



Allgemeine Hinweise zur Datenmigration aus Docuware Dokumentenmanagement

Verfasst von Josep Llord am 4. Januar 2018

In diesem Artikel beschreiben wir die **Migration zwischen DocuWare und OpenKM**. In einer früheren Veröffentlichung habe ich bereits einen Migrationsprozess beschrieben, in diesem Fall zwischen KnowledgeTree und OpenKM. DocuWare ist ein Dokumentenverwaltungssystem, das unter vielen anderen Unterschieden bezüglich OpenKM keine Open-Source-Version hat. Es handelt sich also um eine Anwendung, die unter einer kommerziellen Lizenz angeboten wird. DocuWare GmbH ist der deutsche Hersteller dieser Dokumentenmanagement-Lösung. Als Nächstes werde ich kurz die Hauptmerkmale der DocuWare-Dokumentenverwaltungssoftware überprüfen und sie mit der Enterprise-Content-Management-Lösung von OpenKM vergleichen.

Zunächst möchte ich betonen, dass DocuWare eine Lösung ist, die nur in einem Microsoft Windows-Betriebssystem installiert ist. Das heißt, dass es für seine Architektur eine native Lösung des Microsoft-Ökosystems ist. Im Gegensatz zum **OpenKM DMS**, dessen Architektur auf JAVA basiert und damit Plattformunabhängigkeit unterstützt. Ich denke, wir können es in OpenKM auf die Mehrheit der Hersteller von Dokumentenmanagement-Lösungen, Records Management und Enterprise Content Management erweitern. Wir weisen unsere Kunden darauf hin, wenn möglich Linux als Betriebssystem zu verwenden. Wir empfehlen Ubuntu, Debian, Centos und Red Hat. Der Grund ist einfach; Mit der gleichen Hardware bietet Linux immer eine höhere Leistung als andere Betriebssysteme, da das Eingabe- (E / A-) System effizienter ist. Außerdem werden wir feststellen, dass ein Server mit Windows eine natürliche Tendenz hat, mindestens 2 GB Arbeitsspeicher zu verbrauchen, während in Linux der Ressourcenverbrauch des Betriebssystems standardmäßig sehr viel geringer ist. Wenn wir uns in einem Szenario befinden, in dem "Leistung" ein kritischer Faktor ist, sollten wir immer die Möglichkeit in Betracht ziehen, Linux anstelle von Windows zu installieren. Damit bestreiten wir nicht die Möglichkeit, unter Windows Hochleistungsumgebungen zu installieren. Wir möchten hier natürlich objektiv bleiben und gehen hier von Standard-Konfigurationen aus.

Ein weiterer Unterschied ist, wie bereits erwähnt, die Art der **Lizenz**. Während das OpenKM-Managementsystem über eine doppelte Lizenz ("kommerziell" und "Open Source") verfügt, wird DocuWare nur unter einem einzigen Modell ("kommerziell") angeboten. In Bezug auf die Datenbanken können wir überprüfen, ob sowohl **DocuWare Document Management Solution** als auch OpenKM für die Verwendung mit **Microsoft SQL Server, Oracle und MySQL Server** konfiguriert werden können. Im Falle von OpenKM kann auch **PostgreSQL** verwendet werden. Bei Datenbanken sehen wir, dass sowohl OpenKM als auch DocuWare oder die meisten Hersteller von Content-Management-Lösungen eine dieser vier relationalen Datenbanken wählen. Im Allgemeinen finden wir Benutzer, die die Microsoft SQL Server-Datenbank hauptsächlich in Windows Server-Umgebungen verwenden



möchten. Datenbankbenutzer von PostgreSQL, Oracle, MySQL Server und **MariaDB** finden wir in Linux-basierten Umgebungen.

