

Red Stack

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

Magazin

User Experience



Mehr Performance

Tuning für SQL
und PL/SQL

Im Interview

Paul Bleile, Leiter
Datenbanken,
Coop Schweiz



Oracle Forms

Das neue Release unter
der Lupe

ORACLE®

**Testen Sie die Oracle Cloud
mit Credits im Wert von**

\$300



**Registrieren Sie sich und erhalten Sie eine Gutschrift
für das Pay-as-you-go-Abonnement von Oracle
Cloud-Services in drei einfachen Schritten:**

cloud.oracle.com/de_DE/tryit



Yann Neuhaus
Vorstand der Schweizer
Oracle Usergroup (SOUG)

Liebe Mitglieder, liebe Leserinnen und Leser,

das Zeitalter der digitalen Transformation hat begonnen. Der gegenwärtige Mangel an Informatikern zeigt, dass alle Unternehmen mehr oder weniger groß angelegte Digitalisierungs-Projekte gestartet haben. Es mangelt nicht an Auswahl: Modernisierung der On-Premise-Umgebung oder Cloud-Computing mit Infrastructure as a Service, Platform as a Service oder Firmenlösungen durch Software as a Service. Hinzu kommt das Testen neuer Firmenprozesse dank Big Data und Data Science oder auch bessere Integration der Entwicklung und des Betriebs durch DevOps-Konzepte. Welchen Platz nimmt Oracle in diesem hektischen Wettlauf hin zur Digitalisierung ein? Vor allem: Welchen Platz nehmen Sie ein?

Die Cloud ist erst einmal eine neue Art, IT zu konsumieren, das ist jedoch letztendlich nicht revolutionär. Einfach gesagt, ist die Cloud von der Virtualisierung inspiriert, die seit Jahren in den Unternehmen eingesetzt wird. Diese wird kombiniert mit Hosting-Diensten, ergänzt durch „Back-charging“ und „Provisioning“-Funktionen.

Die Cloud wird sich durchsetzen und unumgänglich werden, das ist klar, aber nicht unbedingt genauso schnell, wie manche Anbieter sich das wünschen – und auch nicht unbedingt im gleichen Umfang. Das Cloud-Angebot ist noch sehr instabil; Oracle hat beispielsweise am 19. September 2017 seine Strategie geändert und Bring Your Own Licences genehmigt. Diese Änderungen werden oft als Verbesserung des Dienstes verkauft, dahinter stecken allerdings nur Verkaufsbemühungen.

Ich wünsche Ihnen viel Erfolg mit Ihren Projekten.

Ihr

MUNIQSOFT

Consulting

Hochverfügbarkeit mit IQ

Sicherheit vor teuren Ausfallzeiten: Mit dem richtigen Konzept sind Ihre Daten und Server vor Systemausfällen optimal geschützt.

Nutzen Sie die Erfahrung von Muniqsoft

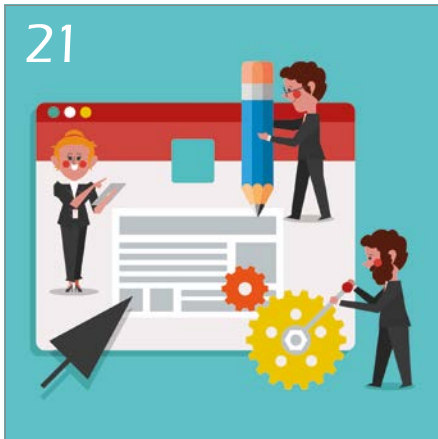
ORACLE® Gold Partner

Specialized
Oracle Database

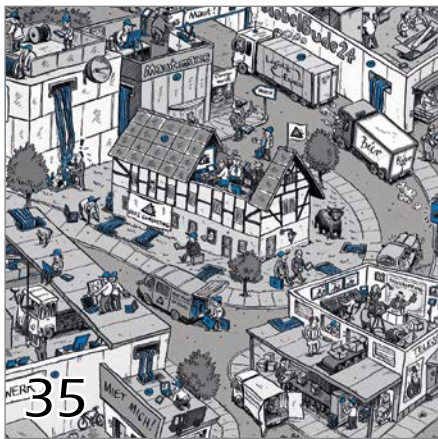


Jetzt Beratungstermin vereinbaren:
+49 (0)89 6228 6789-21

www.muniqsoft.de



Es ist dringend notwendig, den Nutzer wieder ins Zentrum zu rücken



Wie funktioniert ein Chatbot und warum sollten wir auf Chatbots setzen?



Ein tieferes Verständnis für die effiziente Nutzung der erweiterten

Einleitung

- 3 Editorial
- 5 Timeline
- 10 Günther asks Tirthankar
Thema: In-Memory
- 14 „Cloud ist für mich zunächst nur eine andere Formulierung für Outsourcing ...“
Interview: Paul Bleile

- 56 Das neue Forms-Release 12.2.1.3 unter der Lupe
Frank Hoffmann
- 60 Eleganten und effizienten Code schreiben
Jürgen Sieben

User Experience

- 17 User Experience – eine Einführung
Alexej Schneider
- 21 Bessere UX durch Methoden wie Design Thinking und Design Sprints
Maximilian Liesegang
- 26 Das Web ist wieder auf dem Weg zu alter Stärke
Tobias Huber
- 31 Arbeitsagentur 2.0
Michael Adam
- 35 Mit Chatbots und Voice zum flexiblen, Kontext-unterstützten Arbeiten in der digitalen Welt
Christian Ochsenkühn

Intern

- 65 Termine
- 66 Neue Mitglieder
- 66 Impressum
- 66 Inserenten

Entwicklung

- 39 SQL-Performance-Tuning mit „Analytic Functions“
Christian Schwitalla
- 43 WebLogic WLST-Programmierung unter Eclipse mit dem Oracle-OEPE-Plugin und PyDev
Michael Schulze
- 51 PL/SQL-Performance, Bulk Binding und Pipelining
Athanasios Manolopoulos

◆ Timeline

7. September 2017

In lockerer Atmosphäre hören die Teilnehmer des DOAG 2017 IMC Day in Berlin spannende Vorträge rund um Cloud-Themen und Middleware-Trends. Der Tag bietet mit seinem offenen Diskussionsformat eine ideale Plattform für den Wissensaustausch. Nach der Eröffnung durch Jan-Peter Timmermann, Leiter der DOAG Infrastruktur und Middleware Community, berichtet Johannes Michler von Promatis über die Migrationswege der Oracle SOA Suite von 10/11 nach 12c. Dabei klärt er die Teilnehmer über das Upgrade oder eine Out-of-the-place-Migration auf sowie darüber, welche Lizenzen für den Betrieb notwendig sind. Forms und Reports sind dann das Thema der Präsentation von Frank Burkhardt von Quality Technology Solutions. Er geht auf sicherheitsrelevante Dinge bezüglich Forms ein, teilt seine Erfahrungen mit der Anbindung einer Forms-Reports-Umgebung an das Microsoft Active Directory und gibt einen Überblick über die komplexe Architektur, die mehrere Komponenten wie Oracle Internet Directory und WebGate beinhaltet. Auch Jan-Peter Timmermann gibt im Anschluss seine Erfahrungen aus zahlreichen Forms-Projekten weiter und referiert über typische Probleme und Lösungsansätze. Die nächsten zwei Präsentationen befassen sich dann mit der Cloud-Thematik. Jan Brosowski von Oracle gibt einen aktuellen Überblick über die Public- sowie Bare-Metal-Cloud. Danilo Schmiedel von Opitz Consulting vergleicht anschließend unterschiedliche Cloud-Modelle aus der Finanzperspektive. Zum Abschluss wird es dann noch einmal technisch: Grzegorz Lysko von Opitz Consulting zeigt, dass fast alles automatisiert werden kann, und präsentiert einen Lösungsansatz für die automatische Middleware-Installation mittels Ansible. Der nächste Vortrag von Mario Nolte, ebenfalls Opitz Consulting, beschäftigt sich mit der Java Garbage Collection. Der Referent analysiert mit Out of Memory das wohl bekannteste Problem und erläutert unterschiedliche Ursachen sowie mögliche Lösungen.



Die interessanten Präsentationen begeistern die Zuhörer

8. September 2017

Clemens Bleile, Senior Consultant bei dbi services, erläutert in einem einstündigen Webinar, wie man Datenbank-Abfragen nach einem Upgrade schnellstmöglich beschleunigt und typischen Performance-Problemen nach der Migration begegnet. Dabei stellt er mögliche Workarounds vor, denn viele Probleme beim Upgraden einer Oracle-Datenbank lassen sich auf Performance-Engpässe nach der Migration zurückführen.

11. September 2017

Das JavaLand-Programmkomitee trifft sich in Berlin. Aufgrund der deutlich mehr Vortragseinreichungen als in den Jahren zuvor kann ein sehr interessantes und abwechslungsreiches Programm für die JavaLand 2018 zusammengestellt werden.



Die JavaLand 2018 findet vom 13. bis 15. März 2018 im Phantasialand in Brühl statt

14. September 2017

Der DOAG-Vorstand trifft sich zur mehrtägigen Sitzung. Im Mittelpunkt stehen die Strategiethemata für das Jahr 2018. Der Vorstand spricht sich bewusst konträr zur Oracle-Cloud-only-Strategie und für eine hybride Positionierung aus: On-Premise soll demnach weiterhin im Fokus der DOAG stehen, während die Cloud-Themen entlang der vorhandenen Struktur und Produktgruppen auf allen Ebenen mitbehandelt werden sollen.

21. September 2017

Der Zeitpunkt für den DOAG Reporting Day 2017 in Kassel hätte nicht günstiger gewählt werden können: Erst Ende August erschien die neue Version 12.2.1.3 der Oracle Fusion Middleware und in den Release Notes wird auch das endgültige Ende von Oracle Reports bekanntgegeben. Knapp vierzig Teilnehmer diskutieren einen Tag lang über alternative Reporting-Lösungen und tauschen Erfahrungen aus. Mit dem Ende des Premier Support im Oktober 2020 und des Extended Support im Oktober 2023 ist für Kunden von Oracle Reports damit der Planungshorizont festgelegt. „Es gibt viel zu tun“, sagt Jürgen Menge von Der IT-Macher in seiner Einführungspräsentation und gibt dann einen Überblick über mögliche Ablösestrategien für Oracle Reports. Daran schließen sich fünf Präsentationen an, in denen alternative Reporting-Lösungen vorgestellt und live demonstriert werden. Sie reichen vom Oracle BI Publisher, der von Oracle als Alternative zu Oracle Reports empfohlen wird, über die verbreiteten Open-Source-Lösungen JasperReports und Eclipse BIRT bis hin zu Anwendungen, die die Datenbank-Technologien von Oracle stärker nutzen, wie zum Beispiel PL/PDF und APEX Office Print. Sie werden von Referenten präsentiert, die sich in der Vergangenheit bewusst für diese Lösungen entschieden haben und deshalb auch sehr engagiert auftreten. Interessant sind vor

allein die unterschiedlichen Ansätze, um die zahlreichen Anforderungen an das Reporting im Unternehmen abzudecken. Trotz des gefüllten Tagesprogramms gibt es auch während der Vorträge immer wieder zahlreiche interessante Diskussionen, was das hohe Interesse an dem Thema widerspiegelt.

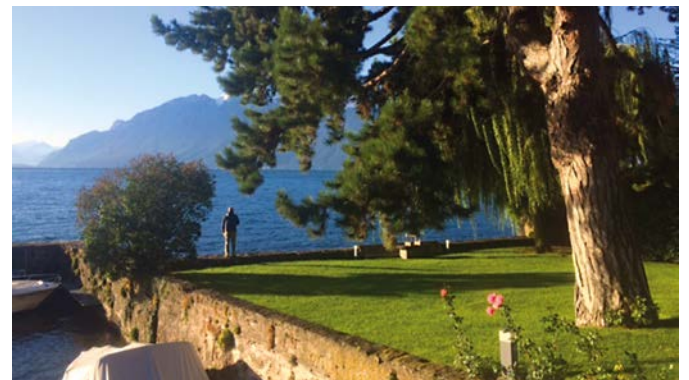
21. September 2017

Die vierten DOAG Big Data Days finden in diesem Jahr mit rund 90 Teilnehmern in Kassel statt. Die Themen „künstliche Intelligenz“, „Data Lakes“ und „Analytics-Lösungen“ sorgen während der zweitägigen Konferenz für viele Gespräche unter den Teilnehmern. Das neue Format mit zwei parallel laufenden Veranstaltungen kommt gut an: Die Besucher können frei zwischen den einzelnen Tracks des Reporting Day und Geodata Day wählen. In der ersten Vortragsreihe „Oracle und Big Data“ stellt Alfred Schlaucher von Oracle das Thema aus Sicht des Herstellers vor. Er führt im ersten Teil in das Thema ein – gut für Big-Data-Anfänger, die vielleicht noch nie Fachbegriffe wie „Hadoop“ gehört haben. Im Mittelpunkt stehen dann Oracle SQL Big Data, die Koexistenz und der Datenaustausch zwischen den unterschiedlichen Systemen. Im nächsten Vortrag stellte Andreas Buckenhofer von Daimler TSS GmbH einen interessanten Praxisbericht über Data Lakes vor. Zahlreiche Erfahrungen aus realen Projekten prägen die Präsentation. Big Data und Data Lakes gehörten mittlerweile zum Projektalltag der Daimler-Tochter und der Referent sieht auch in Zukunft die Rolle des Data Warehouse ungefährdet. Die Frage, ob Big Data in der Cloud oder doch lieber On-Premises bleiben sollte, stellte Guido Schmutz von Trivadis. Basierend auf Projekterfahrungen zeichnet er ein sehr differenziertes Bild: Eine simple Antwort auf diese Frage gibt es nicht. Zumal es sich hierbei nicht um eine Entweder-oder-Frage handele. Hybride Architekturen können durchaus eine interessante Alternative sein. Timm Marschall von der Universität Stuttgart referiert anschließend über agile, analyseorientierte Datenhaltung in Big-Data-Umgebungen. Zum Abschluss des Tages berichtet Ulrike Schwinn von Oracle über Tipps und Tricks zur Speicherung und den Zugriff heterogener Datenmengen in der Datenbank. Am Abend gibt es dann viel Zeit für ein gemütliches Beisammensein und Networking. Der zweite Tag beginnt mit Olaf Nimz von Trivadis. In dem Vortrag „Datenqualität im DWH und Data Lake: Automatische Anomalie-Erkennung“ präsentiert er interessante Methoden, die es erlauben, nicht plausible Daten bereits im Vorfeld einer ETL-Strecke zu erkennen. In einem Doppelvortrag stellen Detlef Schröder und Karin Patenge von Oracle einen spannenden

Showcase vor. Bei der Anki Overdrive handelt es sich um eine Rennbahn für kleine Modellautos. Sie ist während der beiden Tage im Tagungsraum aufgebaut. Das Besondere daran sind die miteinander vernetzten Sensoren. Die Echtzeitdaten fließen in die Cloud; sie können von den Teilnehmern betrachtet und ausgewertet werden. Gesteuert sind die Autos per Smartphone-App. Im Anschluss daran präsentiert Jan Ott von Trivadis das Messaging-System Apache Kafka und im nächsten Vortrag „Vergleich von großen relationalen DBs im Cloud Zeitalter“ von Matthias Fuchs, Esentri, geht es um die Fragestellung, welche alternativen Datenbanksysteme unter dem Einfluss der Cloud-Architektur entstanden sind und welche Stärken und Schwächen sie haben. Er vergleicht die Oracle-Datenbank mit Amazon Aurora und Google Spanner. Mit den letzten zwei Vorträgen rückt die künstliche Intelligenz in den Mittelpunkt und als Abschluss präsentiert Sigrid Keydana von Trivadis weitere interessante Aspekte des maschinellen Lernens. Im Abschluss entwickelt sich eine rege Diskussion um künftige Auswirkungen von künstlicher Intelligenz auf unseren Alltag.

21. September 2017

Die SOUG-Westschweiz feiert den 30. Geburtstag der SOUG im Château de Glérolles am Genfer See. In gemütlicher Atmosphäre profitieren die Teilnehmer von vier spannenden Sessions über Sicherheit (Acceleris), Performanz (dbi services), Oracle Cloud (Trivadis) sowie die Zusammenarbeit von CERN und Oracle (CERN). Nach dem technischen Part wird das Jubiläumsprogramm mit einer Einführung in die Önologie nebst Besichtigung des Weinkellers und Weinverkostung fortgesetzt. Das Event endet im sonnigen Schlosspark am Wasser. Yann Neuhaus, Koordinator der SOUG-Westschweiz, bedankt sich bei allen Teilnehmern und freut sich auf weitere tolle Events.



Bestes Wetter am Genfer See



Die Echtzeitdaten der Rennbahn fließen in die Cloud

26. September 2017

In Baden-Dättwil treffen sich die Schweizer Oracle-Anwender zum dritten SOUG Day 2017. Auf die Teilnehmer warten drei interessante Vortragsstreams zu den Themen „Datenbank“, „Infrastruktur“ und „Security/Performance“. Als Referenten können kompetente Experten gewonnen werden, die teilweise lange Anreisen unternehmen; zudem sind vier Oracle-ACEs vor Ort. Alle Vortragenden geben mit ihrem fundierten Wissen ihre Erfahrungen an die Teilnehmer weiter und tragen mit ihrer Art, Informationen zu vermitteln, zu einem gelungenen Event bei. Nach den technischen Vorträgen endet der SOUG Day mit einem Apéro, bei dem in un-

gezwungener Atmosphäre die Möglichkeit besteht, das Gehörte zu diskutieren und das eigene Netzwerk zu erweitern.



Viel Fachwissen auf dem SOUG Day

30. September 2017

Zum Frühbucherschluss der DOAG 2017 Konferenz sind die Buchungen der Teilnehmer noch etwas verhalten. Die DOAG wird deshalb in den kommenden Wochen nochmals konzentriert die Vorteile eines Konferenzbesuchs hervorheben.

1. Oktober 2017

Beim Auftakt zur OpenWorld in San Francisco, dem größten Oracle-Event des Jahres, fällt vor allem eins auf: Es ist alles etwas weniger pompös. Der Yerba-Buena-Park, in dem früher Bühnen standen, Partys stiegen und ansonsten das Catering platziert war, ist leer und ein ganz normaler öffentlicher Park. Die großen Oracle-Themen in diesem Jahr fokussieren sich, wie nicht anders zu erwarten, auf die Cloud. Larry Ellison, Oracle-CEO bringt in seiner Keynote Kernaussagen zur Zukunft der Datenbank: Die Cloud-Datenbank 18c soll in Zukunft autonom sowie vollautomatisch arbeiten und das besser können als unter menschlicher Administration. Sie soll besser und frühzeitig Tuning-Maßnahmen einleiten, Patches im laufenden Betrieb einspielen, den Storage optimal verwalten und schneller auf Gefahrensituationen reagieren. Vor allem aber soll die neue Datenbank-Generation wesentlich günstiger sein als die der Konkurrenz. Sie wird bis Ende dieses Jahres verfügbar sein. Der ebenfalls neue Blockchain-Cloud-Service soll vor allem für Unternehmenskunden interessant sein, die einen vollständig gemanagten Ansatz bevorzugen. Basis für den Service ist das Open-Source-Projekt Hyperledger Fabric, das auch IBM mit aufgebaut hat. Blockchain arbeitet genau wie Bitcoin mit dem verteilten Speichern von signierten Daten. Dabei geht es um Transaktionen und Codes (Smart Contracts), die aufgrund bestimmter transaktionaler oder zeitlicher Trigger (garantiert) ausgeführt werden. Es gibt eine ganze Reihe von Vorträgen auf der OpenWorld und der JavaOne zu diesem Thema. Analysten gehen davon aus, dass im Jahr 2017 etwa zehn Prozent des Bruttoinlandsprodukts über Blockchain-Transaktionen verwaltet werden und dass die Technologie nicht nur die Finanzbranche umkrempeln wird. Laut den Oracle-Plänen werden zukünftige Anwendungen für die Bereiche Finanzen, Personal, Supply Chain, Fertigung, Handel, Kundenservice, Marketing und Vertrieb AI-basierte Funktionen besitzen. Die Applikationen erweitern AI- und Machine-Learning-Fähigkeiten innerhalb des Cloud-Portfolios. Im Gegensatz zu eigenständigen Anwendungen, die eine Integration benötigen und zusätzliche Schritte im Entscheidungsfindungsprozess erfordern, binden die neuen Applikationen AI-Funktionen direkt in Oracle Enterprise Resource Planning Cloud, Oracle Human Capital Management Cloud, Oracle Supply Chain

Management Cloud und Oracle Customer Experience Cloud Suite ein. Die Apps sollen in Echtzeit reagieren können, lernen und sich anpassen, basierend auf historischen und dynamischen Kundendaten. Ziel ist es, den Anwendern in Unternehmen zu ermöglichen, künstliche Intelligenz direkt in ihren bestehenden Tools zu nutzen, anstatt ihre Daten in eine externe AI-Lösung zu exportieren. Oracle kündigt zudem eine neue regionale Struktur an: Demnach soll es in Europa drei Datacenter geben, die auch isoliert und komplett getrennt vom Rest einzeln oder in Kombination genutzt werden können. Damit wäre dann die Grundlage für eine Nutzung auf Basis des EU-Rechts geschaffen.

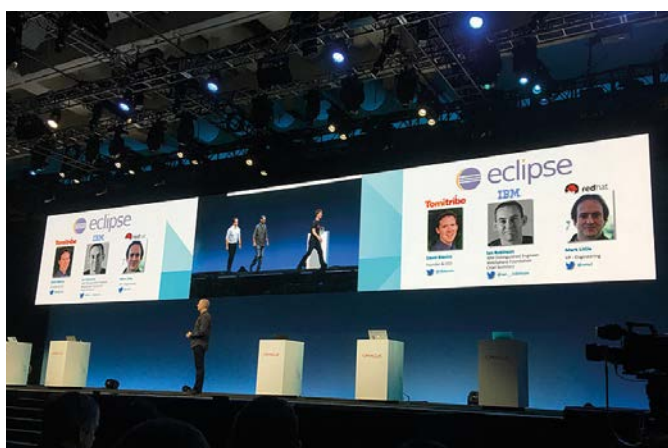


Oracle-Chef Larry Ellison bei seiner Keynote

1. Oktober 2017

Parallel zur Oracle OpenWorld findet ebenfalls in San Francisco die JavaOne statt. Der Sonntag ist hier traditionell der Community-Tag mit einer Mischung aus technischen Vorträgen von und für Java User Groups. Andreas Badelt, stellv. Leitung der DOAG Java Community, berichtet vor Ort: Der Eindruck, dass nicht nur die Oracle OpenWorld, sondern auch die JavaOne etwas weniger Besucher hat als letztes Jahr, scheint sich zu bestätigen. Der große Saal im Moscone North Center wirkt bei der diesjährigen Keynote etwas leerer. Dabei gab es jede Menge zu erzählen und zu zeigen, denn in der Java-Welt ist in den letzten Wochen und Monaten viel passiert: Java 9 und Java EE 8 wurden veröffentlicht und mit dem Projekt Eclipse Enterprise for Java (EE4J) übernimmt die Eclipse Foundation federführend die Weiterentwicklung von Java EE. Georges Saab, Vice President of Software Development bei Oracle, führt durch die Keynote und holt sich zuerst eine Reihe von Personen zur Unterstützung auf die Bühne. Zunächst berichteten Teilnehmerinnen aus dem JDuchess-Programm (die weibliche Duchess als Gegenstück zum Duke), die über das großartige „Community Involvement“ sprechen und natürlich insbesondere mehr Frauen für die Java-Welt und generell für die IT begeistern wollen. Dann erscheint Michael Green, Vice President für System Technologies and Optimization bei Intel, der zunächst zeigt, dass die Intel-Prozessoren Java, verglichen mit 2006, bis zu 110-mal schneller ausführen. Auch die Zusammenarbeit zwischen Intel und Oracle hat Früchte getragen: Durch das Vector-API werden nun die Möglichkeiten paralleler Vektorberechnungen der Intel-Prozessoren genutzt. An einem kleinen Beispiel-Programm für die Bildbearbeitung wurde eine sechsfache Beschleunigung gegenüber Java 8 demonstriert. Mindestens ebenso interessant ist auch, dass Intel eine ganze Reihe von Frameworks zu verschiedenen Themen wie Deep Learning (BigDL) oder Benchmarking (Data-Bench) als Open Source auf GitHub freigegeben hat. Der Trend scheint sich zu verstärken, dass alles, was nicht als Alleinstellungsmerkmal gegenüber dem Wettbewerb taugt, zu Open Source wird. Mark Cavage, Vice President of Product Development bei Oracle, holt sich als Er-

stes Vertreter der MicroProfile-Initiative von Red Hat, IBM und Tomitribe zu einem kurzen Interview auf die Bühne. Diese äußern sich positiv zur Übergabe von Java EE an die Eclipse Foundation und zu der damit verbundenen Freigabe der Test Compatibility Kits als Open Source. Insgesamt soll wohl demonstriert werden, dass die Zusammenarbeit funktioniert, die dann ja in Zukunft komplett innerhalb der Eclipse Foundation stattfinden könnte. Auch das Stichwort „Container“ darf während einer JavaOne-Keynote nicht fehlen. Der Gründer von Heptio spricht insbesondere über Kubernetes und zwei Oracle-Mitarbeiter stellen die neue Version des Java Flight Recorder vor, der mit seinen Diagnose- und Drill-Down-Fähigkeiten für Container einen sehr guten Eindruck macht. Gleich danach wird es „serverless“ mit dem Fn Project. Nach einer kurzen Demo durch den dafür zuständigen Oracle Vice President Software Development, Chad Arimura, gibt dieser zusammen mit Mark Cavage, Vice President of Engineering, das Projekt auf GitHub frei, um der Open-Source-Begeisterung die Krone aufzusetzen. Den letzten Teil der Keynote bestreitet Mark Reinhold, Chief Architect bei Oracle. Das große Thema ist natürlich Jigsaw. Neun Jahre, so Reinhold, hätten er und viele Kollegen daran gearbeitet, die „Classpath Hell“ abzuschaffen und den massiven Monolithen JDK aufzubrechen. Er tritt gleich der Aussage „Jigsaw breaks everything“ entgegen, denn „Jigsaw breaks some things“, so Reinhold. Aber nur bei Zugriffen auf interne APIs. Aber hier sind die Restriktionen ja deutlich gelockert, die meisten dieser Zugriffe werden erst in zukünftigen Releases gar nicht mehr funktionieren, sodass die Projekte Zeit für eine Migration haben, obwohl der Zeitrahmen nicht so groß ist. Die schon vor einiger Zeit angekündigte Änderung in den Release-Zyklen soll nämlich durchgezogen werden. Java erhält einen strikten „Release Train“ mit einem Release alle sechs Monate. Java 10 soll daher im März 2018 kommen. Die ersten fertigen Features, zum Beispiel eine deutliche Verbesserung der Switch-Statements mit implizitem Type Casting für weniger Boilerplate Code und mehr Lesbarkeit, stellt am Ende der Keynote noch Chief Language Architect Brian Goetz vor. Insgesamt ist es eine positive Keynote, auch wenn die ganz großen Knalleffekte ausbleiben – das meiste Pulver ist ja bereits schon im Vorfeld verschossen worden.



Die große Ankündigung: Java EE ist künftig bei der Eclipse Foundation angesiedelt

1. Oktober 2017

Für Dr. Frank Schönthaler, DOAG-Vorstand und Leiter DOAG Business Solutions Community, bringt die Oracle OpenWorld neben „Cloud Security“ und „Autonome Datenbank“ eine Flut

neuer SaaS-Produkte. Besonders beeindruckend ist die Oracle-SCM-Cloud, die nun eine komplette Suite mit Produkten für taktische und operative Planung sowie Management und Execution der Supply Chain einschließlich Transportmanagement und Produktion beinhaltet. Mit neuartigen Funktionen, die sämtliche Potenziale der zugrunde liegenden Oracle-Technologien ausschöpfen – Mobility, In-Memory, Machine Learning, KI, IoT und Big Data –, stößt Oracle das Tor in die digitale Ökonomie ganz weit auf. Gleichzeitig wurden für die Anwender der Oracle-On-Premise-Applikationen Wege in die Cloud aufgezeigt: Migration auf IaaS/PaaS oder Hybrid Coexistence, bei der Oracle-SaaS-Applikationen an die On-Premise-Applikationen angedockt sind. Dafür gibt es Standard-Integrationen im Oracle Integration Cloud Service. Nach Ansicht von Dr. Frank Schönthaler hat Oracle den Kunden und Partnern deutlich gemacht, dass sie auf ihrem Weg in die digitale Ökonomie gut aufgehoben sind. Gleichzeitig wurde jedoch auch offensichtlich, dass ein erheblicher Informationsbedarf bleibt, den der Hersteller alleine nicht abdecken kann. Hier kommen lokale Anwendergruppen wie DOAG, AOUG und SOUG ins Spiel, die diese Lücke mit ihren vielfältigen Veranstaltungs-, Informations- und Networking-Angeboten schließen.

2. Oktober 2017

Stefan Kinnen, Vorstandsvorsitzender der DOAG, lernt Kenneth Johansen, Country Leader Oracle Deutschland, kennen. Sie finden einen guten Draht zueinander, obwohl Stefan Kinnen gleich die brisanten Themen „EU Datenschutzverordnung“ und „Lizenzierung VMware“ anspricht. Beim Datenschutz ist Kenneth Johansen sehr optimistisch mit dem Ausbau des Datacenters Frankfurt. Stefan Kinnen erhält eine Einladung zu einem Termin vor Ort zur Klärung von Details. Zum Thema „VM-Lizenzierung“ will sich Kenneth Johansen rechtzeitig zur DOAG-Konferenz 2017 in Nürnberg vorbereiten und dort Fragen beantworten.

11. Oktober 2017

Das Berliner Expertenseminar mit Chris Antognini zur Oracle-Datenbank 12c ist sehr gut besucht. Es stellt die wichtigsten Neuerungen und Verbesserungen in vier Bereichen vor. Erstens ist der Query Optimizer nicht nur durch die Einführung von Adaptive Query Optimization verbessert, sondern auch durch Verbesserungen bei Objektstatistiken und Query Transformation erweitert. Zweitens stehen neue oder erweiterte Datenstrukturen zur Verfügung, drittens In-Memory-Option Features wie In-Memory Column Store und In-Memory Aggregation. Zudem führt die Version 12.2 Approximate Query Processing ein. Die Teilnehmer sind von den Neuerungen begeistert.

11. Oktober 2017

Fried Saacke, DOAG-Vorstand und Geschäftsführer, führt eine Telefonkonferenz mit Schulungsanbietern über das Vorhaben, eine DOAG-University zu gründen, sodass nach dem Ausstieg von Oracle eine neue zentrale Plattform entsteht, auf der Schulungen für deutschsprachige Oracle-Anwender angeboten werden.



*Dr. Dietmar Neugebauer
Ehemaliger DOAG-Vorstands-
vorsitzender*

Aus der Ferne betrachtet: die Oracle OpenWorld

Für mich als fast Stammgast auf der Oracle OpenWorld war die Eröffnungs-Keynote von Larry Ellison schon immer etwas Besonderes. Der Oracle-Chef hat nie ein Blatt vor den Mund genommen, Dinge erzählt, die seinen Mitarbeiter die Haare zu Berge haben stehen lassen, und die Demos haben auch nicht immer reibungslos funktioniert. Mit seinem Auftritt konnte er das Auditorium mitreißen. So auch in diesem Jahr, auch wenn ich es diesmal nur aus der Ferne auf dem YouTube-Video betrachtet habe.

Gleich zu Beginn wird allen klar, gegen wen es in diesem Jahr geht. Nicht gegen SAP, nicht gegen IBM und nicht gegen Microsoft; natürlich ist der Cloud-Marktführer Amazon das Angriffsziel. Die Ansage ist klar: Oracle ist günstiger (vertraglich garantiert halb so teuer wie Amazon), leistungsfähiger, sicherer und hat eine höhere Verfügbarkeit (99,995 Prozent aller Downtimes eingeschlossen). Was ist die Wunderwaffe von Oracle dafür? Sie nennt sich „Oracle Autonomous Database“ und „Automated Cyber Security“.

Zunächst ist mir nicht ganz klar, was für ein neues Produkt Oracle da auf den Markt bringt. Die Vorzüge einer Datenbank, die ohne jede administrative Eingriffe läuft, sich selbst ohne Downtime-Window patcht, sich völlig selbstständig für jede Abfrage tunt, dynamisch Hardware hinzukonfiguriert und damit völlig frei von menschlichen Fehlern ist – wie ist das möglich?

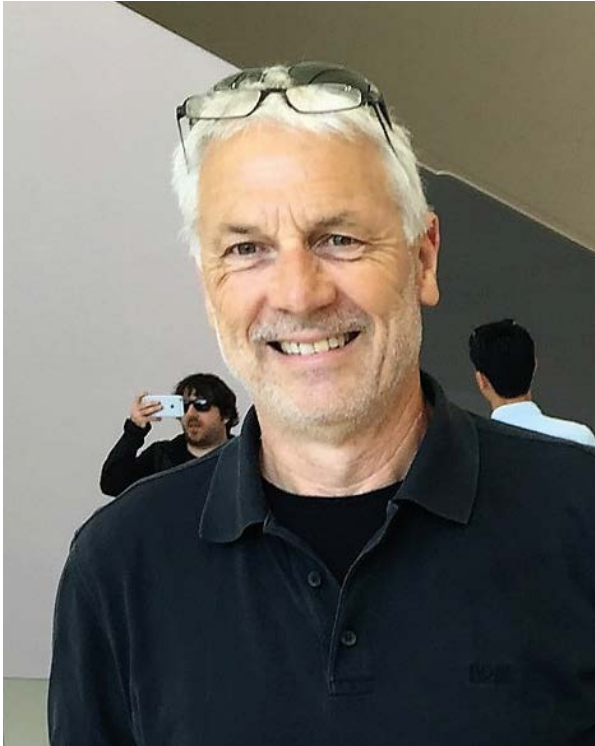
Erst als Larry Ellison die Folie mit dem technologischen Hintergrund des Produkts zeigt, komme ich schnell drauf. Hier sind die uns schon länger bekannten, technologisch immer weiter entwickelten Produkte von Oracle verbaut: RAC für System-Failure und Elastizität, RAC für Rolling Upgrades, Active Dataguard für Ausfallsicherheit, Transient Logical Standby für Release-Wechsel, Online Edition Based Redefinition für Applikationsänderungen sowie Flashback von Transactions, Tabellen und Datenbank, Multitenant, In-Memory, Adaptiv Query Optimization und Zero Data Loss Recovery Appliance.

Als Außenstehender vermittelt mir das alles den Eindruck, dass, seit Oracle selbst Database-Services in der Cloud betreiben will, sich alle Produktverantwortlichen zusammengesetzt und überlegt haben, wie sie ihre technologisch hochstehenden Produkte aus dem eigenen Haus zu einer Einheit zusammenführen können. Ein ganz wichtiger Schritt, den viele Kunden schon seit Jahren gefordert hatten.

Die Autonomous Database noch mit Produkt-Weiterentwicklungen zu versehen und mit dem auszustatten, was Larry Ellison als „Machine Learning“ bezeichnet – dies könnte uns wirklich zur Datenbank aus der Steckdose bringen – schneller vielleicht, als wir dies noch vor Jahren erwartet haben. Dasselbe gilt natürlich auch für die andere Neuerung, die Automated Cyber Security. Auch hier haben die Kunden auf eine Möglichkeit gewartet, möglichst schnell Security-Patches einspielen und auf Hacker-Angriffe reagieren zu können. Voraussetzung für den effektiven Einsatz ist hier natürlich auch eine schnelle Behebung von Sicherheitslücken, was nicht immer eine Stärke von Oracle ist. Aber hier wird ein gesamtheitlicher Ansatz sicherlich auch den notwendigen Druck erzeugen, um den Automatismus zu beschleunigen.

