben alle Methoden einige Eigenschaften gemeinsam:

- Sie schreiben die Informationstypen vor (z.B.: „Concept“, „Task“, „Reference“), welche die Autoren erstellen können.
- Sie modularisieren die Informationen in kleine, wiederverwendbare Komponenten.
- Sie fokussieren die Autoren auf die Erstellung einzelner Komponenten.
- Sie automatisieren die Veröffentlichung und entlasten die Autoren von Entwurfs- und Formatierungsarbeiten.
- Sie lehren die Leser in kürzester Zeit, Informationen in bestimmten Inhaltskategorien und in einer einheitlichen Darstellung zu erwarten.
- Sie ermöglichen die einfache Wiederverwendung von Informationskomponenten in mehreren Dokumenten und den einfachen Austausch zwischen verschiedenen Organisationen.

Mit nur drei grundlegenden Informationstypen für technische Dokumentation bleibt DITA recht unkompliziert. Zwar können Sie jeden dieser Typen für Ihre Branche und Ihre Dokumente spezialisieren, dennoch sollte jedes Buch am Ende nur drei unterschiedliche Informationstypen enthalten.

Nachfolgend finden Sie eine Beschreibung jedes der DITA-Informationstypen:

- „Concept“: Bietet eine Übersicht der Aufgaben („Tasks“) oder Referenzen („References“). Konzepte sind im Allgemeinen kurz und einfach, zum Teil weil die Autoren nur nach einem Abschnitt oder einem Beispiel weitere Abschnitte oder Beispiele hinzufügen können.
- „Task“: Enthält die Schritte, die der Leser im Rahmen einer Vorgehensweise ausführen muss. „Task“ enthält zahlreiche spezialisierte Tags, um vielfältige Darstellungsmöglichkeiten der Anweisungen und der zugrunde liegenden Informationen zu unterstützen.
- „Reference“: Beschreibt die Eigenschaften einer Gruppe von Elementen, beispielsweise von einer Produktliste, von Definitionen oder Programmierbefehlen.

Zwar weist jeder dieser Informationstypen spezielle Eigenschaften auf, alle haben jedoch zahlreiche Tags gemeinsam. Neben den Organisations- und Darstellungs-Tags, die bereits vorgestellt wurden, gibt es auch Tags für die Verknüpfung (Querverweise und Links zu verwandten Themen), für das Suchen, die Navigation, für Schlüsselwörter usw.

Interessanterweise könnte DITA Sie sogar dazu bringen, einige Messgrößen für Ihre Dokumente zu berechnen:

- Die durchschnittliche Anzahl der Wörter in einer Komponente. Anhand dieser Messgröße können Sie Komponenten mit zu wenigen oder zu vielen Wörtern identifizieren (unterschiedliche Informationstypen sollten i.d.R. auch unterschiedliche Zielgrößen aufweisen).
- Das Verhältnis der Informationstypen in einem Buch. Anhand dieser Messgröße können Sie beispielsweise Bücher identifizieren, in denen zu viel Referenzmaterial und zu wenig Konzepte enthalten sind.

Kurz: Durch die Anwendung der DITA-Methode für technische Dokumentation schaffen Sie eine fertige Struktur für Ihre Handbücher. Die drei DITA-Informationstypen „Concept“, „Task“ und „Reference“ lassen sich ganz einfach an Ihre Anforderungen anpassen. Ihre Autoren können mit einem relativ einfachen Satz von Tags schnell versierte XML-Autoren werden und dabei produktiver arbeiten, nicht nur, weil sie von der Verantwortung für die Formatierung befreit werden, sondern auch, weil sie mit sorgfältig definierten und bewährten Vorlagen arbeiten. Am wichtigsten ist jedoch, dass Ihre Leser die benötigten Informationen schneller finden und dafür weniger Wörter und willkürliche Organisationsstrukturen durchsuchen müssen.

