
KAFFEEKLATSCH

Das Magazin rund um Software-Entwicklung

ISSN 1865-682X

10/2011



KAFFEEKLATSCH

— Das Magazin rund um Software-Entwicklung —

Sie können die elektronische Form des KAFFEEKLATSCHS
monatlich, kostenlos und unverbindlich
durch eine E-Mail an

abo@bookware.de

abonnieren.

Ihre E-Mail-Adresse wird ausschließlich für den Versand
des KAFFEEKLATSCHS verwendet.

Das Garagensyndrom (heute)

Nie war es einfacher als heute, sich eine neue Existenz aufzubauen – vorausgesetzt natürlich, man entwickelt Software. Denn das Einzige, was man dafür braucht, ist ein Rechner und ein Internet-Anschluss.

Die Software-Entwicklung ist schon eine privilegierte Zunft. Mit wenigen hundert Euro kann man sich alles besorgen was man braucht. Ein Laptop, den man inzwischen in fast jedem Supermarkt für 'nen Apfel und ein Ei bekommen kann, und ein Internet-Anschluss, der in praktisch jedem Haus verfügbar ist, reichen vollkommen aus, um Software entwickeln zu können.

Man braucht dazu noch nicht einmal das Haus zu verlassen, um beispielsweise eine Bibliothek aufzusuchen. Das Internet ist voll von Tutorien, Büchern, Spezifikationen und Dokumentationen. Und wenn man Lesen kann und „die Gabe“ hat, muss man noch nicht einmal eine Lehre machen oder gar studieren.

Was früher eine Garage notwendig machte, ist überflüssig geworden. Kein Löten und kein Schrauben mehr, nur gelegentlich noch ein Stöpseln des einen oder anderen Kabels, das – ordentlich beschriftet – nur noch in die dafür vorgesehenen Buchsen passt. Wobei sich Letzteres in den meisten Fällen nur noch auf das Stromkabel bezieht.

Der Software-Entwickler kann überall arbeiten. Dank W-LAN ist es egal, ob im Café, im Zug, im Park oder eben Zuhause gearbeitet wird. Im Luxusfall gibt es ein Arbeitszimmer, ansonsten wird einfach auf dem Sofa vorm Fernseher oder an dem kleinen Tisch in der Küche „gehackt“. Ein Büro ist gänzlich überflüssig geworden und dient eher der Kommunikation, als das es einen praktischen Nutzen für die Arbeit hätte.

Denn man kann meist einfach vor sich hin wurschteln und im Falle eines Problems, dieses mit Hilfe der Welt lösen: Entwickler-Foren, News-Groups und nicht zuletzt mit GOOGLE & Co. Oft muss man noch nicht einmal eine Basis aufbauen, denn der Rahmen steht schon. Man muss nur noch die richtige Idee haben und Löcher auffüllen.

Aber das Wichtigste dabei ist: Man kann auch noch Geld damit verdienen. Dank der App-Stores bedarf es noch nicht einmal eines Marketings oder Vertriebs. Einfach das Programm schreiben – und im Bezug auf Apps muss man nicht besonders geistreich sein – und ab in den App-Store. Und wenn man den Geschmack der Zeit trifft, kann man reich damit werden, ein mehrere hundert Euro teures Gerät in eine Tonflöte zu „verwandeln“, die man auf jedem Flohmarkt für zwei Euro fünfzig bekommen könnte.

Ihr MICHAEL WIEDEKING
Herausgeber

Beitragsinformation

Der KAFFEEKLATSCH dient Entwicklern, Architekten, Projektleitern und Entscheidern als Kommunikationsplattform. Er soll neben dem Know-how-Transfer von Technologien (insbesondere Java und .NET) auch auf einfache Weise die Publikation von Projekt- und Erfahrungsberichten ermöglichen.

Beiträge

Um einen Beitrag im KAFFEEKLATSCH veröffentlichen zu können, müssen Sie prüfen, ob Ihr Beitrag den folgenden Mindestanforderungen genügt:

- Ist das Thema von Interesse für Entwickler, Architekten, Projektleiter oder Entscheider, speziell wenn sich diese mit der Java- oder .NET-Technologie beschäftigen?
- Ist der Artikel für diese Zielgruppe bei der Arbeit mit Java oder .NET relevant oder hilfreich?
- Genügt die Arbeit den üblichen professionellen Standards für Artikel in Bezug auf Sprache und Erscheinungsbild?

Wenn Sie uns einen solchen Artikel, um ihn in diesem Medium zu veröffentlichen, zukommen lassen, dann übertragen Sie Bookware unwiderruflich das nicht exklusive, weltweit geltende Recht

- diesen Artikel bei Annahme durch die Redaktion im KAFFEEKLATSCH zu veröffentlichen
- diesen Artikel nach Belieben in elektronischer oder gedruckter Form zu verbreiten
- diesen Artikel in der Bookware-Bibliothek zu veröffentlichen
- den Nutzern zu erlauben diesen Artikel für nicht-kommerzielle Zwecke, insbesondere für Weiterbildung und Forschung, zu kopieren und zu verteilen.

Wir möchten deshalb keine Artikel veröffentlichen, die bereits in anderen Print- oder Online-Medien veröffentlicht worden sind.

Selbstverständlich bleibt das Copyright auch bei Ihnen und Bookware wird jede Anfrage für eine kommerzielle Nutzung direkt an Sie weiterleiten.

Die Beiträge sollten in elektronischer Form via E-Mail an redaktion@bookware.de geschickt werden.

Auf Wunsch stellen wir dem Autor seinen Artikel als unveränderlichen PDF-Nachdruck in der kanonischen KAFFEEKLATSCH-Form zur Verfügung, für den er ein unwiderrufliches, nicht-exklusives Nutzungsrecht erhält.

Leserbriefe

Leserbriefe werden nur dann akzeptiert, wenn sie mit vollständigem Namen, Anschrift und E-Mail-Adresse versehen sind. Die Redaktion behält sich vor, Leserbriefe – auch gekürzt – zu veröffentlichen, wenn dem nicht explizit widersprochen wurde.

Sobald ein Leserbrief (oder auch Artikel) als direkte Kritik zu einem bereits veröffentlichten Beitrag aufgefasst werden kann, behält sich die Redaktion vor, die Veröffentlichung jener Beiträge zu verzögern, so dass der Kritisierte die Möglichkeit hat, auf die Kritik in der selben Ausgabe zu reagieren.

Leserbriefe schicken Sie bitte an leserbrief@bookware.de. Für Fragen und Wünsche zu Nachdrucken, Kopien von Berichten oder Referenzen wenden Sie sich bitte direkt an die Autoren.

Werbung ist Information

Firmen haben die Möglichkeit Werbung im KAFFEEKLATSCH unterzubringen. Der Werbeteil ist in drei Teile gegliedert:

- Stellenanzeigen
- Seminaranzeigen
- Produktinformation und -werbung

Die Werbeflächen werden als Vielfaches von Sechsteln und Vierteln einer DIN-A4-Seite zur Verfügung gestellt.

Der Werbeplatz kann bei Frau NATALIA WILHELM via E-Mail an anzeigen@bookware.de oder telefonisch unter 09131/8903-16 gebucht werden.

Abonnement

Der KAFFEEKLATSCH erscheint zur Zeit monatlich. Die jeweils aktuelle Version wird nur via E-Mail als PDF-Dokument versandt. Sie können den KAFFEEKLATSCH via E-Mail an abo@bookware.de oder über das Internet unter www.bookware.de/abo bestellen. Selbstverständlich können Sie das Abo jederzeit und ohne Angabe von Gründen sowohl via E-Mail als auch übers Internet kündigen.

Ältere Versionen können einfach über das Internet als Download unter www.bookware.de/archiv bezogen werden.

Auf Wunsch schicken wir Ihnen auch ein gedrucktes Exemplar. Da es sich dabei um einzelne Exemplare handelt, erkundigen Sie sich bitte wegen der Preise und Versandkosten bei Alexandra Specht via E-Mail unter alexandra.specht@bookware.de oder telefonisch unter 09131/8903-14.

Copyright

Das Copyright des KAFFEEKLATSCHS liegt vollständig bei der Bookware. Wir gestatten die Übernahme des KAFFEEKLATSCHS in Datenbestände, wenn sie ausschließlich privaten Zwecken dienen. Das auszugsweise Kopieren und Archivieren zu gewerblichen Zwecken ohne unsere schriftliche Genehmigung ist nicht gestattet.