Einen Wermutstropfen hat Larry Ellison noch nebenbei erwähnt. Zunächst wird es die Autonomous Database – unter dem Namen Oracle 18c – nur für das Warehouse geben (ab Dezember 2017) und erst ab Juni 2018 für OLTP. Eine weitere Klarstellung stand ebenfalls auf einer Folie, nämlich dass es diese Produkte nicht nur in der Cloud geben wird, sondern auch On-Premises – allerdings nur auf Exadata-Maschinen.

Gutes Marketing gehört einfach zur Oracle OpenWorld dazu. In einem Tweet stand, dass es sowieso erstaunlich ist, wie Besucher mehr als tausend Dollar Teilnehmergebühr zahlen, um sich die neuen Produkte des Herstellers anschauen zu dürfen. Oracle beherrscht hier das Marketing perfekt, aber für mich präsentiert sich Oracle auch weiterhin als der führende Datenbank-Hersteller – nicht nur aus der Ferne betrachtet.



Günther

asks



Tirthankar

Günther Stürner, dbms publishing, interviewed Tirthankar Lahiri, Vice President for Data and In-Memory Technologies at Oracle, in September 2017

Hallo Tirthankar, thank you for your time! I have a few questions for you ...

Tirthankar: Of course, no problem. As long as they are not hard questions ...

In the last few years you visited Germany, German customers and several DOAG conferences and I am sure many of our readers know you. For all others could you please describe your current role within the Oracle database development organization?

Tirthankar: I am a Vice President in the Oracle Database Organization, responsible for a group known as the Data and In-Memory Technologies. This group is responsible for the Oracle Database Engine, including Transactions, Tables, Indexes, Space management as well as Database In-Memory. I am also responsible for the TimesTen In-Memory Database which is a separate database product.

In Germany you are known as "Mr. In-Memory" but beside that there are a couple of other important themes on your agenda and development groups you have to manage. Is it fair to say that In-Memory-technologies are the most exciting area in these days?

Tirthankar: This is truly a golden age for Databases! There is an explosive growth in data volumes, increasingly stringent real time requirements for data processing, new and emerging areas for data management such as IoT, machine learning, cloud, etc. In-

Memory is definitely a key enabler for this renaissance but data will continue to live on disk and Flash and efficient cross-tier mechanisms are required to enable the most cost-effective solutions. Another very interesting new technology space is with Non-Volatile memory. Previously the realm of academic research, Intel is very close to productizing their "3D Xpoint" non-volatile memory that looks just like DRAM except that its contents are safe from power failures. So lots of activity in the Database engine, spanning memory, non-volatile memory, flash and conventional storage.

Database In-Memory is the youngest member of the Oracle Database Options family but was from the start in June 2014 a kind of 'star' and very successful.

Tirthankar: Yes, Database In-Memory is one of the most successful new technology offerings in Oracle Database. It is definitely motivated by the increasing need for real-time analytics on any data source and with the amount of memory available on modern hardware today, it is definitely very attractive to store hot subsets of a database completely in-memory.

May I ask you to briefly describe the basic principles of the Database In-Memory technology?

Tirthankar: The basic idea is actually very simple and elegant. It's been known for a very long time that transaction processing (OLTP) workloads benefit best from a row-based format, where

issue DDL ALTER commands to populate them into the column store. Once this is done, analytic queries can transparently benefit from the In-Memory format. SAP has also enhanced its schema for its Business Warehouse (BW application) where they take advantage of database in-memory through a greater level of denormalization in the schema. Since the In-Memory format can be scanned at enormous speeds, it is often advantageous to replace joins with scans of denormalized tables.

Oracle Database In-Memory organize data in columnar format in main-memory. This format is known as excellent for analytic workloads but is sub-optimal – to be polite – for transaction workloads. What are the best fitting use-cases for Oracle Database In-Memory?

Tirthankar: What you say is true of pure columnar databases but as I have said earlier, we have a dual format architecture for the fastest possible performance for both OLTP and Analytic workloads. Transactions run against the row store, and changes made by transactions are applied through purely In-Memory operations to the column store. Since the column store is not persistent, maintaining it is very lightweight, cheaper than maintaining indexes. In fact, you can drop analytic indexes altogether once you have the column store since the column store gives you fast analytics on all columns. This can speed up DML operations considerably since they no longer have to maintain indexes. Dropping Analytic indexes also can help to reduce the overall size of the database. We have customers who have used Database In-Memory in OLTP heavy environments because they can replace their analytic indexes with the in-memory column store. Customers like Bosch have found that using the In-Memory column store, not only do their reporting queries run faster, but their DML and batch operations also speed up due to reduced index maintenance.

What is the best way for an Oracle customer to identify the applications and databases which are best for In-Memory?

Tirthankar: We do have an in-memory advisor to assist with this, but the basic thing to look for is a database or application that spends a large fraction of its time on analytic data access (queries that access a large amount of data, and perform index range scans or full table scans). This is of course true for most Data Warehouse or Datamart applications, but is also often true even for OLTP workloads. It is also important to consider how much time the application spends on maintaining analytic indexes – do DMLs run slow because of the large number of indexes on a table? If so, replacing them with the column store would help speed them up. We have a lot of information in AWR reports, segment level statistics, SQL monitor reports, etc. that can be used to identify candidate applications and database objects.

We talked about external competitors and the benefits of the Oracle Database In-Memory implementation. But what's about the internal competitor like Exadata? Why should someone decide for an Exadata machine when Database In-Memory makes his data warehouse so fast with conventional hardware or why should an Exadata customer use In-Memory?

Tirthankar: Exadata is not a competitor and it's unfortunate it's sometimes seen as such. Exadata is the optimal platform for Oracle database and the optimal platform for Database In-Memory. In-Memory complements the benefits of Exadata smart scan: For data

that is the hottest, it is fastest to keep it in the DRAM of the Exadata database nodes and use the full power of memory bandwidth (which is much higher than IO bandwidth) to process that data. For data that is too large to fit In-Memory or not that hot, Exadata smart scans provide very efficient access to that data. In Oracle database 12.2 onwards, if you have the In-Memory option, then on Exadata, the Exadata Storage Flash Cache is also formatted with the same In-Memory format as we have on the Database nodes. This allows offloaded smart scans to achieve similar performance to the scans that run on the Database nodes. To use a car based analogy, you can think of Exadata as a high end sports car, and Database In-Memory like a super-charger to make it run even faster.

In spring-time 2017 when 12.2.0.1 came out you and your team released some very new cool features for Database In-Memory. My personal favorites are In-Memory Expressions, In-Memory Virtual Columns and In-Memory JSON. It seems this is something really unique. Could you please explain this cool stuff?

Tirthankar: Yes, these are definitely very unique capabilities made possible by our dual format approach. Normally, the in-memory column store has data that is present in table columns. But often, workloads need to query a derived value such as "NET_PRICE" which is computed as $(PRICE * (1 - DISCOUNT_RATE) + (1 + TAX_RATE))$. A query that selects the net price must calculate this expression for every row that it accesses. With the In-Memory expressions feature, this net price can be stored as an in-memory virtual column in the column store: When the table is populated into memory, the "net price" column is also pre-populated using the values present within the same row. This avoids the need to recompute the values later on. Also, as the base values are modified, the In-Memory expression columns are automatically recomputed as the column store is maintained. We've seen that this can help complex queries run 3x-5x faster by eliminating repeated computation. Note that this virtual column is not actually present in the database. It is only present within the In-Memory column store. The same idea can be used for JSON. JSON is a flexible text based format for processing web data but it is very inefficient to query in its native text form. Oracle Database 12c added native JSON support but it was previously stored in the column store in that text based representation. In Oracle Database 12.2, we leveraged the In-Memory Expressions infrastructure to create an in-memory optimized binary format for JSON within the column store. With this format, searches on JSON documents (such as find me all the movie titles that contain the word "Rogue") run up to 60x faster!

I am sure some people are now very excited and look for a test-environment to try it out. What is the best way for them to get started?

Tirthankar: Any Oracle Database Enterprise Edition installation from release 12.1.0.2 and beyond has the Database In-Memory functionality. Oracle offers free trial periods for the purposes of evaluating the technology. It is also possible to get a starter experience of Database In-Memory with the Exadata Express Service on Oracle Public Cloud. This is a low-cost way to enable In-Memory with a small sized column store (up to 10 GB, which is adequate for many test and dev, even small departmental, use cases). And of course any customer can use the local Oracle resources from the Sales Consulting groups in Germany, Switzerland or Austria or one of our partners. I am sure they are happy to help and maybe DOAG plans some in Memory Test-Labs

(smile). We also have a lot of online resources including best practice documents, whitepapers, an in-memory blog, and a very active in-memory product management team that can help as needed.

As you mentioned before, Database In-Memory is extremely helpful for analytic workloads. Since ten years Oracle has an high-transaction-rate in-memory database: Oracle Times Ten. Are there plans to enhance this stand-alone product and second, are there plans to integrate it into the Oracle Database similar to the Database In-Memory which is deeply integrated?

Tirthankar: TimesTen is a completely separate In-Memory product for a very different use case: for latency critical OLTP workloads. TimesTen is used by thousands of customers and is pretty much the gold-standard for in-memory databases for OLTP. The use case for TimesTen is primarily when the application needs to get response times within micro-seconds – such as prepaid billing workloads for Telco cellular applications, real-time stock trading, and even real-time fraud detection (should my credit card charge be authorized?). TimesTen is very much an active development product and our plans are to extend it to be a scale out in-memory database for OLTP. At Oracle OpenWorld that concluded on October 5, we previewed the next generation of TimesTen that we are currently referring to as “Velocity Scale” which is currently in Beta. With Velocity Scale we showed it is possible to run at previously unheard of performance: 1.2 Billion SQL Select statements per second! This is why we refer to TimesTen as the Worlds’ Fastest OLTP Database. TimesTen is really a separate product since it runs best when embedded with the application. There are no plans as of now to integrate it within Oracle Database which typically runs on hardware that is separate from the Application-Tier.

It seems that Oracle TimesTen lives a bit in the shadow of the very successful columnar-oriented Database In-Memory. Is there less need for a high-transaction-rate database or is Oracle Database so fast that a specific near-realtime transaction engine is not needed?

Tirthankar: Oracle Database is a very fast OLTP database, but it is not able to deliver micro-second response times since it requires a network round-trip with the application. TimesTen customers in verticals like Telecommunications and Financial Services have very strict SLAs and they often require sub-millisecond database response times. This is a specialized segment of the OLTP market

for which TimesTen is a perfect fit. If Oracle Database were considered to be a high end sports car, TimesTen would be a high end motorcycle – even faster, but not as full-featured.

Currently there is a discussion about the future of database-architectures especially when it comes to NVRAMs – non-volatile RAMs. Some people expect that the end of discs as the main storage media for databases is not too far away. How will such a development influence the future Oracle database architecture?

Tirthankar: NVRAM is a very disruptive technology that will fundamentally influence core database designs. All I can say now is please stay tuned!

Oracle 18 is knocking at the door and I am sure you and your team had some ideas for this next version of the Oracle Database. It would be great if you can share some of the new stuff with us.

Tirthankar: I can't go into too much detail at present, but there are some very interesting new In-Memory features planned for in Oracle 18c that enhance both performance and ease of use. For instance, in keeping with the autonomous theme for Oracle Database 18c, we are planning to deliver an Automatic In-Memory mechanism that automatically identifies the ideal In-Memory candidate tables in a customer workload. There are many additional planned improvements for Hybrid (OLTP + Analytics) workloads to further reduce the overhead of column store maintenance in the presence of heavy OLTP activity. Furthermore, there are also many planned improvements for complex query processing such as automatic parallelization, highly optimized arithmetic operations, etc. More details later!

This is a good statement to finish our talk. Thank you very much. I know you will be in Nuernberg @ DOAG conference 2017 for a presentation and for customer talks. See you there!

Tirthankar: I look forward to it!

Oracle Disclaimer: The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

DOAG Datenbank Community unterstützt DBAs

Die Welt eines Oracle-DBAs unterliegt einem ständigen Wandel. Mit jeder Oracle-Datenbank-Version ändert sich mal mehr, mal weniger. Der letzte große Wechsel war die Architektur-Änderung hin zur Container-Datenbank.

Nun steht das Verlagern von Oracle-Datenbanken in die Cloud bevor. Viele Oracle-Datenbank-Administratoren tun sich damit schwer und schauen besorgt in die Zukunft. Das Team der DOAG Datenbank Community unter Leitung von Christian Trieb hat sich deshalb schon Anfang des Jahres unter dem Titel „Quo vadis, DBA“ inten-

siv damit auseinandergesetzt, um Oracle-DBAs in diesem Änderungsprozess zu unterstützen. Im Mai erfolgte dazu der erste öffentliche Schritt im Rahmen der DOAG 2017 Datenbank in Düsseldorf. Johannes Ahrends und Markus Flechtner leiteten die Session „Quo vadis, DBA“. Dabei beschrieben sie die heutige Situation und zeigten zukünftige Themen auf. Anschließend wurde intensiv mit den Teilnehmern diskutiert. Mit der Ankündigung von Larry Ellison, Oracle CTO, dass Oracle demnächst die „Autonome Datenbank“ bereitstelle, bleibt das Thema aktuell.

Im Oktober 2017 trafen sich die Mitglieder der DOAG Datenbank Community in Ratingen zum Workshop „Sensibilisieren für Veränderungen/Kommunikation“ unter der Leitung von Carsten Firus. Die Datenbank Community wollte sich selbst intensiv weiterbilden, um Oracle-DBAs noch besser unterstützen zu können. Dabei wurde die eigene Haltung hinterfragt und Tipps zur Verbesserung von Kommunikation gegeben.

Christian Trieb
christian.trieb@doag.org



Yann Neuhaus (links) im Gespräch mit Paul Bleile (rechts)

„Cloud ist für mich zunächst nur eine andere Formulierung für Outsourcing ...“

Die Coop-Gruppe ist in der Schweiz tief verwurzelt und blickt auf eine 150-jährige Geschichte zurück. Was als kleine Konsumgenossenschaft begann, ist heute ein international tätiges Detail- und Großhandelsunternehmen. Yann Neuhaus, SOUG-Vorstand, und Wolfgang Taschner, Chefredakteur des Red Stack Magazin, sprachen darüber mit Paul Bleile, Leiter Datenbanken bei der Coop.

In welchem Geschäftsbereich ist Coop aktiv?

Paul Bleile: Die Coop-Gruppe ist im Detailhandel, im Großhandel sowie in der Produktion tätig und betreibt Ladenformate in den Bereichen Food, Non-Food und Dienstleistungen.

Was sind dabei die besonderen Herausforderungen an die IT?

Paul Bleile: Das Wichtigste ist, unsere Waren in der richtigen

Menge, in der richtigen Größe, zum richtigen Preis und zum richtigen Zeitpunkt am richtigen Platz zu haben. Unsere IT muss diesen Prozess bestmöglich unterstützen.

Wie lösen Sie diese Aufgaben?

Paul Bleile: Wir setzen einerseits auf die SAP-Standardsoftware, haben aber andererseits in der Lagerbewirtschaftung, im Bestell-

wesen und in der Bewirtschaftung der Supermärkte zusätzliche Applikationen im Einsatz. Hinzu kommt eine Online-Schiene, die wir mit eigenentwickelten Java-Anwendungen betreiben. Alle Systeme sind mit entsprechenden Schnittstellen gekoppelt.

In welchen Bereichen kommen Produkte von Oracle zum Einsatz?

Paul Bleile: Eines unserer Kernprodukte ist die Oracle-Datenbank, daneben kommt Oracle in der Daten-Replikation zum Einsatz. Zur Administration und Überwachung der Datenbanken und Applikationen verwenden wir den Oracle Enterprise Manager.

Was sind für Sie die Vorteile der eingesetzten Oracle-Produkte?

Paul Bleile: Die Datenbank ist gut skalierbar, bietet umfangreiche Funktionen wie Partitioning oder In-Memory und ist fast für jede Applikation einsetzbar. Damit stehen uns sehr viele Optionen zur Verfügung.

In welchem Umfang ist die Oracle-Datenbank im Einsatz?

Paul Bleile: Wir betreiben derzeit insgesamt etwa fünfhundert Instanzen. Darin sind auch die SAP-, Test- und Entwicklungssysteme enthalten.

Sind Sie schon auf die Version 12.2 migriert?

Paul Bleile: Nein, wir haben es gerade erst geschafft, alle Datenbank-Instanzen zumindest auf 11.2.0.4 zu migrieren, und damit die Voraussetzungen geschaffen, auf einer aktuell unterstützten Version zu arbeiten. Darüber hinaus sind wir damit jetzt auch in der Lage, auf AIX 7.2 zu migrieren. In Sachen „Oracle 12.1.0.2“ sind wir aber auch schon sehr gut unterwegs. Von den mehr als fünfhundert Oracle-Instanzen haben wir bereits rund dreihundertdreißig auf Oracle 12 migriert.

Welche Hardware-Plattform kommt bei Ihnen zum Einsatz?

Paul Bleile: Bis zum Jahr 2000 haben wir sehr viel Eigenentwicklung auf IBM Mainframes betrieben und danach auf die Sun-Solaris-Plattform und SAP R3 gesetzt. Nach der Übernahme von Sun durch Oracle haben wir uns im Rahmen einer Konsolidierung in unseren beiden Rechenzentren für IBM VPower entschieden. In den Verteilzentren stehen noch Solaris-Systeme, die wir derzeit durch andere Rechnern und VMware ablösen.

Wie wichtig ist der Enterprise Manager für Sie?

Paul Bleile: Der OEM ist in der Tat sehr wichtig für uns. Er dient einerseits unseren DBAs für die Administration, Überwachung und das Performance-Tuning der Datenbanken, und andererseits können Superuser unsere Applikationen ebenfalls überwachen, da bei sehr vielen Applikationen sonst keine Möglichkeiten vorhanden sind, um das Lastverhalten zu monitoren.

Wo sehen Sie die Unterschiede zwischen SAP Hana und der In-Memory-Option in der Oracle-Datenbank?

Paul Bleile: Wir haben gerade ein „Proof of Concept“ für eine Applikation mit der In-Memory-Option durchgeführt und sind mit den Ergebnissen durchaus zufrieden. Die Hana-Plattform wird bei uns auf VPower-Hardware betrieben, wobei ein Problem darin besteht, dass es von SAP Restriktionen bezüglich der Auslastung des Systems gibt. Die Aussage von SAP, ab dem Jahr 2025 Oracle nicht mehr zu unterstützen, spricht gegen den Einsatz der

Coop-Gruppe

Die Coop-Gruppe hat als Genossenschaft mit rund 2,5 Millionen Mitgliedern ihren Sitz in der Schweiz und ist im Detailhandel, im Großhandel und in der Produktion tätig. Zur Coop-Gruppe zählen Ladenformate aus den Bereichen Food, Non-Food und Dienstleistungen. Coop bietet die größte Markenvielfalt im Schweizer Detailhandel und zeichnet sich durch ein breites ökologisch und sozial profiliertes Sortiment sowie durch Nachhaltigkeitsleistungen entlang der gesamten Wertschöpfungskette aus. Mit der Transgourmet Holding AG ist Coop die Nummer zwei im europäischen Abhol- und Belieferungsgroßhandel. Insgesamt betreibt die Coop-Gruppe mit mehr als 2.200 Läden das dichteste Verkaufsstellennetz des Schweizer Detailhandels und bietet ihren Gastronomiekunden 139 Cash-and-Carry-Märkte in der Schweiz und im europäischen Ausland. Die Coop-Gruppe beschäftigt rund 85.000 Mitarbeitende, davon mehr als 49.000 in der Schweiz.

In-Memory-Option. Für mich ist auch Exadata eher eine Konkurrenz für Hana.

Welche Rolle spielt die Cloud in Ihrem Unternehmen?

Paul Bleile: Cloud ist für mich zunächst nur eine andere Formulierung für Outsourcing. Das bedeutet, dass Cloud-Anwendungen zunächst deutlich kostengünstiger sein müssen als auf meiner angestammten IT-Plattform. Da wir mehrere Rechenzentren haben, geht es uns allerdings erst einmal darum, diese gut auszulasten. Erst wenn eines dieser Rechenzentren an seine Kapazitätsgrenzen stoßen sollte, wäre eine Diskussion über die Cloud sinnvoll. Hinzu kommt, dass SAP die Oracle-Datenbank nur noch bis zum Jahr 2025 unterstützen möchte. Das hat dazu geführt, dass wir bereits einige kritische Systeme auf SAP Hana migriert haben, was zusätzliche Ressourcen in den Rechenzentren geschaffen hat. Wir werden sicherlich mit dem einen oder anderen Projekt Erfahrungen mit der Cloud sammeln, aber einen größeren Schritt in diese Richtung sehe ich in den nächsten Jahren nicht. Außerdem gibt bei uns in der Regel die Applikation vor, mit welchem Datenbank-Release gearbeitet werden muss. Von daher hilft es wenig, wenn in der Cloud immer nur das aktuelle Release angeboten wird, unsere Applikation hingegen beispielsweise noch das Release 11 benötigt. Solange ich nicht mitbestimmen kann, was wann und wie gepatcht wird, weiß ich auch nicht, wie das gehen soll. Auch die derzeit angekündigte Verfügbarkeit von 99,95 Prozent muss erst noch bewiesen werden.

Wie sind Ihre Erfahrungen mit Oracle Support?

Paul Bleile: Wir haben mit Oracle seit langen Jahren einen Advanced-Customer-Service-Vertrag (ACS). Damit sind Oracle-Mitarbeiter häufig direkt bei uns vor Ort. Sie können uns bei Problemen helfen, sodass wir häufig gar keine Service Requests eröffnen müssen. Sollte dies dennoch erforderlich sein, können diese Leute den Fall sehr gut eskalieren, damit wir schnell und kompetent eine Lösung bekommen.

Unterscheiden sich der Oracle- und der SAP-Support?

Paul Bleile: In beiden Fällen geht es zunächst darum, welcher Kategorie ein Problemfall zugeordnet wird. Dann fordern beide Unternehmen Unmengen von Informationen an, die häufig nicht wirklich benötigt werden, um das Problem zu lösen. Das geht meist eine Zeitlang hin und her, bis das Problem überhaupt verstanden wird. Das ist bei SAP auch nicht anders als bei Oracle.

Wären die Applikationslösungen von Oracle für Sie eine Alternative zu SAP?

Paul Bleile: Ich weiß, es gibt bei Oracle Lösungen wie PeopleSoft, die Mitbewerber zu SAP sind. Ich kenne diese Produkte jedoch nicht so gut, um deren Mächtigkeit einschätzen zu können.

Wie beurteilen Sie die Zukauf-Strategie von Oracle?

Paul Bleile: Das ist sehr unterschiedlich. Bei GoldenGate fand ich die Übernahme zunächst gut. Wenn ich aber jetzt sehe, dass unsere Streams-Lösung irgendwann von GoldenGate abgelöst werden soll, bin ich nicht begeistert, weil GoldenGate für unsere Projekte viel zu umfangreich ist.

Können Sie sich vorstellen, ein Komplettsystem von der Hardware bis zu den Applikationen von einem einzigen Hersteller wie Oracle einzusetzen?

Paul Bleile: Wir setzen seit jeher auf eine Multi-Vendor-Strategie, um nicht von einem einzigen Hersteller abhängig zu sein.

Wäre der Einsatz eines Oracle Engineered System bei Ihnen denkbar?

Paul Bleile: Von Kollegen höre ich oft, dass die Exadata ein gut etabliertes Produkt ist. SAP ist unser strategisches Produkt und wenn es tatsächlich dazu kommen sollte, dass SAP ab dem Jahr 2025 Oracle nicht mehr unterstützt, dann hätte auch die Exadata bei uns keine Zukunft.

In welche Richtung wird sich Ihre IT in den kommenden Jahren entwickeln?

Paul Bleile: In erster Linie wird die Online-Schiene an Bedeutung gewinnen. Ansonsten werden wir weiterhin stark auf SAP setzen. Was erwarten Sie dabei von einem IT-Unternehmen wie Oracle? Paul Bleile: Mit den Oracle-Mitarbeitern aus dem Advanced-Customer-Service-Vertrag sind wir sehr zufrieden. Wir hoffen, dass das auch weiterhin so sein wird.

Wie sehen Sie den Stellenwert einer Anwendergruppe wie der SOUG?

Paul Bleile: Die SOUG halte ich für eine sehr gute Community. Wir werden künftig im Rahmen unserer Firmenlizenz entweder gemeinsam oder abwechselnd an den Events teilnehmen. Es gibt dort immer sehr interessante Themen. Auch der direkte Erfahrungsaustausch auf den Veranstaltungen ist für uns sehr wichtig und wertvoll.



Zur Person: Paul Bleile

Paul Bleile hat im Jahr 1989 in Furtwangen im Schwarzwald im Bereich „Informatik“ den Studiengang „Allgemeine Informatik“ absolviert (Dipl. Informatiker FH). Danach war er in einem DV-Betrieb für die Software-Entwicklung für Apple Macintosh verantwortlich. Seit dem Jahr 1990 arbeitet er in der Schweiz, zunächst bei der Firma Suisselectra im Bereich „Gebäude-Management“ und seit dem Jahr 1992 bei der Coop. Dort hat Paul Bleile verschiedene Funktionen durchlaufen (Daten-Architekt, Informatik-Methodiker, Datenbank-Administrator). Bis zum Jahre 2001 hat er mit DB2 unter OS390 auf dem IBM-Mainframe gearbeitet. Danach wurde ihm die Leitung des Middleware-Teams übertragen, wo er für die Produkte DB2, SQL Server und MQSeries verantwortlich war. Im Jahr 2006 übernahm er die Leitung des Datenbank-Teams und ist derzeit für mehr als fünfhundert Oracle-Instanzen im SAP- und Non-SAP-Bereich sowie für rund fünfzig SQL-Server zuständig.

User Experience – eine Einführung

Alexej Schneider, MT AG



Die heutige Welt ist ohne moderne Technologien nicht mehr vorstellbar. Tagtäglich verwenden wir Computer, Tablets und Smartphones für die unterschiedlichsten Aufgaben. Sei es, um das Wetter zu checken, uns miteinander zu vernetzen oder um schnell mal ein neues Paar Schuhe zu kaufen. Anwendungen im Konsumentenmarkt zeichnen sich oft durch eine gute Bedienbarkeit aus, Business-Anwendungen dagegen eher weniger. Diese sind in der Regel zwar komplexer, doch man kann den Anwender mit entsprechenden Werkzeugen oft bei seiner Arbeit unterstützen.

Große Firmen wie Apple, Google, Facebook oder Amazon machen es uns schon seit vielen Jahren vor. Ihre Produkte sind häufig intuitiv bedienbar, hübsch anzusehen und machen in der Regel sogar Spaß. Für die Unternehmen ist die Hauptmotivation natürlich der Verkauf ihrer Produkte. Der Markt ist hart umkämpft und wenn die Produkte zu schlechten Erlebnissen führen, kann das zum Verlust des Vertrauens und zu einer schlechten Reputation führen. Positive Erlebnisse sorgen hingegen für loyale Kunden und gute Bewertungen. Demzufolge wird viel investiert, entwickelt, geforscht und getestet, um die Anwender zufriedenzustellen. Das Ganze nennt man „User Experience“.

Lets talk about UX

User Experience (UX, zu Deutsch wörtlich „Nutzererfahrung“, besser „Nutzererlebnis“) bezieht sich auf die persönliche Emotion eines Anwenders einer Anwendung, eines Systems oder Service. Die Königsdisziplin einer guten Experience ist, dass man den Nutzer für das Produkt begeistern kann. Dies muss sich in erster Linie nicht unbedingt auf die digitale Welt beziehen. Wir alle kennen gutes und schlechtes UX aus unserem Alltag. Hat man sich nicht auch mal geärgert, als man vor einer Tür stand und nicht auf Anhieb wusste, wie die

se zu öffnen ist? Soll man sie nun drücken, ziehen, drehen oder geht sie vielleicht automatisch auf? Wenn es nicht direkt klappt, so ärgert man sich und sucht anschließend nach einer Anleitung, fragt nach Hilfe oder man lässt es eventuell komplett bleiben.

Dieses Beispiel ist natürlich überspitzt, doch möchte man meinen, dass ein so altes Interaktionsobjekt mittlerweile keinen vor Herausforderungen stellen sollte. Bei unserer Business-Software verhält es sich ähnlich. Die Software soll in erster Linie eine bestimmte Aufgabe effektiv, effizient und zufriedenstellend erfüllen. Eine schlechte UX bringt dabei viele negative Aspekte mit sich:

- **Produktivität**
Wenn eine Anwendung schlecht zu benutzen ist, dauert die Arbeit länger; eventuell muss man sogar mehr Mitarbeiter für eine Aufgabe einsetzen. Insgesamt führt das zu höheren Kosten.
- **Schulungskosten/Einarbeitungszeit**
Mitarbeiter müssen geschult werden, wenn die Anwendung nicht intuitiv zu benutzen ist. Bei schlechter Software dauert es eine längere Zeit, bis die Mitarbeiter produktiv sind.
- **Supportkosten**
Ist die Nutzbarkeit schlecht, muss der Support oft helfen.

- **Entwicklungskosten**
Wenn man die Endanwender spät in die Entwicklung einbezieht (im schlimmsten Fall erst nach der Implementierung), können neue, unerwartete Anforderungen entstehen. Das kann ein komplettes Redesign notwendig machen.
- **Zufriedenheit**
Ein wichtiger Faktor ist auch die Moral der Anwender. Es ist frustrierend, wenn man seine Arbeit nicht sachgemäß durchführen kann. Dadurch kann auch die weitere Produktivität leiden.

Selbstverständlich sind Business-Anwendungen in der Regel komplexer. Gerade deswegen ist es jedoch essenziell wichtig, für eine gute UX zu sorgen. Wenn man anfängt, sich mit UX zu beschäftigen, kann es zunächst doch recht verwirrend sein. Es kommen viele neue Fachausdrücke hinzu, vieles ist auch eher subjektiver Natur und nicht alles ist klar definiert. UX besteht zudem aus zahlreichen Komponenten, die im Zusammenspiel miteinander agieren.

Oft wird UX fälschlicherweise nur mit einer hübschen Benutzeroberfläche (UI, User Interface) in Verbindung gebracht. Nur weil eine Anwendung ein schönes Design besitzt, bedeutet das allerdings nicht gleichzeitig, dass sie zu einem guten Erlebnis führt. Selbstverständlich dient ein gutes

Design der positiven Assoziation mit dem Produkt, es ist aber nur ein Teil von UX.

Eine weitere Komponente (wenn nicht sogar die wichtigste) ist die Nutzerfreundlichkeit („Usability“). Der Anwender des Produkts soll effektiv, effizient und möglichst zufrieden zum Ziel kommen. „Usability“ ist ein subjektiver Begriff. Was für einen Nutzer einfach ist, kann für einen anderen eine unüberwindbare Herausforderung sein. Daher ist es wichtig, während der Entwicklung seine Benutzer genau zu kennen und die Anwendung an deren Erfahrungen und Wünsche anzupassen. Oft spricht man dabei von „User-Centered Design“ (UCD).

In den 1990er-Jahren hat Jakob Nielsen „10 Usability Heuristics for User Interface Design“ [1] formuliert. Auch wenn sie vor mehr als zwanzig Jahren aufgestellt wurden, haben diese Prinzipien nicht an Bedeutung verloren (siehe Abbildung 1):

- **Feedback**
Das System soll den User stets darüber informieren, was gerade passiert.
- **Gemeinsame Sprache**
Das System sollte die Sprache des Benutzers sprechen.
- **Kontrolle und Freiheit**
User sollten Aktionen rückgängig machen oder neu durchführen können.
- **Konsistenz**
Benutzer erwarten gewisse Design-Standards (etwa Logout immer oben rechts).
- **Fehlermeldungen**
Fehlermeldungen sollten für den User verständlich sein.
- **Fehlervermeidung**
Hilfreiche Fehlermeldungen sind gut, Prävention ist besser (etwa durch Client-seitige Validierung).

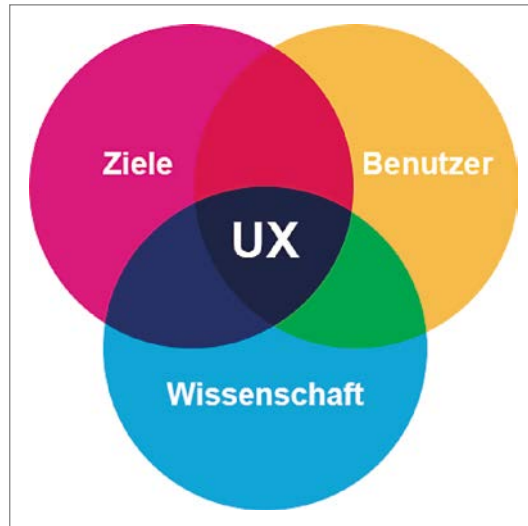


Abbildung 1: Zusammenspiel der Komponenten bei User Experience

- **Flexibilität und Effektivität**
Zusätzliche Funktionen sollen dem Benutzer die Arbeit erleichtern.
- **Erkennen statt Erinnern**
Alle notwendigen Informationen sollten sichtbar sein.
- **Minimalismus**
Design sollte auf das Notwendigste begrenzt werden.
- **Hilfe**
Ein gutes Design ist selbsterklärend, dennoch braucht der User oft Hilfe.

UX-Design

Der Artikel bezog sich bisher darauf, was User Experience bedeutet. Doch wie können wir eine solche nun für unsere Anwendungen umsetzen? Eine gute UX entsteht nicht von einem Tag auf den anderen. Zum Glück ist das jedoch kein neues Thema und es gibt diesbezüglich viele Quellen (Bücher, Blogs, Zeitschriften oder

Tools). Den Vorgang, um eine möglichst gute Erfahrung zu erschaffen, nennt man „User Experience Design“ (UXD).

Im Mittelpunkt jeder Entscheidung stehen die Endanwender. Dementsprechend ist es wichtig, sich mit diesen so früh wie möglich auseinanderzusetzen – allerdings während des kompletten Produkt-Lebenszyklus und nicht nur während der Entwicklung. Im UX-Jargon heißen die Endanwender oft „Persona“. Bei diesen handelt es sich um fiktionale Personen, die einem typischen Vertreter einer Zielgruppe entsprechen. Man fasst die relevanten Merkmale der Personae zusammen und gibt ihnen ein Gesicht. Eine Persona besitzt üblicherweise einen fiktiven Vor- und Nachnamen sowie ein Foto. Außerdem kommen weitere Punkte hinzu, die für das jeweilige Produkt relevant sind:

- Alter/Geschlecht/Herkunft
- Beruf/Ausbildung
- Computerkenntnisse
- Ziele/Wünsche/Erwartungen an das Produkt

	Ähnlichkeit zum Endprodukt	Kosten	Einsatzzweck	Eigenschaften
Skizze	---	++++	Brainstorming	Schnell, einfach
Wireframe	+	++	Feedback	Das Skelett der Anwendung (ohne Design)
Mockup	++	--	Feedback	Das Skelett der Anwendung (mit Design)
Click-Dummy	++++	----	User testing	Interaktiv, aber keine Funktionalität

Tabelle 1: Prototypen

Komplexe Interaktionskonzepte lassen sich nur schwer beschreiben. Von einer Idee zur fertigen Lösung ist es ein langer Weg.

Die Erwartungshaltungen an ein Produkt sind meistens unterschiedlicher Natur zwischen Entwickler, Kunden und Endanwender. Beim UX-Design bekommt das Prototyping daher eine wichtige Rolle. Das agile Vorgehen während des Prototyping unterstützt die Kommunikation und hilft der gemeinsamen Vision vom Endprodukt. Prototypen können dabei unterschiedliche Formen annehmen. Jede hat wiederum ihre Vor- und Nachteile sowie Einsatzzwecke (siehe Tabelle 1).

