



Embedded Workflow

Amazonas Workflow, embedded Version

Whitepaper

Dok.-Name	Embedded Workflow
Dok.-Nr.	OE_MAWP.70502
Version	02.06.2005
Autor(en)	Roman Zulauf

Birmensdorferstrasse 32
CH-8142 Uitikon-Waldegg ZH
Switzerland
Tel +41 (0) 44 400 47 00
Fax +41 (0) 44 400 47 07
E-Mail info@objeng.ch
<http://www.objeng.ch>

Copyright © 2005 by Object Engineering GmbH, Uitikon-Waldegg ZH

Ausgangslage

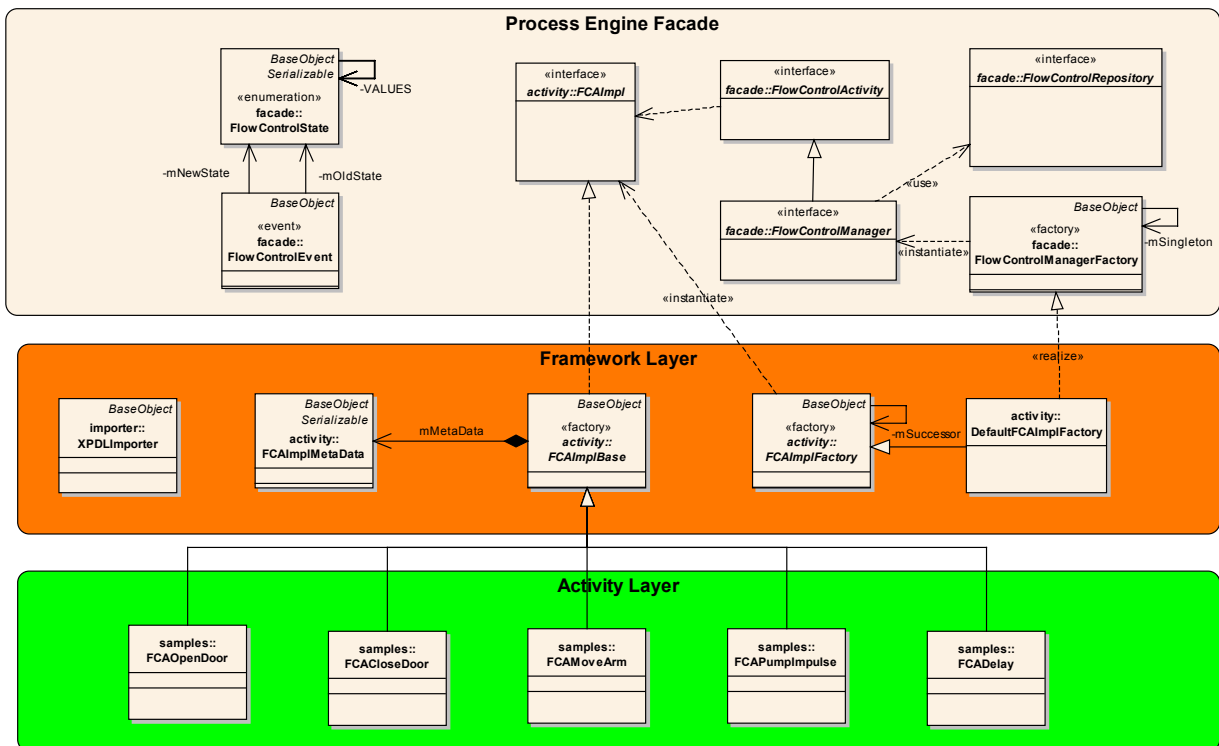
Workflow und Prozess-Management müssen sich nicht notwendigerweise nur auf die Business-Ebene beziehen.

Das Steuern von Abläufen ist auch für die Industrie bis auf die Ebene der Maschinensteuerung ein wichtiges Thema.

In einer Maschinenfabrik war die Ausgangslage die, dass durch ständige Weiter- und Neuentwicklungen von Maschinenmodellen verschiedene Maschinensteuerungen entwickelt wurden, die aber trotz den allgemeinen Ähnlichkeiten keine gemeinsame Architektur hatten. Dem sollte nun durch die Erstellung eines neuen, gemeinsamen Frameworks abgeholfen werden.

Wie wurde dies gelöst?

Bei der Ausarbeitung der Architektur des allgemeinen Frameworks wurde schnell klar, dass auch für das Steuern der Abläufe eine generische Komponente eingesetzt werden konnte. Für diese ergaben sich folgende Schichten:



In der Schicht *Activity Layer* wurde für jedes Gerät eine Zugriffsklasse implementiert, welche spezifisch auf das darunter liegende Gerät zugeschnitten war, die Implementations-Details aber nach oben durch eine einheitliche Schnittstelle abstrahierte. Dies ermöglichte die einfache Anbindung jedes Gerätes in die generische Ablaufsteuerung, wobei die Geräte zu Baukasten-Elementen gemacht wurden, welche dann im übergreifenden Ablauf beliebig zusammengestellt werden konnten.

