****Future Technology

Jangaroo  
Enterprise-JavaScript-Anwendungen  
in ActionScript 3 entwickeln

## Zusammenfassung

Will man heute eine umfangreiche Enterprise-Web-Anwendung entwickeln, ist programming in the large mit JavaScript erforderlich. JavaScript fehlen aber Sprachkonzepte und Werkzeuge, die Schnittstellen, Modularisierung und Qualitätssicherung gezielt unterstützen. Spätestens in Version 3 muss ActionScript™ als die bessere und konsequente Weiterentwicklung von JavaScript für diesen Einsatzbereich gelten. Das Open-Source-Projekt Jangaroo [1] stellt einen Compiler zur Verfügung, der ActionScript-3-Code in JavaScript-Code transformiert, so dass dieser im Browser ohne Plug-In ausgeführt werden kann. Durch weitere Open-Source-Werkzeuge wird ein professioneller, praxisorientierter und modularer Entwicklungsprozess unterstützt. Insgesamt wird damit ein zukunftsweisender Weg angeboten Enterprise-Web-Anwendungen zu entwickeln.

## Programmiersprachen im Browser

JavaScript ist *die* client-seitige Programmiersprache des Webs. Um andere Sprachen im Browser einzusetzen, wird entweder ein Browser-Plug-In benötigt oder sie müssen zuvor in JavaScript übersetzt werden.

## Browser Plug-Ins

Ausführungsumgebungen für programming-in-the-large-geeignete Programmiersprachen wie Java (JRE), C# (SilverlightTM) und Adobe ActionScript (FlashTM) müssen als plattformspezifische Browser-Plug-Ins installiert werden. Alle genannten Plug-Ins übernehmen gleichzeitig die Darstellung (rendering), erschweren damit Installation und nahtlose HTML-Integration und sind weniger portabel.

## JavaScript Code-Generierung

Eine Alternative stellen Compiler dar, die JavaScript aus einer anderen Sprache generieren. Der bekannteste Vertreter dieser Kategorie ist das Google Web Toolkit (GWT), das Java nach JavaScript übersetzt, aber auch weniger bekannte Sprachen wie haXe können nach JavaScript kompiliert werden. Der generierte Code ist üblicherweise weder lesbar noch kann man in ihm ohne weiteres den Quellcode wiedererkennen. Dies erschwert das Debugging vor allem dann, wenn Fehler nur in einer bestimmten Ablaufumgebung auftreten. Weiterhin bleiben wertvolle JavaScript-Spracheigenschaften wie der Funktionsabschluss (closure) und die elegante Objektnotation (JSON) bei diesem Einsatzszenario oft ungenutzt.

## Jangaroo = Enterprise JavaScript ActionScript 3

Statt JavaScript zu einer Maschinensprache zu degradieren, sollte es um die für programming in the large benötigten Sprach-Features aufgewertet werden. Dieses Ziel hat auch Adobe mit ActionScript 3 (AS3) [2] verfolgt, nur dass mit dem Flash Player dafür eine eigene Ausführungsumgebung geschaffen wurde. Der Jangaroo-Compiler dagegen übersetzt eine (große) Teilmenge von AS3 nach JavaScript, so dass der Code ohne Plug-In im Browser lauffähig ist. Jangaroo wurde von der CoreMedia AG Mitte 2008 unter der liberalen Apache-Lizenz, Version 2, als Open Source veröffentlicht.

## Enterprise Web-Anwendungen mit Jangaroo

Jangaroo kann alle zentralen AS3-Sprachkonzepte für programming in the large nach JavaScript übersetzen. Zur Strukturierung großer Code-Mengen werden Klassen (class) und Pakete (package) benötigt. Während Schnittstellen (interface) die statische Unterteilung in entkoppelte Module ermöglichen, sorgen import-Direktiven und automatisches Laden von Klassen für dynamisches Linking modularen Codes. Die optionale statische Typprüfung dient der Qualitätssicherung sowie der besseren IDE-Unterstützung (u.a. Code-Vervollständigung) und kann mit Standardwerkzeugen (FlexTM-Compiler, AS3-IDE) durchgeführt werden. Bewährte JavaScript-Bibliotheken zur Entwicklung von Web-Anwendungen wie beispielsweise Ext JS [3] lassen sich problemlos von Jangaroo-Code aus ansprechen.

Zusammen mit etablierten Open-Source-Werkzeugen unterstützt Jangaroo den gesamten Lebenszyklus professioneller Softwareentwicklung von Enterprise-JavaScript-Anwendungen.

* Eine **IDE** mit AS3-Unterstützung (IntelliJ IDEA 8, FlexBuilder 3, Eclipse 3 + FDT 3, FlashDevelop 3) wird verwendet, um Code zu erstellen, zu analysieren und zu restrukturieren (refactoring).
* Der **automatische Build-Prozess** wird per Kommandozeile, Ant oder Maven [4] gesteuert. Vor allem die Maven-Integration erlaubt modulare Entwicklung und einfache Zusammenstellung des benötigten Codes nur durch Angabe von Abhängigkeiten (dependencies).
* Der Jangaroo **Compiler** übersetzt AS3-Quellcode nach JavaScript. Zusätzlich zum automatisierten Build-Prozess, steht für IDEA ein Plug-In zur Verfügung, das sich in dessen Build-Prozess integriert [5].
* Mit Adobes Dokumentationswerkzeug ASDoc [6] wird HTML**-API-Dokumentation** generiert.
* JooUnit [7], ein minimal angepasstes FlexUnit 3 [8], kann als **Unit-Test-Framework** eingesetzt werden.
* Mit Selenium [9] können Unit- und **UI-Tests** automatisiert werden.
* Zum **Source-Level-Debugging** kann jeder JavaScript-Debugger eingesetzt werden, da der Quellcode im Debug-Compile-Modus zeilengenau erhalten bleibt.

## Referent

Dr. Frank Wienberg  
Software Architect / Project Lead „Jangaroo“  
CoreMedia AG  
[frank@jangaroo.net](mailto:frank@jangaroo.net)  
fon 040 325587-513  
Ludwig-Erhard-Str. 18  
20459 Hamburg

Ich erkläre mich damit einverstanden, dass der Heise Zeitschriften Verlag meine Einreichung redaktionell bearbeitet und veröffentlicht.

## Quellen

[1] Jangaroo: <http://www.jangaroo.net>, <http://twitter.com/Jangaroo>

[2] Adobe ActionScript 3: <http://labs.adobe.com/wiki/index.php/ActionScript_3>

[3] Ext JS: <http://www.extjs.com>

[4] Maven: <http://maven.apache.org/>

[5] Jangaroo IDEA Plugin: <http://plugins.intellij.net/plugin/?id=4281>

[6] Adobe ASDoc: <http://labs.adobe.com/wiki/index.php/ASDoc>

[7] JooUnit: <http://repo.jangaroo.net/maven2/net/jangaroo/joounit/>

[8] FlexUnit: <http://opensource.adobe.com/wiki/display/flexunit/FlexUnit>

[9] Selenium: <http://seleniumhq.org/>

ActionScript, Flex und ASDoc sind Trademarks von Adobe Systems Incorporated.

Silverlight ist ein Trademark von Microsoft.