Das Testen nimmt bei UX eine wichtige Rolle ein. Dabei gibt es auch hierbei zahlreiche Varianten davon, wie und was man testen kann. Die gewonnenen Erkenntnisse können anschließend in eine bessere UX umgesetzt werden:

- **Experten-Review**
Ein UX-Experte untersucht die Anwendung auf Grundlage von Usability-Kriterien und persönlichen Erfahrungen.
- **Usability-Test**
Personen der jeweiligen Zielgruppe werden bei der Benutzung der Anwendung beobachtet.
- **Crowd-Testing**
Es gibt Anbieter, bei denen man seine Anwendungen von vielen Usern testen lassen kann.
- **Eye-Tracking**
Der Blickverlauf einer Person wird sichtbar gemacht; daraus ist zum Beispiel die Empfehlung entstanden, dass man Label rechtsbündig anordnet [2].
- **A/B-Test**
Unterschiedliche Nutzer bekommen unterschiedliche Varianten eines Produkts; anhand von Metriken wird bestimmt, welche Lösung besser geeignet ist.

UI-Design

Das User Interface ist die zentrale Schnittstelle zwischen Mensch und Computer. Eine gute Schnittstelle ist es dann, wenn der Anwender bei seiner Aufgabe unter-

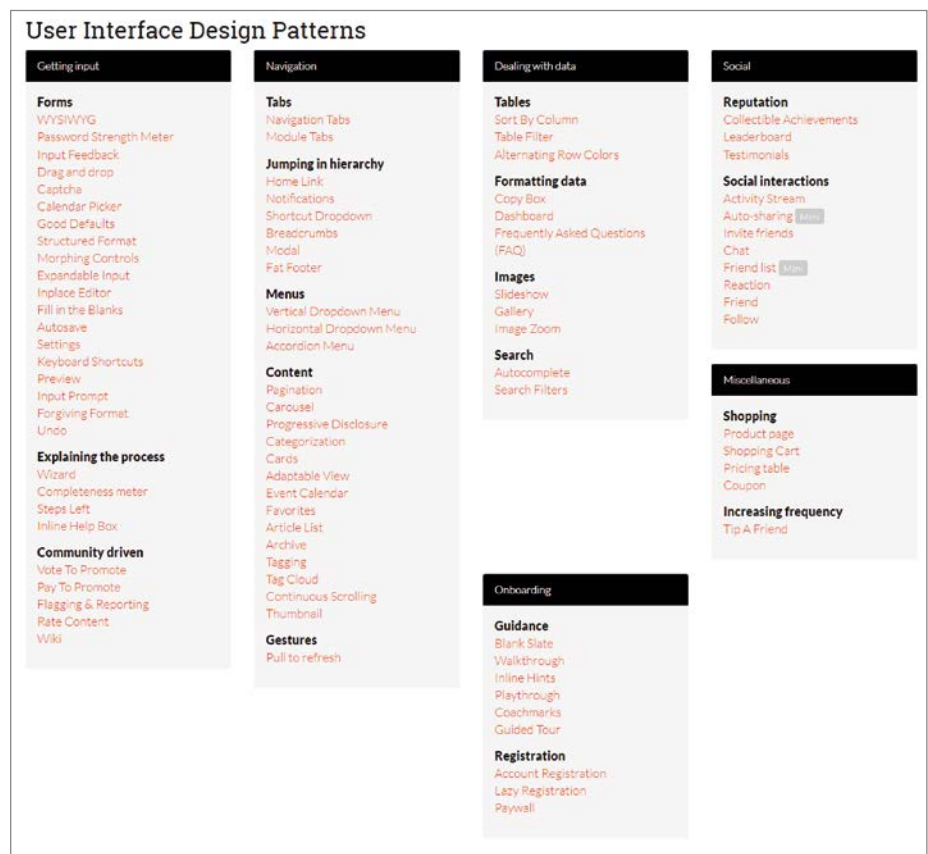


Abbildung 2: Eine Auswahl von UI-Design-Patterns [3]

stützt und ihm keine Steine in den Weg gelegt werden.

Moderne Oberflächen bestehen nicht mehr aus Kommandozeilen, sondern aus einer Vielzahl von Elementen (Fenstern, Formularen, Buttons, Reports, Charts etc.). In der Regel ist es die Aufgabe eines Designers, die zahlreichen Elemente zu einem Interface zu verbinden. Man muss dies allerdings nicht unbedingt studiert haben, um ein ansprechendes UI zu kreieren. Oft reicht es aus, sich von anderen Anwendungen inspirieren zu lassen, oder man verwendet Frameworks à la Bootstrap.

Design hat jedoch mehr mit Regeln und Prinzipien zu tun, als man vielleicht glauben mag. Viele Design-Trends sind auch durch UX-Studien entstanden. Menschen lesen Anwendungen beispielsweise nicht wie ein Buch von links nach rechts. Web-Anwendungen sind dabei ein klassisches Beispiel. Sie sind üblicherweise nach dem „F Pattern Layout“ aufgebaut. Demzufolge werden die wichtigsten Elemente oben links platziert (Logo, Navigation, Handlungsaufforderungen), anschließend die nächste Reihe und so weiter.

Das menschliche Auge sucht standardmäßig nach Mustern. Durch ein entspre-

chendes Design können wir den Benutzer durch die Anwendung führen. Dazu hat man unterschiedliche Design-Werkzeuge wie Positionierung, Größe, Farbe, Kontrast, Abstände oder die Wahl von Schriftarten zur Auswahl. Je dominanter ein Element ist, desto wahrscheinlicher ist es, dass das Auge dahin wandert.

UI-Design-Patterns

Beim Designen von neuen Interaktionselementen muss man nicht jedes Mal das Rad neu erfinden. Es gibt etablierte Best Practices, die man für seine Anwendungen nutzen kann; oft spricht man hier von „UI-Design-Patterns“. In der Programmierung gibt es ein ähnliches Konzept, die sogenannten „Entwurfsmuster“. Dies sind Lösungsschablonen zu wiederkehrenden Entwurfsproblemen.

Bei UI-Design-Patterns handelt es sich um Konzepte, die der User aus seinem alltäglichen Leben kennt. Durch ihren Einsatz muss der Anwender nicht mehr großartig nachdenken und kann im besten Falle die Anwendung intuitiv nutzen (siehe Abbildung 2).

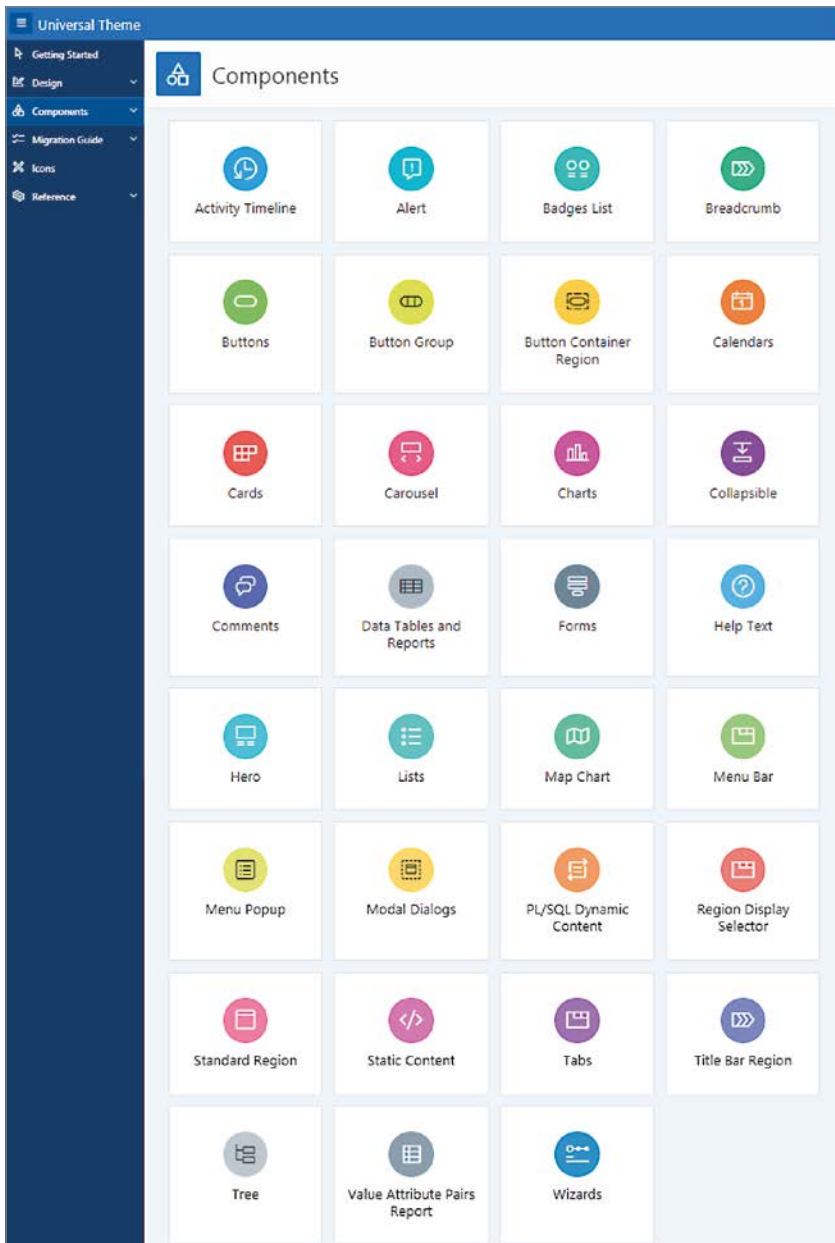


Abbildung 3: Apex-Komponenten [4]

UX am Beispiel von Apex

Oracle Application Express (Apex) ist eine webbasierte Softwareentwicklungs- und Laufzeit-Umgebung. Sie ist standardmäßig Bestandteil jeder Oracle-Datenbank-Edition, wodurch keine zusätzlichen Lizenzkosten entstehen. Bei Apex handelt es sich um ein Rapid-Application-Development-Tool (RAD), mit dessen Hilfe man schnell datenzentrierte Web-Anwendungen erstellen kann. Der Entwickler hat Zugriff auf zahlreiche Komponenten (siehe Abbildung 3) und kann eine Anwendung nach dem Baukasten-Prinzip zusammensetzen.

Viele Funktionen und Eigenschaften lassen sich deklarativ anpassen. Das Layout kann man durch Drag & Drop nach

Belieben verändern. Durch die enge Verbindung zur Oracle-Datenbank sind Prototypen sehr schnell realisiert, die weitaus mehr Funktionalität bieten als einfache Mockups oder Click-Dummys. Dadurch kann man Ergebnisse zügig präsentieren und früh auf Usability testen.

Besonders seit der Apex-Version 5.0 hat sich beim Thema „UX“ sehr viel getan. So wurde das Design komplett überarbeitet und trägt seitdem den Namen „Universal Theme“. Es sieht nun modern aus und ist responsive. Mit dem „Theme Roller“ und den „Template Options“ lassen sich viele Design-Elemente ohne Programmierkenntnisse leicht anpassen. Im Internet findet man weiterhin zahlreiche Plug-ins, die den Umfang von Apex stetig erweitern.

Reicht das nicht aus, muss man bei besonderen Anforderungen mit CSS und JavaScript arbeiten. Auch sind fundamentale Kenntnisse über HTML-DOM und Web-Entwicklung im Allgemeinen zwingend notwendig. Nichtsdestotrotz eignet sich Apex hervorragend, um benutzerfreundliche Anwendungen zu erstellen.

Fazit

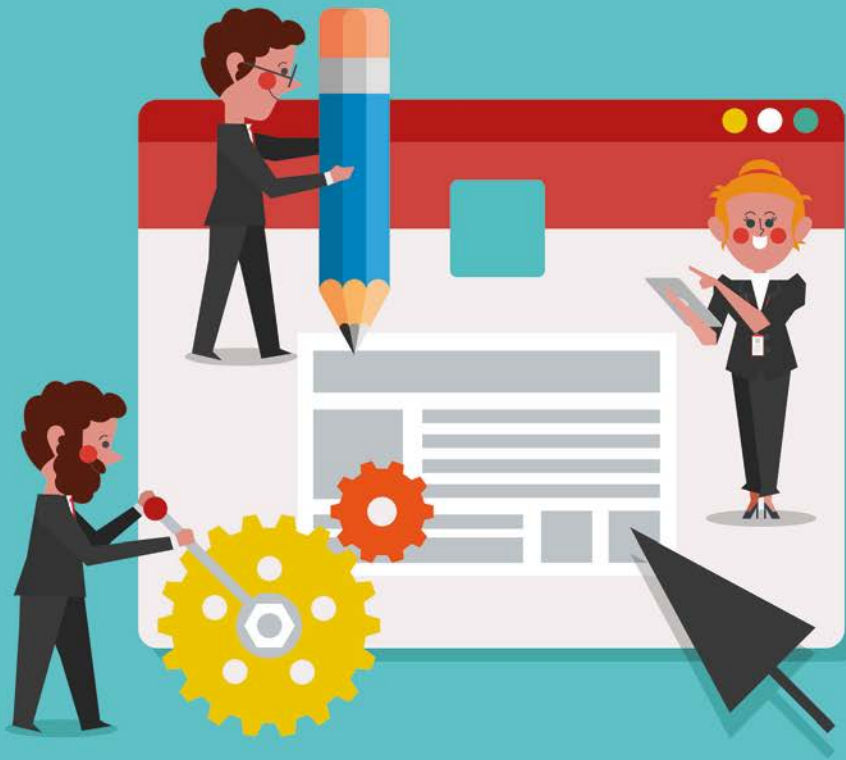
„UX“ ist ein wichtiges und großes Thema. Eine gute UX-Strategie ist im Konsumentenmarkt überlebenswichtig. In der Business-Welt gewinnt es langsam mehr an Bedeutung. Die Anwender sind durch ihre privaten Applikationen mittlerweile einen hohen Standard gewohnt. Diesen Standard erwarten sie auch immer mehr von ihren Anwendungen im Berufsleben. Selbstverständlich erfordert eine gute UX-Strategie mehr Ressourcen. Entwickler, Architekten und Projektmanager müssen sich in diesem Bereich weiterbilden. Außerdem brauchen sie oft Kenntnisse über neue Technologien und Tools. Durch das Feedback der Nutzer kann die Entwicklungsphase auch länger dauern. Auf lange Sicht gesehen, bringt es allerdings überwiegend nur Vorteile, sich mit diesem Thema auseinanderzusetzen. Wie bereits erwähnt, führt eine gute User Experience zu mehr Effektivität, Effizienz und Zufriedenheit der Mitarbeiter. Das ist letztendlich genau das, was wir alle wollen: Gute Software schreiben, die gerne genutzt wird.

Weiterführende Links

- [1] <https://usersnap.com/de/blog/usability-nielsen>
- [2] <https://www.uxmatters.com/mt/archives/2014/09/eye-tracking-in-user-experience-design.php>
- [3] <http://ui-patterns.com/patterns>
- [4] <https://apex.oracle.com/pls/apex/f?p=42>



Alexej Schneider
alexej.schneider@mt-ag.com



Bessere UX durch Methoden wie Design Thinking und Design Sprints

Maximilian Liesegang, esentri AG

Es ist dringend notwendig, den Nutzer wieder ins Zentrum zu rücken. Um diese Erkenntnis gleich in Taten umzusetzen, zeigt der Artikel, wie man das mithilfe von Design Thinking und Design Sprints schafft.

Durch die stetig steigende Digitalisierung unserer Welt stehen viele etablierte Firmen vor einem Problem. Die Dynamik zwischen ihnen und ihren Kunden hat sich radikal verändert. Heute vergleichen die Menschen ein Unternehmen nicht nur mit den direkten Konkurrenten; es wird ebenso anhand der letzten positiven Erfahrung verglichen, die sie online oder auf dem Smartphone hatten. Dort ist der Wettbewerb besonders groß. Es ist also notwendig, dem Benutzer eine großartige Erfahrung – also User Experience – mit der eigenen Anwendung zu ermöglichen.

Gleichzeitig ist man mit einer anderen Herausforderung konfrontiert: Die durchschnittliche Lebensdauer eines Unter-

nehmens sinkt und sinkt. In den 1920er-Jahren verweilte ein Unternehmen durchschnittlich 67 Jahre am „S&P 500“. Bis 1965 sank diese Zahl auf nur noch 33 Jahre und danach weiter auf heutige 15 Jahre. Beispiele von gefallenen Titanen sind Nokia, Kodak, AEG und Quelle. Es gibt allerdings auch gelungene Gegenbeispiele. Das beweisen Firmen, die es geschafft haben, sich neu zu erfinden, etwa IBM, GE und 3M. Es machte sich bezahlt, in die Kreativität und Innovationskraft der eigenen Mitarbeiter zu investieren. Um diesen beiden Herausforderungen, dem Kundenfokus und der Kreativität, gerecht zu werden, bietet es sich an, Design Thinking genauer unter die Lupe zu nehmen.

Design Thinking

Viele kommen zu dem Schluss, dass es sich bei „Design Thinking“ um etwas Neues handelt. Es wurde jedoch bereits im Jahr 1969 zum ersten Mal von Herbert Simon erwähnt und Anfang der 1990er-Jahre perfektionierte die Firma IDEO den Prozess bereits. Die aktuelle Popularität lässt sich auf den Bedarf von Unternehmen und Organisationen nach Kreativität zurückführen. Die Herausforderungen in der Gegenwart sind zu komplex, um von dem einsamen Erfinder in einem Heureka-Moment gelöst zu werden. Allerdings wissen viele nicht, wie man den eigenen Mitarbeitern beibringen kann, kreativer zu sein. Genau hier setzt

Design Thinking als Werkzeugkasten für Kreativität und innovatives Denken an.

Beim Verstehen von Design Thinking ist für viele Leute das größte Problem das Wort „Design“. Zu schnell wird das Wort mit etwas Ästhetischem wie einem Designerstuhl oder -tisch assoziiert. Der nächste Schritt ist die Feststellung, dass man nicht künstlerisch begabt und dass deshalb Design Thinking nichts für einen ist. Um das zu vermeiden, ist es hilfreich, Design als eine Denkweise und nicht als eine Sache zu verstehen. Damit wird aus Design Thinking eine Methode zur innovativen Problemlösung – also wortwörtlich zum Denken wie ein kreativer Designer. Man könnte „Design Thinking“ auch durch „Creative Thinking“ ersetzen.

Es gibt verschiedene Ansätze, die Methode in Phasen zu unterteilen. Von den ursprünglichen sieben Phasen bis zu den drei Is (Inspiration, Ideenfindung, Implementierung) von IDEO. Gemeinsam haben sie aber alle, dass im Zentrum ein tiefes Verständnis der Bedürfnisse des Menschen steht – daraus finden sie die Schnittmenge der Bedürfnisse mit dem, was technisch umsetzbar und wirtschaftlich tragfähig ist (siehe Abbildung 1).

Design Thinking lässt sich in folgende fünf Phasen unterteilen: das Verstehen, die Sichtweise definieren, Ideengenerierung, Prototypen und das Testen (siehe Abbildung 2). Die einzelnen Phasen überlappen sich mit anderen, weshalb bewusst dieser Begriff gewählt wurde. Dabei handelt es sich nicht um strikt nacheinander ablaufende Prozess-Schritte. Es ist sogar explizit gewünscht, dass die Phasen auf der Suche nach neuen Einsichten wiederholt durchlaufen werden und man damit iterativ ein besseres Verständnis der Aufgabe erhält. Eine Gefahr besteht nur darin, sich in einer Art Endlosschleife zu verfangen.

Die Phasen des Design Thinking

Vor dem Beginn, sozusagen als Schritt Null, sollte man sichergehen, dass man wirklich ein Problem formuliert hat. Ein Fehler ist es, nur eine vorgegebene Lösung zu untersuchen. Das Hinterfragen der Problemstellung gleich zu Beginn könnte möglicherweise eine tiefer liegende Frage aufdecken.

In der ersten Phase erfolgt das Verstehen des Benutzers. Es soll dabei ein grö-

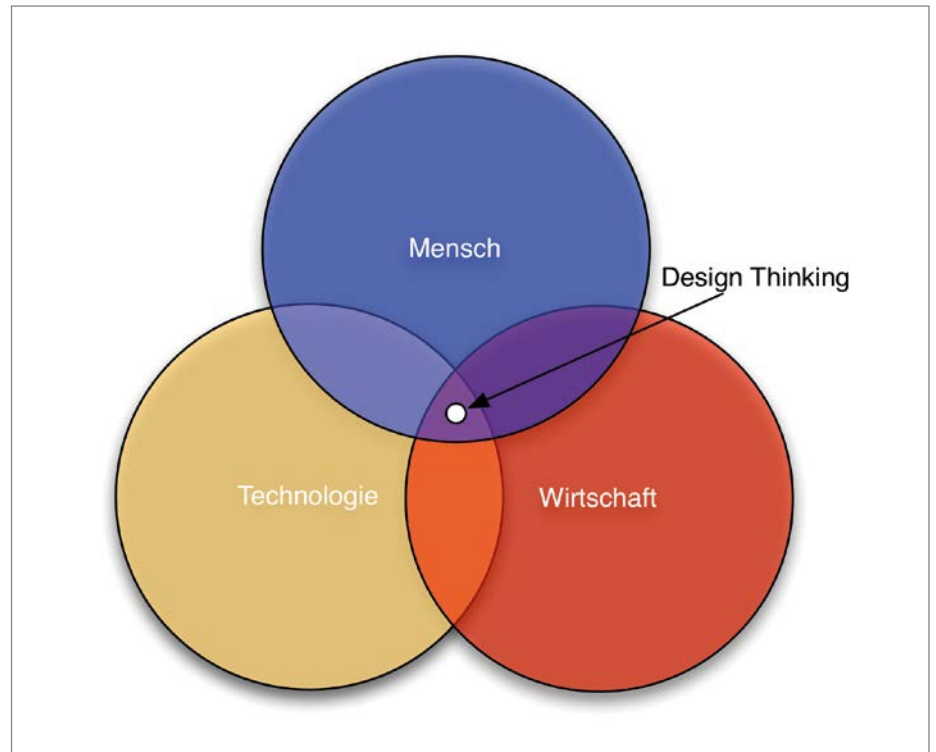


Abbildung 1: Schnittmenge von Menschen, Technologie und Geschäft

ßeres Wissen über die tatsächlichen Bedürfnisse entwickelt werden. Dazu sind folgende Tätigkeiten notwendig: Eintauchen, Interviews und Beobachtungen. Beim Eintauchen geht es darum, sich in die Lage des Benutzers zu versetzen. Man kann zum Beispiel die Ist-Situation am eigenen Leib erfahren. In den Interviews geht es um das gezielte Stellen von Fragen an aktuelle Benutzer. Zudem sollte man die Benutzer noch beobachten. Das ist besonders wichtig, da es häufig Unterschiede gibt zwischen dem, was sie angeben zu tun, und dem, was sie tatsächlich tun.

Ein weiterer Tipp für ein besseres Verstehen ist das genaue Betrachten von Extremfällen unter den Nutzern. Extremfälle sind zum Beispiel die professionellen Nutzer und die Nichtnutzer. Durch das genaue Betrachten dieser können sehr interessante Informationen über das Problem gewonnen werden. An dieser Stelle sollte vor der Gefahr gewarnt werden, sich selbst als die Zielperson zu sehen. Das würde dazu führen, die eigenen Vorlieben in das Endprodukt einfließen zu lassen. Es geht darum, vorurteilsfrei die tatsächliche Zielgruppe zu beobachten.

Nach dem Verstehen geht man daran, das gewonnene Verständnis in Erkenntnisse umzuwandeln. Diese Synthese von Informationen wird auch als „Sichtwei-

se definieren“ bezeichnet. Eine gute Erkenntnis hat folgende Eigenschaften. Zum einen beschreibt sie ein unbefriedigtes Bedürfnis des Nutzers. Dies sind die Stellen, an denen das Erlebnis des Benutzers wirklich verbessert werden sollte. Außerdem darf die Erkenntnis keine Lösung sein, das würde uns beim zukünftigen Divergieren einschränken.

Mit Divergenz und Konvergenz sind die elementaren Techniken des kreativen Denkens beschrieben. In der Phase des „Verstehens“ musste man ja seine Überlegungen möglichst weit fächern sowie verzweigten und unsystematischen Gedanken folgen. Dieser Vorgang wird auch „Divergieren“ genannt. Um dann seine vielfältigen Überlegungen fokussieren zu können, konvergiert man. Je rigoroser Divergenz und Konvergenz exerziert werden, desto bessere Ergebnisse erhält man.

Nachdem man nun eine Reihe von Erkenntnissen gesammelt hat, geht es in der nächsten Phase um das Finden innovativer Lösungen – häufig als „Ideengenerierung“ bezeichnet. Die „Wie-Könnten-Wir“-Technik schafft einen guten Übergang von den Erkenntnissen zu den Lösungen. Hier formulieren wir Erkenntnisse so positiv, dass es sich geradezu anbietet, viele Lösungen auf die Fragen zu finden. Dies bietet eine hervorragende Starttrampe für

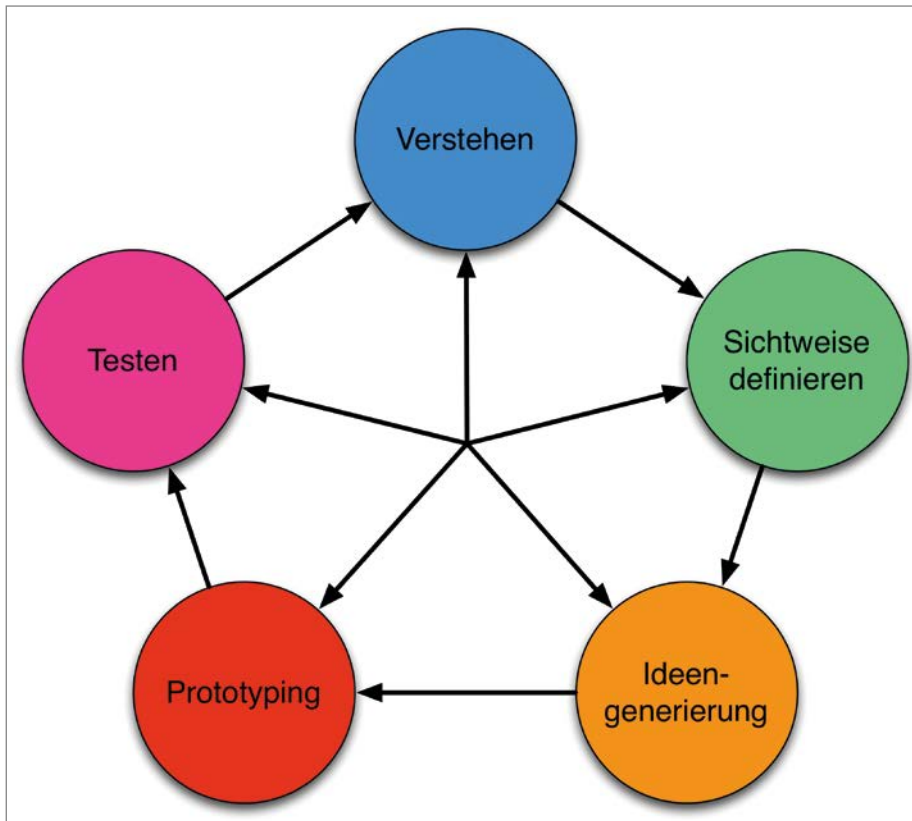


Abbildung 2: Design-Thinking-Phasen

innovationsgeladene Brainstorming-Sitzungen. Ein Beispiel für eine solche Frage ist „Wie könnten wir das Reisen mit der Bahn für Kinder interessant gestalten?“

Nach einigen WKW-Aussagen können wir mit der tatsächlichen Ideengenerierung starten. Hier gibt es eine Vielzahl von Techniken wie Mind Mapping, Sketching oder Bodystorming. Bei all den Techniken gibt es einige Grundregeln, die dabei helfen, sie erfolgreicher durchzuführen: Fördere wilde Ideen, baue auf den Ideen anderer auf, arbeite visuell und achte darauf, dass alle Teilnehmer aktiv an der Unterhaltung teilhaben.

Nachdem man viele Ideen generiert hat, geht es jetzt darum, passende Prototypen zu erstellen. Dabei liegt es nahe, Prototypen zu mehreren Ideen gleichzeitig zu entwickeln, um kein Innovationspotenzial zu verlieren. Doch welche Ideen nimmt man? Hier bietet sich als Auswahlkriterium an, die Ideen unter folgenden Gesichtspunkten auszuwählen: Welche ist die rationale Idee, die überraschendste Idee und die begeistertendste Idee.

Nachdem die ersten Lösungen ausgewählt wurden, geht es daran, Prototypen zu erstellen. Es ist jedoch wichtig, Prototypen zu entwerfen, die man schnell er-

stellen und wieder wegwerfen kann. So können ein Papierentwurf oder ein Storyboard schon reichen, um die ersten Annahmen zu verifizieren. Sie müssen auch keine Ähnlichkeiten mit dem Endprodukt haben, schließlich sind sie nur einer der ersten Schritte dorthin. Darauf basierend kann man Ideen rapide weiterentwickeln. In dieser Phase gilt das Zitat von Tim Brown: „Früh scheitern, um früher Erfolg zu haben“. Sollte es sich herausstellen, dass Annahmen aus den vorherigen Phasen überarbeitet werden müssen, sollte man diese wiederholen. Hier wird sich zunutze gemacht, dass Design Thinking kein linearer Prozess ist.

Iteration ist ein fundamentaler Teil guten Designs, deshalb ist die Phase des „Testens“ und des „Prototyping“ eher als eine Art „Tandem“ zu sehen. Zwischen den beiden Phasen wird fließend gewechselt, um sich iterativ in Feedbackschleifen der Lösung zu nähern. Es wird also wieder konvergiert, nachdem man zuvor mit den vielen Prototypen divergiert hat. Im Unterschied zum Prototyping zeichnet sich das Testen durch konkretere Experimente mit dem Nutzer aus. Dabei sollte man darauf achten, die Tests nicht zu groß anzulegen. Kleine und kostenspa-

rende Nutzertests sind das Ziel. Dadurch behält man den Schwung der iterativen Entwicklung bei.

Eine diversifizierte Zusammenstellung des Teams mit verschiedensten Hintergründen ist elementar wichtig. Durch die größere Anzahl von Blickwinkeln wird die Fähigkeit zum Divergieren verbessert. Außerdem besteht in einem gleich oder ähnlich denkenden Team die Gefahr, dass jeder an derselben Stelle hängen bleiben wird.

Es gibt eine riesige Anzahl von Design Frameworks, die dem oben skizzierten Ablauf ähneln. Die beschriebenen Phasen lassen sich in dieser oder ähnlicher Form in den Alternativen wiederfinden. Schlussendlich muss man sich das Vorgehen zu eigen machen und es an die eigenen Gelegenheiten und Vorlieben adaptieren. Das Wichtigste ist, weiter an der eigenen Innovationsfähigkeit zu arbeiten, dann wird sich die Denkweise eines Designers weiter in den Köpfen der Teilnehmer verfestigen. Diese Transformation zieht sich meistens über längere Zeiträume hin. Aber wie könnte man Design Thinking noch komprimieren und fokussieren?

Design Sprint

In der *Abbildung 3* fallen sofort einige Begriffe ins Auge, die sich so im Design Thinking wiederfinden. Das ist kein Zufall, da Design Sprints das Essenziellste aus Design Thinking benutzen. Das Ziel der Erfinder des Design Sprint war es, in möglichst kurzer Zeit die großen Fragen der Produktentwicklung zu beantworten. Dadurch kann vermieden werden, dass man zu lange in eine falsche Lösung investiert. Im Prinzip springt man in die Zukunft und kann direkt beobachten, wie der Kunde auf einen Prototyp des Projekts reagiert. Bevor man die Leiter des Erfolgs besteigt, sollte man schließlich sichergehen, dass sie an der richtigen Wand lehnt.

Wie die *Abbildung 3* zeigt, besteht ein Design Sprint aus fünf kompakten Tagen. Durch den Stichtag am Freitag besteht auch nicht die Gefahr, sich zu lange in einer der Phasen aufzuhalten. Das Ergebnis der Woche ist ein getestetes Konzept, das als Grundlage für die weitere Produktentwicklung dient. Bisherige Erfahrungen zeigen, dass vier bis sieben Personen eine gute Teamgröße ergeben. Dabei sollte, wie auch beim Design Thinking, darauf

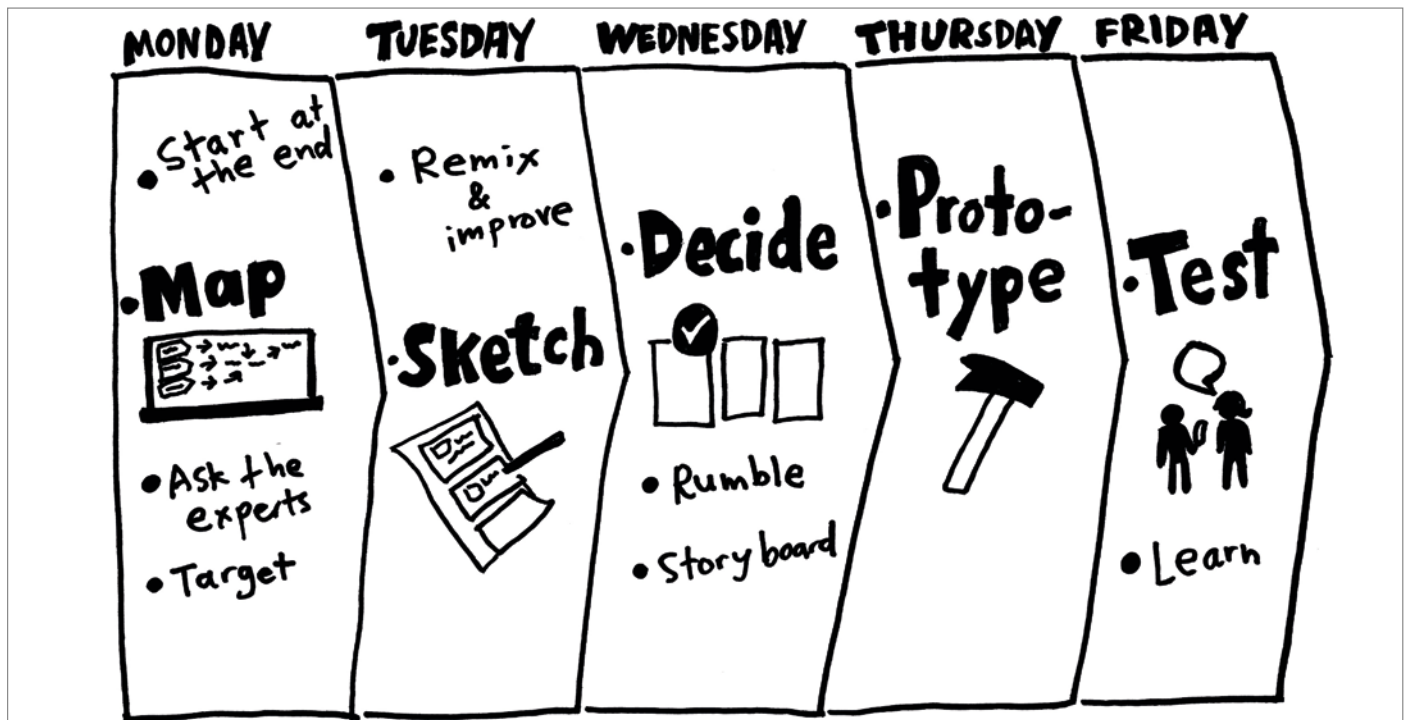


Abbildung 3: Die Design-Sprint-Woche aus dem Sprint-Buch von GV

geachtet werden, ein möglichst diversifiziertes Team zu haben. Eine der Personen übernimmt die Rolle des Entscheiders. Dieser hat das letzte Wort bei allen Entscheidungen während der Woche.

Den Nutzer und die Herausforderung verstehen

Am Montag geht es darum, die Herausforderung und den Nutzer zu verstehen. Bei einigen Teilnehmern wird die Versuchung groß sein, direkt über Lösungen nachzudenken. Man sollte jedoch zunächst im Team ein gemeinsames Verständnis der Herausforderung finden. Als Ergebnis des Montags wird eine Karte entstehen, auf der zu sehen ist, wie der Kunde sich durch den eigenen Service oder das eigene Produkt bewegt. Aus dieser Karte wird eine Aktivität herausgegriffen, die in diesem Sprint verbessert wird. Um dorthin zu kommen, sind aber noch einige Schritte notwendig. Als Erstes wird gemeinsam eine Zukunft beschrieben, in der das Projekt optimal gelaufen ist. Wie genau wird diese Zukunft aussehen?

