



## EasyFlow® for Open Source\* (EF-OS)

Joachim Müller

**Der Markt für Open Source Projekte und das steigende Interesse an Workflow Management Systemen (WfMS) haben dazu geführt, dass sich die TOPAS-IT GmbH entschlossen hat, sich federführend an einem Open Source Projekt zu beteiligen.**

War die Workflow-Engine *EasyFlow*® in der Version 2.6 noch hundertprozentig in einer relationalen Datenbank integriert und von dem Datenbank Produkt Oracle abhängig, so hat sich das mit **EF-OS** geändert. Es wurde auf Basis der über 13-jährigen Erfahrung damit begonnen, eine Architektur zu entwickeln, die folgende **fachliche Anforderungen** erfüllt:

- Genehmigungsrestriktionen („Vier-Augen-Prinzip“) bereits während der Modellierung definierbar
- Standarddialoge für eine effektive Prozesserstellung
- Checklisten-Workflow-Funktionalität für eine prototypische Realisierung bei gleichzeitiger produktiver Einsatzmöglichkeit

- Vererbungs- und Kopierfähigkeit von Vorgängen während der Modellierung und Optimierung.
- Grafische Modellierung über Poseidon (UML), ARIS (eEPK) oder VISIO-Plugin (z.B. [www.EITAM.BIZ](http://www.EITAM.BIZ) oder [www.ITPEARLS.COM](http://www.ITPEARLS.COM))
- Vergabe von Fristen für den Gesamtprozess und Prozesseile. Diese Frist kann relativ (Tage/ Stunden) oder absolut sein
- Automatische Eskalationsmöglichkeiten bei Fristablauf bzw. Eintritt bestimmter Ereignisse
- Ereignisbasierte Ablaufsteuerung
- Anwenderspezifische Wiedervorlage.
- Bereits ausgeführte Schritte können vom Anwender rückgängig gemacht werden. Dabei wird per Definition festgelegt, ob der jeweilige Schritt wieder an die Rolle oder an den letzten Bearbeiter zurück gehen soll.
- Versionierung der definierten Prozesse, d.h. es sind zwei Möglichkeiten wählbar.
  - ein gestarteter Fall muss in seiner alten Form beendet werden, auch wenn der Vorgang in seinem Ablauf zwischenzeitlich geändert worden ist;
  - ein gestarteter Fall muss in der neu definierten Version zu Ende geführt werden. Dies muss entweder durch den Administrator und/oder Anwender entschieden werden.
- Man kann einen Prozessschritt „parken“ (auf „Behalten“ legen), d.h. die Wf-Engine merkt sich beim Verlassen des Schrittes den momentanen Zustand der Aktivitäten.
- Die Prozesshistorie wird automatisch erstellt und ist jederzeit zugreifbar
- Es stehen vier Umgebungen zur Verfügung.
  - Definitionsumgebung
  - Testumgebung
  - Produktivumgebung
  - Historienumgebung
- Es ist eine Weiterleitung des Prozesses bzw. des Schrittes innerhalb der selben Rolle/Gruppe möglich

\* s. Abschnitt Lizenzmodell

- Es gibt eine zeitlich unbegrenzte und begrenzte Vertreterregelung
- Man kann die Bearbeitungsreihenfolge von Aktivitäten in einem Schritt per Definition bestimmen
- Aktivitäten können als Pflichtaktivitäten definiert werden, Unterstützung unterschiedlicher Aktivitätstypen
- Aktivitäteninhalte sind vorgangsglobal, Aktivitäten können je nach Position im Workflow schritt-spezifisch als RW-, WORM- oder RO-Aktivitäten definiert werden
- Es können zu jedem Vorgang / Schritt Notizen hinterlegt werden. Diese Notizen werden chronologisch aufgelistet und sind für jedermann sichtbar. Notizen können auch nach dem Ende eines Vorgangs noch hinterlegt werden
- Es kann ein 24h-7Tage-Betrieb realisiert werden.
- Es gibt die Möglichkeit, einen Fall auszuchecken (Offline-Bearbeitung) inkl. aller Schritte, die für diesen Bearbeiter sind.
- Eine automatische Lastverteilung der Prozessschritte bzw. Fälle ist nach Wertebereichen möglich (z.B. PLZ, Umsatz, Genehmigungsgrenzen usw.)
- Es steht ein Administrations-Tool für die Überwachung der Wf-Engine und der Prozesse zur Verfügung
- Unterstützung einer vollautomatisierten Prozessbearbeitung („Prozessroboter“)
- Kumulierte Erfassung der Bearbeitungszeiten zur Unterstützung der kontinuierlichen Verbesserungsprozesse
- Mandantenfähige Workflow-Engine
- Verwendung von individuellen und Gruppen-Tätigkeitslisten inkl. unterschiedlicher Darstellungsformen