Sie dürfen jedoch die unveränderte PDF-Datei gelegentlich und unentgeltlich zu Bildungs- und Forschungszwecken an Interessenten verschicken. Sollten diese allerdings ein dauerhaftes Interesse am KAFFEEKLATSCH haben, so möchten wir diese herzlich dazu einladen, das Magazin direkt von uns zu beziehen. Ein regelmäßiger Versand soll nur über uns erfolgen.

Bei entsprechenden Fragen wenden Sie sich bitte per E-Mail an copyright@bookware.de.

Impressum

KAFFEEKLATSCH Jahrgang 4, Nummer 10, Oktober 2011
 ISSN 1865-682X
 BOOKWARE – eine Initiative der MATHEMA Software GmbH
 Henkestraße 91, 91052 Erlangen
 Telefon: 0 91 31 / 89 03-0
 Telefax: 0 91 31 / 89 03-55
 E-Mail: redaktion@bookware.de
 Internet: www.bookware.de
 Herausgeber/Redakteur: MICHAEL WIEDEKING
 Anzeigen: NATALIA WILHELM
 Grafik: NICOLE DELONG-BUCHANAN

Inhalt

Editorial	3
Beitragsinfo	4
Inhalt	5
User Groups	16
Werbung	18
Das Allerletzte	19

Artikel

ArchiMate verbindet	
Modellgetriebene Integration mit ArchiMate-Konzepten	6
Der Großstadtindianer	
BIRT in Web-Anwendungen	9

Kolumnen

Auf- und abgezählt	
Des Programmierers kleine Vergnügen	12
Deppenleer Zeichen	
Deutsch für Informatiker	14
Verkehrte neue Welt	
Kaffeesatz	15

ArchiMate verbindet

Modellgetriebene Integration mit
ArchiMate-Konzepten 6
von ERWIN HÖSCH

ArchiMate, ein Wortspiel, das übersetzt bedeutet: Der Freund des Architekten. Besagtes Wortspiel ist eine Modellierungssprache der OPEN GROUP für Unternehmensarchitekturen. Dieser Artikel stellt ArchiMate und die Anwendung seiner Konzepte im Forschungsprojekt *Ein Framework für das Integration Engineering im E-Business (EFIE)* vor.

Der Großstadtindianer

BIRT in Web-Anwendungen 9
von WILLIAM SIAKAM

Die *Business Intelligence and Reporting Tools (BIRT)* von Eclipse sind nicht nur dafür designed, streng gekoppelt mit der Eclipse IDE zu laufen. Neben der Option, eine BIRT-Anwendung *headless* zu verwenden, gibt es für die Internetwelt Möglichkeiten, die Reporterstellung in Web-Anwendungen zu integrieren. Welche Wege Ihnen als BI-Entwickler in der JSF-Welt im Zusammenhang mit BIRT offen stehen, erfahren Sie in diesem Artikel.

Auf- und abgezählt

Des Programmierers kleine Vergnügen 12
von MICHAEL WIEDEKING

Über die Zahlen von 1 bis N oder, wie das bei C-orientierten Sprachen üblich ist, von 0 bis $N - 1$ zu iterieren, ist mehr als trivial. Was ist aber, wenn man dabei gleichzeitig auch über die negativen Zahlen iterieren und damit die Zahlenfolge 0, 1, -1, 2, -2, ... erhalten will? Man iteriert dann mit einer geeigneten Fallunterscheidung – die hier ganz vergnüglich eliminiert wird.

ArchiMate verbindet

Modellgetriebene Integration mit ArchiMate-Konzepten

von ERWIN HÖSCH

Archimate, ein Wortspiel, das übersetzt bedeutet: Der Freund des Architekten. Besagtes Wortspiel ist eine

Modellierungssprache der OPEN GROUP für Unternehmensarchitekturen. Dieser Artikel stellt ArchiMate und die Anwendung seiner Konzepte im Forschungsprojekt *Ein Framework für das Integration Engineering im E-Business (EFIE)* vor.

Modellierungssprache

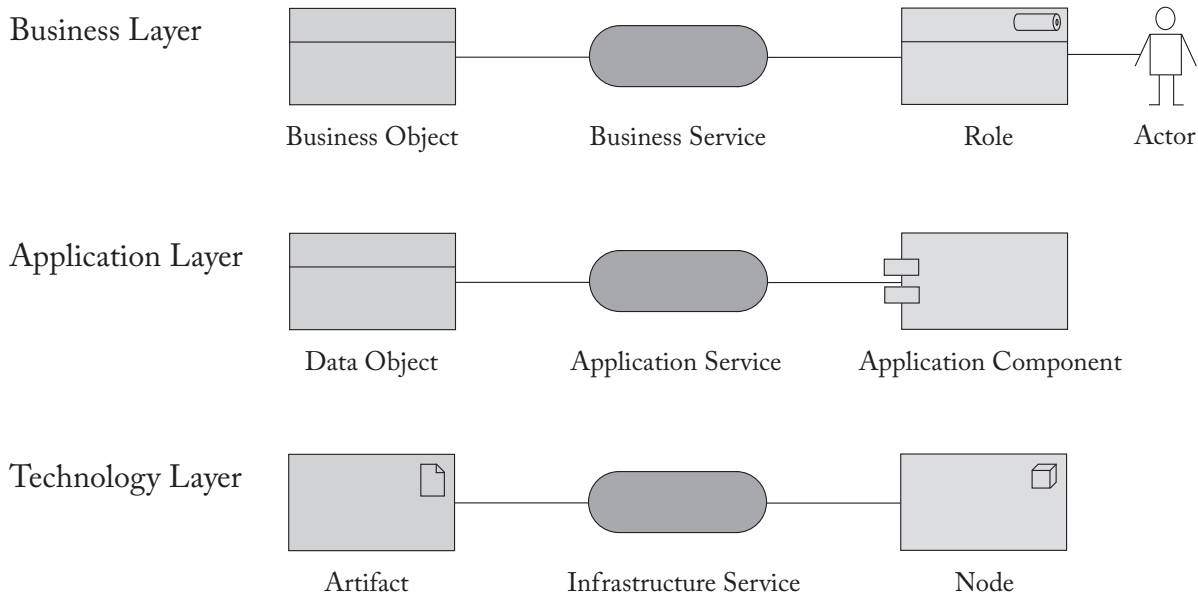
ArchiMate ist in den Niederlanden im Rahmen eines Forschungsprojektes entstanden, bei dem die Regierung,

mehrere Großunternehmen und Universitäten beteiligt waren. Nach dem Ende des Projektes wurde ArchiMate im Jahre 2008 an die OPEN GROUP übergeben und 2009 als Standard der Open Group verabschiedet.

Bevor wir uns dem ArchiMate-Standard zuwenden, noch eine kurze Erklärung zu dem Begriff Unternehmensarchitektur. Eine Unternehmensarchitektur beschreibt das Zusammenspiel von Informationstechnologie und Geschäftsprozessen und hat somit einen ganzheitlichen Blick auf die Informationstechnologie im Unternehmen. Das erste Unternehmensarchitektur-Framework (*The Enterprise Framework*) wurde 1987 von JOHN ZACHMAN vorgestellt. Als weiteres wichtiges Beispiel für ein Unternehmensarchitektur-Framework kann *The Open Group Architecture Framework (TOGAF)* angesehen werden.

Was ist nun das Besondere an ArchiMate?

ArchiMate schlägt eine Brücke zwischen der Geschäftswelt und der Technikerwelt mit Hilfe zweier aufeinander aufbauender Konzepte. Das erste Konzept ist die Definition von drei übereinander angeordneten Modellierungsschichten. Die Reihenfolge der Schichten von oben nach unten ist wie folgt: *Business Layer* (Geschäftsebene), *Application Layer* (die Anwendungsebene) und der *Technology Layer* (Technologieebene). Das zweite Konzept: es werden nicht nur Beziehungen zwischen den Elementen innerhalb einer Ebene definiert sondern auch Beziehungen zwischen Elementen übereinander liegender Ebenen. Damit erlaubt es ArchiMate von einer Ebene in eine andere Ebene zu navigieren und Geschäftsprozesse direkt auf die technischen Prozesse und die technische Infrastruktur abzubilden.