Unterteilen Sie die Repositories des Dokumentenmanagements in zwei große Gruppen: solche wie OpenKM, die eine **Taxonomie** verwenden, und solche, die dieses Paradigma nicht verwenden. Das Wort Taxonomie hat seinen Ursprung in der Wissenschaft, insbesondere in der Biologie, wo ein Mechanismus benötigt wird, um Tier- und Pflanzengruppen zu hierarchisieren und zu systematisieren. Taxonomie ist ein System, das die Klassifizierung oder Anordnung in Gruppen von Dingen ermöglicht, die gemeinsame Merkmale aufweisen. **Dokumentenverwaltungsanwendungen** wie **OpenKM**, **Nuxeo** oder **Alfresco** verwenden das Taxonomiekonzept zur Klassifizierung von Informationen. Kurz gesagt, die Taxonomie in der Volkssprache ist eine Hierarchie von Ordnern, in der Dokumente sortiert und klassifiziert werden können. Andere Dokumentenverwaltungssysteme wie **Documentum** (jetzt **ECM2**) und **OnBase**, zu denen auch DocuWare gehört, verwenden die Katalogisierung, die auf dem **Konzept von "Boxen"** basiert. Im Falle von DocuWare wären die **"Cabinets"** der richtige Name. Ein "Kabinett" wäre in der realen Welt ein Schrank oder eine Schublade, in der ein einziger Dokumenttyp (eine Dokumentenserie) gespeichert ist. Bei diesen Dokumentenverwaltungssystemen werden wir feststellen, dass für jede Art von Dokument eine separate "Box" ("Kabinett") vorhanden ist, mit den entsprechenden Metadaten für den Dokumentartyp, der jeder Box zugewiesen ist.

Beispiel: Ein Unternehmen, das Rechnungen und Verträge speichert. Bei einer **Dokumentenverwaltungslösung** wie OpenKM könnten wir durch eine Verzeichnisstruktur navigieren, um die Informationen zu finden:

```
/ okm: root / Abteilungsmanagement / Verträge / 2019  
/ okm: root / Abteilungsmanagement / Verträge / 2018  
/ okm: root / Abteilungsverwaltung / Rechnungen / 2019  
/ okm: root / Abteilungsverwaltung / Rechnungen / 2018
```

Im Falle von DocuWare, Documentum oder Onbase müssen wir zunächst den Dokumenttyp auswählen, um auf einen Listenbildschirm zuzugreifen, der es uns ermöglicht, die Suche zu verfeinern. Es gibt keine Navigation selbst, aber der Zugriff auf die Dokumentation setzt voraus, dass der Typ des Dokuments, auf das wir zugreifen möchten, bekannt ist. Die ersten **Dokumentenmanagement-Lösungen** wie Documentum und OnBase erschienen unter diesem Paradigma, wahrscheinlich beeinflusst durch die Verlagerung der realen Organisationsstruktur auf Computerlösungen ohne Änderungen. In der realen Welt werden die Informationen nach Art der Dokumentation in den Regalen und in jedem Regal archiviert. Inspiriert von dieser Lösung bot die erste Anwendung in einer digitalen Umgebung dasselbe konzeptionelle Format. In der Folge haben sich die modernsten Anwendungen für Dokumentenmanagement, Content Management und Enterprise Content Management für ein Modell entschieden, das auf der Klassifizierung von Informationen innerhalb einer Taxonomie basiert. Beide Modelle haben meiner Meinung nach Vor- und einige Nachteile. Bei Lösungen, die auf Taxonomien basieren und über die Nutzbarkeit der Benutzer hinausgehen, besteht



der wichtigste Punkt darin, die Isolation der Informationen, die die Kabinette beinhalten, zu durchbrechen. Das heißt; Wenn wir nicht nur einem Problem des Dokumentenmanagements gegenüberstehen, sondern einem Problem des **Datensatzmanagements** (Fallmanagement oder Geschäftsfall), bei dem verschiedene Arten von Dokumenten einen miteinander verknüpften Lebenszyklus haben, wird der Lösungsansatz, der auf den Ergebnissen von "Cabinets" basiert, problematisch.

