JTR05 Fachvortrag für Fortgeschrittene

infoflow – Publikationsstrecken nach dem Baukastenprinzip

Jan Oevermann, Hochschule Karlsruhe

Viele Quellen, viele Ziele, viel Arbeit – das war bisher die Devise beim Erstellen automatisierter Publikationsstrecken. Doch was, wenn man sich den gewünschten Prozess einfach aus einzelnen Bausteinen zusammensetzen könnte? Dieser Frage wurde in einer Masterarbeit methodisch und technisch nachgegangen. Das Ergebnis ist infoflow: Eine modulare Publikationsplattform mit Web-Interface und flexibler Architektur. Multimedia-Inhalte, Word-Dokumente, CMS-Content und mehr können kombiniert, umstrukturiert und angereichert werden, um als PDF, Website, hybride Smartphone-App oder für ein Content-Delivery-Portal ausgegeben zu werden. Im folgenden Beitrag sollen das zugrundeliegende Konzept und die erste technische Implementierung kurz vorgestellt und erklärt werden.

Motivation

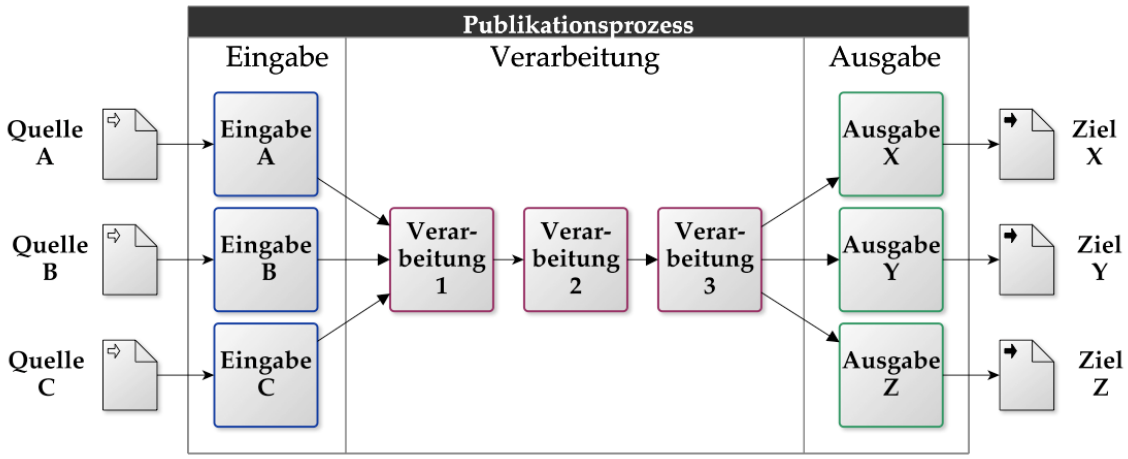
In den letzten Jahren haben anhaltende Trends auf dem Gebiet der Technischen Kommunikation neue Herausforderungen für Personal und Technik mit sich gebracht. Durch eine generell zunehmende Menge an Informationen und einen branchenübergreifenden Trend zur Digitalisierung hat sich die Zahl von Informationsquellen sowie möglicher Ausgabeformen tendenziell erhöht.

Der Technische Redakteur wandelt sich immer mehr zum modernen Medienmanager, der neben seiner traditionell schreibenden Arbeit auch Dritt-Informationen sammeln und bündeln muss, um diese anschließend verschiedenen Ausgabemedien zuzuführen. Diese Entwicklung erscheint konsequent, denn in den TD-Abteilungen einer Firma ist in der Regel das Know-How zur Verarbeitung und Strukturierung von komplexen Informationen angesiedelt. Die Ausgabe modularen Contents in verschiedene Ausgabemedien, das sogenannte Cross-Media-Publishing, ist mit dem Einsatz von Content-Management-Systemen zur Routine geworden und methodisch gut untersucht. Bisherige Ansätze basieren allerdings auf dem Prinzip des Single-Source-Publishing; der Umgang mit einer steigenden Anzahl heterogener und oft unzureichend strukturierter Informationsquellen ist neu.

Für neuartige Informationsprodukte wie Mobile Apps oder Portale werden nunmehr aus unterschiedlichen Abteilungen eines Unternehmens Inhalte zugesteuert, die von Marketing-Videos über Konstruktionsmodellen bis hin zu Programmteilen reichen können. Die Branche reagiert auf diese Bewegung mit der Entwicklung von sogenannten Content-Delivery-Portalen, die Inhalte verschiedener Art aggregieren und webbasiert bereitstellen können; ein Ansatz, der auch als Multi-Source-Publishing bezeichnet werden kann. Nicht betrachtet wird bisher jedoch die Kombination der beiden Ansätze zu einem integrierten und offenen Konzept, das mehrere Quellen aufnehmen, sie verarbeiten und anschließend in verschiedene Zielmedien publizieren kann.

Die Lösung zur Umsetzung einer solchen Multi-Source-/Multi-Target-Plattform können Komponentensysteme sein, deren konkrete Zusammenstellung sich dynamisch anpassen kann. Die Wiederverwendung von in sich abgeschlossenen Einheiten auf Inhaltsebene ist bereits üblich. Deshalb liegt es nahe, ein ähnliches Konzept auch auf Ebene der Inhaltsverarbeitung anzuwenden.