In allen Schichten

ArchiMate verfolgt für alle drei Schichten den gleichen inhaltlichen Aufbau. Fünf Konzepte finden sich schichtenübergreifend wieder.

Erstens: ArchiMate unterscheidet in jeder Ebene zwischen aktiven Strukturelementen, dynamischen Elementen und passiven Strukturelementen. Aktive Elemente spiegeln die Struktur des Modells wieder. Aktive Elemente werden dynamischen Elementen zugewiesen, um zu zeigen, was ausgeführt wird. Passive Elemente werden von dynamischen Elementen benutzt. Diese Strukturierung entspricht einer natürlichen Sprache mit Subjekt (aktives Element), Prädikat (dynamisches Element) und Objekt (passives Element). Einzig die Tatsache, dass der (graphische) Satz mit ArchiMate von rechts nach links geschrieben werden muss, ist ungewohnt.

Zweitens: ArchiMate unterscheidet zwischen einer externen und einer internen Sichtweise. Externe Elemente, *Services* und *Interfaces* werden innerhalb einer Schicht oben angeordnet, interne Elemente werden innerhalb einer Schicht unten angeordnet. Die internen Elemente einer Schicht sollten wiederum die externen Elemente der darunter liegenden Schicht benutzen. Den internen Elementen einer Schicht ist es natürlich erlaubt, ihre externen Elemente zu benutzen.

Drittens: ArchiMate unterscheidet zwischen Verhalten, das von einem einzelnen aktiven Element oder von mehreren aktiven Elementen ausgeführt wird.

Viertens: allen drei Schichten ist gemein, dass ihre externe Schicht Elementpaare von Service und Interface anbietet.

Und fünftens: jede Schicht hat ein zentrales strukturierendes Element. Die zentralen strukturellen Elemente sind die *Business Role* im Business Layer, die *Application Component* im Application Layer und die *Node* im Technology Layer.

Was noch?

ArchiMate definiert ein eigenes Metamodell, eigene graphische Symbole und deren Beziehungen zueinander, ein *Viewpointsystem* und Möglichkeiten für die Sprachenerweiterung.

Wie kann man ArchiMate praktisch nutzen?

Erste Möglichkeit: man kauft ein kommerzielles Werkzeug, das den Standard umsetzt. Eine Liste der kommerziellen Werkzeuge ist auf der ArchiMate-Internetseite [1] zu finden. Zweite Möglichkeit: man benutzt die Open Source Implementierung *Archi* [2]. Dritte Möglichkeit: man implementiert selbst ein Werkzeug, das

dem Standard genügt bzw. sich an ihn anlehnt, wie das beispielsweise im Forschungsprojekt *Ein Framework für das Integration Engineering im E-Business (EFIE)* [3] getan wurde.

Der Unterschied zwischen ArchiMate und der UML

Die *Unified Modeling Language (UML)* ist ursprünglich entstanden, um das Verhalten objektorientierter Systeme zu modellieren. Bei der Entstehung der UML wurden verschiedene Modellierungssprachen für die objektorientierte Programmierung zur UML zusammengefasst. Von diesem Zeitpunkt an wurde der Umfang der UML auf 13 unabhängige Diagrammsprachen erweitert. Sie ist also eine Ansammlung von Modellen, die im Verlauf der Zeit erweitert wurden. Was der UML fehlt, ist ein Metamodell, das definiert, wie die 13 Diagrammsprachen und ihre Elemente in Beziehung zueinander gesetzt werden. Die UML besagt nicht, wie ein Geschäftsprozess zu Software-Modulen und zur Hardware abgebildet wird. ArchiMate hingegen versucht, die Beziehungen zwischen unterschiedlichen Domänen zu modellieren. Mit Hilfe dieser modellierten Beziehungen wird es möglich, zwischen diesen Domänen zu navigieren und das System zu analysieren. ArchiMate konkurriert mit der UML nicht auf Objektebene. Die UML ist ein Werkzeug um Teilsysteme zu modellieren. ArchiMate ist ein Werkzeug um Zusammenhänge zu modellieren.

EFIE

Im Rahmen des Forschungsprojektes EFIE ist ein Modellierungsprototyp auf der Basis von *EMF* [4] und *GMF* [5] entstanden, der sich an die ArchiMate-Konzepte anlehnt. Ziel des Projektes war es, ein modellgetriebenes Integrationsframework zu erstellen, das an die Bedürfnisse des jeweiligen Integrationsdienstleisters angepasst werden kann. Daraus leiten sich folgende Notwendigkeiten ab:

1. Für eine schnelle und erfolgreiche Integration muss das Framework einen Überblick über alle Belange der zu integrierenden Systeme schaffen.
2. Das Framework muss spezifische Erweiterungen eines Dienstleisters erlauben.
3. Das Framework muss es erlauben, auf der Basis der erfassten Daten Integrationsvorschläge zu machen, auf das Fehlen von integrationsrelevanten Daten hinzuweisen und mit jeder erfolgreichen Integration seine Wissens- und Entscheidungsgrundlage zu erweitern und zu verbessern.

Die Konzepte von ArchiMate bedienen genau diese Notwendigkeiten und geben folgende Antworten:

1. Die drei Schichten von ArchiMate sind das ideale Konzept, um sich einen Gesamtüberblick über ein System und dessen Zusammenspiel zu verschaffen.
2. ArchiMate definiert eine grundlegende Modellierungssprache mit dem Ziel der Anpassung und Erweiterung für spezifische Anwendungsgebiete. Im Forschungsprojekt EFIE wurde ArchiMate um Elemente für Software-Integration erweitert.
3. Die Navigierbarkeit in dem Gesamtmodell ist die Basis, um auf den erfassten Elementen, ihren Attributen und Beziehungen Analysen durchzuführen. Der Modellierungsprototyp bietet eine Schnittstelle zur Auswertung und Analyse an. In dem Prototypen wird als Auswertungstechnologie *Drools* [6] eingesetzt. Mit Hilfe von *Drools* wird geprüft, ob integrationsrelevante Daten fehlen und basierend auf der Wissensbasis des Dienstleisters entstehen Integrationsvorschläge. Jedes Integrationsprojekt kann zu neuen Regeln und neuen Attributen führen und somit das System erweitern und verbessern. Neue Attribute verlangen zwangsweise auch neue Regeln.

Mit diesem Ansatz bewegt man sich von dem Integrationsvorgehen: „Wie schaffe ich eine technische Verbindung zwischen den Systemen und welche Daten muss ich dabei austauschen?“ zu einem architekturellen Integrationsansatz. Man dokumentiert seine Integrationslösungen durch Modelle, man dokumentiert das Wissen der eigenen Mitarbeiter und man erhöht die Qualität der Integrationslösung durch die Analysen und wissensbasierten Vorschläge.

Das Forschungsprojekt EFIE war ein Projekt, das sich an kleine und mittelständische Unternehmen richtet hat, die es sich wirtschaftlich nicht leisten konnten, ein solches Werkzeug alleine zu implementieren, um ihre Integrationsdienstleistung effizienter und kostengünstiger anzubieten.

Das Forschungsprojekt hat gezeigt, dass sich der Einsatz eines solchen Werkzeuges nur dann amortisiert, wenn die Integrations szenarien sich ähneln oder einen gleichbleibenden Anteil besitzen, da eine grafische Erfassung des Szenarios mit all seinen Attributen sehr zeintensiv ist. Ein Unternehmen mit einem Produkt, das bei jedem Kunden in die IT-Landschaft integriert werden muss, besitzt diesen gleichbleibenden Anteil.

Im Forschungsprojekt EFIE war das ein Kooperationspartner, der ein ERP-System für kleine und mittelständische Firmen implementiert und vertrieben hat. Der Kooperationspartner musste einmalig sein Produkt modellieren und als Vorlage im Modellierungsprototypen abspeichern. Von Integrationsprojekt zu Integrationsprojekt konnten neue Erkenntnisse mit in die Vorlage übernommen werden. Damit wurde es möglich die Projekte rasch abzuwickeln und die Qualität der Lösung mit jeder Integration zu verbessern.