Die oben aufgeführten fachlichen Anforderungen stehen in Verbindung mit den **technischen Features**:

- Eine WfMC-konforme Architektur (alle Interfaces bis auf Interface 5)
- 100%ige J2EE-Architektur
- Standard-Browser-Client
- Gewährleistung der Sicherheit durch

Zertifikate und Verschlüsselung

- Skalierbarkeit auf Applikationsebene (verteilbar über mehrere Server)
- Datenbank-Unabhängigkeit
- Workflow-Datenhaltung in einer relationalen Datenbank
- Definierte Schnittstellen (J2EE-API) für eine „saubere“ Anbindung von Fremdprodukten (DMS, ERP, usw.)
- Mögliche Anbindung der Engine über WebServices
- Anbindung einer eigenen oder einer integrierbaren User / Rollen / Rechte-Verwaltung über LDAP
- Unterstützung des XPDL-Formates (WFMC) für Interface 2. Somit ist es möglich, dass Vorgangsdefinitionen, welche durch ein etabliertes grafisches Modellierungstools erstellt wurden (wie z.B. Poseidon, Bonapard oder Adonis) über die XPDL-Schnittstelle(WfMC) importiert werden können.
- Datenaustausch zwischen Workflow-Engines:

	<b>Prozessdefinitionen</b>	<b>Laufzeitdaten</b>
Innerhalb von EFOS	auf Tabellenebene	auf Tabellenebene
Mit anderen Wf-Produkten	XPDL-Format	erweitertes XPDL-Format

## Architektur

In der Architektur sind die Vorteile von **EasyFlow®** und die der WfMC-Architektur mit den Anforderungen an eine zukunftssichere Architektur verknüpft worden.



Zeichnung 1: Architektur der Wf-Engine

Dies bedeutet, dass neben den sichtbaren Features wie z.B. Web-Oberfläche und Administrations-Client auch die „unsichtbaren“ Features wie J2EE-Standard und eine modulare Struktur realisiert wurden.

### Verwendete Komponenten bzw. Projekte

Letztendlich weist **EF-OS** neben der spezifischen Betriebssystemunabhängigkeit auch eine allgemeine Produktunabhängigkeit auf. Dies betrifft insbesondere die Datenbank, den Applikationsserver und den HTTP-Server. Im Rahmen des Develop-Frameworks wurde auch großer Wert auf die Verwendung von Open Source Projekten gelegt, die eine effektive und kostengünstige Entwicklung zulassen.

*Betriebssystem* - Da eines der Features die Betriebssystemunabhängigkeit von **EF-OS** ist, ist es dem Kunden freigestellt, ob ein Betriebssystem von Microsoft (Windows XP, Windows 2000) oder das Betriebssystem Linux (z.B. SuSE Distribution) verwendet wird.

*Datenbank* - Aufgrund der Datenbank-technischen Vorgaben hat man sich für die Entwicklung für die PostgreSQL-Datenbank in der Version 8.01 für Windows entschieden.

*Applikationsserver* - Als Applikation Server bietet sich jBoss in der Version 4.0 an. Dieses Open-Source Projekt ist sehr gut dokumentiert, erfüllt die J2EE-Spezifikation 1.4 und erfreut sich großer Akzeptanz (2004 hatte es einen Marktanteil von über 30%).

*HTTP-Server* - Für die schnelle Realisierung des Standard-Client wurde der Apache Web-Server 2.0 mit PHP-Modul gewählt. Später ist für performance-kritische Anwendungen ein Servlet-Container (Tomcat in jBoss) geplant.

Im Rahmen der Entwicklung wurden folgende Frameworks und Design Patterns zur Unterstützung genommen:

- MVC (Struts für Java-Implementierung, Mojavi – für PHP)
- iBatis – DB-Layer
- Log4J bzw. Log4php
- SessionFacade, BusinessDelegate, ValueObjects, DataAccessObjects u. a.