Konzept



Ein Publikationsprozess mit verschiedenen Komponenten (Bilddatei: abb1.pdf)

Publikationsphasen

Das Konzept teilt den Publikationsprozess in drei Phasen ein: Eingabe, Verarbeitung und Ausgabe. In jede dieser Phasen können Komponenten eingesetzt werden, um ein bestimmtes Publikations-Szenario abzubilden. Für jede Quelle wird eine Eingabekomponente bestimmt, die das Quellformat nach bestimmten Regeln in die intern verwendete Informationsstruktur überführt. Nachdem dies geschehen ist, werden alle Datenquellen zusammengeführt und können durch Komponenten in der Verarbeitungsphase manipuliert werden. In der Ausgabephase wird die interne Informationsstruktur wieder in die einzelnen Zielformate exportiert und der Publikationsprozess wird abgeschlossen. Die Informationen „fließen“ dabei durch die Komponenten und werden durch das Publikationssystem automatisch weitergereicht (daher auch der Name „infoflow“).

Die Konstellation der Komponenten in den einzelnen Phasen und die Zuordnung von Quelldaten zu Eingabekomponenten wird in sogenannten Publikationsdefinitionen festgehalten. Diese werden als XML-Dateien entweder vom Web-Client generiert oder von Hand erstellt.

Komponenten

Komponenten sind Verzeichnisse und enthalten eine Manifest-Datei sowie weitere Dateien, die zur Ausführung benötigt werden (z.B. Stylesheets oder Binaries). Das XML-Manifest beschreibt die Komponente und enthält neben Metadaten sowie einer Schnittstellen- und Parameterbeschreibung auch den Prozess, der abgebildet werden soll. Dieser wird aus sequentiellen Schritten aufgebaut, die in zwei Ausprägungen auftreten können: Als direkt eingebetteter Programmcode oder als Referenz auf eine andere Komponente. Referenzen werden bei der Ausführung aufgelöst.

Neben Komponententypen für jede der drei Publikationsphasen existieren auch sogenannte System- und Basiskomponenten. Diese bilden grundlegende Funktionen ab (z.B. kopieren oder transformieren) und können von anderen Komponenten referenziert werden. Nur Basiskomponenten dürfen Schritte verwenden, die direkt Programmcode einbetten.

Alle Komponenten, die in einem System zur Verfügung stehen werden in einer XML-Verzeichnisdatei, dem Repository, registriert. Dieses dient als Grundlage für die Zusammenstellung einer Publikationsdefinition und als Funktionsübersicht des Systems.

Funktionsweise

Bei der Publikation wird zunächst die Publikationsdefinition in einen ausführbaren Ablaufplan umgewandelt („Assemblierung“). Dieser steuert die Umwandlung der Komponenten in ausführbaren Code („Kompilierung“) und die Ausführung mit den entsprechenden Daten zum richtigen Zeitpunkt.

Da die Umwandlung in Programmcode „on-the-fly“ erfolgt und durch die Konvention, dass nur Basiskomponenten Programmcode einbetten können, sind Publikationskomponenten (Eingabe, Verarbeitung, Ausgabe) unabhängig von einer konkreten Implementierung, da sie nur aus Referenzen zu anderen Komponenten bestehen und damit übergeordnete Konzepte und keinen fest definierten Programmcode abbilden. Das fördert Wiederverwendung und technische Unabhängigkeit.

Technische Umsetzung

Architektur

Die Architektur der Publikationsplattform gliedert sich in drei Schichten: Publikationsserver, Web-API und Web-Client. Dies ermöglicht einen skalierbaren Aufbau des Systems und einen Betrieb von verschiedenen Zusammenstellungen der Schichten (z.B. ein Austausch des Clients oder ein alleiniger Betrieb des Publikationsservers).

Der Publikationsserver basiert auf XML-Technologien und wandelt Publikationsdefinitionen und Komponenten in Programmcode um, der von Apache Ant ausgeführt werden kann. Die Web-API basiert auf Node.js und bietet eine HTTP-basierte REST-Schnittstelle. Der Web-Client ist eine AngularJS-Anwendung, die mit der API über das Austauschformat JSON kommuniziert.

Interne Informationsstruktur

Als interne Informationsstruktur kommt eine restriktive Variante von HTML5 zum Einsatz. Dabei wird die Struktur von Dokumenten und Modulen in Form von geschachtelten „sections“ abgebildet. Diese können (neben anderen Elementen) mit Metadaten erweitert werden (RDFa, Microdata) und somit semantische Eigenschaften von Informationsmodellen übernehmen.

Zusammenfassung

In einer ersten prototypischen Umsetzung konnten erfolgreich Komponenten für Eingabe (DITA, HTML, PDF, Word, Excel, Markdown, Video), Verarbeitung (Zusammenführung, Restrukturierung, Indexierung, Validierung, Referenzauflösung) und Ausgabe (PDF, Latex, Markdown, Website, Web-Apps, Hybride Apps, arvato CIM-Portal) erstellt werden. Diese können in beliebigen Konstellationen mit Hilfe eines Web-Clients kombiniert und ausgeführt werden. Der Publikationsserver kann lokal, z.B. auf einem Laptop, oder als Cloud-Anwendung in einem Rechenzentrum betrieben werden.

für Rückfragen:
jan.oevermann@googlemail.com