Fazit

In dem Forschungsprojekt EFIE konnte nachgewiesen werden, dass ein modellgetriebener Integrationsansatz die Integrationskosten senkt. Sehr hilfreich bei der Implementierung eines modellgetriebenen Integrationswerkzeuges war es, sich an die Konzepte aus dem Modellierungsstandard ArchiMate anzulehnen. Wer sich ausführlicher mit ArchiMate auseinandersetzen möchte, dem sei als Einstieg unbedingt das Buch von MARK LANKHORST *Enterprise Architecture at work* [7] zu dem Thema empfohlen.

Referenzen

- [1] ARCHIMATE *Tool Support*
http://www.archimate.nl/en/start_using_archimate/support.html
- [2] ARCHI *Overview*
<http://archi.cetis.ac.uk>
- [3] EFIE *home*
<http://www.ie-framework.de>
- [4] ECLIPSE *Eclipse Modeling Framework (EMF)*
<http://www.eclipse.org/modeling/emf>
- [5] ECLIPSE *Graphical Modeling Project (GMP)*
<http://www.eclipse.org/modeling/gmp>
- [6] JBOSS COMMUNITY *Drools*
<http://jboss.org/drools>
- [7] LANKHORST, MARC et al. *Enterprise Architecture at Work*, Springer, 2005, ISBN: 978-3-642-01309-6

Kurzbiografie



ERWIN HÖSCH arbeitet als Senior Consultant und Trainer für die MATHEMA Software GmbH. Seit 1997 sind seine Kernthemen Java und die Entwicklung verteilter Anwendungen (CORBA). Seit dem Jahr 2005 beschäftigt er sich verstärkt mit dem Entwurf und der Realisierung von Eclipse-Plugins.

COPYRIGHT © 2011 BOOKWARE 1865-682X/11/10/001 Von diesem KAFFEEKLATSCH-Artikel dürfen nur dann gedruckte oder digitale Kopien im Ganzen oder in Teilen gemacht werden, wenn deren Nutzung ausschließlich privaten oder schulischen Zwecken dient. Des Weiteren dürfen jene nur dann für nicht-kommerzielle Zwecke kopiert, verteilt oder vertrieben werden, wenn diese Notiz und die vollständigen Artikelangaben der ersten Seite (Ausgabe, Autor, Titel, Untertitel) erhalten bleiben. Jede andere Art der Vervielfältigung – insbesondere die Publikation auf Servern und die Verteilung über Listen – erfordert eine spezielle Genehmigung und ist möglicherweise mit Gebühren verbunden.

Der Großstadtindianer

BIRT in Web-Anwendungen

von WILLIAM SIAKAM




Die *Business Intelligence and Reporting Tools (BIRT)* von Eclipse sind nicht nur dafür designed, streng gekoppelt mit der Eclipse IDE zu laufen. Neben der Option, eine BIRT-Anwendung *headless* zu verwenden, gibt es für die Internet-Welt Möglichkeiten, die Reporterstellung in Web-Anwendungen zu integrieren. Welche Wege Ihnen als BI-Entwickler in der JSF-Welt im Zusammenhang mit BIRT offen stehen, erfahren Sie in diesem Artikel.

Die Darstellung für Business Intelligence-relevanter Diagramme wurde von JSF nicht explizit vorgesehen. Um also einer Darstellung mit reinen JSF-Komponenten nahe zu kommen, könnte man innerhalb bestimmter *Tags* mit herkömmlichen *Style*-Parametern herumexperimentieren, um deren Verhalten dynamisch zu steuern. So könnte beispielsweise der einfachste JSF-Einsatz für die Darstellung von Balken aussehen:

```
<h:dataTable value="#{bean.reports}" var="report">
  <h:column>
    <f:facet name="header">
      <h:outputText value="Customer" />
    </f:facet>
    <h:outputText value="#{report.name}" />
  </h:column>
  <h:column>
    <f:facet name="header">
      <h:outputText value="sales" />
    </f:facet>
    <h:panelGroup layout="block"
      style="background-color: red ;
      width: #{report.sale}px; height : 20px;" />
  </h:column>
</h:dataTable>
```

Das Ergebnis beim Rendern der Seite könnte dann wie folgt aussehen:

Business Report 2011

Customer	sales
Firma A	
Firma B	
Firma C	

Balkendiagramme (auch in vertikaler Darstellung) lassen sich relativ einfach darstellen, aber spätestens bei anderen Diagrammart, z. B. einem Tortendiagramm, kann sich dieser Weg sehr schnell in einen Jakobsweg [1] wandeln. Im letzten BIRT-Artikel [2] wurde die programmatische Reporterstellung mit Java gezeigt. Mit BIRT kann man zusätzlich einen erstellten Report in verschiedenen Formaten exportieren. Mit der folgenden Methode könnte man einen Report (der im *.rptdesign*-Format gespeichert wurde) im HTML- oder PDF-Format exportieren.

```
public void exportReport(String format)
  throws Exception {
  IREPORTENGINE engine = null;
  ENGINECONFIG config = null;
  try{
    config = new ENGINECONFIG();
    config.setBIRTHome(BIRT_HOME);
    // Platform.startup(config); Falls die Bundles beim
    // Speichern des Reports heruntergefahren wurden müssen
    // sie erneut gestartet werden.
    IREPORTENGINEFACTORY factory =
      (IREPORTENGINEFACTORY) PLATFORM
        .createFactoryObject(
          IREPORTENGINEFACTORY.
            EXTENSION_REPORT_ENGINE_FACTORY );
    engine = factory.createReportEngine(config);

    IREPORTRUNNABLE design;
    design = engine.openReportDesign(REPORT_FILE);
    IRunAndRenderTask task;
    task = engine.createRunAndRenderTask(design);
    if(format.equals("HTML")){
      HTMLRENDEROPTION options =
        new HTMLRENDEROPTION();
      options.setOutputFileName(HTMLEXPORT_FILE);
      options.setOutputFormat("html");
      task.setRenderOption(options);
    }
  }
```

```

else if(format.equals("PDF")) {
    PDFRENDEROPTION options =
        new PDFRENDEROPTION();
    options.setOutputFileName(PDFEXPORT_FILE);
    options.setOutputFormat("pdf");
    task.setRenderOption(options);
}
else {
    throw new undefinedFormatException();
}
task.run();
task.close();
engine.destroy();
} catch (EXCEPTION ex){
    ex.printStackTrace();
} finally {
    Platform.shutdown( );
}
System.out.println(
    "Report erfolgreich in "+ format+ "-Format exportiert!"
);
}

```

Möchte man beispielsweise BIRT als Web-Anwendung laufen lassen, ist bereits unter der zum Download bereit stehenden BIRT-Distribution eine Web-Anwendung vorhanden. Standardmäßig heißt diese *WebViewerExample*. Nachdem sie beliebig umbenannt wurde, kann sie direkt ins *Webapp*-Verzeichnis von Tomcat kopiert werden. Nach einem Neustart vom Tomcat wird sie automatisch deployed und steht zur Anfrage bereit. Eine weitere Web-Anwendung, die ebenso auf dem Tomcat läuft, kann man die Report- bzw. Exportdienste in Anspruch nehmen. Zum Beispiel könnte man den erstellten Report mit dem HTML-Tag *iframe* wie folgt in eine bestehende Seite einbetten:

```

<iframe
    src="http://localhost:8080/birt/run?__report=bi.rptdesign"
></iframe>

```

In der *web.xml* der BIRT-Anwendung gibt es mehrere *Servlet-Mapping*-Möglichkeiten. Die hier verwendete *run-Mapping* ist eine einfache, randlose Seiten-Integration und bietet weder Ajax-Charakter noch Export-Tools an. Das Servlet-Mapping weist entsprechende BIRT-Klassen an, die das *ReportDesign*-Dokument (*bi.rptdesign*) parsen und den *HTML-Output* generieren. Der folgende Ausschnitt der *web.xml* der *WebViewerExample*-Anwendung zeigt die verantwortlichen *Servlet*-Klassen und einige *Mapping*-Möglichkeiten.

```

<!-- Viewer Servlet, Supports SOAP -->
<servlet>
    <servlet-name>ViewerServlet</servlet-name>
    <servlet-class>
        org.eclipse.birt.report.servlet.ViewerServlet
    </servlet-class>
</servlet>
<!-- Engine Servlet -->
<servlet>
    <servlet-name>EngineServlet</servlet-name>
    <servlet-class>
        org.eclipse.birt.report.servlet.BirtEngineServlet
    </servlet-class>
</servlet>
<!-- Servlet Mapping -->
<servlet-mapping>
    <servlet-name>ViewerServlet</servlet-name>
    <url-pattern>/frameset</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>ViewerServlet</servlet-name>
    <url-pattern>/run</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>EngineServlet</servlet-name>
    <url-pattern>/preview</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>EngineServlet</servlet-name>
    <url-pattern>/download</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>EngineServlet</servlet-name>
    <url-pattern>/parameter</url-pattern>
</servlet-mapping>

```

Die interessanteste Mapping-Möglichkeit ist die *frameset*. Sie stellt eine Ajax-fähige Seite dar, die verschiedene Exportmöglichkeiten anbietet. Darunter Microsoft Excel, PDF und CSV.