DITA für Software, Benutzeroberflächen und Formatierung

Domänenelemente sind spezialisierte Tags, die von IBM für spezifische Sachgebiete definiert wurden. Sie bieten nicht nur einen durchdachten Ausgangspunkt für die sachgebietsbezogene Spezialisierung, beispielsweise für Programmierung, Software, Benutzeroberfläche und Formatierung, sondern demonstrieren auch die Nutzung der Spezialisierungsunterstützung von DITA für beliebige Anwendungen.

Meistern der Domäne

DITA ist ein guter Ausgangspunkt für eine Topic-orientierte Informationserstellung – ein wesentlicher Punkt, dass DITA auch in Ihrer spezifischen Situation funktioniert, ist jedoch die „Spezialisierung“ durch Hinzufügen von Tags für Ihre spezifischen Anforderungen.

Wenn Sie beispielsweise Servicehandbücher veröffentlichen, empfiehlt es sich, ein Tag namens „procedure“ zu erstellen, das eine Liste der Aufgaben in der richtigen Reihenfolge enthält. In einer DITA-Anwendung erstellen Sie <procedure> als Spezialisierung eines bestehenden Tags. Da Vorgehensweisen im Wesentlichen wie nummerierte Listen aussehen, wäre <procedure> eine Spezialisierung von (geordnete Liste), das in der Regel als nummerierte Liste dargestellt wird.

Sie könnten <procedure> zwar für die unterschiedlichsten Servicehandbücher verwenden, würden aber wahrscheinlich gern zusätzliche Tags erstellen, die sich speziell auf Ihre Branche oder Organisation beziehen. Software-Unternehmen müssen beispielsweise ihre Anwendungsprogrammierschnittstelle (API) dokumentieren und benötigen zu diesem Zweck mehrere spezialisierte Tags.

Derartige spezialisierte Tags gehören einer „Domäne“ an, was eigentlich nichts anderes bedeutet als ein Fachgebiet. Im Rahmen des Entwurfs von DITA definierte IBM mehrere spezialisierte Tags-Sätze, die so genannten „Domänenelemente“, für spezifische Sachgebiete. Diese Sachgebiete umfassen Programmierung, Software, Benutzeroberfläche und Formatierung. In den folgenden Abschnitten wird der Zweck jedes dieser Domänenelementsätze erläutert.

Verwenden Sie die bestehenden Domänenelemente als Ausgangspunkt, um die Entwicklungsdauer Ihrer Anwendung zu verkürzen und die Wahrscheinlichkeit zu erhöhen, dass Sie Ihren Inhalt mit anderen in derselben Domäne austauschen können. Beispielsweise ist es nicht sinnvoll, ein <parameterlist>-Tag zu erstellen, wenn Ihre Kollegen bereits das <parml>-Tag mit derselben Funktion verwenden.

Klein anfangen

Untersuchen wir zunächst die Domänenelemente für die Formatierung. Diese heißen auch „typografische Elemente“ und sind für Anwendungen verfügbar, in denen der Autor die Formatierung des Texts steuern können soll.

Wie bitte? Verstoßen Tags, die ausschließlich die Darstellung eines Elements steuern, nicht gegen die Trennung des Inhalts von der Formatierung, die von XML gefordert wird? Ja, das stimmt. Wenn es Ihnen widerstrebt, gegen diese Trennung zu verstoßen, können Sie die typografischen Elemente weiter spezialisieren, um ihre Bedeutung zu erweitern. Beispielsweise könnten Sie das <i>-Tag für kursiven Text spezialisieren, indem Sie Tags wie <booktitle>, <emph>, <foreignword> usw. für jede Situation erstellen, in der Text kursiv dargestellt werden soll. Ob es die Mühe wert ist, zwischen diesen unterschiedlichen Bedeutungen von Kursivformatierung zu unterscheiden, hängt von Ihren Anforderungen ab (nicht von der „Reinheit“ von XML).

Ihre neuen Tags, die Sie auch als „semantische“ Tags bezeichnen könnten, weil sie die Bedeutung des Texts betreffen und nicht seine Darstellung, ermöglichen Ihnen die flexible Anpassung ihrer Formatierung zu einem späteren Zeitpunkt ohne Änderung der Tags. Dies ist sicherlich einer der großen Vorteile von XML.