Am Ende soll ein Satz formuliert werden, der diese Zukunft beschreibt. „Wir haben unsere Kunden dazu gebracht, Kaffee auch online zu kaufen“, wäre ein Beispiel für einen solchen Satz.

Danach werden Fragen gesammelt, die positiv beantwortet werden müssen, um die Zukunftsvision zu erreichen. Zum Beispiel „Vertrauen die Nutzer unserer Expertise?“. Auf dieser Basis wird die erste Version der zuvor beschriebenen Karte erstellt. Diese wird danach in einer „Fragen den Experten“-Runde beantwortet. Hier werden Fachleute außerhalb des Teams gebeten, kurze Vorträge über das Problemfeld zu halten; danach wird ein Interview geführt. Durch das neu gewonnene Wissen wird die Karte überarbeitet. Außerdem sammelt jeder Teilnehmer während der Expertenrunde seine beschriebenen „Wie-Könnten-Wir“-Fragen.

Nachdem der letzte Experte den Raum verlassen hat, werden diese Fragen aufgehängt und gruppiert. Um den Tag abzuschließen, legt sich der Entscheider auf ein Sprintziel fest. Durch die intensive Auseinandersetzung mit der Thematik ist es meistens sofort klar, welche Herausforderung im Laufe der Woche bewältigt werden sollte.

Divergieren und Konvergieren

Der Dienstag ist Skizziertag. Jetzt geht es darum, Lösungen zu erarbeiten. Diese sind nun keine abstrakten Ideen mehr,

sondern als konkrete Skizze für jeden zugänglich.

Die Skizzen müssen kein Kunstwerk sein, sondern nur das gewünschte Feature fundamental darstellen. Um diese Features zu finden, wird am Vormittag eine „Lightning Demos“-Runde gestartet. Dort werden bereits bestehende großartige Lösungen für ähnliche Probleme vorgestellt. Die daraus entstehenden Ideen für das Sprintziel werden auf einer Wand gesammelt. Diese Sammlung dient am Nachmittag als Vorlage für die eigenen Skizzen. Die Teilnehmer erarbeiten in Einzelarbeit so viele Lösungen, wie ihnen einfallen, und bringen sie zu Papier. Diese Papiersammlung wird zusammengetragen und für den nächsten Tag auf einen Stapel gelegt.

Am Mittwoch werden die harten Entscheidungen darüber getroffen, welche Lösungen in den Prototypen landen werden. Zunächst wird eine Galerie aller skizzierten Lösungen aufgebaut. Diese werden dann still von allen Teilnehmern analysiert.

Danach wird eine konstruktive Kritikrunde zu jeder Skizze gestartet. Unter diesem Einfluss trifft der Entscheider die Auswahl, welche der Lösungen in den Prototyp wandern. Aus den Lösungen wird nun ein Ablaufplan erstellt, der die Grundlage für den Donnerstag sein wird.

Prototypen und Testen

Am Donnerstag kommen Arbeit und Entscheidungen der ersten drei Tage zum Tragen. Heute geht es darum, eine realistische Fassade für den Nutzer zu erstellen. Diese muss nur ausreichend komplex sein, um das Sprintziel zu validieren. Beispielsweise kann es ein klickbarer Prototyp in PowerPoint oder Keynote sein. Um in dem überschaubaren Zeitraum von einem Tag einen funktionierenden Prototyp zu bauen, teilt man die Arbeit in mehrere Stücke auf. Bis auf eine Person werden diese Stücke auf die Teammitglieder verteilt. Diese eine Person ist nur dafür zuständig, die Geschichte am Ende des Tages wieder zusammenzuführen. Er achtet auch den Tag über darauf, dass alle Teile zusammenpassen.

Am Ende des Tages trifft sich das komplette Team und spielt nochmal den kompletten Prototyp durch. Dabei können auch kleinere Fehler korrigiert werden. Zum Abschluss des Sprints am Freitag

wird getestet. Während der Woche wurden fünf Nutzer mit einem passenden Profil ausgewählt. Diese dürfen nun den zuvor erstellten Prototyp ausprobieren. Durch Einzelinterviews mit den Nutzern werden wertvolle Erkenntnisse über die Annahmen des Sprints gesammelt. Diese bieten die Grundlage für das weitere Vorgehen. Entweder beginnt der Weg zu einem erfolgreichen Produktstart dank der letzten Woche oder man muss noch etwas weiter an der Lösung arbeiten. Hierfür bietet sich ein zweiter Sprint an, der auf den Erkenntnissen des ersten aufbaut.

Fazit

Beide Methoden liefern die idealen Werkzeuge, um schnelles Feedback von den Kunden zu bekommen. Im Falle eines Design Sprint dauert die Feedbackschleife sogar nur eine Woche. Dadurch rücken die tatsächlichen Bedürfnisse der Men-

schen ins Zentrum. Es wird iterativ die Schnittmenge der Bedürfnisse mit dem, was technisch umsetzbar und wirtschaftlich tragfähig ist, gefunden. Darüber hinaus wird auch die vorhandene, aber versteckte Kreativität der Menschen geweckt und genutzt. Die Fähigkeit, das zu tun, wird auch weiterhin einen entscheidenden Wettbewerbsvorteil für jeden bedeuten, der sie nutzt.



Maximilian Liesegang
maximilian.liesegang@esentri.com

Alles, was die SAP-COMMUNITY wissen muss, finden Sie monatlich im E-3 MAGAZIN.
Ihr WISSENSVORSPRUNG im Web, auf iOS und Android sowie PDF und Print:
e-3.de/abo

Wer nichts weiß,
muss alles glauben!

Marie von Ebner-Eschenbach



SAP® ist eine eingetragene Marke der SAP AG in Deutschland und in den anderen Ländern weltweit.

www.e-3.de

nem Smartphone lauffähig ist, sondern auch ganz einfach ein Web-Browser als Container für Apps dienen kann. Schön und gut, aber besitzt ein Web-Browser nicht viel zu wenig native Funktionalität, um mit einer Native App mitzuhalten? Vor ein paar Jahren hätte man genickt und „dito“ gesagt. Heute sieht das jedoch alles etwas anders aus.

Man öffne einfach mal in seinem Smartphone den Browser und tippe die URL „<http://whatwebcando.today>“ ein. *Abbildung 1* zeigt daraufhin die Funktionen, die der Browser beherrscht. Steht da tatsächlich „Push Messages“, „GeoLocation“, „Speech Recognition“ und „Offline Mode“? Google hat tatsächlich den Chrome-Browser dahingehend optimiert, dass er viele mobile native Funktionen nun unterstützt. Es werden zwar noch nicht alle Funktionen bedient, aber es wird davon ausgegangen, dass mit der aktuellen Anzahl an unterstützter nativer Funktionalität rund 90 Prozent der Native Apps auch als PWA entwickelt werden können, und das ist etwas, das alle Web-Entwickler schwärmen lässt.

What Web Can Do Today
Can I rely on the Web Platform features to build my app?
An overview of the device integration HTML5 APIs

✓ Feature available in your current browser ✗ Feature not available in your current browser

Native Behaviors	Seamless Experience	Input
<ul style="list-style-type: none"> LOCAL NOTIFICATIONS ✓ PUSH MESSAGES ✓ FOREGROUND DETECTION ✓ PERMISSIONS ✓ 	<ul style="list-style-type: none"> OFFLINE MODE ✓ HOME SCREEN INSTALLATION BACKGROUND SYNC ✓ INTER-APP COMMUNICATION ✓ 	<ul style="list-style-type: none"> TOUCH GESTURES ✓ SPEECH RECOGNITION ✓ CLIPBOARD (COPY & PASTE) ✓ POINTING DEVICE ADAPTATION ✓
Surroundings	Camera & Microphone	Location & Position
<ul style="list-style-type: none"> BLUETOOTH ✓ NFC ✓ PROXIMITY SENSORS ✗ AMBIENT LIGHT ✓ 	<ul style="list-style-type: none"> AUDIO & VIDEO CAPTURE ✓ ADVANCED CAMERA CONTROLS ✓ RECORDING MEDIA ✓ REAL-TIME COMMUNICATION ✓ 	<ul style="list-style-type: none"> GEOLOCATION ✓ GEOFENCING ✗ DEVICE ORIENTATION ✓ DEVICE MOTIONS ✓
Device Features	Screen & Output	Operating System
<ul style="list-style-type: none"> NETWORK TYPE & SPEED ✓ ONLINE STATE ✓ VIBRATION ✓ BATTERY STATUS ✓ 	<ul style="list-style-type: none"> FULLSCREEN ✓ SCREEN ORIENTATION & LOCK ✓ WAKE LOCK ✗ PRESENTATION FEATURES ✓ 	<ul style="list-style-type: none"> OFFLINE STORAGE ✓ FILE ACCESS ✓ CONTACTS ✗ STORAGE QUOTAS ✓

Abbildung 1: Was mein Browser alles unterstützt

Tap to choose your location stamp

Paper Planes
paperplanes.world

HINZUFÜGEN

1. App Installieren

Paper Planes

2. App ist auf Desktop

Throw and catch paper planes with people around the world

3. Native App Feeling

Abbildung 2: Installation einer PWA

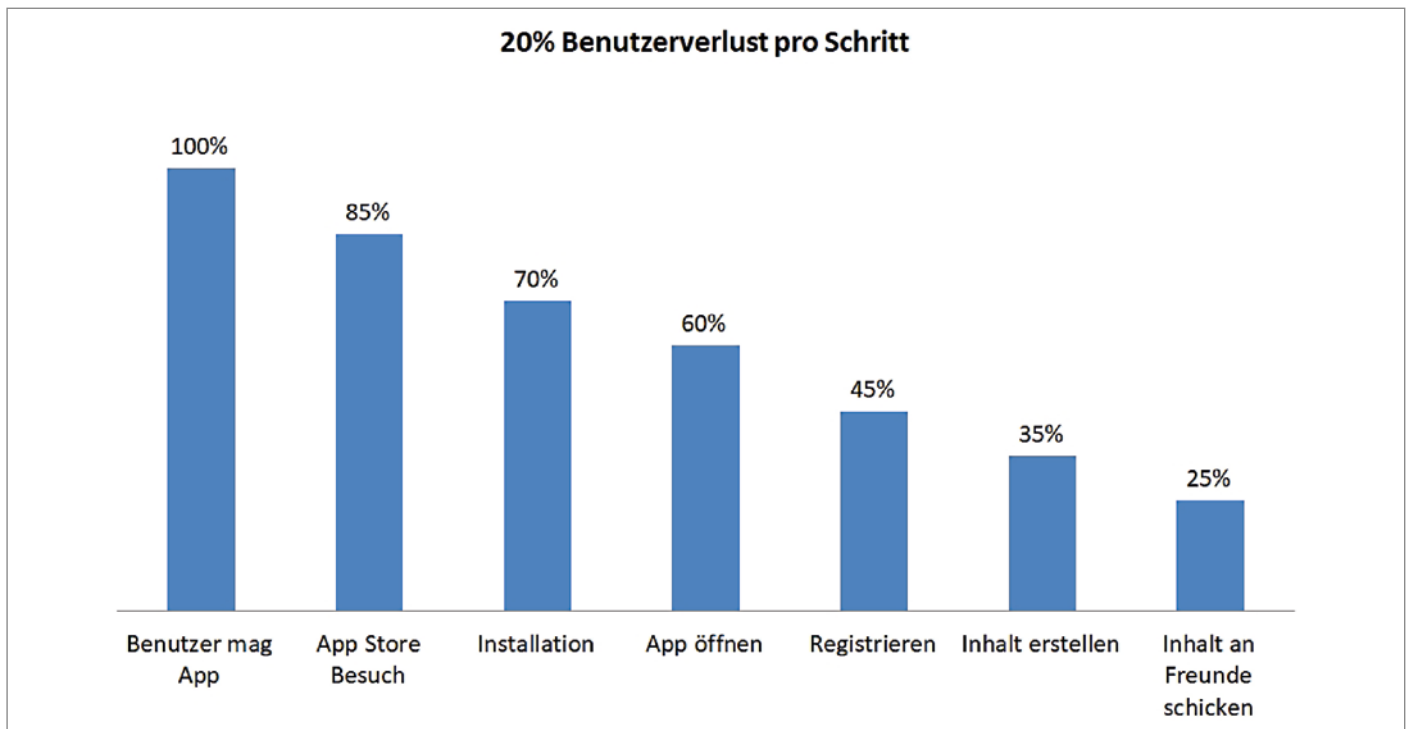


Abbildung 3: User-Verlust bei Erstnutzern einer App

Einziger Wermutstropfen ist aktuell, dass Apple PWAs nicht von Anfang an unterstützt hat. In Cupertino hat man sich lange gegen die Idee von Google gesträubt, was durchaus verständlich ist, da man mit dem Verkauf von Apps über den firmeneigenen App Store jährlich Millionen verdient. Weshalb sollte man also so etwas wie PWA unterstützen, das den eigenen App Store zum Großteil überflüssig macht? Mittlerweile hat man bei Apple die Zeichen der Zeit erkannt und sich dazu entschlossen, ebenfalls PWAs zu unterstützen. Man arbeitet bei Apple aktuell an der Erweiterung des Safari-Browsers und des iOS, damit diese in naher Zukunft für die Idee von Google betriebsbereit sind. Einen iPhone-User freut das natürlich besonders.

Was genau sind denn jetzt Progressive Web Apps?

Es gab so viel Gerede von PWAs und ein Endanwender hat noch immer keinen blassen Schimmer, was mit PWA gemeint ist. Das ist auch gut so, denn dem Endanwender soll der Unterschied zwischen PWA und Native App überhaupt nicht groß auffallen bis auf eine Kleinigkeit, die eine enorme Auswirkung hat. Die App wird zukünftig nicht mehr über den Google Play Store auf dem Handy installiert, sondern beim Besuchen der Web-App wird dem Endanwender die Möglichkeit geboten, die App direkt vom Browser aus zu installieren. Nach der Installation erscheint das typische Icon zum Starten

der App auf dem Desktop. Der Installations- und Startprozess ist in *Abbildung 2* visuell dargestellt.

Diese einfache Installation hat für Endanwender den Vorteil, dass sie sich nicht durch Millionen von Apps und Bewertungen in den App Stores durchklicken müssen, um diese eine App zu finden, die sie haben möchten. Stattdessen sehen sie eine coole Web-App und können sich genau diese mit einem einzigen Klick auf ihrem Smartphone installieren. Für das Deinstallieren der App wird der Standard-Deinstallationsprozess des Smartphones verwendet.

Es sollten nicht nur Entwickler und Endanwender beim Begriff „PWA“ aufhören, sondern auch die Marketing-Abtei-

	PWA	Hybrid	Native	Web-App
Push Notification	+	+	+	-
Offline Support	+	+	+	-
Installation über Browser	+	-	-	-
Installation über App Store	-	+	+	-
Regelmäßige App-Updates nicht notwendig	+	-	-	-
Abdeckung von 100 Prozent nativer Funktionalität	-	+	+	-
iOS-Support	-	+	+	-
Android-Support	+	+	+	-

Tabelle 1

Offline Wikipedia webapp

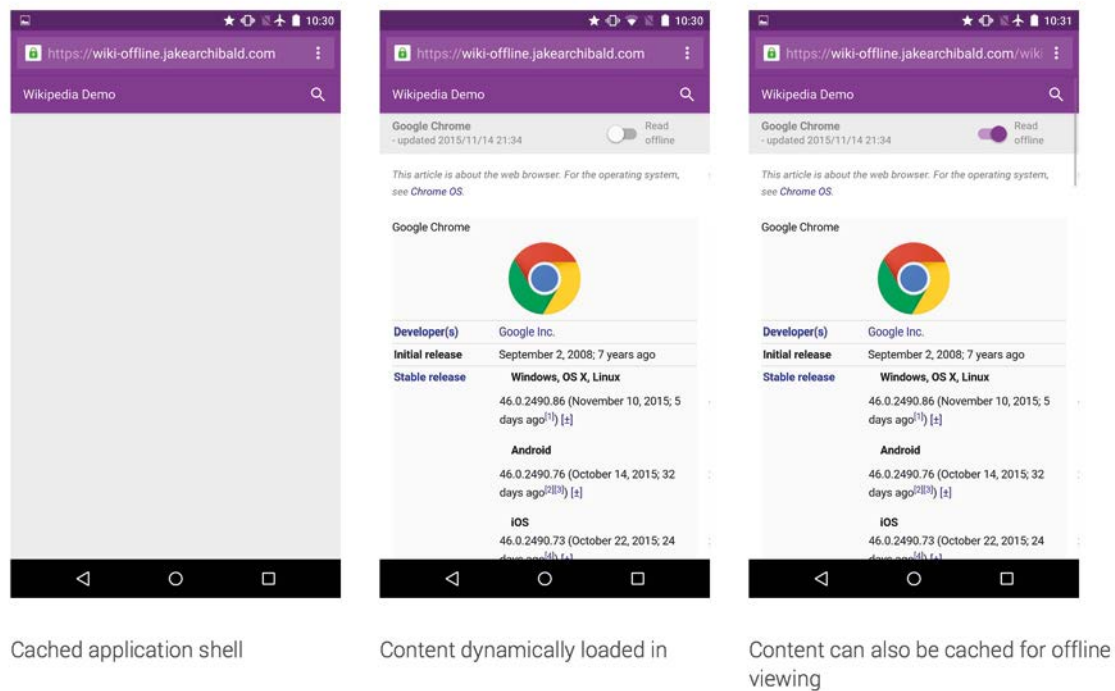


Abbildung 4: Quelle Google – App Shell

lung. Bei PWA können alle drei Parteien profitieren, was unter anderem durch eine interessante Aussage des Google-Managers Gabor Cselle untermauert wird. In einem Blog im Jahr 2012 hat er behauptet, dass mit jedem Schritt, den der Endanwender machen muss, bevor er einen Nutzen von der App hat, rund 20 Prozent der Endanwender und damit auch der potenziellen Kunden abspringen. Mithilfe von PWA fallen da zumindest schon mal die Installationsschritte weg. *Abbildung 3* zeigt den User-Verlust einer sozialen App wie Instagram anhand einer Statistik auf.

Web-App vs. PWA vs. Native App

Würde den Autor jemand fragen, was der Unterschied zwischen Web-Apps und PWAs ist, würde er sagen, dass eine PWA eine Web-App ist, die Responsive Design, Push Notifications und Offline-Support bietet. Zusätzlich gibt es noch die Möglichkeit, dass man eine PWA auf dem Smartphone recht einfach installieren kann, was mit einer Web-App so nicht möglich ist.

Der Unterschied zwischen PWA und Native Apps ist, dass PWAs noch nicht 100 Prozent der nativen Funktionalität beherrschen, was dazu führt, dass die eine oder andere App, die nach besonderer Funktionalität schreit, nativ entwickelt werden muss. Diese Lücke wird in den nächsten Monaten kleiner werden und spätestens dann muss sich jeder Anbieter bei einer Neuentwicklung mit dem Gedanken beschäftigen, ob er noch Native App gegenüber PWA verwenden möchte und welchen Vorteil eine native Entwicklung mitbringt.

Wie viel mit PWA schon möglich ist, sieht man daran, dass große Unternehmen wie Twitter ihre Anwendung bereits als PWA anbieten. Für den Anbieter ist der Kostenfaktor auch ein großer Vorteil einer PWA gegenüber einer Native App. Es gibt bei einer PWA nur einen Code zu entwickeln und zu pflegen, der sowohl für das Web als auch für iOS und für Android verwendet werden kann.

Tabelle 1 stellt nochmals die Möglichkeiten der einzelnen Entwicklungsvarianten dar. Da sich bei PWAs aktuell die Unterstützung nativer mobiler Funktionalität rasend schnell erweitert, sind die fehlenden Funktionen nicht explizit in

der Liste angegeben, da diese zum Zeitpunkt der Veröffentlichung dieses Artikels wahrscheinlich schon wieder veraltet wären.

Zu einigen Punkten aus der Tabelle sei kurz noch etwas erwähnt. Die Installation über einen App Store ist bei PWA aktuell nicht möglich. Wie man damit umgeht, ist noch nicht ganz klar. Viele sehen den App Store auch als Werbe-Plattform und gerade für große Firmen ist es Pflicht, dass ihre App eben auch im App Store vorhanden ist. Auch wenn es Twitter nun als PWA gibt, so kann es sich Twitter nicht leisten, die Entwicklung der Native App einzustellen, weil Twitter dann nicht mehr im App Store wäre.

Regelmäßige App-Updates sind bei PWA nicht mehr notwendig, da es sich unter der Haube noch immer um eine Web-Anwendung handelt. Die PWA erkennt automatisch, dass es eine neue Version gibt, die dann auch gleich verwendet wird. Dieser Update-Workflow ist in dieser Art eben auch möglich, weil PWAs nur einen Bruchteil an Speicherplatz einer nativen App oder gar einer Hybrid App benötigen.

Fehlender iOS-Support bedeutet, dass beim Aufruf einer PWA mit einem iOS-

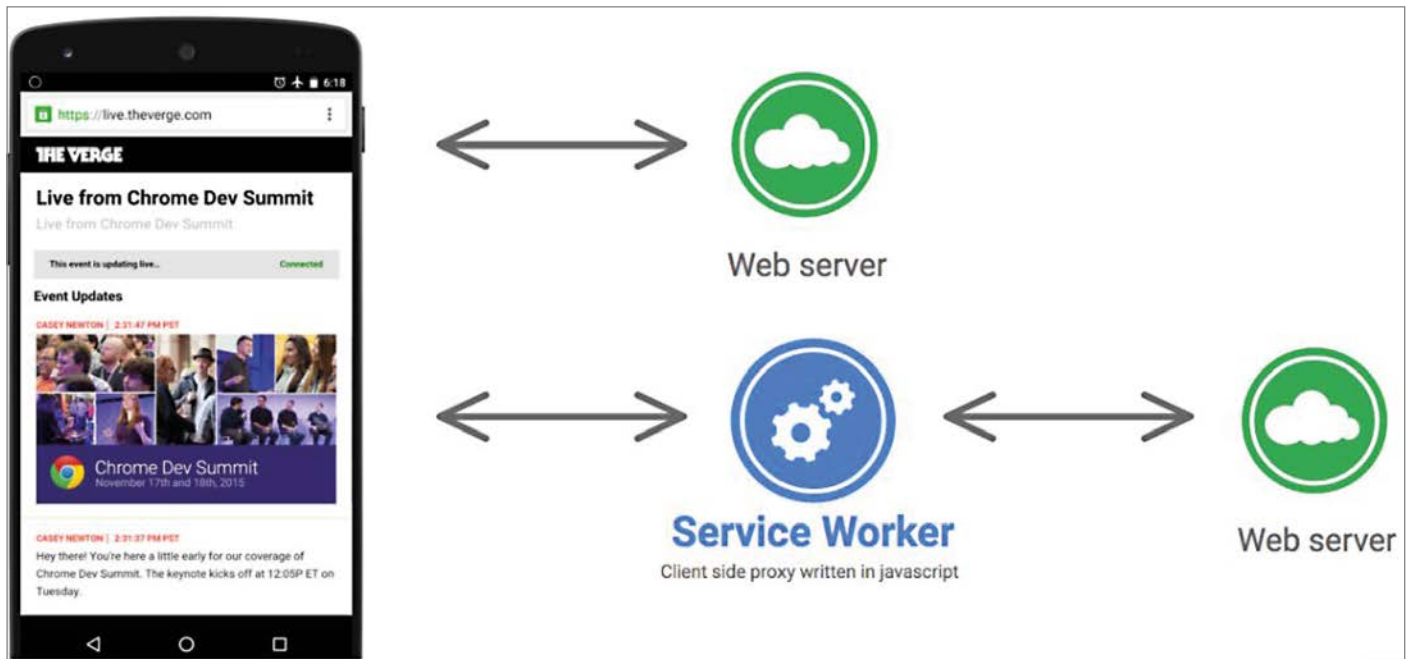


Abbildung 5: Quelle Google – Service-Worker

Browser die PWA dem Endanwender ganz normal als Web-App dargestellt wird. Es kommt hier zu keiner Fehlermeldung.

Alles dreht sich um das Caching

Für den Endanwender liegt ein großer Vorteil einer PWA gegenüber einer Web-App darin, dass eine PWA wesentlich schneller geladen wird, weil Google für das Caching gute Lösungen gefunden hat. Zum einen gibt es eine App-Shell, bei der es sich um statischen Inhalt, bestehend aus HTML, CSS, Bildern und JavaScript handelt, der im Cache geladen ist. Das ist sinnvoll, weil sich dieser Inhalt nur selten ändert. Das Cachen der App Shell sorgt auch dafür, dass dem Endanwender eine PWA schnell angezeigt werden kann, auch wenn nur eine 3G-Verbindung besteht. *Abbildung 4* deutet die Idee einer App Shell nochmal an. Wenn die PWA dann zum ersten Mal aufgerufen wird, wird gecacht. Beim zweiten Aufruf wird die App Shell schon aus dem Cache geladen und der Content asynchron nachgezogen.

Bei PWAs ist es allerdings nicht nur möglich, statischen Content zu cachen, sondern auch, die komplette Antwort von GET-Requests. Dies führt dazu, dass, je länger ein Nutzer sich auf einer Seite aufhält, diese desto performanter wird, weil

der Cache immer weiter gefüllt wird. Im Normalfall bedeutet dies dann auch, dass selbst bei einer fehlenden Internetverbindung etwas angezeigt wird. Bei einer Newsseite wäre dies dann eben die Ausgabe mit den Artikeln vom Vortag anstatt einer 404-Fehlermeldung.

Das Geheimnis des hochwertigen Caching bei PWA liegt in den Service-Workern, die im Hintergrund arbeiten. Ein Service-Worker ist eine JavaScript-Datei, die in der Anwendung läuft, aber in einem separaten Prozess. Dieser hat den Vorteil, dass der Service-Worker weiterläuft, auch wenn die Anwendung beendet wird. Er läuft die ganze Zeit im Hintergrund und kann frische Daten asynchron nachziehen oder auch auf Push Messages reagieren, die vom Server geschickt werden. *Abbildung 5* zeigt das einfache Basisprinzip eines Service-Workers ganz gut, indem dargestellt wird, dass er als Proxy zwischen Web Server und PWA dient.

Fazit

Der Autor war vor einigen Jahren von Hybrid Apps begeistert und ist nun von PWAs angetan, weil es Schritte in die richtige Richtung sind – mehr erreichen mit weniger Arbeit. Eine Frage, die er sich aktuell allerdings immer wieder stellt, ist, weshalb Google mit PWA etwas vor-

antreibt, das bei Erfolg den eigenen App Store überflüssig machen könnte. Immerhin verdient Google damit sehr viel Geld. Wenn PWAs allgegenwärtig sind, kann rein theoretisch jeder seinen eigenen App Store aufbauen, weil die Installation einer PWA nicht an Android oder iOS gebunden, sondern lediglich ein kompatibler Browser notwendig ist.



Tobias Huber
huber@lumabit.de

Willkommen, wie können wir Sie weiterbringen?

PRIVATPERSONEN

UNTERNEHMEN

INSTITUTIONEN



Arbeitslos und Arbeit finden

Jobsuche, So beantragen Sie Arbeitslosengeld, So beantragen Sie Arbeitslosengeld II



Schule, Ausbildung und Studium

Berufswahl, Studienwahl, Schulabschlüsse nachholen, Freiwillige Dienste



Familie und Kinder

Kindergeld beantragen, Kinderzuschlag beantragen, Familie und Beruf vereinbaren



Menschen mit Behinderungen

Wichtige Schritte nach Unfall oder Erkrankung, Spezielle Hilfe und Unterstützung, Persönliche Beratung



Karriere und Weiterbildung

Karrierepfad, Kurse finden, Online-Trainings



Für Menschen aus dem Ausland

Deutsch lernen, Job suchen, Praktikum suchen, Infos rund um finanzielle Leistungen, Ausländische Abschlüsse anerkennen lassen



Arbeitsagentur 2.0

Michael Adam, Bundesagentur für Arbeit

Als einer der größten Dienstleister im öffentlichen Sektor hat sich die Bundesagentur für Arbeit im Zeichen der Umsetzung eines kundenzentrierten eGovernment-Portals neu aufgestellt. Grundlage ist, alle Dienstleistungen und Anliegen der Kunden durch eine Plattform zu bündeln und einfach für den Kunden nutzbar bereitzustellen.

Die strategische Quelle und der gesetzliche Auftrag der Bundesagentur für Arbeit (BA) finden sich in der Digitalen Agenda des Bundes und genauer im daran ausgerichteten

Entwicklungsprogramm „BA 2020“ wieder. Kern ist hier das Leistungsversprechen, den Kundinnen und Kunden „dort zu begegnen, wo sie sind“, und damit neue, benutzer-

freundliche Online-Angebote zu realisieren. Zusätzlich sollen über die Digitalisierung und Automatisierung von End-to-End-Prozessen die Kundenanliegen schneller, ef-

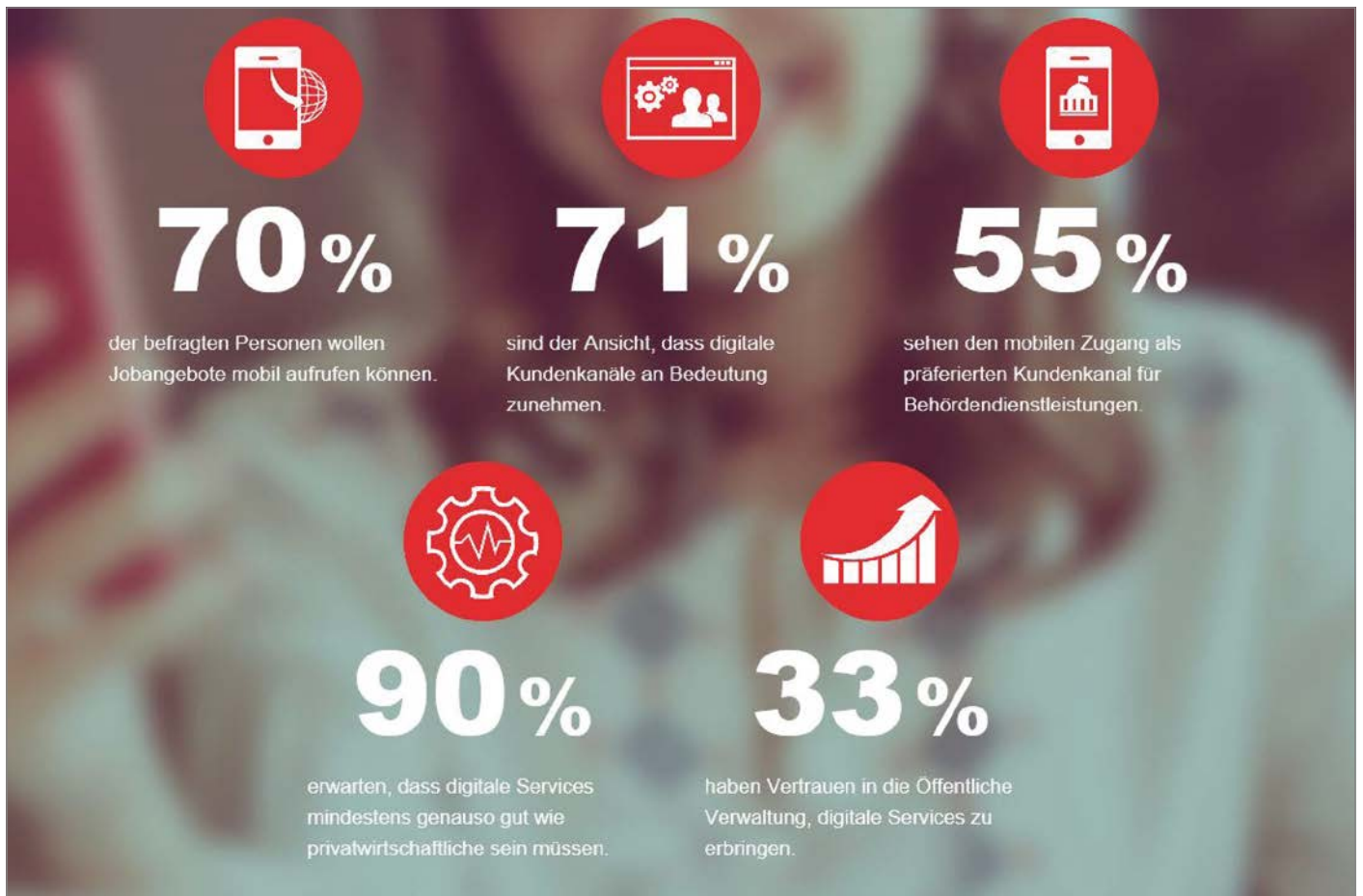


Abbildung 1: Echte Internetseiten statt PowerPoint-Slides

fizienter und in höherer Qualität erledigt werden. Der Mensch wird gezielt in den Mittelpunkt der Konzipierung und Produktentwicklung gestellt. Eine erlebbar bessere User Experience als Schlüssel zur besseren Akzeptanz und Nutzung von ehemals unangenehmem Papierkram ist Programm.

Mit dem an verschiedenen Lebenslagen der Kundinnen und Kunden ausgerichteten Online-Auftritt möchte die BA den Nutzererwartungen hinsichtlich digitaler Behördendienstleistungen mehr als

nur gerecht werden. Ziel ist es, die individuellen Bedürfnisse und Situationen, in der sich die Nutzerinnen und Nutzer befinden, zu erkennen und durch reibungslose Abläufe zu bedienen.

„Ist“ vs. „Soll“

Bürgerinnen und Bürger wie Unternehmerinnen und Unternehmer erwarten auch vom Staat zunehmend digitale Dienstleis-

tungen. Zur IT-Strategie 2020 der BA gehören daher individualisierte, einfach nutzbare Angebote. Der Kunde findet sofort Themen, die zu seiner persönlichen Lebenslage passen, und bekommt mit drei Klicks die gewünschten Informationen.

Die BA hat ein Internetportal entwickelt, das dem Nutzer die Kontaktaufnahme erleichtert und viele Fragen sofort beantwortet. Es funktioniert als One-Stop-Shop: Die Startseite listet alle Anliegen auf, die Bürgerinnen und Bürger,



Abbildung 2: Definition von Personas strukturiert die Inhalte

Unternehmen oder Institutionen in einer bestimmten Lebenslage haben können. Aus diesen Anliegen heraus erfolgt eine exakte Hinführung zu den richtigen Informationen oder dem Fragenkatalog für die entsprechende Leistung. Einheitliche Gestaltung und zeitgemäße Navigation ermöglichen es, sich arbeitssuchend zu melden oder Antworten, etwa auf Fragen zu Themen wie „Kindergeld“, „Berufsberatung“ oder „Weiterbildung“, zu erhalten.

Neben Informationen und Wegweisern sind seit Januar 2017 erste Services im BA-Portal im Bereich der Unternehmen vollständig online zur Verfügung gestellt. Das Beantragen einer Betriebsnummer für Unternehmen erfolgt seitdem vollständig online und sofort. Weitere Angebote werden folgen.

Laut Dr. Markus Schmitz, dem Generalbevollmächtigten und Chef der IT der BA, war der alte Internetauftritt wenig intuitiv und zu sehr aus der Sicht einer Behörde gedacht. „Mit dem neuen Onlineportal haben wir einen wichtigen Schritt gemacht. Wir werden es um neue Angebote erweitern.“ Gemeint ist mit diesem Schritt der BA die Umkehr ihrer Perspektive beziehungsweise deren Darstellung: Die Bundesagentur erzählt den Menschen nicht, was sie kann, sondern wie sie den Kundinnen und Kunden intuitiv helfen kann (siehe Abbildung 1).