Der Vorteil der Verwendung der *WebViewerExample*-Anwendung liegt darin, dass man einfach eine „fertige“ Anwendung benutzt bzw. in eine bestehende einbindet und somit weniger Konfigurationsschweiß investieren muss.

Eine ästhetischere Möglichkeit BIRT in die JSF-Welt zu integrieren, liefert *JSF4birt* [3], eine von Exadel entwickelte Bibliothek, die die BIRT-Klassen so *wrapp*t, dass sie durch Tags angesprochen werden können. Die Eleganz der Lösung zeigt sich darin, dass die Report-Erstellung in den JSF-Lebenszyklus integriert ist und im Gegensatz zu der *Iframe*-Lösung keine separaten, un schönen *Render*-Effekte existieren. Hier ein Beispiel für die Verwendung des *jsf4birt*-Frameworks:

```
<birt:birtWrapper id="businessReport"
  reportDesign="/bi_report.rptdesign">
  <f:param name="jahrParam" value="2011"/>
</birt:birtWrapper>
```

Das Kleingedruckte

Ein Nachteil ist, dass diese Variante nicht gerade dafür sorgt, dass die Anwendung schlanker wird. Die ganzen BIRT-Laufzeit-Bibliotheken müssen im Klassenpfad der Anwendung liegen, und das sind nicht gerade wenig. Außerdem wird eine bestimmte Projektstruktur von *jsf4birt* vorgeschrieben. Ein Ordner namens *plattform* muss unter */WEB-INF/* liegen und dorthin müssen auch u. a. die *eclipse-Bundles* kopiert werden, die in der BIRT-Distribution mitgeliefert werden. Bei einer Reporterstellungsaktion werden die entsprechenden *eclipse-Bundles* gestartet, deshalb trägt sie nicht viel zur Performanz der Anwendung bei und schmiert den Hardware-Ressourcen somit keine Butter auf die Brezel.

Und für Unterwegs?

Für die Freunde unter uns, die unterwegs zur Abwechslung mal, anstatt ihre Zeit auf einer bekannten Sozialnetzwerk-Plattform zu „vergeuden“, auf die Idee kommen, ihre Daten zu visualisieren, gibt es eine erfreuliche Nachricht: Dafür gibt's eine App!



BIRT Mobile Viewer [4]

Der App-Client greift auf eine *Cloud*-Anwendung zu und kann sie komfortabel durch Report-Parameter bedienen. Die Nutzung dieser (kostenlosen) App erfolgt ausschließlich in Verbindung mit einem Account bei der (kostenpflichtigen) SaaS-Lösung *BIRT onDemand* [5], die für die ersten 45 Tage kostenlos genutzt werden kann. Dort kann jeder User seine persönlichen Report-Dokumente hochladen und „mobil“ machen (*Reports mobilization*), zumindest wenn man unterwegs darauf Zugriff haben möchte. Ein großer Gewinn bei der *onDemand*-

Lösung ist, dass man seine Reports mit ausgewählten anderen *onDemand*-Usern teilen kann.

Fazit

Die Einbindung von BIRT in Web-Anwendungen funktioniert relativ problemlos. Wie man sehen konnte, gibt es mehrere Möglichkeiten BIRT-Reports in eine bestehende Web-Anwendung zu integrieren. Möchte man die komplette Kontrolle über die Reportersstellungsphasen haben (dynamisches Bestimmen der Datenquellen und Datensätze, sowie Format und Design der Reports), könnte man die im letzten Artikel [2] gezeigte Variante in *Backing-Beans* einer JSF-Anwendung implementieren und das erstellte *rptDesign*-Dokument im Web-Ordner der *WebViewExample*-Anwendung ablegen. Diese kann anschließend in die Zielseite integriert werden.

Die umfangreiche API des BIRT-Frameworks bietet den Business Intelligence-Entwicklern eine sehr flexible Möglichkeit seine Reports zu bestimmen, wobei die Komplexität der Anwendung dieser API an den eigenen Anspruch an die Reports gekoppelt ist. Ein Punkt, in dem sich BIRT als Report-Framework von anderen vorteilhaft differenziert, ist das Exportieren der Reports in unterschiedlichen Formaten. Legt man eher viel Wert auf schnelle graphische Darstellung von Anwendungsdaten, ohne zwangsläufig an Exportmöglichkeiten zu appellieren, lohnt es sich einen Blick auf JSF-Frameworks wie *Primefaces* [6] oder *Fiji* [7] zu werfen.

Referenzen

- [1] WIKIPEDIA *Jakobsweg*
<http://de.wikipedia.org/wiki/Jakobsweg>
- [2] SIAKAM, WILLIAM *Berichterstatter Teil 2*
KAFFEEKLATSCH, Februar 2011
- [3] EXADEL *Exadel jsf4birt*
<http://exadel.org/jsf4birt>
- [4] BIRTEXCHANGE *BIRTMobile*
<http://www.birt-exchange.com/be/marketplace/resources/?articleid=19497>
- [5] BIRTONDEMAND *home*
<https://www.birtondemand.com/bod/home>
- [6] PRIMEFACES *home*
<http://www.primefaces.org>
- [7] EXADEL *Exadel Fiji*
<http://exadel.org/fiji>

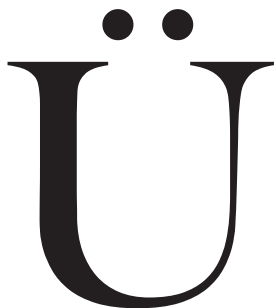
Kurzbiografie



WILLIAM SIAKAM arbeitet als Software-Entwickler und -Consultant für die MATHEMA Software GmbH. Er interessiert sich besonders für die Java Standard und Enterprise Edition (Java SE und Java EE). Dabei liegt sein Schwerpunkt auf dem Design und der Anwendung von Web-Frameworks.

Auf- und abgezählt

von MICHAEL WIEDEKING



Über die Zahlen von 1 bis N oder, wie das bei C-orientierten Sprachen üblich ist, von 0 bis $N - 1$ zu iterieren, ist mehr als trivial. Was ist aber, wenn man dabei gleichzeitig auch über die negativen Zahlen iterieren und damit die Zahlenfolge $0, 1, -1, 2, -2, \dots$ erhalten will? Man iteriert dann mit einer geeigneten Fallunterscheidung – die hier ganz vergnüglich eliminiert wird.

Neulich hatte ich das Problem, dass ich über die ganzen Zahlen, also positive wie negative Zahlen, iterieren wollte. Die Frage nach der Reihenfolge der Zahlen stellte sich dabei nicht wirklich, weil es meist genügt, bei 0 anzufangen und dann betragsmäßig aufsteigend erst die positive und dann die dazugehörige negative Zahl zu liefern:

$0, +1, -1, +2, -2, +3, -3, \dots$

Dies lässt sich relativ einfach mit einer bedingten Anweisung bewerkstelligen. Hat man es mit einer positiven Zahl zu tun, dann muss diese einfach negiert werden. Hat man es hingegen mit einer negativen Zahl zu tun, so muss diese erst durch eine Negation in eine positive Zahl umgewandelt werden, um diese dann zu erhöhen.

