AsciiDoc - medienneutrales, technisches Publizieren

Wilhelm Meier <wilhelm.meier@fh-kl.de>

1. Kurzfassung

AsciiDoc ist eine sehr leichtgewichtige Auszeichnungssprache. Sie eignet sich hervorragend zur Dokumentation von technischen Produkten, insbesondere von Softwareartefakten. Mit dem `asciidoc`-Prozessor kann aus einer Quelle (single-source publishing) ein sehr große Vielzahl von Medien erzeugt werden (z.B. eBook-Formate, Print-Formate, Online-Formate). Im Vortrag soll zum einen `AsciiDoc` als Sprache besprochen werden, zum anderen soll aber auch ein besonderer Editor inkl. Präprozessor (`app`) vorgestellt werden. Dieser ermöglicht durch eine Vielzahl von kontextsensitiven Hilfsfunktionen ein ganz besonders schnelles Erzeugen von Softwaredokumentation.

2. Inhalt

Technische Dokuentationen zu erstellen, ist nicht selten eine recht große Herausforderung. Gilt es doch einerseits einen äußerst präzisen Text zu schreiben, der aber auch andererseits dadurch gekenntzeichnet ist, dass viele nicht-immanente Inhalte wie etwa Bilder, spezielle Grafiken, Quelltext bzw. Fremddateien und Systemausgaben sowie etwa Fremddokumente in den Text zu integrieren sind. Alles dies zusammen bildet eine semantische Einheit und soll gemeinsam verwaltet und auch versioniert werden können. Weiterhin soll das Endprodukt in den unterschiedlichsten Formaten publiziert werden können: von der reinen *Druckversion* etwa als `pdf` oder *PostScript* über Online-Formate (`html`, `html5`) bis zu *eBook*-Version (`epub`, `mobi`), wobei selbstverständlich jedwedes Format aus ein und derselben Quelle automatisch erzeugt werden soll.

AsciiDoc bietet genau die Voraussetzungen, um die o.g. Ziele zu erreichen:

- `AsciiDoc` ist eine **leichtgewichtige** Auszeichnungsprache[1]. Das ist eine wichtige Voraussetzung für schnelles Dokumentieren.

- `AsciiDoc` bietet über einen *Filter*-Mechnismus die Möglichkeit, nicht-immanente Inhalte aus dem Text heraus zu generieren — bspw. Grafiken wie UML- oder Prozess-Diagramme — oder Fremddokumente (etwa Quelltext, etc.) zu inkludieren / zu formatieren.

- `AsciiDoc` gestattet es, über verschiedene *Backends* aus dem einheitlichen Text diverseste andere elektronische Formate zu generieren wie etwa `html` oder `docbook`, die natürlich durch zusätzliche Werkzeuge beliebig weiterverarbeitet werden können (etwa `xslt`-Prozessoren).

Der Vortrag wird diese Möglichkeiten an *hands-on* Beispielen demonstrieren und zudem einige reale Anwendungsfälle (Buch- bzw. Dokumentationsprojekte) präsentieren.

Obgleich der originäre Text natürlich mit jedem beliebigen Editor erstellt werden kann, wünscht man sich bei der Bearbeitung von aufwändigeren Projekte vor allem

- einen `AsciiDoc` sensitiven Editor, der die Möglichkeiten der *Auszeichnungsprache* sowie der *Filter* kontextsensitiv[2] unterstützt, und

[1] Wie sie in vielen Wikis (an `MarkDown` angelehnt) verwendet wird.
[2] Die Editoren `vim` und `emacs` bieten nur Syntaxhervorhebung.

- eine Möglichkeit, eigene Erweiterungen (Templates) einzufügen und auch weitergehende Inhaltsprüfungen[3] (content-asserts) durchführen zu können.

Um diese Lücke zu füllen, sollen zwei zusätzliche Werkzeuge ebenfalls präsentiert werden:

- `aedit`: einen auf `AsciiDoc`-spezialisierten Editor zur Erfüllung der o.g. Aufgaben
- `app`: einen `AsciiDoc`-*Präprozessor* für eigene Erweiterungen und Prüfungen

Beide Werkzeuge sind unabhängig voneinander, so dass die zusätzlichen Funktionsmerkmale durch `app` auch *ohne* den Editor `aedit` verwendet werden können.

Auch hier werden die Möglichkeiten wiederum durch Beispiele und reale Projekte dargestellt.

[3] Der *Filter*-Mechanismus von `AsciiDoc` ist ein *pre*-processing und kann damit keine `AsciiDoc`-Elemente verwenden.