Es gibt ein übliches Szenario, in dem der Dokumentenmanager kein einfacher Beweisbehälter für das Unternehmen ist, sondern über einen Workflow verfügt, der den **Lebenszyklus der Dokumente** steuert. Gleichzeitig sind Dateien in Bearbeitung, an denen mehrere Dokumente beteiligt sind (**Geschäftsfall**). In diesem Szenario ist die auf "Cabinets" basierende Lösung aus meiner Sicht nicht die Optimalste. Im Allgemeinen gehen Unternehmen immer mehr über einen einfachen Container von Dokumenten hinaus und wollen Informationen aktiver steuern. Dort werden sowohl die **Workflows** als auch der Dateiplan (Plan und Anordnung der Dateien) angezeigt. Das Problem ist nicht mehr nur Speicherung, sondern Kontrolle der Gültigkeit, Änderungen sowie der Verteilung und des Zugriffs auf Informationen, wenn dies erforderlich ist. Und das alles ist aktiv in die Tätigkeit des Unternehmens integriert (hier finden sich typische Fälle für die Integration in **CRM** und **ERP** ua). Dokumentations-Management-Anwendungen werden vom reinen Container zum aktiven Teil des Software-Ökosystems, mit dem das Unternehmen arbeitet. Weitere Informationen finden Sie im DocuWare Knowledge Center.

Im Falle von DocuWare noch etwas aus der Praxis. Wir haben mehrere Migrationen zu OpenKM durchgeführt. In dem Moment, in dem ich diesen Artikel schreibe, haben wir insbesondere Migrationen von Version 6.x und Version 7.x durchgeführt. Wie üblich haben wir keine Informationen vom Hersteller gefunden, die ihren Kunden helfen sollen, ihre Daten aus ihrem Repository zu extrahieren. Dieser Punkt wird wie bei den Migrationen wiederholt, die wir mit anderen Dokumentenmanagementlösungen durchgeführt haben. Der Client ist in der Computerlösung eingesperrt. Deshalb möchte ich von hier aus erneut Berufung einlegen, wie ich es in dem Artikel über die Migration von **KnowledgeTree** zu OpenKM getan habe. Aus meiner Sicht konzentrieren sich die zukünftigen Benutzer dieser Tools beim Erwerb einer **Dokumentenverwaltungssoftware** auf die Funktionalitäten der Tools. Die Technologie, auf der sie basieren, sowie die Kosten. Abgesehen von der Bewertung eines Themas, das ich für den Schlüssel halten würde; wie Sie unsere Daten in Zukunft aus diesem System herausholen können. Der Fall von DocuWare ist zumindest konzeptionell sehr ähnlich zu OnBase. Für jedes "Kabinett", das heißt für jeden Dokumenttyp, wird ein Satz spezifischer Tabellen erstellt, in denen die Informationen separat gespeichert werden. Das heißt, wenn wir 50 Arten von Dokumenten haben, generiert die Anwendung mindestens 50 Tabellen, von denen jede die Informationen und Metadaten der einzelnen Arten von Dokumenten enthält. Bei DocuWare ist es auch von Bedeutung, wie Daten im Dateisystem gespeichert werden. Es besteht die Möglichkeit, dass jeder Dokumenttyp an einem anderen Ort gespeichert wird, d. H. Binäre Dokumente können je nach Typ in unterschiedlichen Einheiten gespeichert werden. Diese Flexibilität in der Konfiguration kann den Ablauf des **Migrationsprozesses** verkomplizieren, da abhängig von der Art des Dokuments berücksichtigt werden muss, wo sich die Informationen physisch auf der Festplatte befinden. Eine



weitere Kuriosität von DocuWare ist, wie Dateien im Dateisystem gespeichert werden. Zum Beispiel wird eine PDF-Datei mit 32 Seiten in 32 Dateien auf jeweils einer Seite separat gefunden. Wenn wir eine PDF-Datei hochladen (abhängig von der Konfiguration von Docuware), wird diese in Form von 32 Dokumenten mit jeweils einer Seite verarbeitet und gespeichert. In Bezug auf die Art der Speicherung von Dokumenten haben wir Abweichungen festgestellt, je nachdem, ob die Version 6.x oder 7.x war und auch je nach Art des PDF-Dokuments oder mehrseitiger TIFF-Dateien.