$(i > 0) ? (-i) : (-i + 1)$

Da die 0 weder positiv noch negativ ist, muss man sich noch überlegen, wo diese eingeordnet werden soll, wobei recht günstig ist, dass sich eine Negation auf die 0 nie auswirkt. Da die oben angegebene Folge $0, 1, -1, \dots$ lautet, kann die 0 einfach zu den negativen Zahlen zählen; damit bleibt es bei der angegebenen Bedingung $(i > 0)$ und der Nachfolger von 0 ist damit wunschgemäß $(-0 + 1) = 1$.

```
// Ausgabe der Folge 0, 1, -1, ..., n - 1, -(n - 1)
for (int i = 0; i < N; i = (i > 0) ? (-i) : (-i + 1)) {
    print(i);
}
```

Aber dies hier wäre nicht ganz so vergnüglich, wenn sich diese fürchterliche Bedingung nicht eliminieren ließe. Formt man nämlich die Ausdrücke um, so kann man für die beiden Fälle auch $(0 - i)$ bzw. $(1 - i)$ schreiben. Die beiden Ausdrücke sind bis auf die 0 bzw. 1 gleich. Würde es nun gelingen, die Bedingung in etwas umzuwandeln, dass entsprechend eine 0 für $(i > 0)$ und eine 1 für $(i \leq 0)$ liefern würde, wäre das Problem schon gelöst.

Die C-Programmierer werden sich hier ins Fäustchen lachen, liefert bei denen doch jede Bedingung eine 1, wenn diese erfüllt ist und 0, wenn diese nicht erfüllt ist. Damit lässt sich das Iterieren – nicht verständlicher, aber deutlich kürzer – wie folgt beschreiben:

$(i > 0) - i$

Was machen aber die Programmierer, denen dieses Mittel nicht zur Verfügung steht? Die erinnern sich daran, dass sich diesem Problem schon einmal ein kleines Vergnügen [1] gewidmet hat. Darin wurde nämlich beschrieben, wie man beliebige ganzzahlige Vergleiche in Ausdrücke um-

wandeln kann, die wie in C eine 1 für *true* und eine 0 für *false* liefern.¹

Für $(x > 0)$ steht dort beispielsweise, dass man $(-x \ \& \ \sim x) \ggg w$ verwenden kann, wobei in diesem Fall w die Wortbreite $- 1$ ist, für ein 32-Bit-Wort also 31. Einen einfachen Test in vier Anweisungen umzuwandeln kann tatsächlich immer noch schneller sein, ist aber schon sehr grenzwertig. Ganz neidisch blickt man da auf $(x < 0)$, das in nur einer Operation mit $(x \ggg w)$ berechnet werden kann, oder $(x \leq 0)$, für das $(\sim x \ggg w)$ gilt. Eine oder zwei Operationen sind deutlich schneller als vier.

Hier bietet sich also an, den Ausdruck $(x > 0)$ so umzuformen, dass anstatt des größer-Operators der kleiner- oder kleiner-oder-gleich-Operator zum Einsatz kommen kann. Das lässt sich recht einfach bewerkstelligen, denn x ist nämlich genau dann größer als 0, wenn $-x$ kleiner oder gleich 0 ist. Also kann man für $(x > 0)$ auch $(-x \leq 0)$ schreiben und damit $(\sim(-x) \ggg w)$.

Dadurch kann man eine Operation einsparen. Wandelt man den Ausdruck $\sim(-x)$ auch noch um, so erhält man wegen $\sim(-x) = x - 1$ schließlich $(x - 1) \ggg w$ und braucht lediglich zwei statt der ursprünglichen vier Operationen. Alles in Allem erhält man nun für die Iteration

```
for (int i = 0; i < N; i = ((i - 1) >>> 31) - i) ...
```

Anstatt der einen Operation, die man beim Iterieren über die positiven ganzen Zahlen braucht, hier, beim Einbeziehen der negativen Zahlen, nur drei Operationen zu brauchen, ist mehr als zufriedenstellend. Wie schon erwähnt, ist das nicht unbedingt etwas für die Lesbarkeit, aber auf jeden Fall etwas für ein gutes Gefühl.

Referenzen

[1] WIEDEKING, MICHAEL

Des Programmierers kleine Vergnügen – Vergleichsweise einfach

KAFFEEKLATSCH, Jahrgang 3, Nr. 3, Bookware, März 2010

<http://www.bookware.de/kaffeeklatsch/archiv/KaffeeKlatsch-2010-03.pdf>

Kurzbiographie



MICHAEL WIEDEKING (michael.wiedeking@mathema.de) ist Gründer und Geschäftsführer der MATHEMA Software GmbH, die sich von Anfang an mit Objekttechnologien und dem professionellen Einsatz von Java einen Namen gemacht hat. Er ist Java-Programmierer der ersten Stunde, „sammelt“ Programmiersprachen und beschäftigt sich mit deren Design und Implementierung.

¹ Dazu wird neben dem bitweisen Und $(x \ \& \ y)$ und dem bitweisen Oder $(x \ | \ y)$ das einstellige $\sim x$ benötigt, mit dem alle Bits eines Worts invertiert werden. Für die Negation $-x$ gilt dann im Falle des Zweierkomplements, das hier stillschweigend vorausgesetzt wird, dass $-x = \sim x + 1$ ist. Darüber hinaus ist $x \ggg n$ ein vorzeichenloses Verschieben von x um n Stellen nach rechts, wobei garantiert nur Nullen nachgeschoben werden.

Wissenstransfer par excellence

3.– 6. September 2012
in Nürnberg

Deutsch für Informatiker

Deppenleer Zeichen

VON ALEXANDRA SPECHT

Mit dieser Überschrift sind wir auch schon Mitten im Thema dieser Kolumne. Und nein, es geht nicht um Deppen und Leerzeichen, sondern um das Deppenleerzeichen. Über den Deppenapostroph haben wir ja schon geschrieben.¹

Wie ist es denn mit den Leerzeichen, die gemacht werden und da nicht hingehören? Es ist ja immer so eine Sache mit zuviel und zuwenig. Wir wollen ja alle gern viel. Aber wenn dann etwas zuviel ist, ist es nicht schön; ist etwas zu wenig, ist es nicht viel besser. Ein kleines Leerzeichen wegzulassen erscheint als nicht so kapitaler Fehler, aber wenn dann aus einer Unihalle die Uni Halle wird, ist das dann schon verwirrend. [1]

Leerzeichen in Komposita, also in Zusammensetzungen von Substantiven, Adjektiven, Verbstämmen, Pronomen oder Partikeln mit Substantiven ebenso wie von mehrteiligen Substantivierungen sind nach der geltenden Rechtschreibung im Deutschen grundsätzlich nicht vorgesehen. Nach § 37 der amtlichen Regelung werden diese Zusammensetzungen zusammengeschrieben. Diese Leerzeichen werden umgangssprachlich als Deppenleerzeichen bezeichnet.

Doch keine Regel ohne diverse Ausnahmen: Falls es das Verständnis erfordern sollte, dass einzelne Bestandteile besonders gekennzeichnet, gegeneinander abgesetzt oder hervorgehoben werden sollen, dann darf ein Bindestrich verwendet werden; ebenso beim Zusammentreffen von drei gleichen Buchstaben. Zum Beispiel: *Ich-Erzählung*, *Software-Angebotsmesse*, *Drucker-Zeugnis*, *Kaffee-Ersatz*.

Aus anderen Sprachen stammende Verbindungen aus Substantiv plus Substantiv, die sich im Deutschen grammatisch wie Zusammensetzungen verhalten, können

¹ Heute erst hat mir ein Kollege wieder von einem ganz schönen erzählt: *Zum Günter's Trödelmarkt*. Nicht nur ein Deppenapostroph, sondern auch noch eine kleine Schwäche in der Grammatik.

zusammen- oder mit Bindestrich geschrieben werden: *Desktoppublishing* oder *Desktop-Publishing*.

Aus dem Englischen stammende Substantivierungen aus Verb plus Adverb schreibt man mit Bindestrich, zum Beispiel: *Make-up*. Wobei aber auch eine Zusammenschreibung erlaubt ist, sofern die Lesbarkeit nicht beeinträchtigt ist: *Stand-by* oder *Standby*.

Man sollte sogar einen Bindestrich setzen bei Zusammensetzungen mit Einzelbuchstaben, Abkürzungen, Initialwörtern oder Ziffern: *x-beliebig*, *Dipl.-Ing.* (= *Diplomingenieur*), *8-Zylinder* die *n-te*.

