Oliver Meyer: **aTool: Typographie als Quelle der Textstruktur**

Promotion: RWTH Aachen
Erstgutachter: Prof. Dr.-Ing. Manfred Nagl, RWTH Aachen
Zweitgutachter: Prof. Dr. Ulrik Schroeder, RWTH Aachen
Datum der Prüfung: 6. April 2005
Veröffentlichung: 308 S., Shaker-Verlag, 2006

Kurzfassung:

Um Texte schneller publizieren, nutzbringender archivieren und effektiver erschließen zu können, muss ihre innere Struktur Werkzeugen zugänglich gemacht werden. Dies geschieht heute i.d.R. nach der Texterstellung durch eine aufwändige Auszeichnung in SGML oder XML. Zur Zeit sind es noch die Verlage, die diese nachträgliche Konvertierung durchführen. Die heute verfügbaren XML-Werkzeuge sind entweder reine XML-Editoren, die für XML-Laien ungeeignet sind, oder Batch-Konverter, die letztlich ihrem Anspruch der vollautomatischen Konvertierung nicht gerecht werden können.

Mit dem in dieser Arbeit entwickelten Werkzeug aTool, das als Aufsatz auf MS-Word implementiert wurde, kann der Autor die formale Struktur eines Textes bereits erzeugen, während er den Text erstellt. Dabei nutzt aTool u.a. die Typographie, um halbautomatisch die XML-Struktur abzuleiten und so den Autor zu entlasten. Die Arbeit stellt aTool, seine Konzepte und seine Anwendungen vor.

Auch wenn aTool in erster Linie von Autoren verwendet wird, mindert es doch den Arbeitsaufwand der Verlage. Die Anforderungen an aTool wurden für diese Arbeit in Zusammenarbeit mit dem Springer-Verlag formuliert. Im primären Anwendungsgebiet von aTool, den wissenschaftlichen Zeitschriftenpublikationen, erhält der Springer-Verlag als Einreichungen eine Vielzahl von MS-Word-Dokumenten. Um den dadurch notwendigen Konvertierungsaufwand zu verringern, möchte er von seinen Autoren unmittelbar XML gemäß seinen Vorgaben erhalten. Die Autoren sind i.d.R. jedoch XML-Laien, die an gewohnten Arbeitsweisen festhalten wollen. aTool muss die Autoren daher stark unterstützen und darf ihre gewohnte Arbeitsweise nur wenig ändern. Gleichzeitig soll es flexibel an unterschiedliche Vorgaben anpassbar sein und diese bereits beim Autor überprüfen. Findet es Fehler, sollen verständliche Fehlermeldungen den Autor bei der Korrektur anleiten.

aTool erstellt die Struktur halbautomatisch gemäß den Strukturvorgaben aus dem Verlag, während derAutor den Text mit MS-Word erstellt. Dabei arbeitet aTool minimal-invasiv und eng mit MS-Word zusammen. Der Autor kann weiterhin instanzbasierte Formatierung verwenden, um Texte auszuzeichnen und zu strukturieren. Noch während der Erstellung parst aTool den Text und erstellt eine integrierte getypte XML-Struktur.

Dazu abstrahiert aTool die konkrete Formatierung zu einem definierten Formattupel und fasst gleich formatierte Zeichen zu einem Token zusammen. Das Parsing erzeugt allein aus diesem Tokenstrom einen Strukturbaum, dem im Mapping Elementtypen zugewiesen werden. Das Mapping bildet die Synthese aus Formatierungsabbildung und Strukturvorgaben. Da der Fließtext immer im Sinne der Strukturvorgaben formal korrekt, aber inhaltlich falsch interpretiert werden kann, wird dabei der Formatierung ein größeres Gewicht gegeben als den Strukturvorgaben.

Eine solche Abbildung ist per se mehrdeutig, da Autoren nicht eindeutig formatieren. aTool arbeitet daher nur halbautomatisch und verschiebt Entscheidungen, bis der Autor den Problemfall auflöst.