Mit der richtigen Methode zum Erfolg

Eine der wegweisenden Methoden für die Entwicklung war, dass durch Design-Thin-

king stets das Interesse des Kunden im Mittelpunkt stand – ein für Behörden eher untypisches Vorgehen. Binnen kürzester Zeit entwickelten Experten sogenannte „Personas“ und „Customer Journeys“ – konsequent aus der Perspektive der Nutzerinnen und Nutzer. Neben dem Design-Thinking- und dem Rapid-Prototyping-Ansatz (der schnellen Entwicklung von Klick-Prototypen, Mock-ups oder Design-Entwürfen) kam detailliertes Wissen über den Umgang der Kundinnen und Kunden mit digitaler Kommunikation sowie ihre Erwartungen an diese zum Tragen. Erfolgsentscheidend war auch die schnelle Visualisierung – binnen sechs Wochen gab es bereits einen klickbaren Prototyp.

Die inhaltliche Struktur basiert auf der Definition von 35 Personas. Hierbei steht jede Persona für eine idealtypische Kun-




USABILITY TESTING	KUNDENBEFRAGUNG	RÜCKMELDUNG VON MITARBEITERN
 <p>In einem Testlabor in Berlin wurden mit 15 Probanden „Thinking aloud“ und „Eyetracking“-Tests durchgeführt.</p>	 <p>In den BIZen der ausgewählten Pilotagenturen in Trier, Coesfeld und Oldenburg-Wilhelmshaven wurden Kundenbefragungen durchgeführt.</p>	 <p>Im Rahmen einer Open Innovation Kampagne konnten Mitarbeiter/innen der Pilotagenturen Rückmeldungen zum neuen Webportal geben.</p>

Abbildung 3: Nutzer laufend an der Verfeinerung beteiligt



**WEITERGEDACHT,
UM SIE WEITERZUBRINGEN!**

Die neue Website rund ums Leben. ▶

Abbildung 4: Entscheidender Schritt zur Digitalisierung

engruppe der BA und deren ganz speziellen Anliegen. Die Personas selbst wurden vom Projekt im engen Austausch mit Angestellten der BA entwickelt, die ihre praktische Erfahrung aus Beratungsgesprächen einbrachten (siehe Abbildung 2). „In den Workshops waren Mitarbeiterinnen und Mitarbeiter, die noch am Tag zuvor mit Arbeitssuchenden am Tisch gesessen hatten und genau wussten, wie und wonach diese Zielgruppe sucht“, sagt Bernd Karl, Projektleiter Accenture Consulting und Leiter Digitale Transformation in der öffentlichen Verwaltung. „Durch unsere besondere Herangehensweise konnten wir das Konzept auf greifbare und emotionale Weise visualisieren und so alle Stakeholder sehr schnell von der Idee überzeugen.“

Für die BA bedeutete dies mehr als nur die Einführung agiler Techniken und Methoden auf Teamebene, sondern vor allem auch die möglichst reibungsfreie Einbindung agiler Firmenbereiche in das Gesamtunternehmen. Für diese Aufgabe benötigte es Experten, die den Überblick über das gesamte Spektrum der Agilität haben – angefangen bei Design-Thinking über SAFe, Scrum und Kanban bis hin zu DevOps. Diese Methoden mussten an die jeweiligen Unternehmensbedürfnisse angepasst werden.

Der Schlüssel zum agilen Erfolg lag darin, unternehmenseigene agile Coaches in der Bundesagentur für Arbeit aufzubauen und ein tiefes Verständnis für Agilität im gesamten Management zu schaffen. So wurden die notwendigen Veränderungen fest in der Organisation verankert.

Eine schnelle Entwicklung des Produkts zum Anfassen

Viele Nutzerinnen und Nutzer fanden sich in der alten BA-Onlinewelt, die aus 95 verschiedenen Angeboten bestand, nur schwer zurecht. Das neue Portal bündelt jeden Tag mehr Themen, leitet strukturiert durch die jeweiligen Prozesse und bereitet so das Klären individueller Fragen im persönlichen Kontakt vor, etwa via geplante Telefonat oder Text-Chat.

Im Kontakt mit der BA bewegt sich der Nutzer damit durchgehend in einer einheitlichen Umgebung und erfährt so eine „seamless customer experience“. So kann auf sein Anliegen kanalübergreifend op-

timal reagiert werden. Ein Servicekonto und eine nach seinem Surfverhalten individualisierte Startseite verbessern Servicequalität und Bedienerfreundlichkeit im nächsten Schritt weiter. Grundsätzlich ist eine Personalisierung im gesamten BA-Portal angedacht.

Mit dem Einstieg in die Customer Journey auf der Hauptseite startet ein durchdachter End-to-End-Zyklus, der sich in die individuellen Anliegen aufteilt. Schritt-für-Schritt-Anleitungen führen Nutzer durch einen ansonsten komplexen und schwer fassbaren Prozess, schaffen Verständnis und bieten direkte Einstiege in Onlineprozesse. Um die entsprechenden Prozesse anlegen zu können, wurde der Design-Thinking-Ansatz angewendet, mit dem sich Motivation und Bedürfnisse von Kunden verstehen lassen. Wichtig war etwa, wo jemand seine Reise bei der BA beginnt, welches Ziel er hat und wie er es am liebsten erreichen möchte – online oder im persönlichen Gespräch.

Ein entscheidender Schritt zur Digitalisierung

Das positive Feedback wird als Antriebsfeder der weiteren Entwicklung genutzt. Wesentlich zur Entwicklung trugen nämlich eben diese praxisorientierten Anregungen und die permanente Rückmeldung von BA-Mitarbeitern sowie Unternehmen und Bürgern bei, die das Portal nutzen. Dazu setzte die Bundesagentur unter anderem Usability-Tests und eine Open-Innovation-Werkstatt ein.

Wertvollen Input lieferten objektive Usability-Tests durch externe Prüflabore ebenso wie kreative Workshops, in denen BA-Angestellte und auch deren Kunden die aktuelle Version des Portals ausprobieren und Verbesserungsvorschläge machen konnten. Weitere Anregungen wurden beispielsweise an Präsentationsständen in Berufsbildungszentren und in der Nürnberger Fußgängerzone gesammelt oder im Innovationslabor JOSEPHS des Fraunhofer-Instituts durch co-kreative Prozesse erarbeitet. Alle Informationen flossen in die agile Entwicklung des Internetauftritts ein, der auch weiterhin über die Open-Innovation-Plattform auf Funktionalität und Akzeptanz überprüft sowie kontinuierlich im laufenden Betrieb verbessert wird (siehe Abbildung 3).

Seit Sommer 2017 wird das Nutzungsverhalten (wie Verweildauer, häufig verwendete Suchbegriffe und Klick-Verhalten) mit dem Gedanken analysiert, die Benutzung direkt zu verbessern. Im Januar 2018 ist eine Ausweitung der Analyse auf weitere Bestandteile des BA-Portals vorgesehen. Zudem werden ab Frühjahr 2018 einzelne Funktionen in ausgewählten Bereichen im Pilotbetrieb produktiv gesetzt, um schon im Vorfeld die Akzeptanz der Bürgerinnen und Bürger direkt im Portal nachweisen zu können. Der gegebenenfalls langwierigen Entwicklung ohne direkte Berührungspunkte zum Nutzer wird so entgegengewirkt (siehe Abbildung 4).

Fazit

Insgesamt wird das Portal als großer Erfolg und entscheidender Schritt zum digitalen Wandel angesehen. Nach dem erfolgreichen Start des Portals werden jetzt weitere Services schrittweise integriert und ans einheitliche Design angepasst – bisher sind nur die ersten drei bis vier Ebenen harmonisiert. Parallel beginnt die Optimierung der Prozesse, beispielsweise durch die automatische Zuteilung der Betriebsnummer für Firmenkunden. Für die BA-Mitarbeiter gab es eine Kampagne mit Flyern, Postern und Beiträgen im Intranet. Die Kunden wurden zum Start mit TV-Spots auf das neue Portal aufmerksam gemacht. Kundinnen und Kunden sowie Mitarbeiterinnen und Mitarbeiter der BA loben das neue Webportal und in einer Nutzerbefragung halten es 80 Prozent für gelungen.



Michael Adam
michael.adam@arbeitsagentur.de



Mit Chatbots und Voice zum flexiblen, Kontext-unterstützenden Arbeiten in der digitalen Welt

Christian Ochsenkühn, OPITZ CONSULTING Deutschland GmbH

Dieser Artikel widmet sich einigen grundlegenden Fragen, die sich Unternehmen stellen, wenn sie im aktuellen Hype um Chatbots und/oder Sprachassistenten vorne mit dabei sein wollen: Wie funktioniert ein Chatbot? Warum sollten wir auf Chatbots setzen? Was bringt das unseren Nutzern?

Auf der Hauskonferenz „inspire|IT“ hat der Autor in diesem Jahr zum ersten Mal „Alexa Skill“ vorgestellt. Dazu wurde einen Tisch vorbereitet, auf dem ein Amazon Echo, ein Laptop und ein Poster mit einem sogenannten „Wimmelbild“ auf die Besucher warteten. Letzteres ist ein Bild, auf dem es vor Details nur so wimmelt; daher der Name. In diesem Fall zeigte es die unterschiedlichsten Projekte, die das Unternehmen des Autors bereits im Digitalisierungs-

kontext durchgeführt hat (siehe Abbildung 1). An dem Tisch konnten die Besucher nun von Alexa etwas zu diesen Projekten erfahren, etwa indem sie nach der Aufforderung „Erzähle mir etwas zum Flughafen“ anschließend die Frage stellten: „Welche Technik steckt dahinter?“. Alexa merkte sich in diesem Fall, dass wir zuvor über das Flughafen-Projekt gesprochen hatten und nannte die zugehörigen Informationen zu einigen technischen Details.

Wie ist man innovativ?

Nach einiger Zeit kam ein älterer Herr im Anzug auf den Autor zu und ließ sich die verschiedenen Sprachbefehle erklären. Interessiert fragte er: „Wie hat das alles angefangen?“ Eine gute Frage. Der Autor wollte direkt loslegen und erklären, dass er einen ersten Prototyp in Python entwickelt hatte und später – als ihn ein Kollege bei der Entwicklung unterstützte

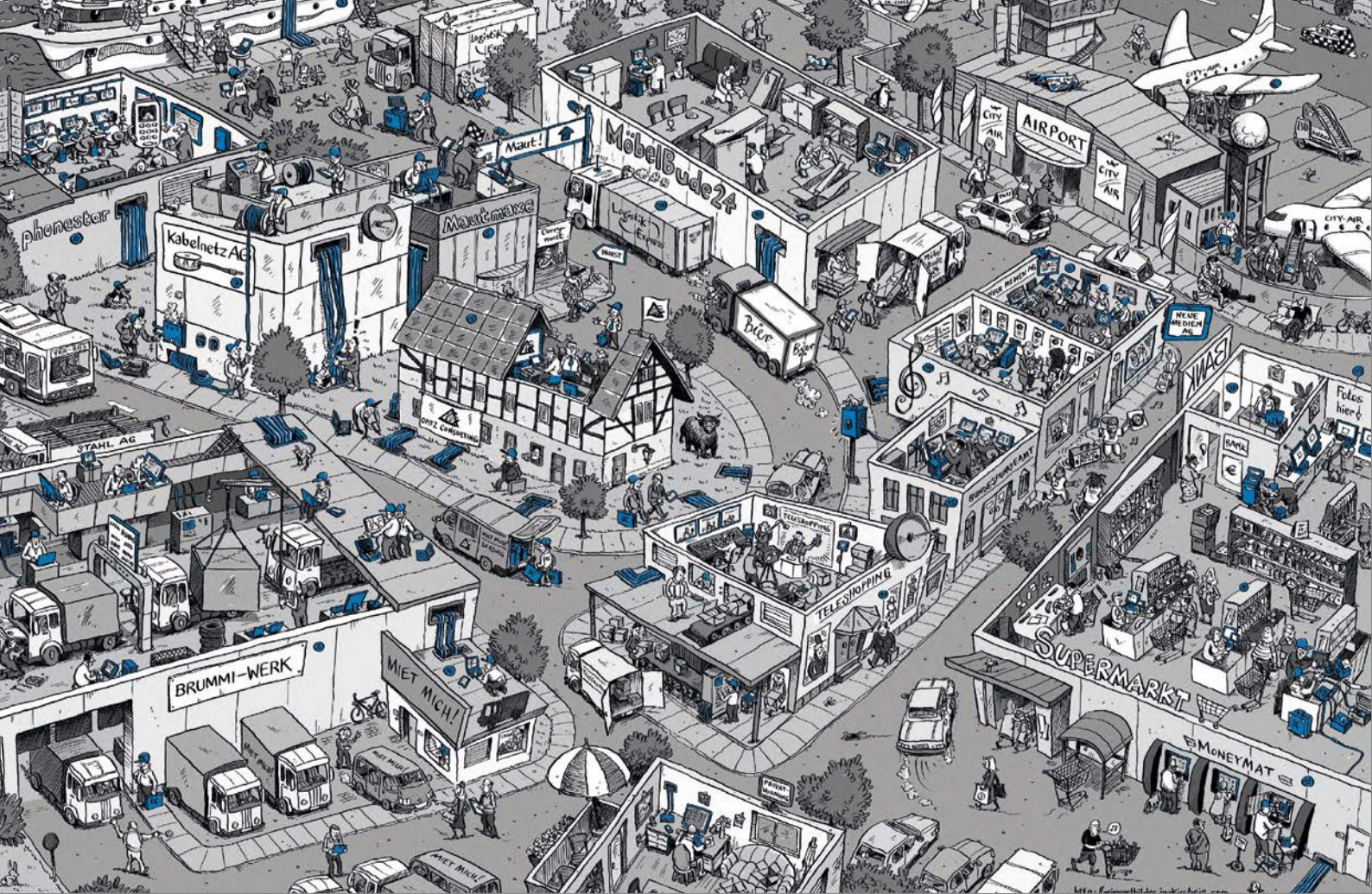


Abbildung 1: Zu diesem Wimmelbild konnte man den Alexa Skill befragen

– mangels Dokumentation und auf Anraten seitens Amazon auf JavaScript umgeschwenkt war. Da unterbrach der Besucher ihn auch schon und meinte: „Nein, nein, ich meine noch einen Schritt zuvor. Woher kommt die Idee dafür? Wie ist man innovativ?“

Eine schwierige aber gleichzeitig ebenso gute Frage. Die Idee zum Skill ist dem Autor damals einfach so eingefallen, oder? Das ist natürlich nur die halbe Wahrheit. Grundsätzlich ist es zwar richtig: Ideen entstehen überall. Egal, ob unter der Dusche, beim Joggen durch den Wald oder wenn man auf dem Balkon stehend den Blick in die Ferne schweifen lässt.

Damit es in diesen entspannten Momenten aber zum Klick für eine neue Idee kommt, muss einiges an Vorarbeit geschehen sein. So hatte sich der Autor vor der eigentlichen Entwicklung des Skills einige Fragen gestellt: „Was kann die Technologie bereits, die ich einsetzen möchte?“, „Wie funktioniert sie?“, „Was macht die Konkurrenz bereits damit?“ und „Was bringt das Ganze dem Nutzer?“ Diese Fragen sollen uns bei der Entscheidung helfen, wie und ob wir einen Chatbot oder

ein Sprach-Interface für einen Business Case einsetzen möchten.

Wie ein Chatbot funktioniert

Grundsätzlich gibt es zwei Arten von Chatbots, die sprach- und die textbasierten. Beide sind sich (auch technisch) sehr ähnlich. Sie ermöglichen uns, mit verschiedenen Endgeräten in natürlicher Sprache zu kommunizieren. Endgeräte für sprachbasierte Chatbots – meist Sprach-Assistenten genannt – sind beispielsweise Amazons Echo mit Alexa, Apples Siri, Microsofts Cortana oder auch Google Home mit dem Google Assistant, der bereits auf allen neueren Android-Geräten zu finden ist. Textbasierte Chatbots laufen hingegen häufig auf dem Smartphone in diversen Messengern wie dem Facebook Messenger, dem Chatdienst Slack oder auch dem vor allem in China verbreiteten WeChat.

Die Sprach-Assistenten unterstützen von Haus aus bereits einige Funktionen wie die Frage nach dem Wetter oder das Abspielen von Musik. Deren Funktionsumfang lässt sich erweitern, indem sogenannte „Skills“ (ähnlich einer App auf dem

Smartphone) aktiviert werden. Die grundlegende Funktionsweise dieser Skills und aller verfügbaren Chatbots ist sehr ähnlich (siehe Abbildung 2).

Am Anfang jeder Konversation steht ein Nutzer, der per Sprache mit einem der oben genannten „Clients“ (wie Alexa oder Facebook Messenger) interagiert. Von dort wandert dieser Sprach- oder Textbefehl zunächst in die Cloud, wo er übersetzt wird. Für Alexa findet das bei Amazon statt, bei textbasierten Chatbots übernehmen dies Anbieter wie Oracle Intelligent Bots, „api.ai“ (von Google) oder auch „wit.ai“ (von Facebook). Dort wird der eingehende Satz – nachdem er eventuell von Sprache zu Text transformiert wurde – analysiert und einer Aktion aus einem Pool von Aktionen zugeordnet.

Dieser Pool ist allgemein als „Intent-Schema“ bekannt und wird vom Chatbot-Entwickler beschrieben. Jede Aktion, die unser Skill oder Chatbot ausführen oder beantworten können soll, ist dort festgelegt. Damit nun die eigentliche Intelligenz dieser Übersetzungen erkennt, welchen Intent wir mit einem Satz ansprechen möchten, müssen die Entwickler zuvor pro Intent einige Beispielsätze angeben.



Abbildung 2: Die grundlegende Funktionsweise von Chatbots und Spracheingaben

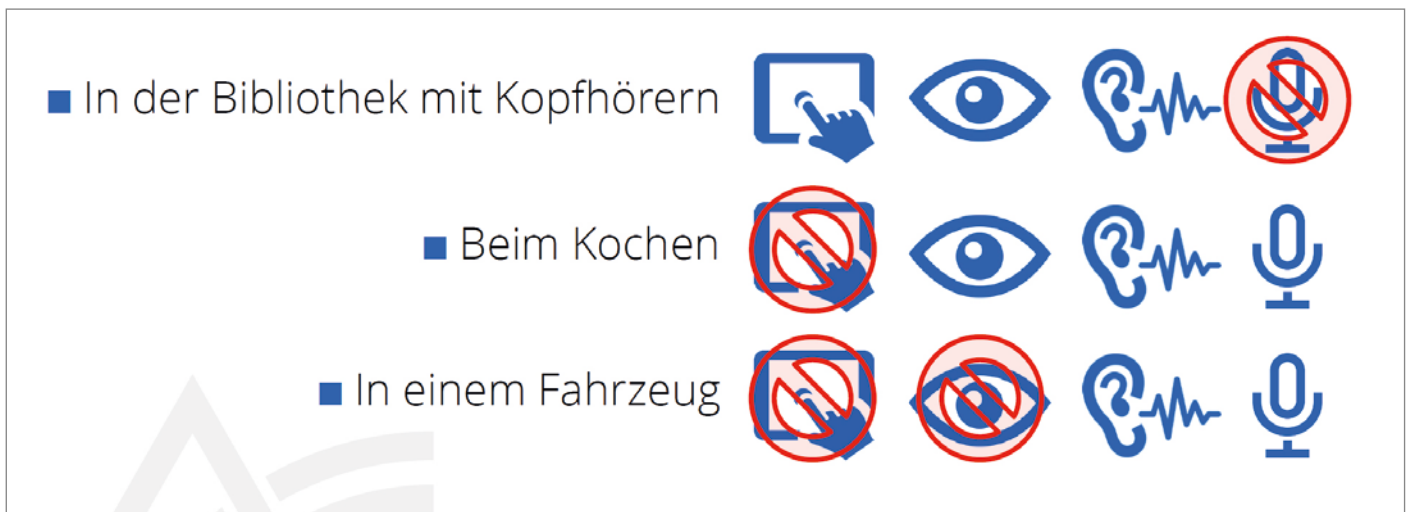


Abbildung 3: Je nach Kontext sind unterschiedliche Interaktionen (wie Spracheingabe) sinnvoll

Daraus sollen sie lernen und auch ähnliche Sätze dem richtigen Intent zuordnen.

Erst jetzt kommt von dort eine Anfrage beim Code des Chatbots an. Dies geschieht über eine Web-Schnittstelle. Man kann den Code also grundsätzlich dort hosten, wo man möchte. In der eingehenden Anfrage erhalten wir aber nicht etwa den gesprochenen oder geschriebenen Satz, sondern den zuvor erkannten Intent, also die Aktion, die ausgeführt werden soll. Zusätzlich gibt es noch einige Informationen aus dem Request sowie einen Status, eine User-ID oder bestimmte Worte aus dem initialen Satz, den man weiterverarbeiten will. Eventuell möchten wir zum Beispiel eine Zahl, ein Datum, einen Ort oder ein anderes bestimmtes Wort aus dem Satz verwenden, um eine bestimmte Aktion auszulösen.

Auf diesen Intent kann man nun reagieren und eine passende Antwort kreieren. Was unsere Aktion genau auslöst, hängt vollkommen vom Anwendungsfall ab. Der Bot kann direkt eine ganz einfache

Antwort zurückgeben, eine Audio-Datei abspielen, einen weiteren (Web-)Service ansprechen, der wiederum weiterführende Informationen bereitstellt, oder gar eine zusätzliche künstliche Intelligenz nutzen, um eine dynamisch generierte und wirklich intelligente Antwort liefern zu können. Egal welche Aktion wir auslösen, letztendlich wird dazu eine Antwort generiert, die wieder im Client und somit beim Nutzer landet. Der Kreis schließt sich.

Sind Bots die neuen Apps?

Jetzt, wo wir wissen, wie Chatbots grundsätzlich funktionieren, stellen sich neue Fragen: „Brauchen wir überhaupt einen Bot?“ oder „Sollen wir ausschließlich Bots einsetzen, um unseren Kunden Mehrwert zu bieten?“ Die Antwort lautet: „Ja und nein“. Die Interaktion mit menschlicher Sprache bietet eine gute Möglichkeit, das Angebot schnell um einen weiteren Ser-

vice beziehungsweise ein weiteres Produkt zu erweitern, indem man bereits vorhandene Daten und Schnittstellen im Unternehmen nutzt und diese wiederverwendet. Jedoch ist nicht jeder Service dafür geeignet, um zum Beispiel per Sprache bedient zu werden. Vielmehr ist der Kontext wichtig, damit ein Chatbot einen Mehrwert zu bewährten Benutzeroberflächen schaffen kann.

Das klingt erst einmal plausibel, aber was sagt uns das? Bots sind nicht die neuen Apps, sondern eine On-Top-Erweiterung, die im richtigen Kontext eine zusätzliche und komfortablere Möglichkeit bietet, mit einem digitalen Produkt oder Service zu interagieren.

Haben wir bisher via Auge (Display), Händen (Tastatur, Touch) oder Ohren (Sound) interagiert, ist das Sprechen beziehungsweise Schreiben menschlicher Sprache der nächste logische Schritt in dieser Reihe. Die folgenden „Placeonas“ (Idee von [1]: Place + Persona) bieten einen Einblick in das Zusammenspiel von

Kontext und in diesem Fall Spracheingabe (siehe Abbildung 3).

Man stelle sich vor, man wäre in einer Bibliothek und hätte Kopfhörer auf (1). Dort kann man sein Smartphone mit Touch-Gesten bedienen, auf dem Display etwas sehen und durch die Kopfhörer auch auf Sprachausgabe hören. Laut seinem Gerät sollte man an diesem Ort aber eher nicht sprechen (2). Anders ist es, wenn man in einem Fahrzeug sitzt. Dort sollte man die Augen auf den Straßenverkehr richten und auch seine Hände am Lenkrad lassen, statt ein Touch-Gerät zu bedienen. Hören und vor allem das Sprechen von Befehlen ist hier absolut sinnvoll (3). Ähnlich ist es in der Küche, wo man vielleicht vom Teig verklebte Hände hat und daher lieber per Sprachbefehl einen Timer startet oder laut nach dem nächsten Schritt des Rezepts fragt.

Diese einfachen Beispiele lassen bereits erahnen, wohin die Reise geht. Für jeden Kontext gibt es das passende Mittel zur Interaktion. Lange Texte in sein Smartphone eingeben? Nein, danke. Den Text hingegen sprechen und in ein digitales Schriftstück umwandeln lassen, klingt wiederum sehr angenehm. Vorlesen lassen möchte man sich einen langen Text, ein Diagramm oder eine Tabelle nicht. Hier ergibt eine grafische Oberfläche am meisten Sinn, am besten auf einem größeren Screen als auf dem Smartphone. Das Ziel, dem uns Chatbots und vor allem die Spracheingabe einen großen Schritt näherbringen, muss somit sein, ein flexibles und Geräte-übergreifendes Arbeiten zu ermöglichen.

Flexibles, Kontext-unterstützendes Arbeiten in einer digitalen Welt

Die Interaktion mit einem Chatbot oder einem Sprachassistenten kann entweder so ablaufen, dass man einen Befehl absetzt und dieser direkt einen Prozess anstößt, der letzten Endes eine Lösung und eine Antwort bietet. Häufiger wird es aber vorkommen, dass der Befehl Teil eines größeren und Geräte-übergreifenden Ganzen ist. Der folgende fiktive Teil eines Arbeitstags soll diesen Mehrwert verdeutlichen: Ein Geschäftsmann sitzt morgens im Hotel beim Frühstück und bekommt eine E-Mail auf seine Smartwatch.

Da dort eingestellt ist, dass nur Mails mit einer hohen Priorität ankommen sollen, öffnet er sie, scrollt kurz durch den Text und verschafft sich einen Überblick.

Antworten wird er nicht sofort. Erstens wäre das auf der Smartwatch viel zu mühsam, sein Smartphone liegt noch oben im Hotelzimmer, und zweitens möchte er zuerst sein Frühstück genießen. Später fährt er mit dem Auto vom Hotel in sein Büro vor Ort und spricht währenddessen mit dem Chatbot: „Nimm das Excel-Sheet aus der heutigen E-Mail von Tobias und erstelle ein Diagramm nach Template drei.“

Der Bot erstellt das Diagramm anhand eines vorher definierten Templates und legt es in der Cloud für ihn ab. Wenn er dann im Büro ankommt, sind seine Informationen bereits erstellt. Er analysiert die Tabellen aus der E-Mail dort auf einem großen Bildschirm, um den besten Überblick zu haben. Anschließend macht er sich zu Fuß auf den Weg zum Kunden, wo er dieses Wissen und das Diagramm brauchen wird.

Unterwegs schickt er jenes Diagramm noch schnell per Smartphone an einen Kollegen, der später zum Kundentermin hinzustoßen wird und sich so vorher einen Überblick verschaffen kann. Zum Verschicken nutzt er den unternehmensinternen Chat, weil er seinen Kollegen darüber am schnellsten erreicht.

Dieses Beispiel zeigt eindrucksvoll, wie Chatbots eine weitere Lücke schließen, sodass man wirklich flexibel und vor allem passend zum jeweiligen Kontext mit digitalen Produkten und Services agieren kann.

Fazit

Bots sind also nicht die neuen Apps, sondern eine logische Erweiterung der bisherigen Interaktionsmöglichkeiten mit digitalen Produkten und Services. Wo anfangs die Interaktion nur per Kommandozeile stattfand, folgte später die Maus. Sie ersetzte die reinen Textbefehle allerdings nicht (komplett). Ebenso wie heute Sprachassistenten hatte es die Maus bei ihrer Einführung nicht leicht und wurde von vielen nur als nettes Gimmick gesehen. So schrieb die PC Week im April 1984: „Mice are nice ideas, but of dubious value for business users.“ Wie das heute aussieht, wissen wir alle.

Ähnlich erging es dem Smartphone, als es mit seinen Touch-Gesten Einzug hielt. Apps, die ein digitales Feuerzeug simulierten – indem sich die Flamme passend zur Smartphone-Neigung bewegte – wirkten zunächst ja auch wirklich nur wie eine Spielerei. Doch zeigten sie schon eindrucksvoll, wohin die Reise gehen sollte. Heute nutzen wir Smartphones massiv im Alltag wie auch im Businesskontext. Verdrängt haben sie Laptops, Maus und Tastatur aber dennoch nicht, auch wenn wir diese mittlerweile deutlich weniger nutzen. Ebenso wird es mit Chatbots und Spracheingabe kommen. Sie werden sowohl Apps als auch grafische Oberflächen nicht ersetzen, sondern diesen Mix aus Interaktionsmöglichkeiten erweitern.

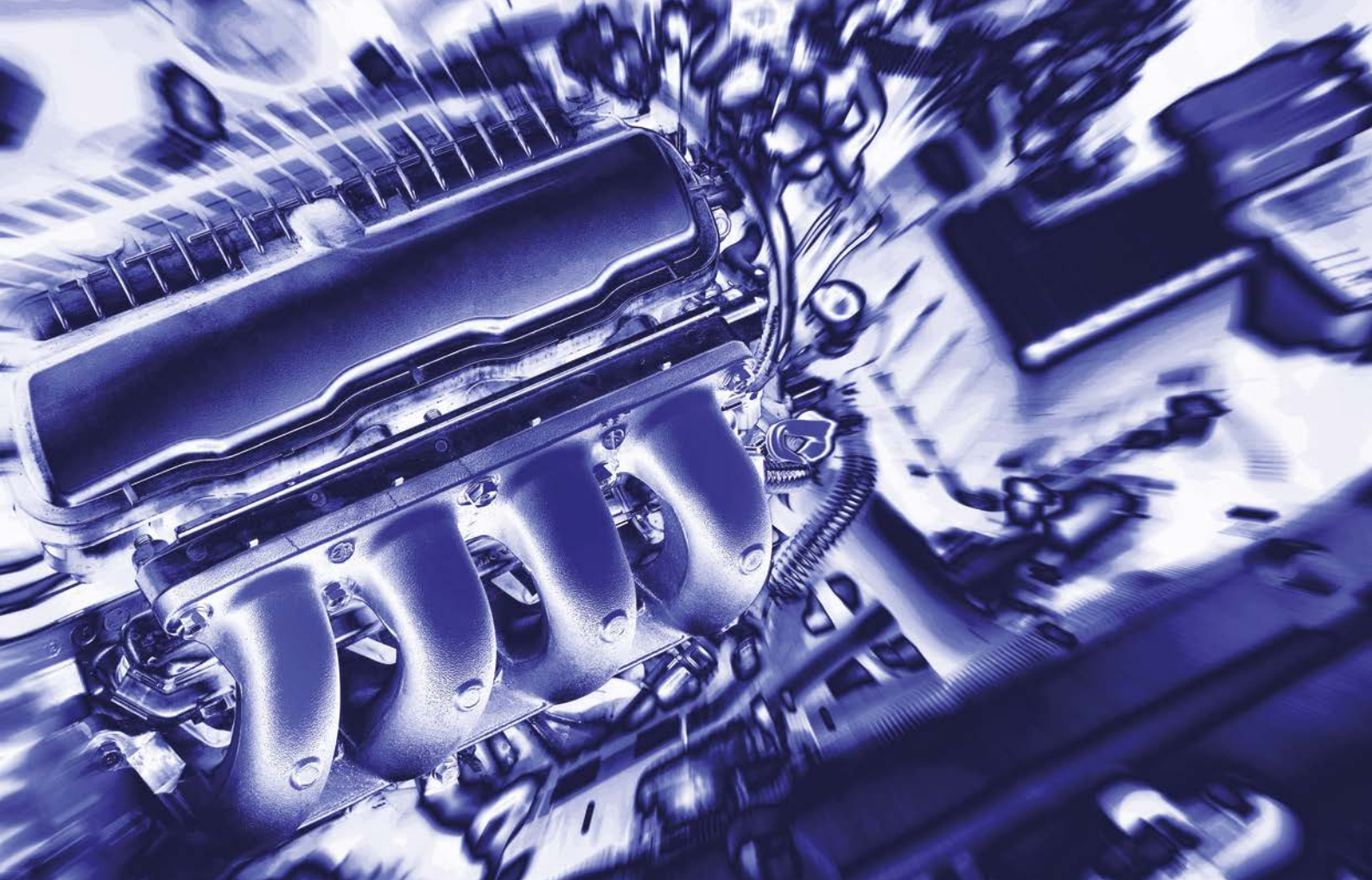
Der Autor hat große Hoffnung, dass Chatbots und Spracheingabe uns einen Schritt näher an eine digitale Welt bringen, in der das, was wir nicht gerne tun, automatisiert und das, was wir gerne tun, digital angereichert („augmented“) wird [2]. Dieser Satz verdeutlicht genau das, was Digitalisierung für ihn bedeutet und weshalb er auf die weitere digitale Zukunft mit freudiger Erwartung blickt.

Weitere Informationen

- [1] D. Traynor, What voice UI is good for: <https://blog.intercom.com/benefits-of-voice-ui/>
- [2] U. O'Broin. Chatbots in the Enterprise – Not Just All Talk: <https://chatbotmagazine.com/chatbots-in-the-enterprise-not-just-talk-de385ec6f2>



Christian Ochsenkühn
christian.ochsenkuehn@opitz-consulting.com



SQL-Performance-Tuning mit „Analytic Functions“

Christian Schwitalla, Apps Associates GmbH

Die momentan auf Hochtouren laufende Marketing-Maschinerie könnte man vielleicht dahingehend missverstehen, dass Software, die in der Cloud läuft, immer performant ist. Ergo: Warum sollte man sich überhaupt noch Gedanken zu dem Thema Performance machen?

Performance-Probleme, die auf einer schlechten Architektur oder suboptimalen Implementierung beruhen, werden nicht dauerhaft durch die Cloud geheilt. Es ist also nach wie vor sinnvoll, die Performance einer Anwendung schon während der Entwicklung im Auge zu behalten. Dabei sollte man nicht erst dann einschreiten, wenn es offensichtliche Probleme mit der Performance einer SQL-Abfrage gibt. Scheinbar zufriedenstellend laufende Abfragen können sich schon

morgen als echte Probleme erweisen. Mögliche Gründe dafür könnten in folgenden Bereichen liegen:

- Verbrauch von System-Ressourcen
- Parallelisierbarkeit
- Skalierbarkeit bei wachsenden Datenmengen

Daher ist es ratsam, Abfragen von Anfang an performant zu gestalten: „Do not tune – write fast SQL“.

SQL Analytic Functions

Analytical Functions bilden eine Familie von SQL-Funktionen. Sie sind auch als „Fenster-“ oder „Windowing“-Funktionen bekannt, weil sie eine Frame-Sicht auf die Daten bieten. Sie arbeiten auf der Ergebnismenge einer Abfrage und verfügen dabei über besondere Eigenschaften. Die Oracle-Dokumentation sagt dazu: „Analytic functions compute an aggregate value based on a group of rows. They dif-

fer from aggregate functions in that they return multiple rows for each group. The group of rows is called a window and is defined by the analytic_clause. For each row, a sliding window of rows is defined. The window determines the range of rows used to perform the calculations for the current row. Window sizes can be based on either a physical number of rows or a logical interval such as time.“, und listet 46 Funktionen auf (siehe „<http://docs.oracle.com/database/121/SQLRF/functions004.htm#SQLRF06174>“).

Analytic Functions sind nach wie vor wenig bekannt – eigentlich schade. Das Einsatzgebiet wird eher in den Bereichen „Business Intelligence“ oder „Data Warehouse“ vermutet, weniger in Performance Tuning normaler SQL-Abfragen. Dabei sind sie seit der Datenbank-Version 8i (8.1.6, Januar 2000) in allen Datenbank-Editionen verfügbar. Darüber hinaus wurden sie längst als ANSI/ISO SQL 2003 und SQL 2008 („Windowing Functions“) in den SQL-Standard aufgenommen. Auch relationale Datenbanken anderer Hersteller (DB2, Oracle, Sybase, PostgreSQL und SQL Server) bieten sie inzwischen an.

Besondere Eigenschaften

Viele Analytical Functions kennt man auch als Aggregationsfunktionen, etwa die Funktion „MAX()“ = „returns maximum value of expr. You can use it as an aggregate or analytic function“. Listing 1 zeigt die Aggregationsvariante von MAX().