Bei Verbindungen aus Ziffern und Suffixen im vorderen Teil der Zusammensetzung, setzt man nach dem Suffix einen Bindestrich: in den *80er-Jahren* (aber auch in den *80er Jahren*).

Und man setzt Bindestriche in substantivisch gebrauchten Zusammensetzungen insbesondere bei substantivisch gebrauchten Infinitiven mit mehr als zwei Bestandteilen, wenn es das Verständnis fördert. Zum Beispiel: *Entweder-oder*, *An-den-Haaren-Herbeiziehen*, *Von-der-Hand-in-den-Mund-Leben*, *Trimm-dich-Pfad*, aber: *Ballspielen*, *Inkrafttreten*.

Und dann gibt es noch schöne Beispiele über den regen Gebrauch des Deppenleerzeichens bei *Apple*, wo offensichtlich konsequent auf jegliche Koppelungsstriche im deutschsprachigen Raum verzichtet wird. Geschweige denn, dass die deutsche Rechtschreibung beachtet und die entsprechenden Wörter zusammen geschrieben werden:

Ein *echter Durchbruch* (Da ist die Welt noch in Ordnung, kein *Durch Bruch*), neueste *Quad-Core Prozessoren*, *Thunderbolt Technologie*, *FaceTime HD Kamera*, *Multi-Touch Trackpad* [2].

Richtig wäre: *Quad-Core-Prozessoren*, *Thunderbolt-Technologie*, *FaceTime-HD-Kamera*, *Multi-Touch-Trackpad* oder *Multitouchtrackpad*.

Oder was auch möglich ist, wenn man nicht so der Bindestrichfan ist, den Begriff in Anführungszeichen zu setzen und nur einen Koppelungsstrich zu verwenden, zum Beispiel: „*FaceTime HD*“-Kamera.

Nun ja. Wenn man sich also nicht ganz sicher ist und nicht alle Regeln parat hat, dann einfach zusammenschreiben. Und wenn das dann ganz komisch aussieht oder was anderes bedeuten kann, als das, was man ursprünglich meinte, dann einfach einen Bindestrich setzen. Allemaal besser als ein Deppenleerzeichen.

Referenzen

- [1] WIKIPEDIA *Datei:Deppenleer Zeichen cropped.jpg*
http://de.wikipedia.org/w/index.php?title=Datei:Deppenleer_Zeichen_cropped.jpg&filetimestamp=20080415134926
- [2] APPLE *Ein echter Durchbruch, durch und durch*
<http://www.apple.com/de/macbookpro/features.html>

Kaffeesatz

Verkehrte neue Welt

von MICHAEL WIEDEKING

Die JavaOne in San Francisco zu besuchen ist immer ein Privileg. Nicht nur wegen der zu erwartenden Vorträge, sondern auch wegen des Drumherums. Wann immer ich dann in San Francisco bin, freue ich mich nämlich sehr darüber, dass ich in der Fremde einkaufen kann, vor allem Bücher, Hörbücher und CDs. So ist es für mich immer eine Wonne in der fremden Welt zu stöbern, um zu sehen, was denn gerade so literarisch und musikalisch „in“ ist.

Der Buchladen meiner Wahl ist meist BORDERS, sehr zentral am Union Square gelegen, da wo die Cable Cars vorbeifahren und an der Ecke, wo sich Detective Lt. MIKE STONE (KARL MALDEN) und Inspector STEVE KELLER (MICHAEL DOUGLAS) gelegentlich mit dem gut informierten Schuhputzer in den „Straßen von San Francisco“ (1972 – 1977) getroffen haben. Über vier Etagen fand sich gut sortiert alles, was das Herz begehrt: Belletristik, Computerbücher, Bildbände, Postkarten, Zeitschriften, Filme, Musik und vieles mehr und darüber hinaus noch das obligatorische Café.

Das schöne an dem Laden waren auch die Öffnungszeiten: Das Geschäft hatte bis abends zehn Uhr offen und am Wochenende sogar bis elf Uhr. Selbstverständlich bedeutete Wochenende auch, dass der Sonn-

tag mit eingeschlossen war und so konnte man während oder nach der Konferenz beliebig viel in den Regalen stöbern, in den Büchern schmökern und in die Musik Reinhören.

So wollte ich auch dieses Jahr wieder meinen jährlichen Großeinkauf starten. Weil ich schon ein bisschen früher in San Francisco angekommen war, habe ich mir einen frühen Nachmittag für ein erstes Stöbern ausgesucht. Wie überrascht war ich dann, als ich vor geschlossenen Türen stand. Da fiel mir auch wieder ein, dass ich von der Insolvenz der Buchladenkette gelesen hatte, aber offensichtlich war mir da nicht ganz klar, was das bedeuten würde.

Also bin ich wieder zurück ins Hotel und habe nachgesehen, was es sonst noch so gibt, mir die Adressen notiert und habe mich am nächsten Tag aufgemacht, die Läden zu finden. Die wenigen Bücherläden waren zwar allesamt sympathisch und mehr oder weniger gut sortiert, hatten aber so gut wie keine Computerbücher im Sortiment. Also setzte ich meine Hoffnung auf BARNES & NOBLE.

Nachdem ich deswegen über eine halbe Stunde vom Embarcadero zum Fisherman's Wharf gelaufen bin, musste ich leider feststellen, dass an der angegebenen Adresse nur ein Kleidergeschäft zu finden war – BARNES & NOBLE hatte kurz zuvor die Filiale in San Francisco geschlossen. Das war insgesamt nicht ganz so schlimm, denn der Weg ist sehr schön. Anschließend habe mir dann auch noch den Luxus geleistet, mit dem Cable Car zurückzufahren.

Wenn man also je ein Buch über Computer, Software-Entwicklung und IT kaufen möchte, dann sollte man nicht nach San Francisco fahren, denn das ist in dieser 800 000-Einwohner-Stadt leider nicht (mehr) möglich. Es scheint tatsächlich so zu sein, als dass es dafür nur noch AMAZON und Co. gibt.

Schade eigentlich.

Kurzbiographie



MICHAEL WIEDEKING (michael.wiedeking@mathema.de) ist Gründer und Geschäftsführer der MATHEMA Software GmbH, die sich von Anfang an mit Objekttechnologien und dem professionellen Einsatz von Java einen Namen gemacht hat. Er ist Java-Programmierer der ersten Stunde, „sammelt“ Programmiersprachen und beschäftigt sich mit deren Design und Implementierung.

User Groups

Fehlt eine User Group? Sind Kontaktdaten falsch?
Dann geben Sie uns doch bitte Bescheid.

BOOKWARE

Henkestraße 91, 91052 Erlangen
Telefon: 0 91 31 / 89 03-0
Telefax: 0 91 31 / 89 03-55
E-Mail: redaktion@bookware.de

Java User Groups

DEUTSCHLAND

JUG Berlin Brandenburg

<http://www.jug-bb.de>
Kontakt: Herr Ralph Bergmann (orga@jug-bb.de)

Java User Group Saxony

Java User Group Dresden
<http://www.jugsaxony.de>
Kontakt: Herr Torsten Rentsch (torsten@jugsaxony.de)
Herr Falk Hartmann (falk@jugsaxony.de)
Herr Kristian Rink (kristian@jugsaxony.de)

rheinjug e.V.

Java User Group Düsseldorf
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf
<http://www.rheinjug.de>
Kontakt: Herr Heiko Sippel (info@rheinjug.de)

ruhrjug

Java User Group Essen
Glaspavillon Uni-Campus
Universitätsstr. 12, 45127 Essen
<http://www.ruhrjug.de>
Kontakt: Herr Heiko Sippel (heiko.sippel@ruhrjug.de)

JUGF

Java User Group Frankfurt
<http://www.jugf.de>
Kontakt: Herr Alexander Culum
(jvausergroupfrankfurt@googlemail.com)

JUG Deutschland e.V.