Jede Veränderung der Struktur prüft aTool kontinuierlich und inkrementell gegen die vom Verlag vorgegebenen Regeln in Form einer erweiterten DTD. Unterschiedliche Verlage und verschiedene Zeitschriften bedeuten wechselnde Vorgaben für die Texte. aTool ist daher in hohem Maße parametrisierbar und kann neben dem Dokumentformat sowohl an die Vorbildung und Formatierungsvorlieben einer Benutzergruppe als auch an den einzelnen Autor angepasst werden.

Der Autor kann über die Struktur im Text navigieren oder sie direkt manipulieren. Dabei helfen ihm konstruktive Operationen. Sie fügen automatisch sowohl ganze Teilbäume mit Elementen in ihrer üblichen Verwendung als auch umfassende Elemente ein.

Verstößt ein Autor gegen die Vorgaben des

Verlags, erzeugt aTool konstruktive Fehlermeldungen, die weit über das hinausgehen, was andere Werkzeuge im Fehlerfall an Unterstützung anbieten. Es berechnet den Abstand des aktuellen Dokuments zum erlaubten Dokument als Länge einer Editierfolge und fasst dann alle minimalen Editierfolgen zusammen. Die generierte Fehlermeldung gibt eine konkrete Anleitung, wie der Autor seine Dokumentstruktur korrigieren kann.

Auch wenn aTool aus den Anforderungen des Publikationsszenarios entstanden ist, reichen die möglichen Einsatzgebiete darüber hinaus. Sein Anwendungsfeld liegt überall dort, wo Textdokumente in XML mit bekannter DTD erstellt werden sollen, die Autoren jedoch vorzugsweise formatierten Fließtext in MS-Word erstellen.

Die Arbeit untersucht daher auch die Eignung für die Unterstützung einer Dokumentenfamilie. Dies geschieht am Beispiel des Benutzerhandbuchs für ein Softwaresystem, das die T-Systems GEI GmbH in unterschiedlichen Varianten vertreibt. Während bisher nur die Programmvarianten automatisch parametrisiert wurden, können mit dem hier entwickelten Konzept auch die Handbücher parametrisiert werden. Bei der Erstellung einer gemeinsamen Handbuchquelle hilft die XML-Struktur, gemeinsame und variantenspezifische Dokumentanteile zu erkennen. Diese werden methodisch zur Quelle zusammengesetzt. Anhand der Konfigurationsbeschreibung der Software generiert daraus eine Erweiterung von aTool konsistente spezifische Handbücher. Da auch in Zukunft Änderungen in der Software und den Handbüchern zu erwarten sind, geht die Arbeit ebenfalls differenziert auf die Wartung einer solchen Handbuchquelle ein. Auch hier hilft die zusätzliche XML-Struktur.

Durch die Formatierung ausgezeichnet und direkt in den Text eingebettet werden kann jedoch nur die einfache, formale, syntaktische Struktur. Die komplexe, semantische Struktur, welche die Inhalte und ihre Abhängigkeiten beschreibt, muss als Graph neben dem Text modelliert werden. Dies ist in der Kodissertation [Gat04] zu dieser Arbeit geschehen. Als Top-down-Ansatz bietet sie ein ausgefeiltes Dokumentenmodell und mächtige Analysen im Graphwerkzeug CHASID.

Als drittes Anwendungsfeld beschreibt diese Arbeit, wie aTool diese semantische Modellierung unterstützen kann. CHASID benötigt eine engere Integration mit dem konkreten Text und eine Anbindung an verbreitete Autorenwerkzeuge. Die Arbeit entwickelt daher ein Konzept zur Kopplung von aTool und CHASID, das die aus dem Fließtext abgeleitete syntaktische Struktur als Zwischenschritt zur semantischen Struktur nutzt. aTool dient als Front-End für CHASID und stellt Parametrisierungsfunktionalität bereit. CHASID kann dann unverändert für beliebige XML-Anwendungen verwendet werden. aTool bildet dazu Sichten auf die syntaktische Struktur und entscheidet anhand der Parametrisierung, wie es XML-Elemente in das semantische Modell von CHASID abbildet.

Die Integration stellt für beide beteiligten Werkzeuge einen Gewinn dar: aTool gewinnt ein umfassendes semantisches Modell hinzu, das erweiterten Analysen dient. CHASID erhält ein feingranulareres Textmodell und kann so differenziertere Aussagen über die Struktur machen als bisher.