### Lizenzmodell

Zu beachten ist, dass Open Source Software in der Definition nach der OSI ([www.opensource.org](http://www.opensource.org)) allen offen steht. Dadurch kann weder ein einzelner Programmierer noch ein Unternehmen die Richtung der Entwicklungen vorgeben.

Aufgrund der oben aufgeführten Freiheiten auf der einen Seite und Einschränkungen für die Verwendungen von Open Source Software für den kommerziellen Gebrauch auf der anderen Seite wurde besonders auf die Lizenzmodelle der verwendeten Klassenbibliotheken geachtet.

EFOS wird in die folgenden Module aufgeteilt:

- Core Workflow-Engine
- Enterprise Workflow-Engine
- Erweiterungen, welche die Dialoge wie Standard-Client, Administrator-Client usw. enthalten
- Modellierungswerkzeug
- Lösungen für den Kunden (Prozesse mit den kundenspezifischen Dialogen)

Diese Module unterliegen unterschiedlichen Lizenzen. Als Basis dient die Core Workflow-Engine, die unter der LGPL steht. Die Core Workflow-Engine und deren Sourcecode wird frei verfügbar sein. Alle weiteren Module werden als proprietäre Lösungen angeboten.

Daraus folgt, dass die Core Workflow-Engine selber ein Open Source Projekt ist und das WfMS eine proprietäre Anwendung.

Mit dem Erwerb des WfMS erhält man den Sourcecode und kann diesen im Rahmen der Lizenzvereinbarungen verwenden. Dies macht **EF-OS** besonders für Systemhäuser mit einer eigenen Software-Entwicklungsabteilung und Produkthersteller, die ihr Produkt durch ein 100%iges Workflow Management System ergänzen wollen, interessant.

### Fazit

**EF-OS** erfüllt alle Anforderungen eines zukunftsicheren WfMS. Durch die Architektur, das Lizenzmodell und der bestehenden Projekterfahrung ist **EF-OS** und das Team um **EF-OS** die ideale Lösung für die Einführung eines WfMS in Unternehmen, die wachsen und die Qualität der Geschäftsprozesse verbessern wollen sowie die Kosten ihrer IT nicht nur optimieren, sondern überhaupt einen Einblick erhalten wollen. Die oben genannten Features von **EF-OS** sind aber auch eine ideale Ergänzung, um z.B. das eigene ERP-Produkt oder ein Lagerverwaltungsprodukt gegenüber der

Konkurrenz besser positionieren zu können.

Die TOPAS-IT GmbH und der Autor haben sich auf die Automatisierung der IT-Prozesse insbesondere nach dem ITIL-Referenzmodell spezialisiert und können für die Kunden auf diesem Gebiet auf eine besonders umfangreiche Projekterfahrung zurückgreifen. Dadurch ist gewährleistet, den Kunden in Projekten erfolgreich bei der Modellierung, Automatisierung und Reorganisation unterstützen zu können.

#### **Autor**

Joachim Müller beschäftigt seit 1995 mit WfMS sowohl in der Produktentwicklung als auch in der Umsetzung von Projekten. Er konnte in den letzten fünf Jahren umfangreiche Erfahrungen in großen IT-Projekten als Projektleiter sammeln. Diese Projekte hatten zum Ziel, die IT-Prozesse in großen Systemhäusern zu definieren, zu etablieren, ggf. zu automatisieren und zu optimieren. In den letzten drei großen Projekten kam das Prozessframework ITIL zum Einsatz. Die Erfahrungen sind nicht zu letzt in dem Buch „Workflow-based Integration“ (Springer, 2004) und dem zugehörigen Vorgehensmodell eingeflossen.



Joachim Müller ist der Projektleiter des Open Source Projektes und steht für weitere Fragen unter [buch@wf4you.de](mailto:buch@wf4you.de) zur Verfügung.

#### **TOPAS-IT GmbH**

Als kompetenter Lösungsanbieter mit inzwischen 13 Jahren Erfahrung sieht die TOPAS-IT GmbH ihre Aufgabe in einer umfassenden und qualitativ herausragenden IT-Serviceleistung. Den Kunden wird ein ganzheitliches Dienstleistungsspektrum von der Beratung bis hin zur Entwicklung und Betreuung von integrierten IT-Lösungen. Schwerpunkte bilden dabei die Bereiche Geschäftsprozessoptimierung, Workflow, Business Process Automation und Enterprise Process Management unter besonderer Berücksichtigung von ITIL.  
Kontakt: [info@topas-it.de](mailto:info@topas-it.de)