Ein weiteres Problem ist die Art und Weise, in der **Metadaten** gespeichert werden, insbesondere die des Datumstyps. Dies ist ein klassisches Problem, das wir in praktisch allen Migrationsprozessen anderer Dokumentenverwaltungsanwendungen gefunden haben und das besondere Aufmerksamkeit erfordert, um den Wert des Datums im entsprechenden Format zu erfassen und dann in OpenKM im Format ISO8601 zu speichern. Die letzte Kuriosität von DocuWare besteht schließlich darin, dass das Festplatten-Repository für jedes Dokument eine Datei mit allen Informationen im Dokument enthält. Das heißt, alles deutet darauf hin, dass das lokale DocuWare-Repository als **Live-Export des Repositories** mit seinen Metadaten fungiert. Dies bedeutet, dass jede Änderung, die wir an der Anwendung in einem Dokument vornehmen, sowohl auf Datenbankebene als auch auf diesen lokalen Dateien Auswirkungen hat. Dies hat den Vorteil, dass wir im Repository bereits eine Sicherung haben (heißen **Export**); Gleichzeitig duplizieren wir jedoch alle Daten (Datenbank und Dateisystem) mit dem entsprechenden Hardware-Verbrauch, der all dies impliziert. Dies bedeutet auch eine komplexe Logik auf der Anwendungsseite, um diese Daten regelmäßig zu synchronisieren.

Als die R & D - Abteilung von OpenKM die Reverse Engineering - Analyse zur Durchführung der Datenmigration durchführte, war es für uns viel bequemer, den Migrationsprozess mit einem JAVA - Skript durchzuführen, das in Kombination mit der Datenbasis und der Dateisystem, das es über das JAVA SDK mit OpenKM verbindet. Wir verwerfen die Eingabe und verarbeiten die Textdateien, die die DocuWare-Metadatenstruktur enthalten, da das Format, in dem diese Informationen gespeichert sind, im Gegensatz zu OpenKM nicht sichtbar war. Wenn wir die Struktur von **Metadaten** in Dateien exportieren, machen wir dies mit Dateien im **JSON-Format**, was die nachfolgende Verarbeitung erheblich erleichtert. Von der Forschungs- und Entwicklungsabteilung untersuchen wir auch die Unterstützung von **CMIS** in DocuWare. Obwohl wir uns nicht als CMIS-Fans deklarieren, ist dies eine Option, die zu berücksichtigen ist, und zum Zeitpunkt der Erstellung dieses Artikels ist dieser Artikel nicht verfügbar. Wir haben einige Beispiele für die Verbindung über .NET mit der **DocuWare-API** gefunden, wie wir in diesem Beispiel der DocuWare-API sehen können. Es ist ein Pfad, den wir einmal verworfen haben, teilweise aufgrund der gesamten **Microsoft-Entwicklungsumgebung**, die zusammengestellt werden muss, die wir jedoch als Option betrachten, die für alle diejenigen in Betracht zu ziehen ist, die ihr Repository migrieren möchten. Alles deutet darauf hin, dass das DocuWare Platform API-Paket eine Reihe von Bibliotheken umfasst (ähnlich wie in OpenKM unter dem Namen **SDK für JAVA, .NET und PHP**), die den problemlosen Zugriff auf alle Funktionen des Repositories über die **Webservices ermöglichen in REST**.

Wenn möglich, und wenn wir eine API haben, über die ein Repository abgegriffen werden kann, ist es generell ratsam, es als ersten Ansatz zur Lösung des Problems zu verwenden. Wir konnten keine



Dokumentation online finden. In OpenKM haben wir uns wie andere Hersteller für die Option entschieden, dass die REST-Servicedokumentation selbst über **Swagger.io** verfügbar ist.

Ich hoffe, dass dieser Artikel als Einführung für alle Benutzer dienen kann, die in Zukunft ihre Daten von DocuWare migrieren möchten. Und die Erfahrung, die wir hier zu teilen versuchen, wird in gewisser Weise die Lösung des Problems erleichtern. Der Prozess der Migration der DocuWare-Repositories, die wir abgeschlossen haben, war sehr ähnlich, jedoch nicht derselbe. Daher können wir kein magisches Rezept bereitstellen, das in allen Fällen nützlich ist. In diesem Artikel habe ich versucht, das "**allgemeine**" **Wissen**, das ich als wiederverwendbar erachte, in allen Szenarien zu teilen, in denen wir uns zumindest begegnet sind.

Aus dem englischen übersetzte Quelle: <https://www.openkm.com/blog/migration-from-docuware.html>