Nutzen für den Anwender

Durch den Einsatz der Embedded Process Engine wurde der Programmieraufwand erheblich verringert. Statt dass die Logik einzelner Abläufe in einer Programmiersprache hart codiert wurde, konnte für jeden dieser Abläufe ein Prozess definiert werden. Diese Vorgehensweise reduzierte die Fehleranfälligkeit, da weniger Code geschrieben werden musste und die Steuerung des Ablaufs in einer gemeinsam genutzten Komponente abgehandelt wurde.

Ein weiterer Nutzen wurde dadurch erreicht, dass durch den Einsatz einer standardisierten Prozess-Definitions-Sprache Abläufe ohne Programmieraufwand angepasst werden konnten.

Technische Anforderungen

Eine Embedded Process Engine sollte in 100% Standard Java implementiert sein. Damit wäre gewährleistet, dass die Software auf allen gängigen Hardware-Plattformen und Betriebssystemen ohne Anpassung lauffähig wäre.

Die Grösse einer Embedded Process Engine inkl. allen benötigten Bibliotheken (ausser J2RE) sollte so klein wie möglich sein. Dies wird durch ein gutes Design und durch möglichst wenig Abhängigkeiten zu externen Bibliotheken erreicht. Durch diesen kleinen Footprint wäre sie auch für kleinere Systeme geeignet.

Eine Engine könnte sich stark an das Workflow Referenz-Modell der Workflow Management Coalition anlehnen. Prozesse würden somit in der standardisierten XML Definitions-Sprache XPDL erstellt. Personen die bereits mit den Konzepten und Begriffen der Workflow Management Coalition vertraut sind, wären sehr schnell auch mit der Embedded Process Engine vertraut.

Prozess-Design

Da für die Prozess-Definition ausschliesslich auf Standards aufgebaut würde, könnte jegliches Werkzeug verwendet werden, welches einen Export im XPDL-Format unterstützt. Es sind diverse Open-Source Designer erhältlich, wobei das Tool JaWE bevorzugt wird.

Prozess-Elemente

Folgende Elemente sollten innerhalb eines Prozesses unterstützt sein:

- Aufruf von selbst erstellten Prozess-Aktivitäten (Java Klassen)
- Prozess-Aufruf (Subprozess)
- Route (keine Aktion, wird für komplexe Aufteilungen verwendet)

Die Engine sollte sowohl AND- wie auch XOR-Aufteilungen und –Vereinigungen interpretieren. Diese Funktionalität würde das parallele Abarbeiten mehrerer Aktivitäten ermöglichen. Übergangs-Bedingungen bei Kanten würden in einer einfachen, typensicheren Ausdrucks-Sprache definiert.

Java API

Die Prozess Engine sollte dem Entwickler ein einfaches, intuitives API bieten. Über dieses API könnten u.a. folgende Funktionen aufgerufen werden:

- Auflistung aller Aktivitäts-Instanzen einer gestarteten Prozess-Instanz
- Starten, Suspendieren, Terminieren und Abbrechen von Aktivitäts- und Prozess-Instanzen

- Setzen und Lesen von Prozess-Variablen

Durch Sicherheitsstufen könnte gewährleistet werden, dass keine unerlaubten Aktivitäten ausgeführt werden.

Eine Ereignis-basierte Implementation würde erlauben, dass registrierte Listeners über jegliche Status-Änderungen innerhalb der laufenden Prozesse informiert werden könnten.

Java Aktivitäts-API

Zur Implementation und Integration eigener Prozess-Aktivitäten sollte eine einfache Java-Schnittstelle implementiert werden. Je nach Anforderungen könnte so von einer Basis-Klasse abgeleitet werden oder die Schnittstelle direkt implementiert werden.

Zwischen Prozess und einzelnen Prozess-Aktivitäten sollten über Parameter Prozess-Variablen in den verschiedenen Modi (IN-, OUT-, INOUT-Mode) ausgetauscht werden können. Zusätzlich sollte jede Prozess-Aktivität direkten Zugriff auf den Variablen-Pool des Prozesses haben, was eine grosse Flexibilität für die Implementierung dieser Aktivitäten bieten würde.

Fazit

Eine Embedded Process Engine bietet als einfach integrierbare Komponente allgemeine Workflow-Funktionalität, die manche Applikation bereichern kann.

Durch ihre Flexibilität und Einfachheit eignet sich eine solche Engine für den Einsatz in Web-Applikationen bis zur Maschinensteuerung.