Es wird der Höchstwert von „salary“ ermittelt. Da keine „WHERE“-Klausel vorkommt, ist das der Höchstwert in der gesamten Tabelle „employees“. Listing 2 zeigt die analytische Variante von MAX().

Es wird pro Window, in diesem Fall pro „manager_id“, der Höchstwert von „salary“ ermittelt. Hier ist bereits eine interessante Eigenschaft der Analytic Functions sichtbar: Sie bieten Zugriff auf Werte aus anderen Zeilen der Ergebnismenge.

Ein weiteres Beispiel einer Analytic Function ist „LAG()“: „LAG is an analytic function. It provides access to more than one row of a table at the same time without a self join. Given a series of rows returned from a query and a position of the cursor, LAG provides access to a row at a given physical offset prior to that position“ (siehe Listing 3).

```
SELECT MAX(salary) "Maximum" FROM employees;
Maximum
-----
24000
```

Listing 1

```
SELECT manager_id, last_name, salary,
MAX(salary) OVER (PARTITION BY manager_id) AS mgr_max
FROM employees
ORDER BY manager_id, last_name, salary;
```

MANAGER_ID	LAST_NAME	SALARY	MGR_MAX
100	Zlotkey	10500	17000
101	Baer	10000	12000
101	Whalen	4400	12000
102	Hunold	9000	9000
103	Austin	4800	6000

Listing 2

```
SELECT hire_date, last_name, salary,
LAG(salary, 1, 0) OVER (ORDER BY hire_date) AS prev_sal
FROM employees
WHERE job_id = 'PU_CLERK'
ORDER BY hire_date;
```

HIRE_DATE	LAST_NAME	SALARY	PREV_SAL
18-MAY-03	Khoo	3100	0
24-JUL-05	Tobias	2800	3100
24-DEC-05	Baida	2900	2800
15-NOV-06	Himuro	2600	2900
10-AUG-07	Colmenares	2500	2600

Listing 3

Hier wird „salary“ des Vorgängers (im Sinne der Sortierung des Analytic Window) ermittelt. Wieder verblüffen die Analytic Functions, denn sie schaffen dies ohne ein mehrfaches Lesen der Daten, beispielsweise durch Subselects.

Eine sehr nützliche Analytic Function ist „RANK“: „RANK calculates the rank of a value in a group of values. As an aggregate function, RANK calculates the rank of a hypothetical row identified by the arguments of the function with respect to a given sort specification. As an analytic function, RANK computes the rank of each row returned from a query with respect to the other rows returned by the query, based on the values of the value_exprs in the order_by_clause. „RANK()“ kann also auch als Aggregationsfunktion eingesetzt werden (siehe Listing 4).

Hier wird ermittelt, welchen Rang der Wert von 15500 als „salary“ bei absteigender Sortierung annehmen würde. Listing 5 zeigt die analytische Variante von RANK(). Es wird für alle Mitarbeiter mit „department_id = 60“ anhand der Spalte „salary“ ein Rang gebildet. Mit „RANK()“ geht es kinderleicht.

Einsatzbeispiel aus einem Projekt

In einem Projekt stand der Autor vor der Aufgabe, eine Reihe von SQL-Statements zu tunen. Es handelte sich um sogenannte „Langläufer“, die das nächtliche Batch-Fenster immer häufiger gesprengt haben. Die Aufgabe der Statements bestand darin, Daten aus Quell-Tabellen in Materialized Views („Snapshots“) zu laden. Die

Statements waren als Views gekapselt und nach einem gemeinsamen Schema aufgebaut. Ihre Aufgabe bestand darin, die korrekte Version eines Objektes aus der Quell-Tabelle herauszufinden. Um das Prinzip zu veranschaulichen, betrachten wir folgendes fiktives Beispiel:

- Lesen von Adressen
- Pro Kunde nur eine Adresse
- Auswahl über TYP der Adresse:
 - Bevorzugte Adresse mit „TYP = FIRMA“
 - Wenn nicht vorhanden, dann Adresse mit „TYP = POSTFACH“
 - Wenn nicht vorhanden, dann Adresse mit „TYP = PRIVAT“
- Tabelle „ADRESSEN“ mit folgenden Elementen:
 - ID (PK)
 - KUNDEN_ID (UK)
 - TYP [FIRMA | POSTFACH | PRIVAT] (UK)
 - STRASSE
 - HAUSNUMMER
 - POSTFACH
 - PLZ
 - ORT

Listing 6 zeigt die Beispieldaten. Das bisherige SQL-Statement ist in Listing 7 dargestellt. Listing 8 zeigt den Execution-Plan des bisherigen SQL-Statements.

Die Laufzeiten des bisherigen SQL-Statements in Sekunden (Performance-Tests mit Oracle Database 12c Enterprise Edition Release 12.1.0.2.0, 64bit unter VirtualBox auf einem Office-Laptop) betragen:

- 100 Rows: 0,02
- 10 000 Rows: 0,21
- 1 000 000 Rows: 12,58
- 100 000 000 Rows: 1190,00

Jeder Satz in der Tabelle „ADRESSEN“ muss daraufhin überprüft werden, ob er in der richtigen Version vorliegt oder ob gegebenenfalls zu dem gleichen Kunden eine andere Adresse vorliegt, die Vorrang vor dem aktuellen Satz hat. Daher die beiden „NOT EXISTS“-Klauseln. Durch die Index-Scans skaliert die Antwortzeit recht gut, jedoch waren die Antwortzeiten immer länger geworden, da die Anzahl der zu betrachtenden Sätze stetig gewachsen ist. Der Einsatz der Analytic Function RANK() hat das Problem gelöst (siehe

```
SELECT RANK(15500) WITHIN GROUP (ORDER BY salary DESC) AS "Rank of 15500"
FROM employees;

Rank of 15500
-----
4
```

Listing 4

```
SELECT department_id, last_name, salary,
RANK() OVER (PARTITION BY department_id ORDER BY salary) AS RANK
FROM employees
WHERE department_id = 60
ORDER BY RANK, last_name;
```

DEPARTMENT_ID	LAST_NAME	SALARY	RANK
60	Lorentz	4200	1
60	Austin	4800	2
60	Pataballa	4800	2
60	Ernst	6000	4
60	Hunold	9000	5

Listing 5

KUNDEN_ID	TYP
90810846361	PRIVAT
90810846361	POSTFACH
217537839264	PRIVAT
481491166609	PRIVAT
481491166609	POSTFACH
481491166609	FIRMA

Listing 6

```
SELECT A.KUNDEN_ID, A.TYP,
A.STRASSE, A.HAUSNUMMER, A.POSTFACH, A.PLZ, A.ORT
FROM ADRESSEN A
WHERE
( ( A.TYP = 'FIRMA' ) OR
( A.TYP = 'POSTFACH'
AND NOT EXISTS ( SELECT 1 FROM ADRESSEN B
WHERE B.KUNDEN_ID = A.KUNDEN_ID
AND B.TYP = 'FIRMA' ) ) OR
( A.TYP = 'PRIVAT'
AND NOT EXISTS ( SELECT 1 FROM ADRESSEN C
WHERE C.KUNDEN_ID = A.KUNDEN_ID
AND C.TYP IN ( 'FIRMA' , 'POSTFACH' ) ) ) )
```

Listing 7

Vorgang	Optionen	Objekt
SELECT STATEMENT		
FILTER		
TABLE ACCESS	STORAGE FULL	ADRESSEN
INDEX	UNIQUE SCAN	ADRESSEN_UK
INDEX	RANGE SCAN	ADRESSEN_UK

Listing 8

```

SELECT KUNDEN_ID, TYP, STRASSE, HAUSNUMMER, POSTFACH, PLZ, ORT
FROM
(
  SELECT KUNDEN_ID, TYP,
    STRASSE, HAUSNUMMER, POSTFACH, PLZ, ORT,
    RANK() OVER (PARTITION BY KUNDEN_ID
      ORDER BY CASE TYP WHEN 'FIRMA' THEN 1
        WHEN 'POSTFACH' THEN 2
        WHEN 'PRIVAT' THEN 3 END )
    AS RANK
  FROM ADRESSEN
)
WHERE RANK = 1
    
```

Listing 9

Vorgang	Optionen	Objekt
SELECT STATEMENT		
VIEW		
WINDOW	SORT PUSHED RANK	
TABLE ACCESS	STORAGE FULL	ADRESSEN

Listing 10

	bisher	neu	Differenz
• 100 Rows:	0,02	0,02	0%
• 10 000 Rows:	0,21	0,09	-50%
• 1 000 000 Rows:	12,58	6,23	-50%
• 100 000 000 Rows:	1190,00	630,00	-50%

Listing 11

	bisher	neu
• consistent gets:	9.366	193
• logical read bytes from cache:	76.726.272	1.581.056
• CPU used by this session:	1	2

Listing 12

Listing 9). Listing 10 zeigt den Execution-Plan des neuen SQL-Statements.

Jetzt werden die Daten lediglich in einem Full-Table-Scan gelesen; auf die Daten der Tabelle „ADRESSEN“ ist kein weiterer Zugriff notwendig. Der Eintrag „WINDOW SORT PUSHED RANK“ zeigt, dass die Analytic Function ausgeführt wird. Dies geschieht In-Memory – abhängig von der Datenmenge gegebenenfalls auch im TEMP-Tablespace. Das spiegelt sich in den Ausführungszeiten wider (in Sekunden, siehe Listing 11). Listing 12 zeigt einige Statistiken dazu.

Die Ausführungszeit ist signifikant gesunken, im Beispiel um den Faktor 2. Im

Projekt wurden sogar Verbesserungen um den Faktor 8 beobachtet. Damit war das Performance-Problem der zu lange laufenden Batch-Jobs erfolgreich gelöst. Die Eingriffe in das bestehende System waren auf die SQL-Statements begrenzt.

Die Tests der neuen Statements waren recht einfach: altes „Select“ minus neues „Select“. In beiden Fällen ist die leere Menge als Ergebnis zu erwarten. Weitere Vorteile des neuen SQL-Statements sind:

- Es ist einfacher aufgebaut, dadurch verständlicher und wartbarer – zwei Subselects wurden eliminiert

- Es ist in der Ausführung weniger ressourcenintensiv – nur ein Full-Table-Scan findet jetzt statt
- Es wird mit wachsender Datenmenge besser zurecht kommen

Fazit

Analytic Functions bieten dank besonderer Eigenschaften Power (fast) zum Nulltarif. Sie sind ein Geheimfavorit für so manches Performance-Problem. Es gibt kaum Gründe, die gegen ihren Einsatz sprechen. Sie sind ein Beispiel für die stetige Evolution von SQL. Ähnliche Ansätze kann man auch in der Literatur oder im Internet finden:

- Artikel „2000 Zeilen Java oder 50 Zeilen SQL“ von Lukas Eder in der Ausgabe 01/2016 von Java aktuell
- http://www.dba-oracle.com/t_rewrite_sql_exists_subqueries_rank_partiton_analytics.htm

Analytic Functions sind nicht nur für Statistiker, sondern für alle SQL-Entwickler interessant und aus Sicht des Autors das am meisten unterschätzte SQL-Feature. „Es gibt SQL vor Fenster-Funktionen und SQL nach Fenster-Funktionen“, so Dimitri Fontaine (PostgreSQL, siehe „<http://tapoueh.org/blog/2013/08/20-Window-Functions>“).



Christian Schwitalla
christian.schwitalla@appsassociates.com



WebLogic WLST-Programmierung unter Eclipse mit dem Oracle-OEPE-Plugin und PyDev

Michael Schulze, OPITZ CONSULTING Deutschland GmbH

Dieser Artikel setzt die WLST-Entwicklung mit der Eclipse-Umgebung fort, deren Erstellung im ersten Teil dieser Artikelserie beschrieben ist [1]. Anhand von einfachen Beispielen wird dargestellt, wie die IDE dabei mit dem Oracle WebLogic Server interagiert und welche Vorteile es für Entwickler haben kann, sich mit der WLST-Programmierung unter Eclipse (OEPE) zu beschäftigen. Die Screenshots geben einen Eindruck von den in der Umgebung enthaltenen Funktionalitäten. Zudem wird die Konfiguration der Git-Funktionen (Versionsverwaltung) erläutert, die Application-Server-Sicht beschrieben und kurz auf im WLST-Plugin enthaltene Features (Syntax Highlighting, WLST-Help etc.) eingegangen.

Abbildung 1 gibt einen Überblick über das weitere Test-Szenario und zeigt schematisch die einzelnen Komponenten, die teilweise noch erstellt werden müssen. Auch die Verbindungen der Komponenten untereinander im Entwicklungskontext mit der IDE sind hier dargestellt.

Die folgenden WLST-Online-Test-Szenarien teilen sich thematisch in folgende Kategorien auf:

- Extending Domain: Erweitern einer Domain um weitere WebLogic-Komponenten
- Monitoring
 - Ermitteln der Heap-Memory-Auslastung von WLS-Servern einer Domain
 - Status-Ermittlung WLS-Server
- Node Manager: Connect-Beispiel und Status-Ermittlung WLS-Server
- Configuration: Auslesen von JDBC-Datasources als Konfigurationsbeispiel

Beispiel 3: Erweitern der erstellten WebLogic-Domain um weitere Komponenten wie Server, Cluster und Node Manager (online, lokal)

Im ersten Beispiel des ersten Artikels ging es um das Erstellen einer lokalen WebLogic-Domain im WLST-Offline-Modus. Analog dazu wollen wir nun ein nächstes Skript erstellen, mit dessen

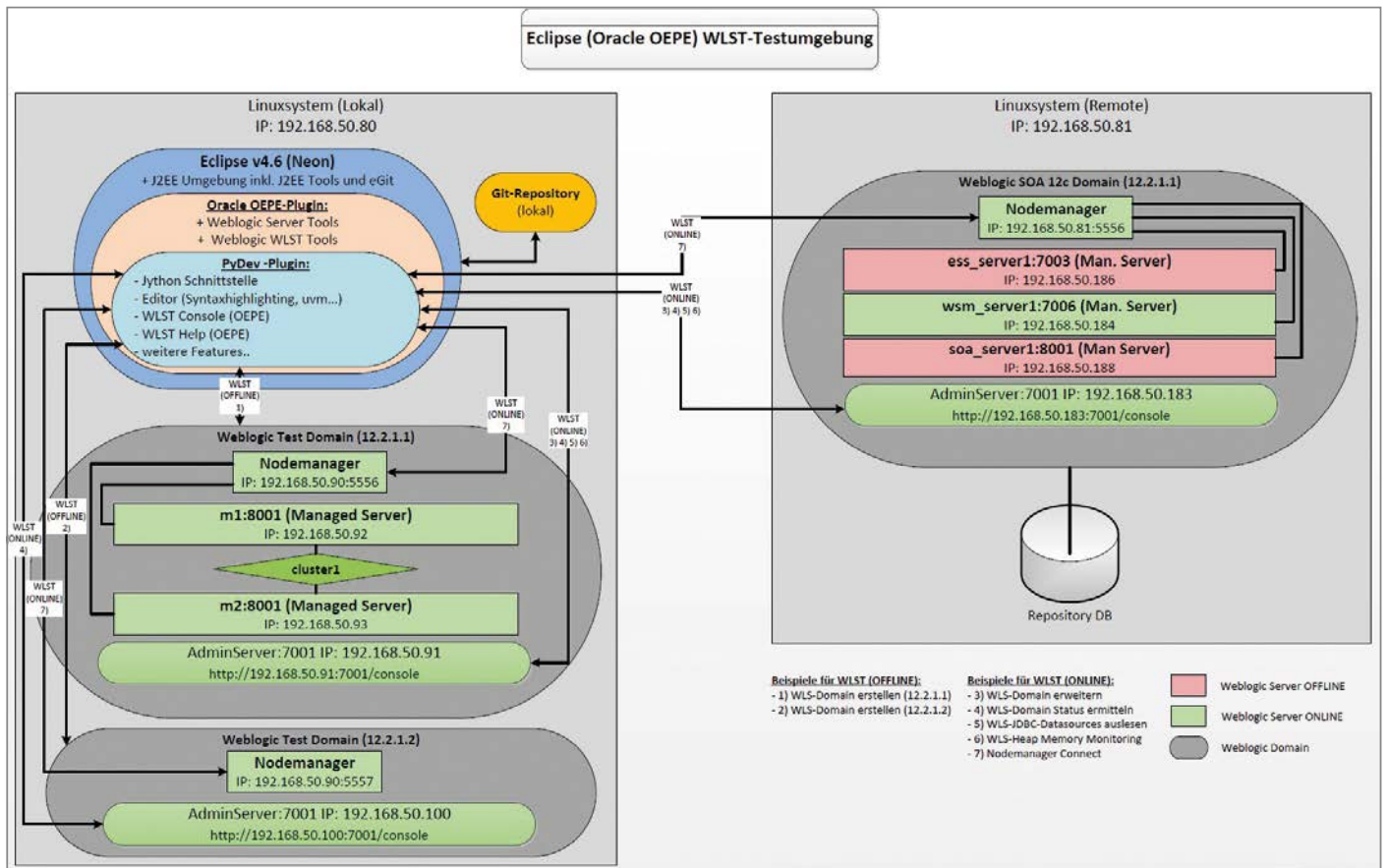


Abbildung 1: Überblick Testszenario (eigene Darstellung)

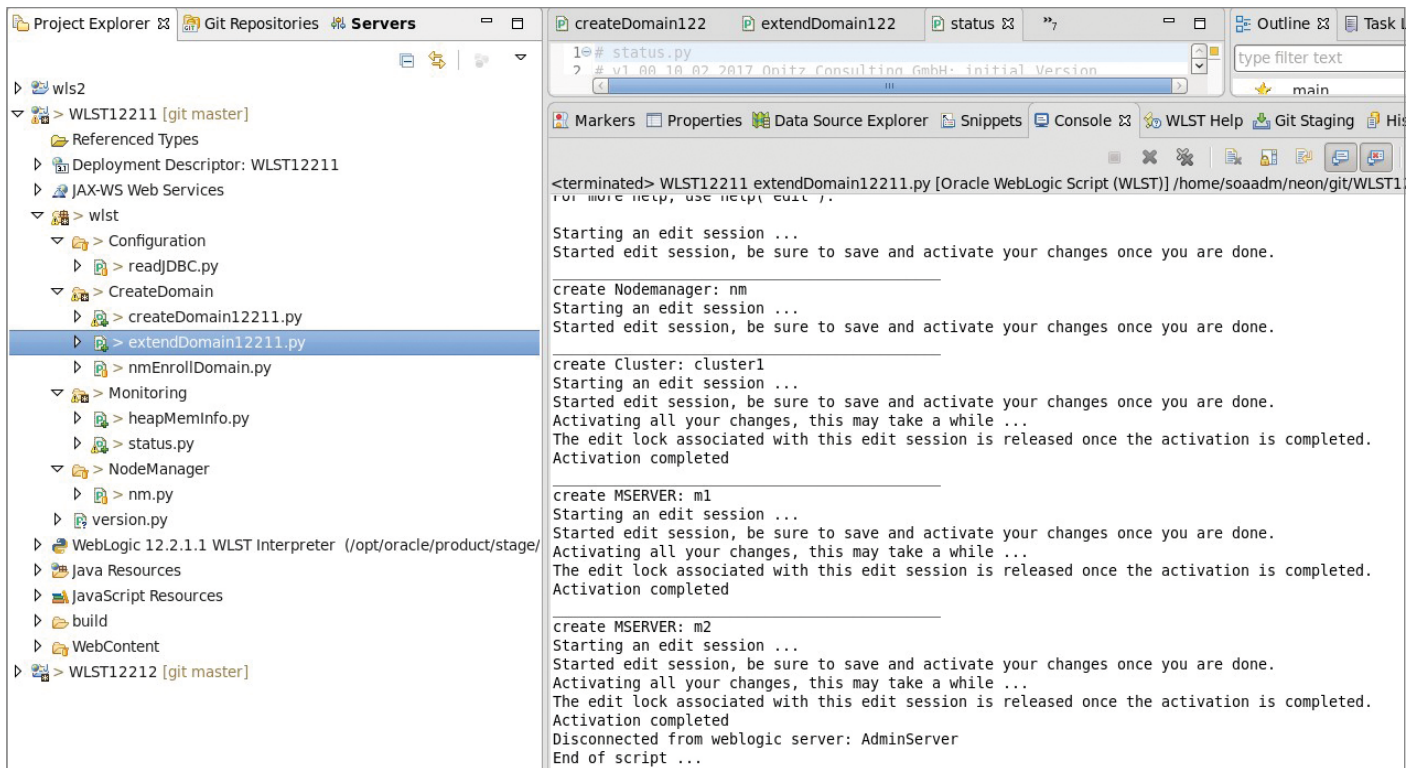


Abbildung 2: Domain-Erweiterung

Hilfe die Domain um verschiedene Komponenten erweitert werden soll (siehe Abbildung 1). Die erforderlichen Schritte sind:

- create Node Manager: nm
- create Cluster: cluster1
- create MSERVER: m1, Zuordnung Cluster und Node Manager

- create MSERVER: m2, Zuordnung Cluster und Node Manager

Das Skript dazu (siehe Listing 1) befindet sich

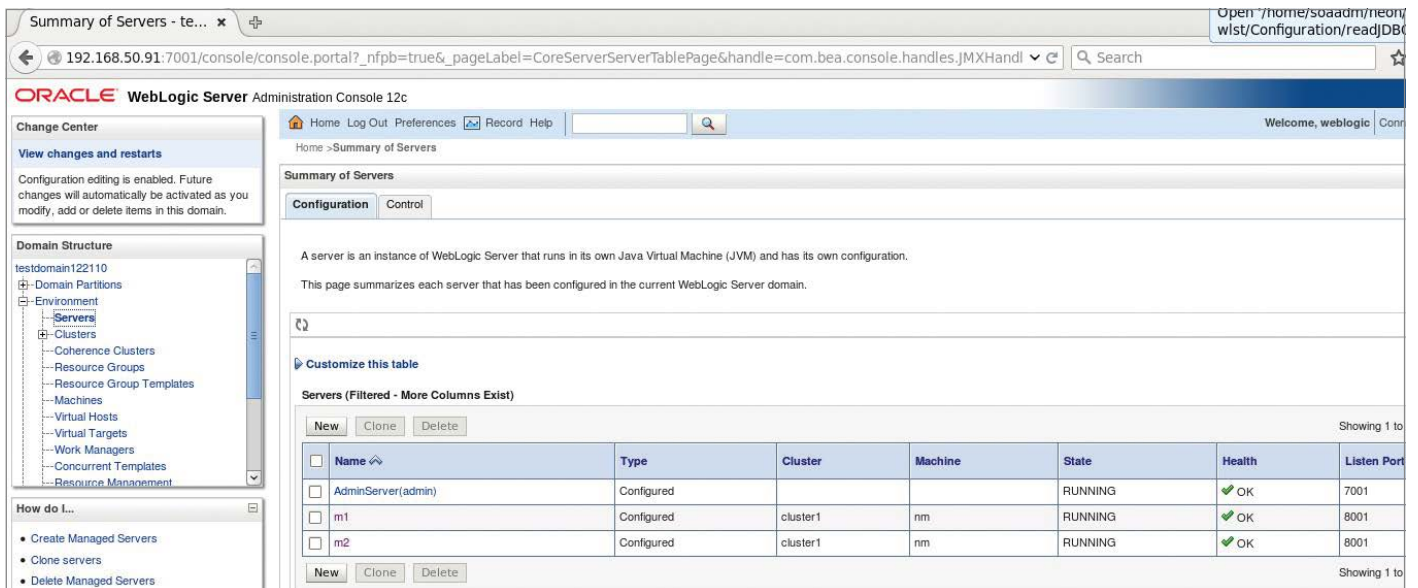


Abbildung 3 zeigt das Ergebnis nach erfolgreichem Startup.

```

if __name__ == '__main__':
    from wlstModule import *#@UnusedWildImport
    print 'starting the script ....'
    username = 'weblogic'
    password = 'welcome1'
    url='t3://192.168.50.91:7001'

    connect(username,password,url)
    domainRuntime()
    cd('ServerRuntimes')
    servers=domainRuntimeService.getServerRuntimes()
    for server in servers:
        print server.getName()+': '+server.getState()
    disconnect()
    exit()

```

Listing 2

aufgrund der Größe unter folgender URL: <http://www.doag.org/go/redstack201706listing1>. Im Project Explorer wird dieses Skript nun mit „rechte Maus“/„Run As“ /„WLST Run“ ausgeführt.

Die Domain ist nun erweitert, es müssen noch folgende manuelle Anpassungen und Aktionen getätigt werden, bevor die Domain vollständig zur Verfügung steht (siehe Abbildung 2):

- Enrolling der erstellten Domain zu Node Manager

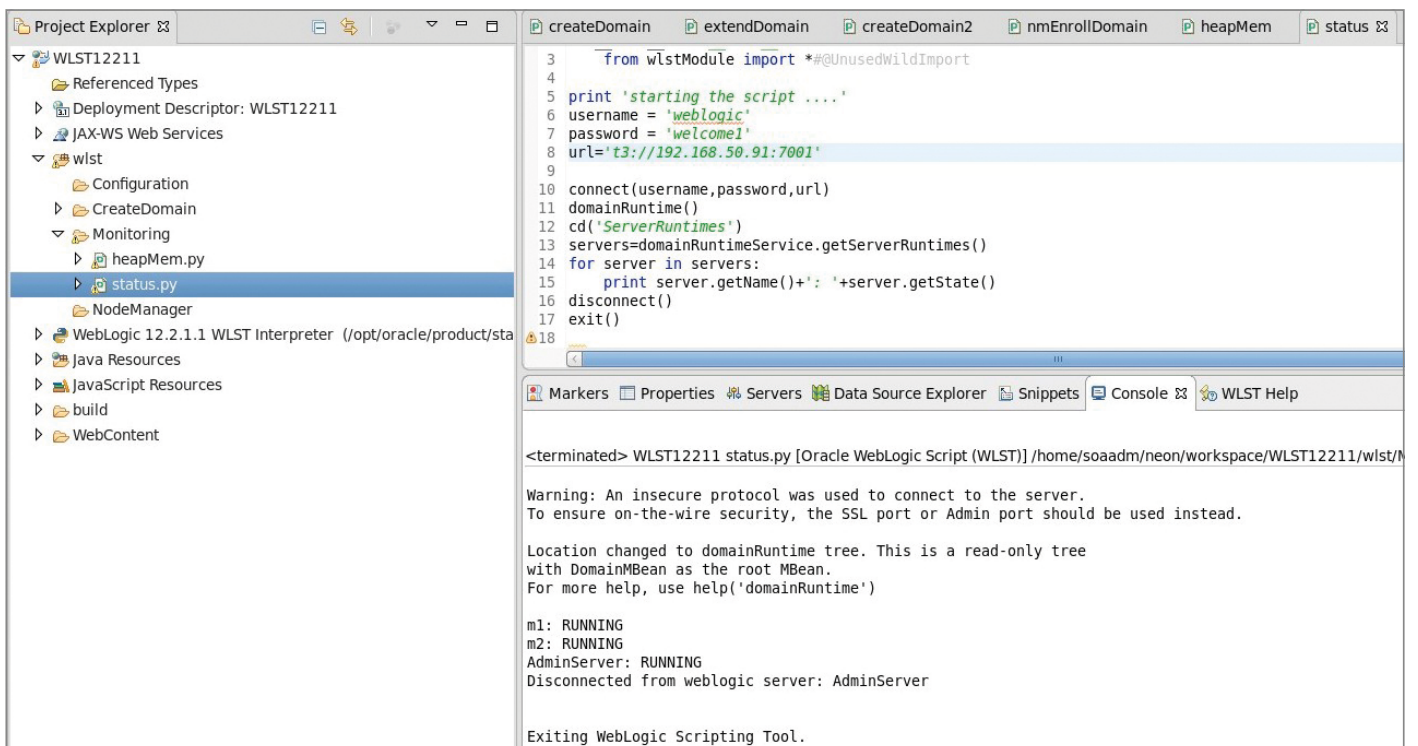


Abbildung 4: Domain-Status checken über WLST

- Node Manager auf „IP:Port“ anpassen und starten
- Startup der Managed Server „m1 + m2“ über AdminConsole

Beispiel 4: Status prüfen und Heap Memory (WLST lokal)

In diesem Test soll es darum gehen, ein

WLST-Skript für verschiedene Monitoring-Aktionen zu erstellen. Zunächst wird der Server-Status ermittelt, dazu soll auch der Java-Heap-Memory analysiert werden,

```

if __name__ == '__main__':
    from wlstModule import *#@UnusedWildImport

print 'starting the script ....'
username = 'weblogic'
password = 'welcome1'
# local Server
url='t3://192.168.50.91:7001'
# remote Server
#url='t3://192.168.50.183:7001'

connect(username,password,url)
domainRuntime()
servers = domainRuntimeService.getServerRuntimes();
print('=====')
print('# Java Heap Memory Ueberblick:')
print('=====')
print('%20s %10s %8s %8s %4s' % ('Server','Heap-Aktuell','Heap-Frei','Heap-Maximal','Heap-Prozent'))
for server in servers:
    free = int(server.getJVMRuntime().getHeapFreeCurrent())/(1024*1024)
    freePct = int(server.getJVMRuntime().getHeapFreePercent())
    current = int(server.getJVMRuntime().getHeapSizeCurrent())/(1024*1024)
    max = int(server.getJVMRuntime().getHeapSizeMax())/(1024*1024)
    print('%20s %7d MB %5d MB %5d MB %3d%%' % (server.getName(),current,free,max,freePct))
disconnect()
exit()
    
```

Listing 3

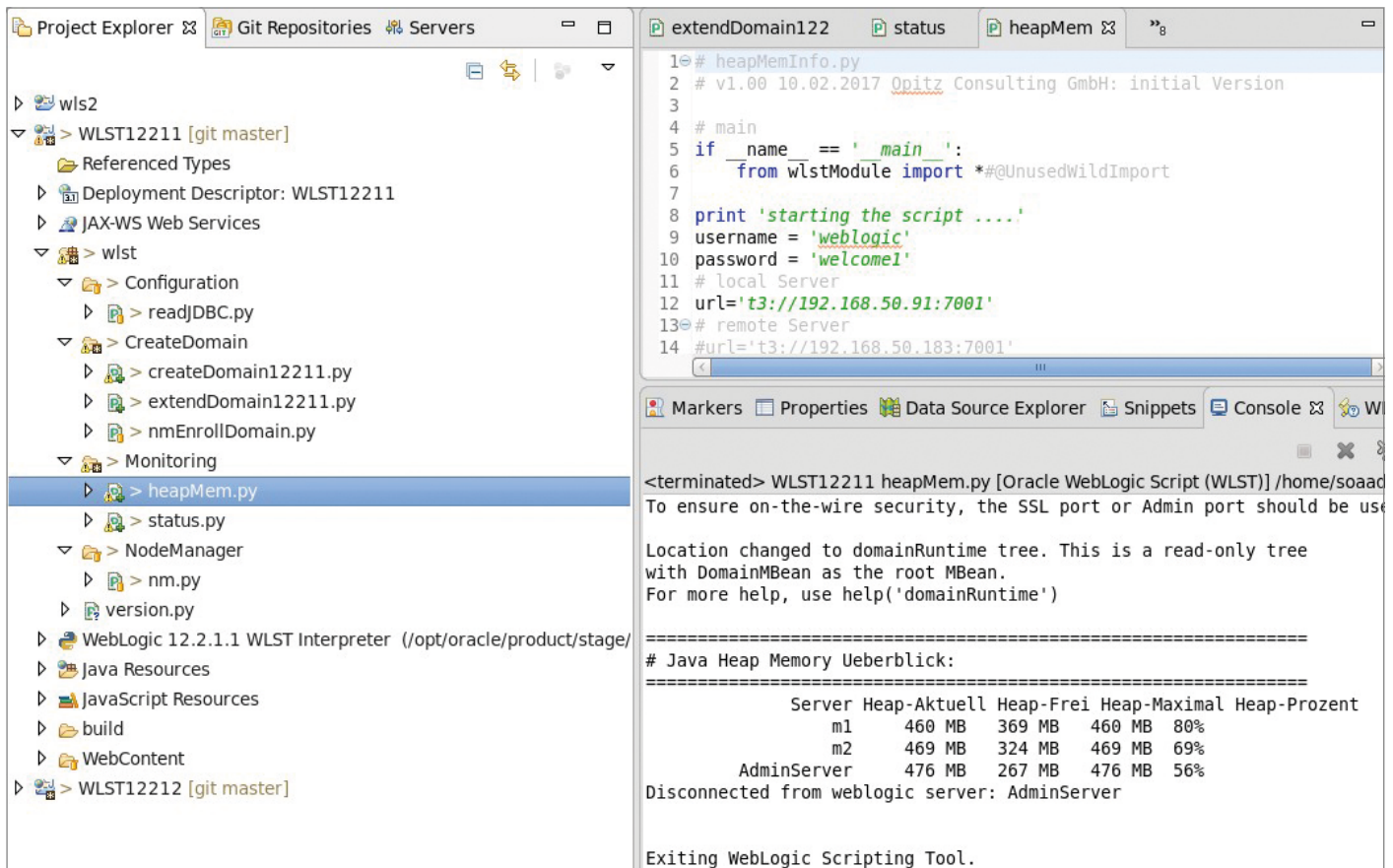


Abbildung 5: Java-Heap-Auslastung unserer Domain

insbesondere die Auslastung. Auch hier muss analog zum ersten Beispiel [2] ein WLST-Skript erstellt werden (siehe Listing 2). Im Project Explorer kann das neue Skript

nun wieder mit „rechte Maus“/„Run As“/„WLST Run“ ausgeführt werden. Abbildung 4 zeigt die Ausgabe. Jetzt folgt eine skriptgesteuerte Heap-Analyse. Dazu er-

stellen wir ein neues WLST-Skript (siehe Listing 3).