Es gibt jedoch Situationen, in denen es einfach den Aufwand nicht wert ist, ein spezielles Tag zu erstellen, um die Bedeutung des Texts zu erfassen – in diesem Fall sind typografische Elemente eine durchaus akzeptable Alternative. Sie können Ihren Autoren die typografischen Elemente direkt zur Verfügung stellen – dies kann jedoch auch seine Nachteile haben.

Nachfolgend eine vollständige Liste:

Typografische Elemente	
Tag-Name	Beschreibung
	Fett
<i>	Kursiv
<u>	Unterstrichen
<tt>	„Fernschreiberschriftart“ – wählt eine Festbreitenschriftart wie Courier aus, bei der jedes Zeichen dieselbe Breite aufweist. Dies kann nützlich sein, wenn die Zeichen vertikal ausgerichtet sein sollen.
<sup>	Hochgestellt – verkleinert den Text und platziert ihn oberhalb des umgebenden Texts. Sinnvoll für Fußnoten und mathematische Ausdrücke wie $2^2 = 4$.
<sub>	Tiefgestellt – verkleinert den Text und platziert ihn unterhalb des umgebenden Texts. Sinnvoll für chemische Formeln wie H_2O .

Software-Elemente

Die folgenden Domänenelemente sind nützlich zur Beschreibung der Funktion eines Software-Programms:

Software-Elemente	
Tag name	Beschreibung
<msgph>	Meldungssatz – enthält eine von einem Programm ausgegebene Meldung.
<msgblock>	Meldungsblock – enthält eine mehrzeilige Meldung oder mehrere Meldungen.
<msgnum>	Meldungsnummer, falls vorhanden
<cmdname>	Befehlsname
<varname>	Variable – enthält eine Variable, die als Eingabe für ein Programm dient.
<filepath>	Dateipfad – enthält den Namen und optional den Speicherort einer Datei.
<userinput>	Benutzereingabe – enthält Benutzereingaben in ein Programm.
<systemoutput>	Systemausgabe – enthält beliebige Ausgaben eines Programms; (<msgph> ist spezifischer als <systemoutput>)

Für die Verwendung dieser Elemente gelten einige Regeln. Beispielsweise kann „varname“ gemäß der Definition dieser Domänenelemente in „msgph“, „msgblock“, „filepath“, „userinput“ und „systemoutput“ enthalten sein, nicht aber in „filepath“.

Diese Strukturregeln sind mindestens genauso wichtig wie die Tag-Namen. Es ist relativ einfach, die Tags ineinander zu konvertieren, hingegen ist es viel schwieriger, Strukturen ineinander zu konvertieren.

Benutzeroberflächen- und Programmiererelemente

Die Domänenelemente werden durch Elemente für Benutzeroberflächen und Programmierung abgerundet.

Verwenden Sie die Benutzeroberflächenelemente, um ausführlich die Anwendereingaben in ein Computerprogramm zu beschreiben. Diese Elemente umfassen Menüs, Tastenkombinationen, Computerbildschirme und Benutzeroberflächen-Steuerelemente wie Schaltflächen und Menüelemente.

Im Gegensatz zu den anderen Domänen mit jeweils einigen wenigen Elementen umfasst die Programmierdomäne 25 Elemente, da für die Dokumentation der Arbeit mit einer Programmiersprache eine

Vielzahl von Teilen erforderlich ist. Mit Elementen u.a. für Schlüsselwörter, Code, Variablen und Parameter können Sie eine Programmiersprache in all ihren Details beschreiben.

Die Umsetzung von DITA

Konzeptgemäß besteht DITA-Inhalt aus Informationskomponenten, nicht aus Büchern. Wie lassen sich diese Informationskomponenten also in einen zusammenhängenden Satz von Informationen umwandeln, beispielsweise in eine Online-Hilfdatei oder ein gedrucktes Dokument? Hier kommen Maps ins Spiel.

Und wo schließlich erhalten Sie Tools, die den Aufbau von DITA-Anwendungen, das Erstellen von DITA-Inhalten, das Zusammenstellen des Inhalts zu zusammenhängenden Sammlungen sowie die Veröffentlichung für unterschiedliche Medien unterstützen? Bei PTC natürlich!