Java User Group Deutschland e.V.
c/o asc-Dienstleistungs GmbH
Ehrengard-Schramm-Weg 11, 37085 Göttingen
<http://www.java.de> (office@java.de)

JUG Hamburg

Java User Group Hamburg
<http://www.jughh.org>

JUG Karlsruhe

Java User Group Karlsruhe
Universität Karlsruhe, Gebäude 50.34
Am Fasanengarten 4, 76131 Karlsruhe
<http://jug-karlsruhe.de>
jug-karlsruhe@gmail.com

JUGC

Java User Group Köln
<http://www.jugcologne.org>
Kontakt: Herr Michael Hüttermann
(michael@huettermann.net)

jugm

Java User Group München
Jupiterweg 8, 85586 Poing
<http://www.jugm.de>
Kontakt: Herr Andreas Haug (ah@jugm.de)

JUG Münster

Java User Group für Münster und das Münsterland
<http://www.jug-muenster.de>
Kontakt: Herr Thomas Kruse (tkjugi@sforce.org)

JUG MeNue

Java User Group der Metropolregion Nürnberg
c/o MATHEMA Software GmbH
Henkestraße 91, 91052 Erlangen
<http://www.jug-n.de>
Kontakt: Frau Alexandra Specht
(alexandra.specht@jug-n.de)

JUG Ostfalen

Java User Group Ostfalen
(Braunschweig, Wolfsburg, Hannover)
Siekstraße 4, 38444 Wolfsburg
<http://www.jug-ostfalen.de>
Kontakt: Uwe Sauerbrei (info@jug-ostfalen.de)

JUGS e.V.

Java User Group Stuttgart e.V.
c/o Dr. Michael Paus
Schönaicherstraße 3, 70597 Stuttgart
<http://www.jugs.org>
Kontakt: Herr Dr. Micheal Paus (mp@jugs.org)
Herr Hagen Stanek (hs@jugs.org)

SCHWEIZ

JUGS

Java User Group Switzerland
Postfach 2322, 8033 Zürich
<http://www.jugs.ch> (info@jugs.ch)
Kontakt: Frau Ursula Burri

.Net User Groups

DEUTSCHLAND

.NET User Group OWL

http://www.gedoplan.de/cms/gedoplan/ak/ms_net
% GEDOPLAN GmbH
Stieghorster Str. 60, 33605 Bielefeld

.Net User Group Bonn

.NET User Group "Bonn-to-Code.Net"
Langwartweg 101, 53129 Bonn
<http://www.bonn-to-code.net> (mail@bonn-to-code.net)
Kontakt: Herr Roland Weigelt

Dodned

.NET User Group Franken
<http://www.dodned.de>
 Kontakt: Herr Bernd Hengelein
 Herr Thomas Müller (consulting@tom-mue.de)

.net Usergroup Frankfurt

c/o Thomas Sohnrey, Agile IService
 Mendelssohnstrasse 80, 60325 Frankfurt
<http://www.dotnet-ug-frankfurt.de>
 Kontakt: Herr Thomas 'Teddy' Sohnrey
 (thomas.sohnrey@gmx.de)

.NET DGH

.NET Developers Group Hannover
 Landwehrstraße 85, 30519 Hannover
<http://www.dotnet-hannover.de>
 Kontakt: Herr Friedhelm Drecktrah
 (friedhelm@drecktrah.de)

INdotNET

Ingolstädter .NET Developers Group
<http://www.indot.net>
 Kontakt: Herr Gregor Biswanger
 (gregor.biswanger@web-enliven.de)

DNUG-Köln

DotNetUserGroup Köln
 Goldammerweg 325, 50829 Köln
<http://www.dnug-koeln.de>
 Kontakt: Herr Albert Weinert (info@der-albert.com)

.Net User Group Leipzig

Brockhausstraße 26, 04229 Leipzig
<http://www.dotnet-leipzig.de>
 Kontakt: Herr Alexander Groß (agross@dotnet-leipzig.de)
 Herr Torsten Weber (tweber@dotnet-leipzig.de)

.NET Developers Group München

<http://www.munichdot.net>
 Kontakt: Hardy Erlinger (hardy.erlinger@hotmail.com)

.NET User Group Oldenburg

c/o Hilmar Bunjes und Yvette Teiken
 Sachsenstr. 24, 26121 Oldenburg
<http://www.dotnet-oldenburg.de>
 Kontakt: Herr Hilmar Bunjes
 (hilmar.bunjes@dotnet-oldenburg.de)
 Yvette Teiken (yvette.teiken@dotnet-oldenburg.de)

.NET User Group Paderborn

c/o Net at Work Netzwerksysteme GmbH,
 Am Hoppenhof 32, 33104 Paderborn
<http://www.dotnet-paderborn.de>
 (raacke@dotnet-paderborn.de)
 Kontakt: Herr Mathias Raacke

.Net Developers Group Stuttgart

Tieto Deutschland GmbH
 Mittlerer Pfad 2, 70499 Stuttgart
<http://www.devgroup-stuttgart.de>
 (GroupLeader@devgroup-stuttgart.de)
 Kontakt: Frau Catrin Busley

.net Developer-Group Ulm

c/o artiso solutions GmbH
 Oberer Wiesenweg 25, 89134 Blaustein
<http://www.dotnet-ulm.de>
 Kontakt: Herr Thomas Schissler (tschissler@artiso.com)

ÖSTERREICH**.NET Usergroup Rheintal**

c/o Computer Studio Kogoj
 Josefgasse 11, 6800 Feldkirch
<http://usergroups.at/blogs/dotnetusergrouprheintal/default.aspx>
 Kontakt: Herr Thomas Kogoj (thomas@kogoj.com)

.NET User Group Austria

c/o Global Knowledge Network GmbH,
 Gutheil Schoder Gasse 7a, 1101 Wien
<http://usergroups.at/blogs/dotnetusergroupaustria/default.aspx>
 Kontakt: Herr Christian Nagel (ug@christiannagel.com)



Die Java User Group
 Metropolregion Nürnberg
 trifft sich regelmäßig
 einmal im Monat.

Thema und Ort werden über
www.jug-n.de
 bekannt gegeben.

Weitere Informationen
 finden Sie unter:
www.jug-n.de

■ Web-Anwendungen mit Apache Wicket

Umsetzung von Web-Projekten mit
Apache Wicket
Termine nach Vereinbarung,
1.180,- € (zzgl. 19% MwSt.)

■ Design Patterns und ihre Anwendung

Überblick über die wichtigsten Entwurfsmuster
und deren Konzepte
835,- € (zzgl. 19% MwSt.),
7. – 8. November 2011

■ Neues in Java 7

Die nächste Java Generation
835,- € (zzgl. 19% MwSt.),
7. – 8. November 2011

■ Testkonzepte

Software-Tests zur
kontinuierlichen Sicherung
der Qualität
30. Nov. – 2. Dez. 2011
1.315,- €
(zzgl. 19% MwSt.)



MATHEMA

MATHEMA Software GmbH | Telefon: (09131) 89032-0 | Internet: training.mathema.de
Henkestraße 91, 91052 Erlangen | Telefax: (09131) 89032-55 | E-Mail: training@mathema.de

Einstellungssache

Software-Entwickler (m/w)

Software-Architekt (m/w)

Selbstständiges, motiviertes Arbeiten und Denken im Team ist Ihnen wichtig? Sie haben einen gesunden Ehrgeiz und Lust, in internen wie externen Projekten Verantwortung zu übernehmen? Für uns die richtige Einstellung!

Wenn Sie zudem über mehrjährige Berufserfahrung in den Schwerpunkten verteilte Technologien, Komponenten- und Objektorientierung (insbesondere im Umfeld der Java EE) verfügen oder offen sind, etwas darüber zu lernen, dann sollten wir uns kennen lernen.

Wir freuen uns auf Ihre Bewerbung.



MATHEMA

Das Allerletzte

Hardware

Neue 2,5-Zoll Festplatte mit
15000 U/min von TOSHIBA

Mit der MK3001GRRB/R kündigt
TOSHIBA eine 2,5 Zoll große Festplatte
an, die 15.000.000 Umdrehungen
pro Minute schafft.

Dies ist kein Scherz!

Diese Ankündigung wurde tatsächlich in der freien
Wildbahn angetroffen.

Ist Ihnen auch schon einmal ein Exemplar dieser
Gattung über den Weg gelaufen?
Dann scheuen Sie sich bitte nicht, uns das mitzuteilen.

Der nächste KAFFEEKLATSCH erscheint Anfang November 2011.



Herbstcampus

Wissenstransfer par excellence

3.– 6. September 2012
in Nürnberg