Im Project Explorer kann das neue Skript nun wieder mit „rechte Maus“/„Run

```

if __name__ == '__main__':
    from wlstModule import *#@UnusedWildImport
    redirect('/dev/null', 'false')

print 'starting the script ....'
username = 'weblogic'
password = 'welcome1'

print '_____ '
print 'Example NM-Connect Local: Test-Domain:'
nmConnect(username, password, '192.168.50.90', '5556', 'testdomain122110', '/opt/oracle/config-local/stage/do-
mains/testdomain122110', 'plain')
print 'm1='+nmServerStatus('m1')
print 'm2='+nmServerStatus('m2')
nmDisconnect()

print '_____ '
print 'Example NM-Connect Remote: SOA-Domain:'
nmConnect(username, password, '192.168.50.81', '5559', 'soa_stage', '/opt/oracle/config-local/stage/domains/
soa_stage', 'plain')
print 'soa_server1='+nmServerStatus('soa_server1')
print 'ess_server1='+nmServerStatus('ess_server1')
print 'wsm_server1='+nmServerStatus('wsm_server1')
nmDisconnect()

```

Listing 4

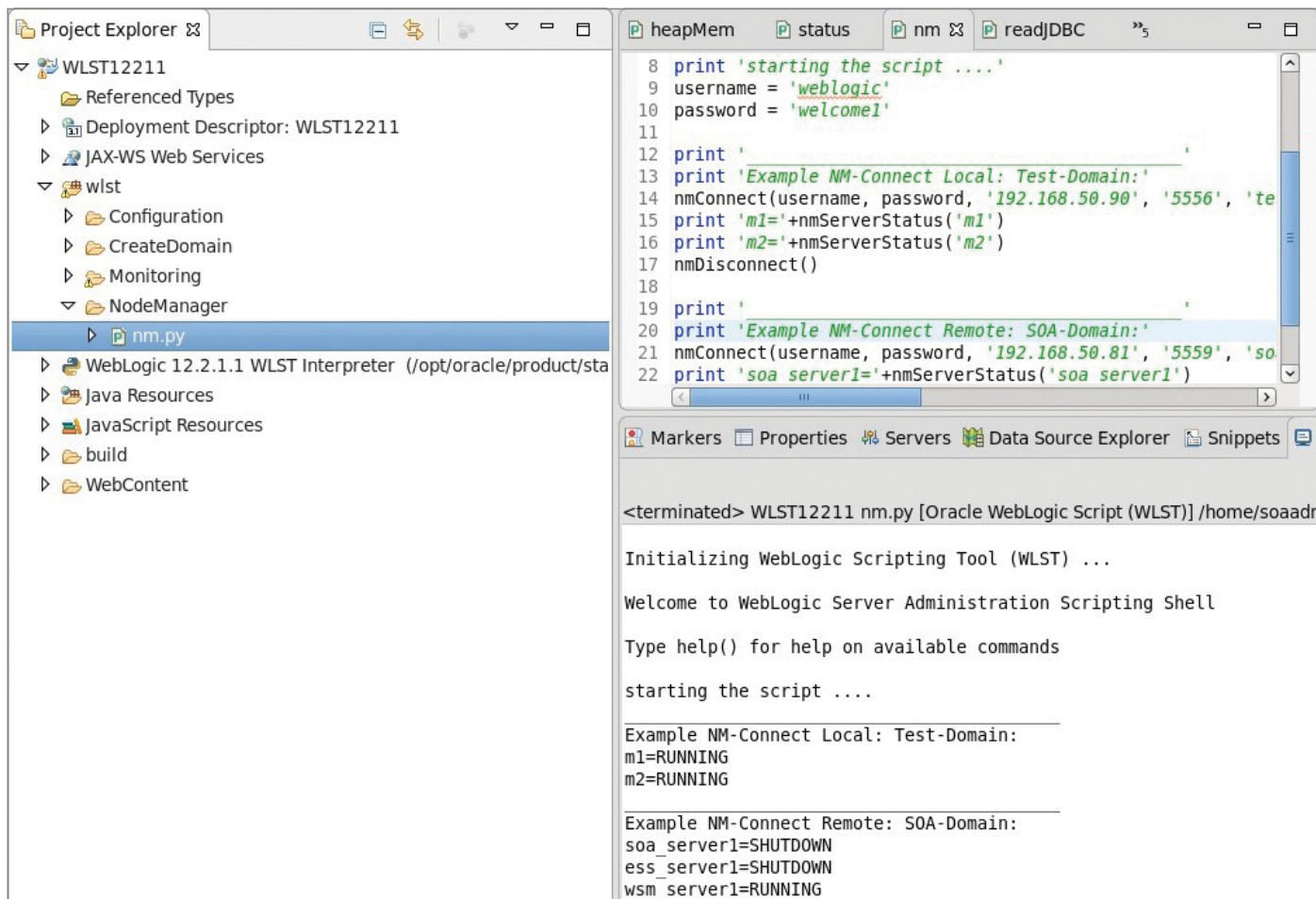


Abbildung 6: Node Manager: Status der Managed Server prüfen (lokale Domain, SOA 12c Testdomain)


```
if __name__ == '__main__':
    from wlstModule import *#@UnusedWildImport

print 'starting the script ....'
username = 'weblogic'
password = 'welcome1'
url='t3://192.168.50.183:7001'
#url='t3://192.168.50.91:7001'

connect(username,password,url)
print '_____ '
print 'Liste der JDBC-Datasourcen:'+url
allJDBCResources = cmo.getJDBCSystemResources()
for jdbcResource in allJDBCResources:
    dsname = jdbcResource.getName()
    print dsname, jdbcResource.getJDBCResource().getJDBCDataSourceParams().getJNDINames()[0], jdbcResource.getJDBCResource().getJDBCDriverParams().getUrl()
```

Listing 5

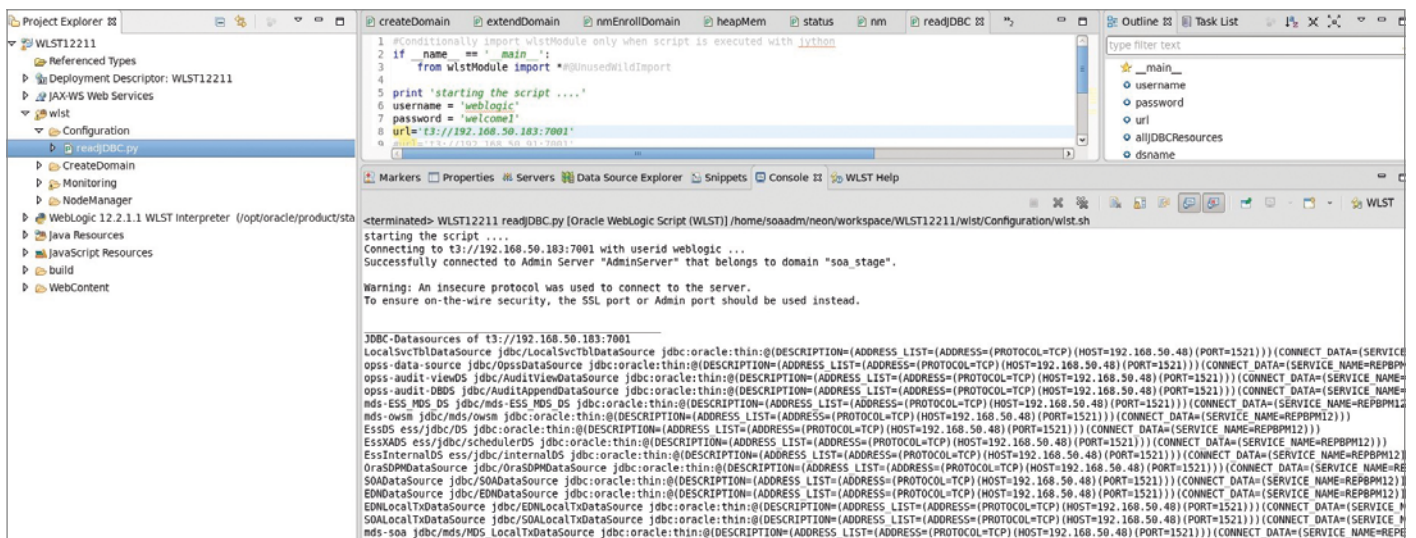


Abbildung 7: Auslesen der JDBC-Data-Sources

As/„WLST Run“ ausgeführt werden. Abbildung 5 zeigt die Ausgabe.

Beispiel 5: Serverstatus der Managed Server ermitteln

Analog zu Beispiel 3 erfolgt nun über die Node-Manager-Verbindung eine Statusermittlung der Managed Server, sowohl auf der erstellten lokalen Domain als auch auf einer weiteren aktiven Testdomain auf SOA 12c. Dazu ist ein neues WLST-Skript mit folgendem Inhalt zu erstellen:

Im Project Explorer kann das neue Skript nun wieder ausgeführt werden: „rechte Maus“/„Run As“/„WLST Run“. Abbildung 6 zeigt die Ausgabe.

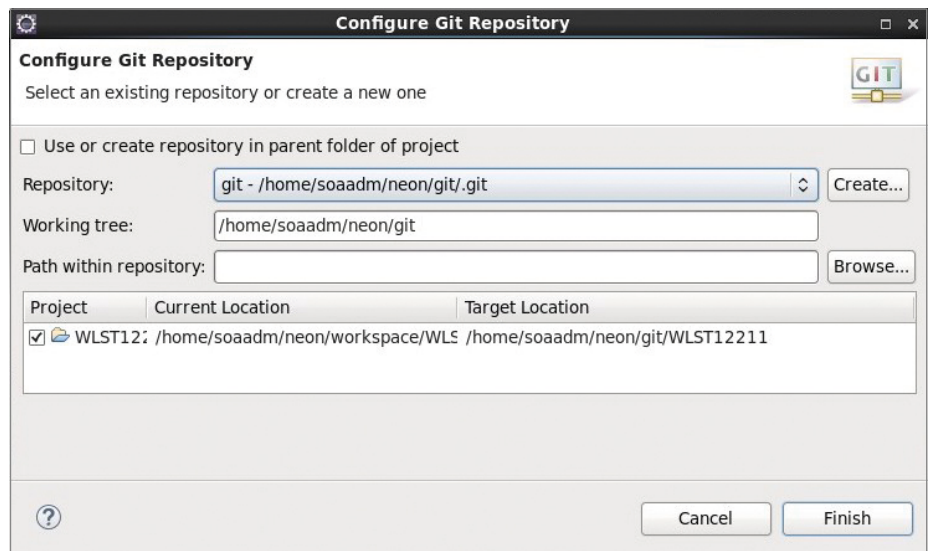


Abbildung 8: Projekt in Git einbinden: Wahl des Repository

WLST-Beispiel 6: Auslesen der JDBC-Data-Sources im Remote-System zu Konfigurations- und Dokumentationszwecken

Da in der vorhandenen SOA-12c-Testdomain mehrere Data-Sources enthalten sind, wurde dieses System für den Test gewählt. Zunächst ist wieder ein WLST-Skript zu erstellen (siehe Listing 5). Nach dessen Ausführung kommt es zu einer Ausgabe, in der alle enthaltenen JDBC-Strings der SOA-12c-Testdomain gelistet sind (siehe Abbildung 7).

Im weiteren Schritt und für eine ernsthaftere Entwicklung unabdingbar, wird nun ein lokales Git-Repository erstellt und in Eclipse eingebunden. Das Egit-Plug-in existiert bereits und kann nun konfiguriert werden. Hierzu öffnen wir in Eclipse die Perspektive Git unter „Window/Perspective/Open Perspective/Other/Git“. Dort wird nun mittels „Create a new local Git repository“ ein neues Git-Repository zum Beispiel unter dem Verzeichnis „\$HOME/neon/git“ angelegt. Nun erscheint ein leeres Git-Repository im Tree.

Jetzt geht es darum, die existierenden WLST-Projekte in das erstellte Git-Repository zu übernehmen. Hierzu ist im Project Explorer das Projekt zu wählen, anschlie-

ßend im Menü mit der rechten Maustaste auf Team/„Share Project“ zu gehen und im folgenden Dialog das vorher bereitgestellte (noch leere) Git-Repository auszuwählen (siehe Abbildung 8). Dann mit „Finish“ abschließen, und das Projekt ist mit der Git-Versionsverwaltung verbunden.

Um Änderungen der Projektdateien in Git zu übernehmen, arbeitet man am besten mit der Git-Staging-Perspektive „Window/Show View/Other/Git/Git Staging“. Hier können via Commit zunächst alle erstellten WLST-Projektdateien übertragen werden. Jede weitere Änderung wird analog behandelt. Der Vorteil: Man verliert keine Änderungen, alles ist in Git verzeichnet.

Application-Server-Sicht

Eine weitere enthaltene Funktionalität ist die Application-Server-Sicht. Um diese zu nutzen, muss zunächst über „Window/Show View/Servers“ die Perspektive des Servers hinzugefügt werden. Daraufhin erscheint eine leere Ansicht. Hier kann nun der WebLogic-Server hinzugefügt werden.

Sobald die Verbindung besteht, erscheinen im Tree alle der Domain zugehörigen Server. In der Application-Server-Sicht der einzelnen Server stehen nun

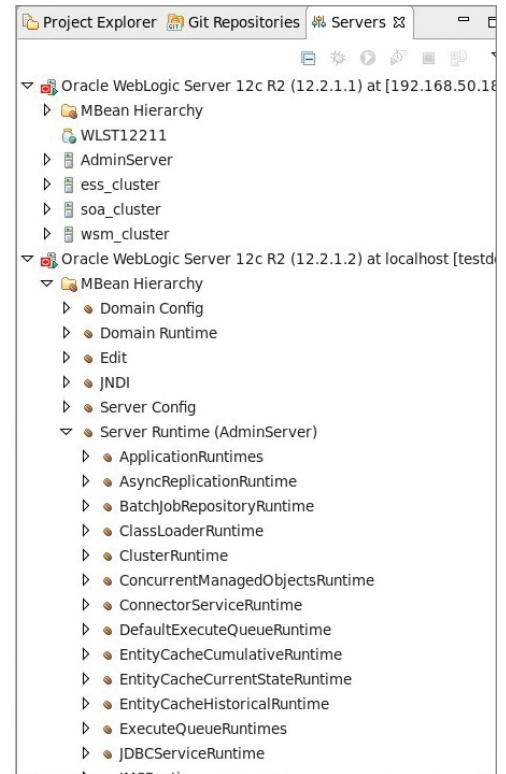


Abbildung 9: Eingebundene WebLogic-Server und geöffneter MBean-Tree

Funktionen wie „Server Start/Stop“ zur Verfügung. Interessant ist in dieser Ansicht auch der Bereich „MBean Hierarchy“. Dort kann auf den kompletten Web-

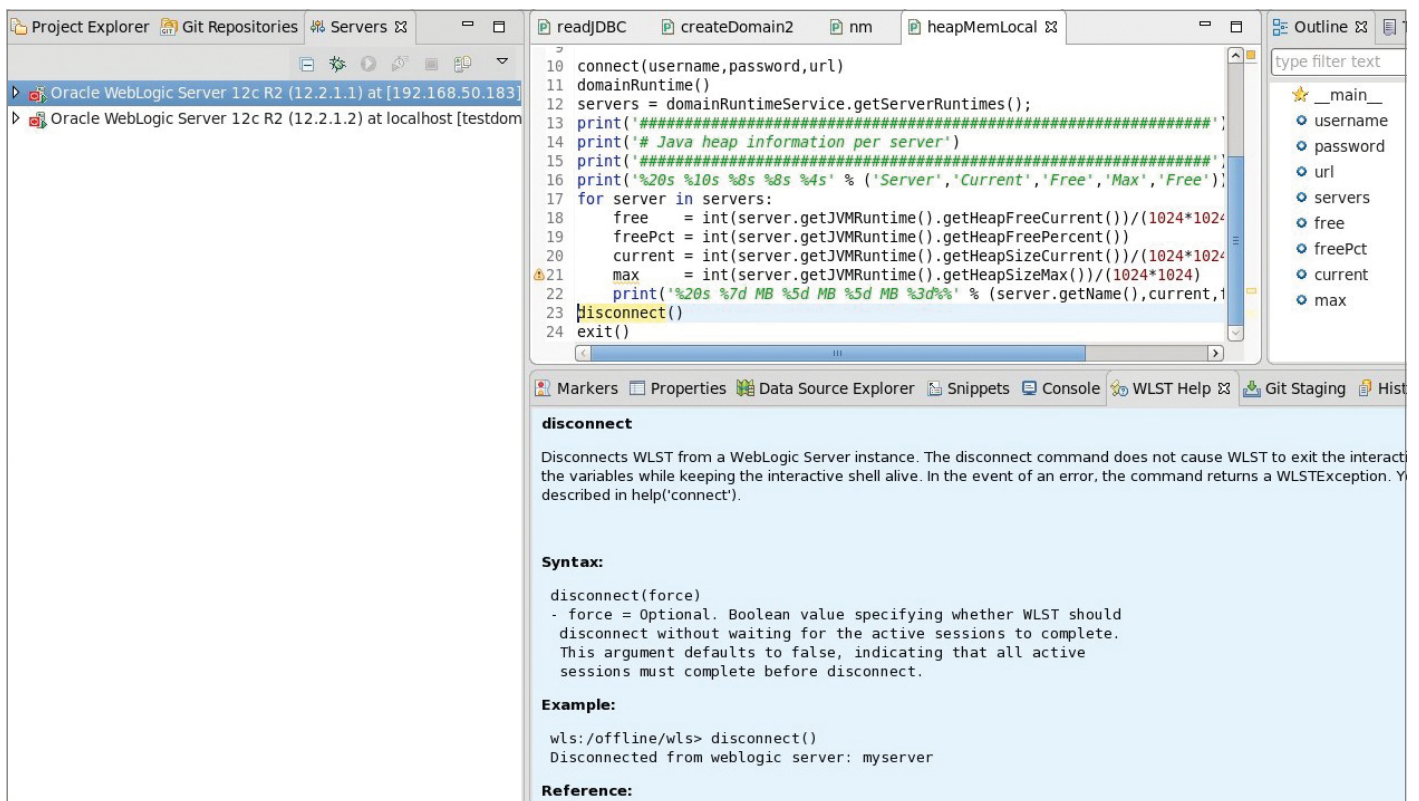


Abbildung 10: Das Feature „eingebaute WLST-Hilfe“

Logic-MBean-Tree des Admin-Servers zugegriffen werden (siehe Abbildung 9).

Zusätzliche Features

In der Programmier-Oberfläche stecken neben Core-Funktionalitäten wie Syntax Highlighting und Code Completion noch viele weitere interessante Features. Einige seien hier noch kurz erwähnt. Über die Snippet-Funktionalität können Code-Fragmente gesammelt werden, die integrierte WLST-Hilfe bietet eine Hilfestellung bei der WLST-Syntax und -Funktion (siehe Abbildung 10) und das PyDev-Plug-in enthält einen funktionierenden Debug-Modus mit Variablenverfolgung.

Fazit

Dieser Artikel zeigt die Entwicklung von WebLogic-WLST-Skripten unter Verwen-

dung der bekannten Eclipse IDE mit dem Oracle-OEPE-Plug-in. Die gesamte WLST-Funktionalität (Online/Offline) steht auch hier zur Verfügung. Entwickler profitieren durch die integrierte PyDev-IDE, die mit dem Oracle-Plug-in mitgeliefert wird, von Vorteilen wie Syntax Highlighting, Auto Complete sowie der direkten Server-Anbindung an bestehende WebLogic-Umgebungen.

Über die vorhandene Jython-Funktionalität ergeben sich neue Möglichkeiten, die dazu beitragen können, die Entwicklung mit WLST zu vereinfachen und strukturierter zu gestalten. Die beschriebene Umgebung eignet sich insbesondere für Test Cases mit verschiedenen WebLogic-Releases, da alle Server Runtimes bis einschließlich 12.2.1.2.0 unterstützt werden. Sie ist auch für Einsteiger, die sich bezüglich dieser Themen noch im Lernprozess befinden, oder für Schulungszwecke geeignet.

Anmerkungen

- [1] Der erste Teil zu diesem Artikel erschien in Ausgabe 2/2017 des Red Stack Magazin: http://www.opitz-consulting.com/fileadmin/user_upload/Collaterals/Artikel/red-stack-magazin-2017-02_WebLogic-WLST-Programmierung-unter-Eclipse-mit-dem-Oracle-OEPEPlug-in-und-PyDev_Schulze_sicher.pdf
- [2] Die Beispiele 1 und 2 sind im ersten Teil des Artikels zu finden (siehe [1]).



Michael Schulze
michael.schulze@opitz-consulting.com

*Erfahren Sie, wie Sie als DBA
APEX gewinnbringend für Ihre tägliche Arbeit
einsetzen können.*



APEX für den DBA



18.+ 19. Januar 2018 in Münster



www.doag.org/go/noon2noon



PL/SQL-Performance, Bulk Binding und Pipelining

Athanasios Manolopoulos, Sopra Steria

Ein tieferes Verständnis für die effiziente Nutzung der erweiterten PL/SQL-Funktionen erhalten Entwickler bei der TÜV Rheinland Akademie. Mit acht Oracle Approved Education Centern (OAE) ist die TÜV Rheinland Akademie der größte autorisierte Weiterbildungsanbieter in Deutschland und bietet Oracle-Schulungen zu verschiedenen Produkten an.



Oracle-Seminare wie zum Beispiel „Oracle Database 12c: Advanced PL/SQL“ vermitteln die Nutzung der fortgeschrittenen PL/SQL-Features, um einen effizienten PL/SQL-Code zu schreiben und ihn als Schnittstelle zur Datenbank oder für Anwendungen bereitzustellen. Für die Schulungen setzt die TÜV Rheinland Akademie erfahrene, zertifizierte Trainer ein (siehe „<https://akademie.tuv.com/page/informationstechnologie/oracle>“). Seit vielen Jahren arbeitet sie beispielsweise mit dem Autor zusammen. Er ist Experte für die Programmiersprache PL/SQL.

Context-Switches und Bulk Binding

Die Abfragesprache SQL und die Programmiersprache PL/SQL laufen innerhalb einer Oracle-Datenbank in zwei verschiedenen Engines. Jeder Aufruf von SQL-Anweisungen aus PL/SQL-Code heraus verursacht einen Engine-Wechsel, den sogenannten „Context-Switch“. Im Hinblick auf die Performance gilt: Context-Switches kosten Zeit. In ihnen ist die Hauptursache für PL/SQL-Performance-Einbußen zu suchen. Ihre Minimierung sollte infolgedessen das erklärte Ziel eines jeden PL/SQL-Entwicklers sein.

Der Artikel zeigt anhand von Code-Beispielen, in welcher Größenordnung Context-Switches die Performance von PL/SQL-Programmen beeinträchtigen können. Außerdem werden die PL/SQL-Sprachelemente vorgestellt, die zur Minimierung der Context-Switches zur Verfügung stehen. In den Code-Beispielen kommt die Tabelle „SALES“ aus dem Beispielschema „SH“ zur Anwendung, die bei einer Oracle-Datenbank mit installiert wird, wenn man die Checkbox „Sample Schemas“ im Installationsassistenten anklickt. Sie lassen sich auch nachinstallieren (siehe „<https://docs.oracle.com/database/121/COMSC/installation.htm#COMSC001>“).

Die Tabelle „SALES“ enthält Informationen über die Umsatz-Historie eines fik-

tiven Unternehmens (welches Produkt an welchem Datum an welchen Kunden verkauft und welcher Umsatz dabei erzielt wurde). Listing 1 zeigt die Tabellenstruktur.

Nehmen wir folgende fiktive Aufgabenstellung an: Auf Basis des Inhalts der Tabelle „SALES“ soll durch die nachfolgend beschriebenen Daten-Transformationen eine Auswertungstabelle namens „SALES_VAT“ erstellt werden, deren Struktur Listing 2 zeigt.

Diese neue Tabelle besteht lediglich aus zwei Spalten „SALES_ID“ und „VAT“ (Umsatzsteuer). Der Wert der Spalte „SALES_ID“ soll aus der Verkettung der Spaltenwerte „prod_id“, „cust_id“, „time_id“, „channel_id“ und „promo_id“ der Tabelle „SALES“ entstehen. Der Wert der Spalte „VAT“ soll aus

```
describe sales
```

Name	Null	Type
PROD_ID	NOT NULL	NUMBER
CUST_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
CHANNEL_ID	NOT NULL	NUMBER
PROMO_ID	NOT NULL	NUMBER
QUANTITY_SOLD	NOT NULL	NUMBER(10,2)
AMOUNT_SOLD	NOT NULL	NUMBER(10,2)

Listing 1: Struktur der Tabelle „SALES“

dem Spaltenwert „amount_sold“ der Tabelle „SALES“ durch Anwendung einer Umsatzsteuerformel, die den absoluten Umsatzsteuerbetrag zu den getätigten Verkäufen ermittelt, entstehen.

Die Listings 3 und 4 zeigen zwei Implementierungsvarianten und vergleichen ihre Ausführungszeiten. Die Variante von Listing 3 verwendet eine „Cursor-FOR“-Loop und verursacht, wie wir sehen werden, eine große Menge Context-Switches. Die Variante in Listing 4 verwendet das sogenannte „Bulk Binding“ und verursacht eine sehr geringe Anzahl davon. Die genaue Erklärung folgt.

Als Erstes nehmen wir den Code aus Listing 3 unter die Lupe. Beim Durchlaufen der „FOR“-Loop wird jede Zeile einzeln aus der Tabelle „SALES“ geholt („Cursor-FETCH“). Danach erfolgen die beschriebenen Daten-Transformationen. Anschließend wird durch die „INSERT“-Anweisung eine Zeile mit den gebildeten beziehungsweise berechneten Werten in die Tabelle „SALES_VAT“ eingefügt. Hier finden zwei der erwähnten Context-Switches pro Schleifen-Durchgang statt; einer bedingt durch den „Cursor-FETCH“-Vorgang und ein zweiter durch die „INSERT-Anweisung“.

Da die Cursor-Zeilenmenge aus 918.843 Zeilen besteht, ergibt sich hier die doppelte Anzahl Context-Switches (1.837.686) für den Gesamtvorgang. Das schlägt sich erwartungsgemäß auf die Performance nieder. Die Ausführung des betroffenen PL/SQL-Blocks dauert rund 19 Sekunden. Hinweis: Die Ausführungszeiten sämtlicher Anweisungen dieses Artikels wurden mit dem Notebook des Autors gemessen.

Wir nehmen jetzt den Code aus Listing 4 unter die Lupe, der die Bulk-Binding-Variante demonstriert (Klauseln „BULK COLLECT“ und „FORALL“). Im ersten Schritt werden die Daten der Tabelle „SALES“ abgerufen. Die Klausel „BULK COLLECT“ in der „SELECT“-Anweisung sorgt dafür, dass die gesamte Zeilenmenge der Tabelle „SALES“ in einem Rutsch (nur ein Context-Switch) aus der Datenbank geholt und in eine lokale PL/SQL-Tabelle („vt_sales“) gespeichert wird.

Im zweiten Schritt wird diese Zeilenmenge mit einer „FOR“-Loop umformatiert und in eine weitere PL/SQL-Tabelle („vt_sales_vat“) eingefügt. Diese „FOR“-Loop verursacht keinen einzigen Context-Switch, da die gesamte Schleife innerhalb der PL/SQL-Engine ausgeführt wird und kein Zugriff auf die Datenbank erfolgt.

```
create table sales_vat (sales_id varchar2(100), vat number);
describe sales_vat
Name          Null          Type
-----
SALES_ID      VARCHAR2(100)
VAT           NUMBER
```

Listing 2: Erstellung der Tabelle „SALES_VAT“ und Beschreibung ihrer Struktur

```
declare
  v_sales_id varchar2(100);
  v_vat number;
begin
  for r in (select * from sales) loop
    v_sales_id := r.prod_id
              || r.cust_id
              || r.time_id
              || r.channel_id
              || r.promo_id;
    v_vat := round(r.amount_sold * 19 / 119, 2);

    insert into sales_vat values (v_sales_id, v_vat);
  end loop;
  commit;
end;
/
-- Überprüfung
select count(*) from sales_vat;
```

Listing 3: Zeilenweise Verarbeitung in einer Cursor-Schleife

```
truncate table sales_vat;

declare
  type tt_sales is table of sales%rowtype;
  vt_sales tt_sales;
  type tt_sales_vat is table of sales_vat%rowtype;
  vt_sales_vat tt_sales_vat := tt_sales_vat();
begin
  -- Schritt 1: Bereitstellen der Quelldaten
  select * bulk collect into vt_sales from sales;

  -- Schritt 2: Datentransformation
  for i in vt_sales.first..vt_sales.last loop
    vt_sales_vat.extend;
    vt_sales_vat(i).sales_id := vt_sales(i).prod_id
                              || vt_sales(i).cust_id
                              || vt_sales(i).time_id
                              || vt_sales(i).channel_id
                              || vt_sales(i).promo_id;
    vt_sales_vat(i).vat := round(vt_sales(i).amount_sold * 19 / 119, 2);
  end loop;

  -- Schritt 3: Einfügen der transformierten Daten
  forall i in vt_sales_vat.first..vt_sales_vat.last
    insert into sales_vat values vt_sales_vat(i);

  commit;
end;
/
-- Überprüfung
select count(*) from sales_vat;
```

Listing 4: Bulk-Binding mit den Klauseln „BULK COLLECT“ und „FORALL“

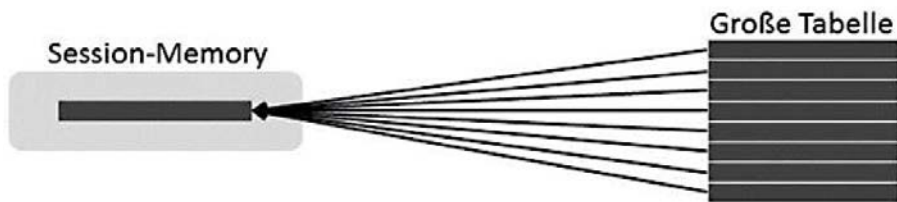


Abbildung 1: Einteilung in „Chunks“

Im dritten Schritt wird die umformatierte Zeilenmenge („vt_sales_vat“) in die Ziel-Datenbanktabelle („SALES_VAT“) eingefügt. Die „FORALL“-Klausel der entsprechenden „INSERT“-Anweisung sorgt ebenfalls dafür, dass dies in einem Rutsch geschieht. Bei Betrachtung dieses Programmablaufs wird klar, dass sich die Anzahl der Context-

Switches auf insgesamt zwei reduziert hat. Auch bei der Bemessung der Ausführungszeit wird der große Unterschied zum *Listing 3* bemerkbar. Der Vorgang dauerte hier nur zwei Sekunden, also nur ein Zehntel der im *Listing 3* bemessenen Zeit.

Aus diesem Zeitvergleich wird klar, welches Performance-Verbesserungspo-

tenzial im Bulk Binding steckt. Für die Verarbeitung großer Datenmengen ist Bulk Binding in Hinsicht auf die Performance unabdingbar.

Ressourcenprobleme bei der Anwendung des Bulk Binding

Der Code im *Listing 4* hat gezeigt, wie durch das Bulk Binding der gesamte Inhalt der Tabelle „SALES“ in einem Rutsch im lokalen Arbeitsspeicher des PL/SQL-Programms geladen wurde. Dass das so geklappt hat, liegt auch daran, dass die Tabelle „SALES“ relativ klein ist (ca. 40 MB). Bei größeren Tabellen entsteht diesbezüglich ein Ressourcenproblem. Der lokale Arbeitsspeicher kann nur Ergebnismengen bis zu einer bestimmten Größe aufnehmen. In so einem Fall wäre die vorgestellte Methode nicht anwendbar. Dafür gibt es glücklicherweise eine weitere elegante Lösung, die Chunk-Verarbeitung. Sie besteht darin, dass die Tabelle in Stücke („Chunks“) geteilt und dadurch häppchenweise in die Session Memory geladen und verarbeitet wird (*siehe Abbildung 1*). *Listing 5* zeigt die Implementierung dieses Prinzips.

Wir betrachten den Code vom *Listing 5* etwas genauer. Die „LIMIT“-Option der „BULK COLLECT“-Klausel definiert die Chunk-Größe. Sie ist über die Variable „v_limit“ frei wählbar. Dadurch können wir sie an die Größe unseres Arbeitsspeichers anpassen und das Ressourcenproblem in den Griff bekommen. Allerdings handeln wir uns dadurch auch einen Performance-Nachteil ein, da sich die Anzahl der Context-Switches wieder erhöht. Dies liegt daran, dass die Tabellenzeilen nicht alle auf einmal, sondern stückweise („Chunk“-weise) aus der Datenbank geholt werden (*siehe Listing 5*, „FETCH“-Vorgang in der äußeren LOOP).

Die Tabelle „SALES“ besteht aus 918.843 Zeilen. Die Chunk-Größe wurde auf 100.000 Zeilen eingestellt. Infolgedessen erfolgt die Verarbeitung in zehn „LOOP“-Schritten. Innerhalb jedes Loop-Durchlaufs wird einmal die Anweisung „FETCH ... BULK COLLECT INT ...“ und einmal die Anweisung „FORALL ...INSERT ...“ aufgerufen. Das macht zwei Context-Switches pro Loop-Schritt. Dies betrifft allerdings nur die äußere Loop. Die innere („FOR“-Loop) verursacht keine Context-Switches, da sie innerhalb der PL/

```
truncate table sales_vat;
declare
  type tt_sales is table of sales%rowtype;
  vt_sales tt_sales;

  type tt_sales_vat is table of sales_vat%rowtype;
  vt_sales_vat tt_sales_vat := tt_sales_vat();

  cursor cur_sales is select * from sales;
  v_limit pls_integer := 100000;
begin
  open cur_sales;

  loop
    fetch cur_sales bulk collect into vt_sales limit v_limit;
    if vt_sales.count > 0 then
      -- Datentransformation
      for i in vt_sales.first..vt_sales.last loop
        vt_sales_vat.extend;
        vt_sales_vat(i).sales_id := vt_sales(i).prod_id
                                || vt_sales(i).cust_id
                                || vt_sales(i).time_id
                                || vt_sales(i).channel_id
                                || vt_sales(i).promo_id;
        vt_sales_vat(i).vat := round(vt_sales(i).amount_sold * 1 / 119
                                   , 2);
      end loop;

      -- Bulk Insert
      forall i in vt_sales_vat.first..vt_sales_vat.last
        insert into sales_vat values vt_sales_vat(i);

      commit;

    end if;
    exit when vt_sales.count < v_limit;
  end loop;

  close cur_sales;

end;
/
-- Überprüfung
select count(*) from sales_vat;
```

Listing 5: Chunk-Verarbeitung

SQL-Engine und ohne jeglichen Zugriff auf die Datenbank läuft. Es gibt daher zwanzig Context-Switches für den Gesamtvorgang. Die Ausführungsdauer des geänderten PL/SQL-Blocks erhöht sich wieder auf 4,5 Sekunden und hat sich gegenüber dem Code aus *Listing 4* verdoppelt.

Parallelisierung und Pipelined-Funktionen

Wie wir gesehen haben, hat die Chunk-Verarbeitung den Nachteil, dass sie eine etwas erhöhte Anzahl von Context-Switches verursacht. Hier könnte jedoch eine weitere Technik zum Einsatz kommen, die diesen Nachteil mehr als wettmacht, die Parallelisierung. Parallelausführung von PL/SQL-Code lässt sich unter anderem durch „PIPELINED“-Funktionen implementieren. Der Code der *Listings 6 und 7* zeigt, wie man den anonymen Block aus *Listing 5* in eine „PIPELINED“-Funktion integrieren kann. Der Code aus *Listing 8* zeigt zusätzlich, wie sich diese Funktion aufrufen und ausführen lässt (siehe „SELECT“-Anweisung mit „TABLE“-Operator in der „FROM“-Klausel).

Wir schauen uns den Code der *Listings 6 bis 8* etwas genauer an. Die „PIPELINED“-Funktion ist hier innerhalb eines Package namens „pck_sales_vat“ und mit zwei Eingabe-Parametern namens „p_cursor“ und „p_limit“ definiert (siehe *Listing 6*). Der erste Parameter „p_cursor“ legt die Eingangsmenge und den Parallelitätsgrad der Verarbeitung fest (siehe „PARALLEL“-Hint in der „SELECT-FROM-sales“-Anweisung in *Listing 8*). Der zweite Parameter „p_limit“ definiert die Chunk-Größe.