Der DITA-Inhalt ist bereit! Wie geht es nun weiter?

Einer der wichtigsten Vorteile Topic-orientierter Inhalte, wie DITA sie fördert, ist die Möglichkeit für Autoren, relativ kleine Informationskomponenten zu schreiben. Anstelle eines Auftrags über ein 1.000-seitiges Dokument bearbeiten die Autoren Informationskomponenten von nur wenigen Seiten Länge.

Für die Zusammenstellung dieser Komponenten zu Veröffentlichungen definiert DITA eine „Map“, in der Sie die einzuschließenden Komponenten, deren Reihenfolge und ihre hierarchische Beziehung festlegen können.

Angenommen, Sie haben die folgenden Informationskomponenten verfasst:

- Topic: Vergaser
- Concept: Die Funktionsweise von Vergasern
- Task: Einstellen des Vergasers
- Task: Austauschen des Vergasers
- Reference: Das Verhältnis des Vergasers zu anderen Motorbauteilen

Wenn Sie diese Informationskomponenten zu einer Veröffentlichung zusammenstellen möchten, führen Sie sie zunächst in einer Map auf. In der Map können Sie nicht nur die einzuschließenden Komponenten angeben, sondern auch hierarchische Beziehungen definieren.

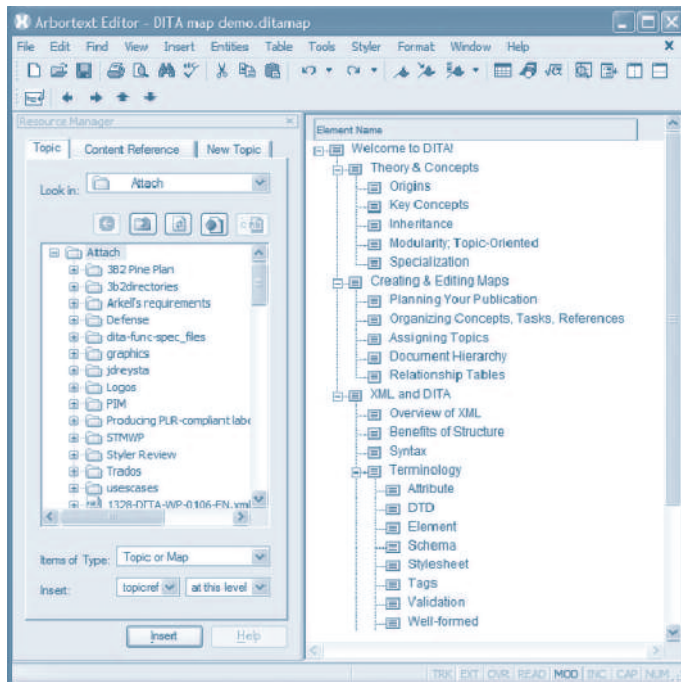
In diesem Beispiel sind die vier letzten Komponenten der ersten untergeordnet, wie die folgende Hierarchie zeigt:

- Topic: Vergaser
 - Concept: Die Funktionsweise von Vergasern
 - Task: Einstellen des Vergasers
 - Task: Austauschen des Vergasers
 - Reference: Das Verhältnis des Vergasers zu anderen Motorbauteilen

Das Hauptelement einer Map ist <topicref>, das als Einschlussmechanismus (vergleichbar mit Dateientitäten oder XInclude) für Informationskomponenten in das aktuelle Dokument fungiert.

Die folgende Abbildung zeigt eine Map der Veröffentlichung zum Thema Vergaser. Um die Hierarchie zu definieren, können Sie `<topicref>` in sich selbst „verschachteln“: Das innere `<topicref>` ist dann dem ersten `<topicref>` untergeordnet.

Im folgenden Beispiel enthält `<topicref>` für Vergaser die `<topicref>`-Elemente für alle übrigen Informationskomponenten. Diese inneren Komponenten sind also der äußeren Komponente untergeordnet:



Als Teil von Arbortext 5.3 bietet Arbortext Editor einige Tools, die das Erstellen von Topics (oder die Spezialisierung von „Topic“, z.B. „Concept“, „Task“ und „Reference“) sowie von „Maps“ mit einer Liste der in einen Informationssatz (z.B. ein Buch oder eine Online-Hilfedatei) einzuschließenden Topics und deren Navigation vereinfacht.

Alles unter einem Hut – DITA-Unterstützung durch PTC

Zwar war die ursprüngliche Absicht von DITA die Einhaltung bestehender Standards. Dennoch wurden einige Merkmale neu eingeführt, die zusätzliche Verarbeitungsschritte erfordern. Es ist zu erwarten, dass sich DITA für vielfältige Anwendungen durchsetzen wird. PTC hat sich aus diesem Grund zu beträchtlichen Investitionen in zusätzliche DITA-spezifische Funktionalitäten entschlossen.

PTC hat die Arbortext Produktlinie um mehrere Merkmale ergänzt, die DITA-Unterstützung bereitstellen oder verbessern:

- **Erstellen:** Arbortext Editor enthält mehrere Funktionen, die ihn zu einer leicht zu erlernenden und äußerst produktiven Umgebung für das Erstellen und Bearbeiten von Topics und Maps machen. Hierzu gehören: Hinzufügen von Komponenten zu einem Topic oder von Topics zu einer Map mit Hilfe von Ziehen-und-Ablegen; ein neuer Ressourcen-Manager, der wiederverwendbare Komponenten sofort verfügbar macht; und eine neue Spaltenansicht, mit der eine Map wie ein Inhaltsverzeichnis dargestellt werden kann.
- **Zusammenstellen und Veröffentlichen:** Arbortext Publishing Engine nutzt die DITA-Map zum Zusammenstellen und anschließenden Veröffentlichen der Dokumente für mehrere Medientypen (sowohl gedruckt als auch elektronisch).
- **Spezialisierung:** Die Arbortext Produktlinie unterstützt die Spezialisierungsmöglichkeit von DITA. Daher können Sie mit einer einzelnen Formatvorlage den Stil von „Topic“, „Task“, „Concept“, „Reference“ und anderen Spezialisierungen steuern. XSL-Formatvorlagen stellen zwar bereits einen Mechanismus zur Unterstützung von Spezialisierungen bereit. Wir mussten aber dennoch die Unterstützung für Spezialisierungen zu unserem internen Formatvorlagenformat für die Bearbeitung hinzufügen.
- **Semantische Tabellen:** Die DITA-DTD enthält einige Elemente, die in Form einer Tabelle und nicht als Fließtext dargestellt werden müssen. Arbortext Produkte ermöglichen Ihnen daher die Auswahl der Tags, die als Tabelle angezeigt und gedruckt werden sollen.

Beispielsweise erstellen Sie für einen Katalog möglicherweise die Tags `<partnum>`, `<description>` und `<price>` und möchten, dass diese als Tabelle dargestellt werden. Ohne Unterstützung für semantische Tabellen müssten diese Tags als Fließtext dargestellt werden:

`<partnum>` 123-456 `</partnum>`

`<description>` Vergaser `</description>`

`<price>` 45,99 € `</price>`

Manchmal lassen sich Informationen sinnvoller als Tabelle darstellen (siehe unten), nicht als Liste oder ganzer Satz.

Um die Bearbeitung zu erleichtern, können Sie in Arbortext Editor einstellen, dass Tags als Tabellenzellen angezeigt werden sollen:

Teilenummer	Beschreibung	Preis
123-456	Vergaser	45,99 €

- **Conref-Unterstützung:** In DITA werden Informationskomponenten mit Hilfe von „conref“ in ein Dokument eingeschlossen. Dies ist ein Ersatz für den Mechanismus mit Dateientitäten und XInclude und erfordert eine spezifische Funktionalität, die PTC in der Arbortext Produktlinie ergänzt hat.
- **Topicref-Unterstützung:** In DITA werden Topics (oder Spezialisierungen von „Topic“ wie „Concept“, „Task“ und „Reference“) mit Hilfe von „topicref“ in eine Map eingeschlossen. Dies ist ein Ersatz für den Mechanismus mit Dateientitäten und XInclude und erfordert eine spezifische Funktionalität, die PTC in der Arbortext Produktlinie ergänzt hat.
- **Dokumenttypen:** Die Arbortext Software beinhaltet eine Reihe von Dokumenttypen für „Topic“, „Concept“, „Task“ und „Reference“. Ein Dokumenttyp umfasst die DTD, eine Formatvorlage und weitere geeignete Anpassungen für den Dokumenttyp.

Zusammenfassung

Einer der wichtigsten Vorteile eines XML-basierten Veröffentlichungssystems ist die Möglichkeit, Informationen innerhalb des Unternehmens wiederzuverwenden und auszutauschen. In den meisten Organisationen hat jedoch jede Abteilung eigene Dokumentationsregeln und -vorgaben. Die Erstellung von Datenmodellen, die einerseits für die Autoren leicht zu verwenden sind und andererseits die vielfältigen Anforderungen der einzelnen Abteilungen unterstützen, kann ziemlich kompliziert sein. Die Verwaltung dieser Datenmodelle gestaltet sich nahezu unmöglich, wenn sich die Anforderungen der Autoren und des Unternehmens weiterentwickeln und verändern.

DITA bietet eine Lösung für diese Interoperabilitätsprobleme, indem es eine Methode bereitstellt, individuelle Datenmodelle für die spezifischen Anforderungen jeder einzelnen Abteilung in der Organisation zu erstellen. Dadurch wird die Größe der einzelnen Datenmodelle reduziert und deren Handhabung für die Autoren vereinfacht. Da jedoch alle Datenmodelle auf einem einzelnen globalen Datenmodell basieren, können die Autoren dennoch Informationen organisationsweit austauschen und wiederverwenden. Dies ist dadurch möglich, dass Formatvorlagen und Anwendungen unbekannte Tags in den individuellen Datenmodellen auf Grundlage ihrer globalen Entsprechungen verarbeiten können.

Autoren können produktiver arbeiten, da sie bestehende Informationen beliebiger Abteilungen ihrer Organisation wiederverwenden können und bei Änderungen an der Quelle automatisch alle Instanzen aktualisiert werden, an denen diese Quelle verwendet wurde. So verwenden die Autoren weniger Zeit darauf, bestehende Informationen immer wieder neu zu erstellen und mehrfach vorkommende Informationen zu aktualisieren. Durch die Wiederverwendung werden Aktualisierungen schneller, einfacher und global durchgeführt – das Ergebnis sind einheitliche und stets aktuelle Informationen. Dadurch, dass XML-Formatvorlagen und Anwendungen unbekannte Tags basierend auf den globalen DITA-Tags verarbeiten können, von denen sie abgeleitet wurden, lässt sich die Veröffentlichung fehlerfrei und ohne zeitaufwändige Aktualisierungen automatisieren.

Zusammenfassend lässt sich sagen, dass Organisationen mit DITA die Erwartungen an XML in die Tat umsetzen können – das heißt: produktivere Autoren, hochwertigere Informationen und Veröffentlichungsprozesse, die nicht nur schneller, sondern auch kostengünstiger sind.

Copyright © 2006, Parametric Technology Corporation (PTC). Alle Rechte vorbehalten. Die Inhalte dieser Seiten werden ausschließlich zu Informationszwecken bereitgestellt und beinhalten keinerlei Gewährleistung, Verpflichtung, Bedingung oder Angebot seitens PTC. Änderungen der Informationen vorbehalten. PTC, das PTC Logo, The Product Development Company, Arbortext, das Arbortext Logo, Pro/ENGINEER, Wildfire, Windchill und alle PTC Produktnamen und Logos sind Warenzeichen oder eingetragene Warenzeichen von PTC und/oder Tochterunternehmen in den Vereinigten Staaten und anderen Ländern. Alle anderen Produkt- oder Firmennamen sind Eigentum ihrer jeweiligen Besitzer.