In dem Beispiel werden ein Parallelitätsgrad von „3“ und eine Chunk-Größe von „100.000“ verwendet (siehe *Listing 8*, Funktionsaufruf). Das bedeutet: Es werden drei Parallelprozesse gestartet und infolgedessen wird die Eingangsmenge (Inhalt der Tabelle „SALES“) in drei Partitionen geteilt. Jeder Parallelprozess verarbeitet jeweils eine Partition. Jede Partition wird wiederum nicht als Ganzes, sondern Chunk-weise in Stücken von jeweils einer maximalen Größe von 100.000 Zeilen verarbeitet (siehe *Listing 7*, Implementierungscode der Funktion „fill_sales_vat“ im Package Body). Zusammengefasst kann man sagen, dass es hier eine zweistufige Aufteilung der Eingangsmenge gibt, ein-

```
-- Package-Spezifikation
create or replace package pck_sales_vat
as
    type tt_num is table of number;

    -- Definition der Pipelined Funktion
    function fill_sales_vat(p_cursor in sys_refcursor, p_limit in number)
    return tt_num pipelined
    PARALLEL_ENABLE(PARTITION p_cursor BY ANY);
end;
/
```

Listing 6: Package-Spezifikation

```
-- Package-Body
create or replace package body pck_sales_vat
as
    type tt_sales is table of sales%rowtype;

    PROCEDURE execute_business_logic (pt_sales tt_sales)
    is
        PRAGMA AUTONOMOUS_TRANSACTION;
        type vt_sales_vat is table of sales_vat%rowtype;
        vt_sales_vat tt_sales_vat := tt_sales_vat();
    begin
        for i in pt_sales.first..pt_sales.last loop
            vt_sales_vat.extend;
            vt_sales_vat(i).sales_id := pt_sales(i).prod_id
                || pt_sales(i).cust_id
                || pt_sales(i).time_id
                || pt_sales(i).channel_id
                || pt_sales(i).promo_id;
            vt_sales_vat(i).vat := round(pt_sales(i).amount_sold * 19 /
119, 2);
        end loop;

        forall i in vt_sales_vat.first..vt_sales_vat.last
            insert into sales_vat values vt_sales_vat(i);

        commit;
    end execute_business_logic;

    FUNCTION fill_sales_vat(p_cursor in sys_refcursor, p_limit in number)
    return tt_num pipelined
    PARALLEL_ENABLE(PARTITION p_cursor BY ANY)
    is
        vt_sales tt_sales;
    begin
        LOOP
            FETCH p_cursor BULK COLLECT INTO vt_sales LIMIT p_limit;
            if vt_sales.count > 0 then
                execute_business_logic(vt_sales);
            end if;
            EXIT WHEN vt_sales.count < p_limit;
        END LOOP;
        close p_cursor;

        -- Dummy Zeile zurückgeben:
        pipe row (1.0);
    end fill_sales_vat;
end;
/
```

Listing 7: Package Body

```

truncate table sales_vat;

-- Hier findet der Aufruf der Pipelined-Funktion statt
SELECT *
FROM TABLE(pck_sales_vat.fill_sales_vat(
    p_cursor =>
    CURSOR(SELECT /*+ full(s) parallel(s, 3) */ * FROM sales
s)
    ,p_limit => 100000)
);

-- Überprüfungen
select 'SALES', count(*) cnt from sales
union
select 'SALES_VAT', count(*) from sales_vat;
select * from sales_vat;

```

Listing 8: Funktionsaufruf und Überprüfung

```

SQL> set timing on
SQL> SELECT *
  2  FROM TABLE(pck_sales_vat.fill_sales_vat
  3  (p_cursor =>
  4  CURSOR(SELECT /*+ full(s) parallel(s, 3) */ * FROM sales s)
  5  ,p_limit => 100000)
  6  );

COLUMN_VALUE
-----
          1
          1
          1

Abgelaufen: 00:00:01.58

```

Listing 9

mal durch die Partitionierung seitens der Parallelprozesse und einmal durch die Limitierung der „FETCH“-Menge innerhalb jedes Prozesses.

Die eigentliche Transformations- und „INSERT“-Logik ist an eine Prozedur namens „execute_business_logic“ ausgelagert, die die Direktive „PRAGMA AUTONOMOUS_TRANSACTION“ enthält, um die Betätigung von Commits zu ermöglichen (siehe Listing 7, Implementierungscode der Prozedur „execute_business_logic“ im Package-Body). Nach jeder Verarbeitung eines einzelnen Chunk erfolgt ein Commit. Diese regelmäßigen Commits bewirken, dass immer wieder Ressourcen frei werden (Sperrern, Speicherplatz im Undo Tablespace etc.). Außerdem können wir dadurch den Fortschritt der „INSERT“-Operation in einem getrennten SQL*Plus-Fenster durch „SELECT-FROM-<Zieltabelle>-Anweisungen verfolgen. Die Commits sind die Lebenszei-

chen des Systems und der Nachweis dafür, dass die Operation noch im Gange ist. Bei Ausführungen, die mehrere Minuten dauern, ist eine Fortschrittskontrolle ein nicht zu vernachlässigender Aspekt.

Diese Ausführungen werden anschaulicher, wenn wir das Paket „pck_sales_vat“ kompilieren und anschließend die „PIPELINED“-Funktion in einem SQL*Plus-Fenster aufrufen (siehe Listing 9). Die Struktur der Ergebnismenge besteht aus nur einer numerischen Spalte („COLUMN_VALUE“). Das Ergebnis besteht aus lediglich drei Zeilen (gleich mit der Anzahl der Parallelprozesse). In jeder Zeile hat die Spalte „COLUMN_VALUE“ den Wert „1“. Dieses Ergebnis ist fachlich völlig bedeutungslos. Es wird durch die Anweisung „pipe row“ im Implementierungscode der Funktion „fill_sales_vat“ produziert (siehe Listing 7). Diese Anweisung beinhaltet in unserem Fall reinen Infrastrukturcode und er-

füllt lediglich die Konvention, dass eine „PIPELINED“-Funktion mindestens eine „PIPE-ROW“-Anweisung enthalten muss. Die fachlich relevanten Aktionen sind dagegen die Daten-Transformationen und die Bulk-Inserts, die innerhalb der Prozedur „execute_business_logic“ stattfinden. Die Ausführungsdauer der obigen Anweisung (und infolgedessen der Befüllung der Ziel-Tabelle „SALES_VAT“) beträgt nur 1,58 Sekunden. Wir beobachten hier einen deutlichen Performance-Gewinn gegenüber der sequenziellen Bulk-Insert-Variante aus Listing 5 (ein Drittel der Zeit).

Der Anwender dieser Technologie sieht sich immer mit der Fragestellung nach dem richtigen Mix aus Parallelitätsgrad und Chunk-Größe konfrontiert. Der Autor hat diese Methode auf einem Datenbank-Server mit 32-Prozessoren angewendet. Schon bei einem Parallelitätsgrad von acht Prozessen und einer Chunk-Größe von fünfhundert Zeilen ließen sich erstaunliche Performance-Gewinne bei sehr großen Tabellen erzielen. Bei der Erstellung einer Auswertungstabelle mit neunzig Millionen Zeilen und mehrere Dutzend GB groß wurde eine Bearbeitungszeit von zwanzig Minuten bemessen. Diese Tabelle konnte innerhalb einer kurzen Downtime der Applikation erstellt werden.

Fazit

Bulk Binding und Pipelining sind zwei mächtige Features der Sprache PL/SQL, die uns zur Behebung von PL/SQL-Performance-Problemen zur Verfügung stehen. Die kombinierte Verwendung beider Technologien führt zu einer deutlichen Reduzierung der Laufzeiten und kann bei zeitkritischen Prozessen äußerst wirksam sein.



Athanasios Manolopoulos
a.manolopoulos@soprasteria.com



Das neue Forms-Release 12.2.1.3 unter der Lupe

Frank Hoffmann, Cologne Data GmbH

„We did it again. Plz welcome FMW 12.2.1.3.“, postete Michael Ferrante, Produktchef für Oracle Forms am 30. August 2017 über seinen Twitter-Account „@PMOracleForms“. Mit diesem Post hat Oracle ein weiteres Mitglied der Forms-12c-Familie eingeführt. Damit sind wieder Modernisierungen, Erweiterungen und Bugfixes umgesetzt worden. Zudem ist das angekündigte „Terminal Release“ für Reports erschienen, also die letzte Version der schon im Jahr 2015 abgekündigten Reports-Technologie.

Schritt für Schritt modernisiert Oracle sein in die Jahre gekommenes Produkt für eine lange Lebensdauer. Dabei ist festzustellen, dass Forms mit der Datenbank-Version und dem WebLogic-Server laufend weiterentwickelt wird sowie mehr und mehr die von den Anwendern gewünschten Funktionen und Erweiterungen bekommt. Vieles, was früher aufwendig mit Java Beans umgesetzt werden musste, ist jetzt Teil der ausgelieferten Standard-Funktionalität.

Oracle hört aber auch den Anwendern aufmerksam zu. So wurde das gesamte

Installationsverfahren vereinfacht und optimiert. Die Dokumentation ist verbessert und am Tage der Veröffentlichung stand bereits alles unter „<https://docs.oracle.com/middleware/12213/formsandreports/index.html>“ bereit, was ein Entwickler über das Release wissen muss. Selbst die Online-Hilfe im Forms Builder wurde verbessert und beinhaltet nun alle neuen Features.

Eine für den Autor wichtige Neuerung ist die neue PL/SQL-Version. Das neue Forms nutzt intern PL/SQL 12.1.0.2 und betreibt auch den Datenbank-Client Version 12.1.0.2.

Mit PL/SQL 12.1.0.2 ist der PL/SQL-Funktionsumfang in Forms nun noch einmal deutlich erweitert. Die Masken können daher auch unter 12.2.1.3 gespeichert, jedoch mit älteren Versionen (PL/SQL 11) geöffnet werden. Mit dem Trick „`compile_all=special`“ kann man die Forms-Sourcen unberührt lassen und die neue Version testen, ohne die Sourcen direkt umstellen zu müssen. Weitere neue Features sind unter anderem:

- JavaScript-Integration for Java Web Start and Forms Standalone Launcher


```

WEBUTIL_FILE.FUNCTION FILE_MODIFIED_EPOCH
(file_name Varchar2) return NUMBER;
WEBUTIL_FILE.FUNCTION FILE_MODIFIED_DATE
(file_name Varchar2) return Varchar2;

```

Listing 1

- Forms-Application-Deployment-Services (FADS) – Verteilung von Forms-Versionen
- Support for Setting Smart Bar Size – Dynamische Toolbar-Größen-Anpassung
- Row Banding Support – Mehrfarbige Zeileneinfärbung
- Gradient Color Canvas Support – Farbverläufe im Hintergrundbild (Leinwand)
- Icon Name Property and Cursor Style Property – Eigenschaften in Grafikdarstellung
- Forms Advanced Configuration for Oracle Enterprise Manager
- Form Builder Multiple Monitor Support
- Form Builder File Open, Save, and Save As Dialog Box Modernized

- Support for Obtaining Modified Date through Forms WebUtil – Möglichkeit eines Dateivergleichs von Client und Server (siehe Listing 1)

Oracle Forms 12.2.1.3 läuft unter Java 9

Auch wenn noch nicht offiziell unterstützt, läuft die neue Version bereits mit der Java Runtime-Umgebung 9. Das hat der Autor sowohl mit dem Browser-Plug-in als auch mit Forms Standalone Launcher (FSAL) über eine JRE-Installation oder ein JRE-Filesystem erfolgreich testen können; es lässt sich auch mit dem Demoserver auf „<https://forms12c.com>“ ausprobieren. Mit der Version 12.2.1.2 kam es noch zu Fehlern. Folgen-

de Proxy-Optionen mit FSAL sind bereits möglich und werden in die nächste Forms-Dokumentation mit einfließen:

- **HTTP Proxy**
`java -Dhttp.proxyHost=<proxy-host> -Dhttp.proxyPort=<proxy-port> -jar frmsal.jar -url "<URL>"`
- **HTTPS Proxy**
`java -Dhttps.proxyHost=<proxy-host> -Dhttps.proxyPort=<proxy-port> -jar frmsal.jar -url "<URL>"`
- **Exceptions**
`java -Dhttp.nonProxyHosts="localhost|example.com" -Dhttps.proxyHost=<proxy-host> -Dhttps.proxyPort=<proxy-port> -jar frmsal.jar -url "<URL>"`
- **Use System Settings (nicht immer kompatibel)**
`java -Djava.net.useSystemProxies="true" -jar frmsal.jar -url "<URL>"`

Weitere Hinweise zur Proxy-Nutzung stehen unter „<https://docs.oracle.com/javase/8/docs/technotes/guides/net/proxies.html>“.

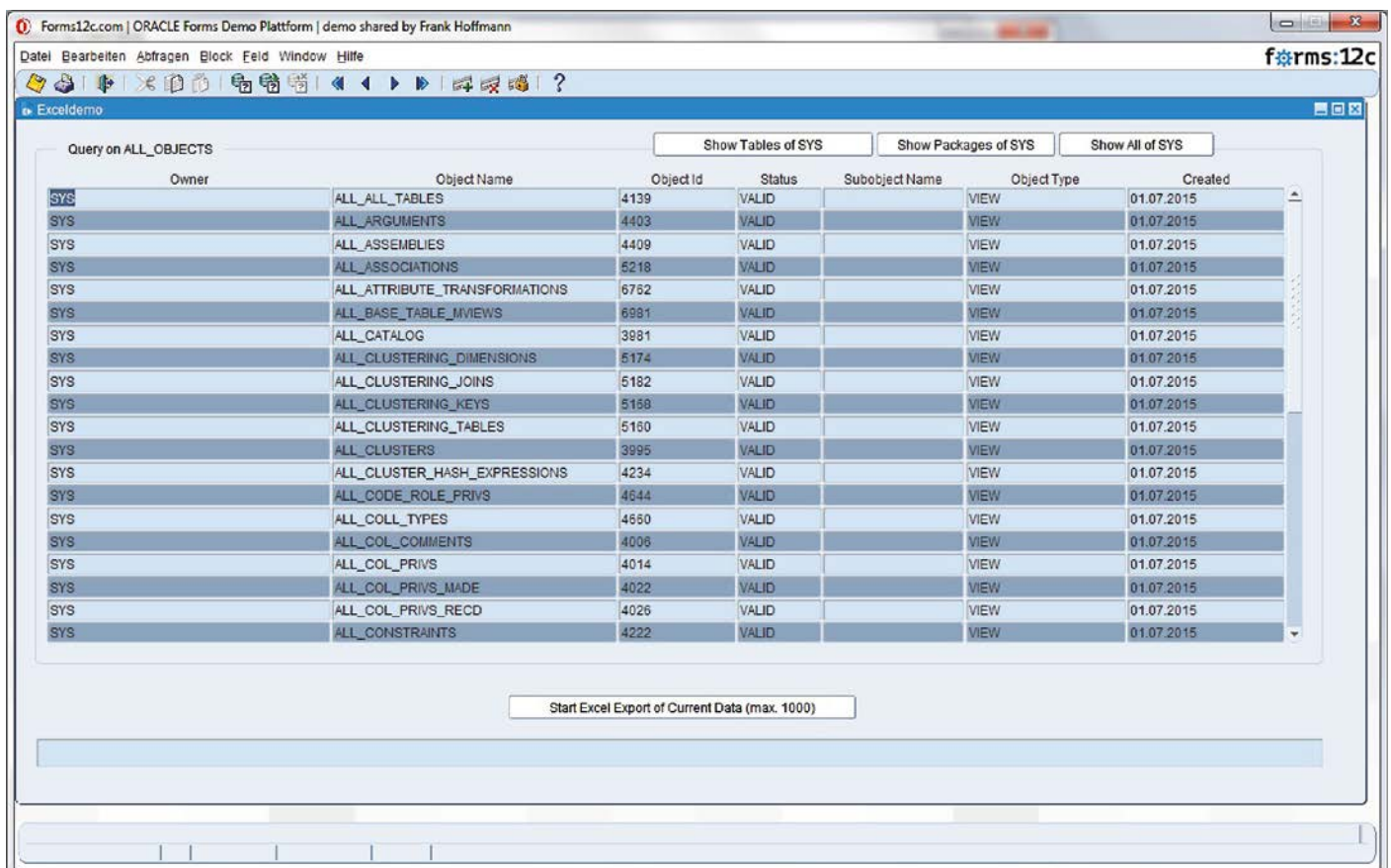


Abbildung 1: Row Banding

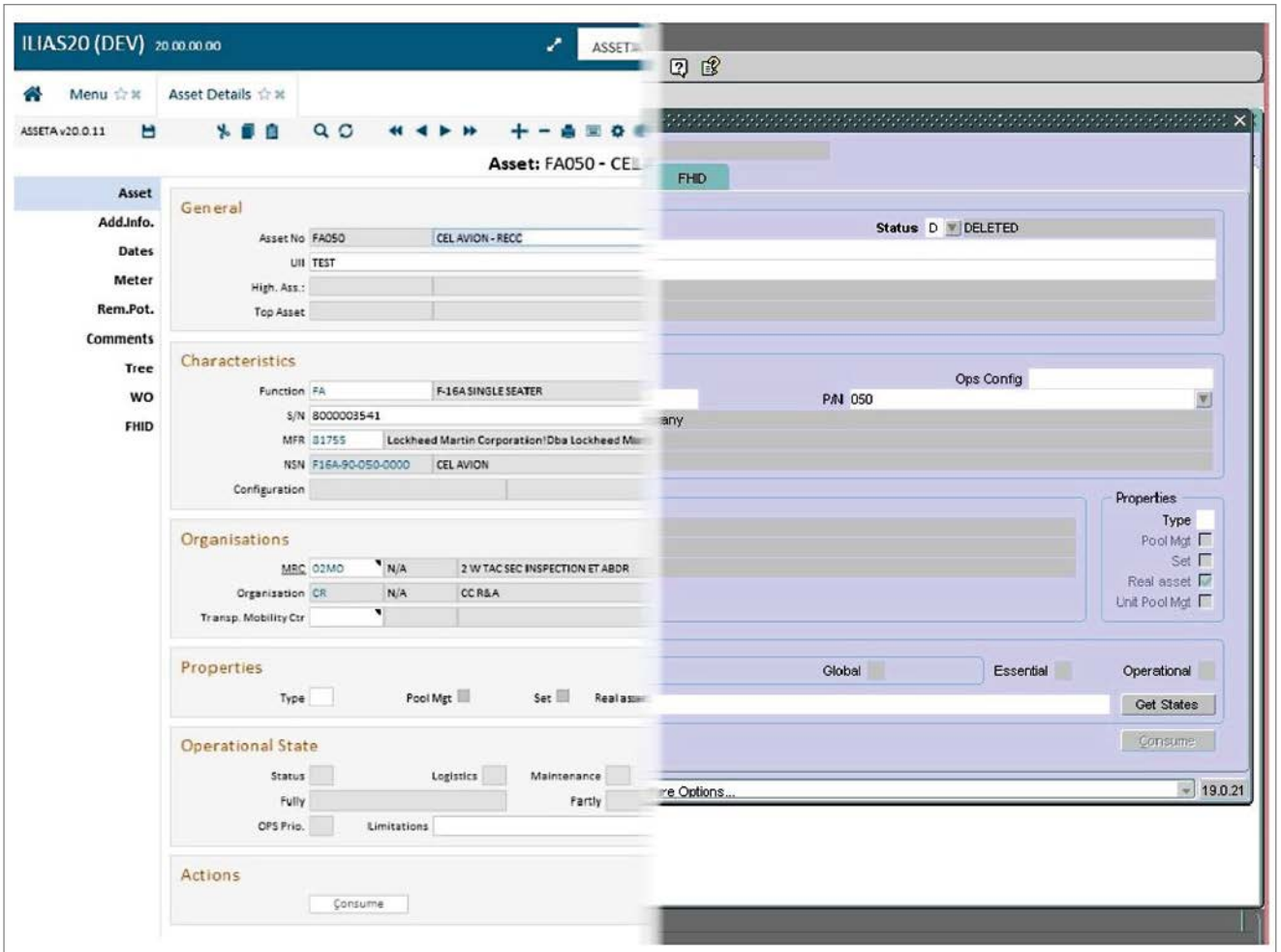


Abbildung 2: Modernisierungsbeispiel, Quelle: <http://www.ilias-solutions.com/sites/default/files/ILIAS%20Forms%20modernization%20v1%202.pdf>

Forms-WebSocket-JavaScript-Integration (WJSI)

Oracle hat eine Möglichkeit geschaffen, einen schlanken Webserver (Eclipse/Jetty Version 9.4.5) auf dem Client zu integrieren. In einer Webseite lässt sich der WebSocket sehr einfach mit „<script type=’text/javascript’ src=’/forms/java/frmwebsocketjsi.js’></script>“ einfügen.

Mit der Version 12.2.1.3 kann damit sogar JavaScript in vollem Umfang mit FSAL ohne Java-Browser-Plug-in betrieben werden. Möglich macht dies das Jetty-Java-Plug-in, das jedoch noch vom Forms-Entwickler mit „<http://central.maven.org/maven2/org/eclipse/jetty/aggregate/jetty-all/9.4.5.v20170502/jetty-all-9.4.5.v20170502-uber.jar>“ signiert werden muss.

Die neue Version ist bereits mit dem Windows Server 2016 zertifiziert und kann ab einem JDK 1.8.131 betrieben werden.

New Features von Michael Ferrante

Auch im August 2017 hat Oracle sein „Forms 12c New Features“-Dokument aktualisiert (Version 2). Hier sind nun alle neuen Features ausführlich mit Beispielen beschrieben. Herausgekommen sind vierzig Seiten, die für jeden Forms-Entwickler zu empfehlen sind (siehe „<http://www.oracle.com/technetwork/developer-tools/forms/documentation/oracleforms-12210-newfeatures-2906037.pdf>“).

Support-Zeiträume wurden nicht verlängert

Für den Autor etwas enttäuschend, wurde der Support nicht verlängert. Oracle sieht die neue Version als Teil einer Familie und bleibt bei den Terminen der ersten Version 12.2.1.0 vom Oktober 2015. Das bedeutet also weiterhin ein Ende des

Supports für Forms/Reports 12c im Oktober 2020 und Extended Support bis zum Jahr 2023. Der Autor hatte auf die Jahre 2022 und 2025 gehofft.

Der Forms-Demo-Server wurde schon kurz nach der Veröffentlichung auf die neue Version migriert. Dafür wurde eine Parallel-Installation zur alten Version auf dem Server durchgeführt.

Erfahrung bei der Installation von Forms 12.2.1.3

Die Dialoge bei der Installation sind auch verbessert. Hat der Autor mit der ersten Version 12.2.1.0 noch Wochen gekämpft, konnte er diesmal in knapp zwei Stunden die Version 12.2.1.3 auf dem Demoserver (siehe „<https://forms12c.com/>“) installieren, starten und alle Module neu kompilieren. Einige Beispiele der neuen Funktionen wurden auch bereits in den Demo-Modulen umgesetzt, etwa Row Banding (hier

SWAN look&feel) unter „<https://forms12c.com/forms/frmservlet?config=DEMO0007>“ (siehe Abbildung 1).

Ausblick auf zukünftige Versionen

Die Vorbereitung auf die nächsten Java-Versionen (18, 19 ...) laufen und Forms wird nach Ende des IE Plug-in-Supports im Jahr 2020 weiter mit FSAL lauffähig bleiben. Wie Oracle Forms mit Version 19 oder 20 auf dem Client aussehen wird, wissen wir noch Durch Bereitstellung einer Browser-Plug-in-freien Version (Webstart/FSAL) kann Forms schon heute schlank als Filearchiv installiert werden und mit allen neuen Features und einem modernen Look and Feel gestartet werden, das sich nicht verstecken muss.

Auch wenn das Oracle-Forms-Entwicklerteam klein ist, hat es einen großen Vorteil. Es arbeitet eng mit dem Java-, WebLogic- und Datenbank-Versionen zusammen und zeigt, dass es mit den aktuellen Java-Versionen, WebLogic-Server und Datenbank-Releases Schritt halten kann. Jetzt, wo ein Release ohne Reports ansteht, kann sich das Forms-Entwicklerteam voll und ganz auf eine neue Deployment-Technologie kon-

DOAG Forms Day 2018

Der Autor freut sich auf den DOAG Forms Day, der nach aktueller Planung am 21. Februar 2018 wieder in Berlin bei Oracle mit dem Thema „Forms 12c in der Praxis“ stattfinden soll. Forms-Experten aus Europa, so auch die beiden Ilias-Entwickler der Modernisierung aus *Abbildung 2* werden ihr Wissen teilen. Der letztjährige DOAG Forms Day war mit mehr als 100 aktiven Teilnehmern ausgebucht. Es macht viel Spaß mit Jürgen Menge, Jan Timmermann und Gerd Volberg die Formsflagge bei der DOAG hochzuhalten.

zentrieren. Die Forms-Community ist zwar nicht so agil und laut wie die der anderen Produkte, wächst aber wieder nach einer langen Stagnation. Das schafft Vertrauen und erhöht die Spannung auf einen möglichen und gewünschten Client-Technologiewechsel für das nächste Release. Viele Kunden sind in diesem Jahr auf Forms 12c gewechselt. Jetzt geht es an die Modernisierung, die Reports-Ablösung und die Optimierung der Software-Techniken.

Fazit

Oracle Forms hat mit der Familie 12c und der neuen Version 12.2.1.3 eine Reihe von neuen Funktionen und Modernisierungen eingeführt, die auch in der Vergangenheit angekündigt wurden. Das schafft Vertrauen und erhöht die Spannung auf einen möglichen und gewünschten Technologiewechsel für das neue Major-Release im Oktober 2020. Oracle Forms 12c lässt sich bereits heute so modern entwickeln, dass man es sich von anderen modernen Web-Anwendungen nicht mehr unterscheiden kann.



Frank Hoffmann
frank.hoffmann@cologne-data.de



Data Analytics 2018

Data als Motor der Digitalisierung

13. Data
Warehouse
Konferenz

19. & 20. März | Brühl bei Köln
analytics.doag.org

ORACLE®

DOAG



Eleganten und effizienten Code schreiben

Jürgen Sieben, ConDeS GmbH & Co. KG

Jede Programmiersprache hat – über ihre syntaktischen Details hinaus – eine spezifische Denkweise, die man verstanden haben muss, um in dieser Programmiersprache wirklich effizient und elegant programmieren zu können. Für Java und andere Sprachen ist dies das objektorientierte Paradigma, für JavaScript das funktionsbasierte.

Oftmals herrscht die Meinung vor, für PL/SQL als einfache, prozedurale Sprache gelte dies nicht in gleichem Maße. Daher werden Programme in PL/SQL auch häufig auf einfache, prozedurale Weise angegangen und gelöst: SQL liefert Daten, PL/SQL verarbeitet sie. Obwohl dies oftmals ein richtiger Ansatz ist, gibt es doch auch Strategien, die ein Pro-

blem auf elegantere Weise lösen, als andere Programmiersprachen dies könnten. Diese Strategien werden allerdings nicht durch besondere Befehle in PL/SQL erreicht, sondern nur in Kombination mit SQL, also durch die Strategie, mit der Datenbank und ihren Möglichkeiten und nicht ohne oder gar gegen diese zu programmieren.

Dieser Artikel zeigt einige dieser Strategien, obwohl mir klar ist, dass es nicht möglich ist, dieses komplexe Thema in wenigen Sätzen umfassend darzustellen. Deshalb werde ich im Red Stack Magazin in loser Folge eine kleine Kolumne zu diesem Thema anbieten, die verschiedene Aspekte dieser Strategie streiflichtartig beleuchtet, und dieser Artikel ist der Einstieg in diese Kolumne. Er richtet sich an fortgeschrittene PL/SQL-Programmierer, die sich fragen, was als Nächstes kommt, zeigt aber auch dem langjährigen Entwickler eventuell noch Alternativen und bislang ungenutzte Möglichkeiten.

Worum es mir geht, möchte ich Ihnen an einem Codebeispiel erläutern. Dieses Beispiel habe ich einem der Workshops entnommen, die ich für mein Buch „Oracle PL/SQL – das umfassende Handbuch“ erstellt habe. Sie alle haben schon mit der Konkatenation von Texten zu tun gehabt, und ich wette, keiner von Ihnen hat es wirklich gemocht. Die Verwendung des doppelten Pipe-Operators führt zu extrem unübersichtlichem Code (siehe Listing 1).

Diesen Code habe ich einer Apex-Anwendung entnommen, die durch das Apex-Entwicklerteam erstellt wurde, doch Code wie diesen haben mit Sicherheit alle schon gesehen und wahrscheinlich auch bereits selbst erstellt. Doch welche Alternative gibt es zu dieser Schreibweise? Denkbar wäre eine Extraktion der konstanten Anteile der Zeichenketten in eine Konstante mit Ersetzungszeichenfolgen, die dann durch mehrere „replace“-Anweisungen ersetzt würden (siehe Listing 2).

Besser, meine ich, ist vor allem, dass die Zeichenketten-Konkatenation nicht wiederholt werden muss und zudem durch die Auslagerung in eine Konstante

```
if (instr(l_endofUrl, substr(l_string, n+j, 1)) > 0) then
  l_url := rtrim(substr(l_string, n, j), '.' || chr(32) || chr(10));
  if p_shorten_url = 'Y' and length(l_url) > 100 then
    l_url := '<a href="' || l_url || '" target="_blank">'
      || substr(l_url,0,60) || '...' || substr(l_url,-30,30) || '</
a>';
  else
    l_url := '<a href="' || l_url || '" target="_blank">' || l_url ||
'</a>';
  end if;
  l_string := substr( l_string, 1, n-1 ) || l_url || substr( l_string,
n+j );
  l_current_pos := n + length(l_url);
end if;
```

Listing 1

```
...
  c_tmpl constant varchar2(1000) := q'^
<a href="#URL# target="_blank">#URL_DISPLAY#</a>^';
  c_end_pattern constant varchar2(5) := '.' || chr(32) || chr(10);
...
begin
  l_url := rtrim(substr(l_string, n, j), c_end_pattern);
  l_url_display := l_url;
  if p_shorten_url = 'Y' and length(l_url > 100) then
    l_url_display := (l_url,0,60) || '...' || substr(l_url,-30,30);
  end if;
  l_url := replace(
    replace(c_tmpl, '#URL#', l_url),
    '#URL_DISPLAY#', l_url_display);
...
end;
```

Listing 2

auch übersichtlicher ist: Man sieht, was am Ende herauskommen soll.

Dieses kurze Beispiel offenbart aber auch Schwächen, falls zum Beispiel viele Ersetzungsanker vorhanden sind. Die Schachtelung der „replace“-Anweisungen wird dann äußerst unschön. Doch haben wir hier keine andere Wahl, denn PL/SQL erlaubt es nicht, einer selbstgeschriebenen Funktion beliebig viele Parameter zu übergeben, um beliebig viele Ersetzungsanker auf einmal zu ersetzen. Oder doch?

Einen Ausweg bietet die Möglichkeit, ein Nested-Table-Objekt als Parameter zu übergeben. Dazu ein einfaches Beispiel: In SQL definieren wir einen benutzerdefinierten Typ, der eine beliebig lange Liste von Zeichenketten entgegennehmen kann, und verwenden eine Instanz dieses Typs als Parameter für eine selbstgeschriebene Ersetzungsfunktion (siehe Listing 3).

Besonders schön – gerade auch im Vergleich zu einer PL/SQL-Tabelle – ist die einfache Instanziierung mithilfe einer Konstruktor-Funktion. Interessant ist die Verwendung solcher Typen mit der „table“-Funktion in SQL, wie in der zweiten „select“-Abfrage, die eine Instanz dieses Typs als Zeilen einer Tabelle darstellen kann. Auf diese Weise lässt sich zum Beispiel eine Liste von Suchbegriffen zur Filterung einer „select“-Anweisung übergeben.

Nested Tables müssen nicht auf skalare Datentypen wie „VARCHAR2“ in unserem Beispiel beschränkt sein, sondern können wiederum komplexe Objekt-Strukturen repräsentieren, doch führt uns das für unser Beispiel zu weit. Da sich der SQL-Typ „CHAR_TABLE“ nun auch als Parameter in PL/SQL verwendet lässt, können wir eine einfache Hilfsmethode schreiben, die es uns erlaubt, beliebig viele Ersetzungen vornehmen zu lassen. Wir einigen uns hierzu auf die Konvention, dass in einer Zeichenkette, die als erster Parameter übergeben wird, der jeweils erste, dritte, fünfte ... Eintrag durch den zweiten, vierten, sechsten ... Eintrag ersetzt werden wird, ähnlich einer „decode“-Anweisung in SQL (siehe Listing 4).

Nun könnte der Code umgeschrieben werden, was bei zwei Ersetzungsankern allerdings noch keinen sehr großen Gewinn bringt (siehe Listing 5).

Man erkennt aber den Vorteil bei vielen Ersetzungsankern. Es gibt den Einwand, dass die Realität nicht so einfach ist, da viele Fallunterscheidungen zu treffen sind (was wir im Beispiel der darzustellenden

URL ja auch mussten) und dass daher eine Vereinfachung dieser Form nicht möglich ist. Das mag so sein, doch werden wir möglicherweise viele Fallunterscheidungen eliminieren können, wenn wir noch eine Kleinigkeit hinzufügen.

Zunächst eine weitere Konvention: Wir benennen die Ersetzungsanker wie die Spalten einer Tabelle, die die Ersetzungs-Zeichenfolgen beinhaltet. Möchten wir also den Spaltenwert „ENAME“ in einen Text einfügen, machen wir das

```
SQL> create or replace type char_table as table of varchar2(4000);
2 /
Type created.

SQL> select char_table('A', 'B', 'C')
2 from dual;
CHAR_TABLE('A','B','C')
-----
CHAR_TABLE('A', 'B', 'C')

SQL> select *
2 from table(char_table('A', 'B', 'C'));
COLUMN_VALUE
-----
A
B
C
```

Listing 3

```
SQL> create or replace function bulk_replace(
p_text in varchar2,
p_chunks in char_table)
return varchar2
as
l_result varchar2(32767);
begin
l_result := p_text;
if l_result is not null and p_chunks is not null then
for i in p_chunks.first .. p_chunks.last loop
if mod(i, 2) = 1 then
l_result := replace(l_result, p_chunks(i), p_chunks(i+1));
end if;
end loop;
end if;
return l_result;
end bulk_replace;
/
```

Listing 4

```
l_url := bulk_replace(c_tmpl, char_table(
'#URL#', l_url,
'#URL_DISPLAY#', l_url_display));
```

Listing 5

```
SQL> select kua_text_template tmpl,
2 kun_titel titel,
3 kun_vorname vorname,
4 kun_nachname nachname
5 from kunden kun
6 join kunden_art kua
7 on kun_kua_id = kua_id;
```

Listing 6

durch einen Textanker „#ENAME#“. Dann kommt das Entscheidende: Wir verlagern die Text-Templates in eine Tabelle und sorgen dafür, dass diese Texte in der ersten Spalte unserer Abfrage übergeben werden. Dadurch ist es möglich, mehrere Text-Templates zu verwenden, die sich über die „select“-Anweisung auswählen lassen. Die weiteren Spalten enthalten unsere Ersetzungs-Zeichenfolgen. Das Prinzip lässt sich an einem stark vereinfachten Beispiel untersuchen. Wir stellen uns vor, dass wir für eine Gruppe von Kunden eine Anrede erzeugen möchten. Welche Anredeform wir wählen, hängt von der Art des Kunden ab.

Die Anweisungen zur Erstellung des Kunden und seiner Kunden-Arten habe ich Ihnen im Download zum Artikel hinterlegt, das ist zum Verständnis nicht weiter spannend. Die Tabelle „KUNDEN_ART“ enthält – und das ist das Entscheidende – für jede Kundenart ein Text-Template zur Erzeugung der Anrede. Ein einfacher Join zwischen den Tabellen „KUNDEN“ und „KUNDEN_ART“ gibt uns die entsprechenden Text-Templates sowie die Werte für die Ersetzungsanker, mit den zu den Textankern passenden Spalten-Bezeichnern (siehe Listing 6).

Der Punkt ist: Wenn wir nun eine Funktion hätten, die diese SQL-Abfrage als Cur-

sor oder als Anweisung entgegennehmen könnte und die Konvention kennt, die wir festgelegt haben, würde diese Funktion ohne weitere Fallunterscheidung für jeden Kunden die korrekte Anrede erstellen; einfach, indem sie das entsprechende Kunden-Template nähme und alle vorkommenden Ersetzungsanker durch die zugehörigen Spaltenwerte ersetzt.

Wir kontrollieren über die „select“-Anweisung die Namen der Spalten über Alias und können sie so auf das jeweilige Text-Template abstimmen. Die Anzahl der Spalten ergibt sich aus der Summe der unterschiedlichen Ersetzungsanker in allen Text-Templates. Voraussetzung dafür, dass so ein Ansatz funktioniert, ist, dass wir in der Lage sind, die SQL-Anweisung oder den übergebenen Cursor zu verstehen, also eine Liste seiner Spaltenbezeichner zu erhalten und nicht nur die Daten. Genau das ist mit dem Package „DBMS_SQL“ möglich.

Die Implementierung dieses Package ist wieder im Download zu finden, hier nur die entscheidenden Teile: Wir werden im Package eine Überladung für die Methode zur Generierung des Codes anbieten, die entweder mit einer SQL-Abfrage als Zeichenkette oder aber mit einem (geöffneten) Cursor arbeiten kann. Im Package besteht die Aufgabe darin, die Metadaten des Cursors in eine PL/SQL-Tabelle zu speichern, damit man einen generischen Mechanismus besitzt, um den Spaltennamen und den Spaltenwert zu erfahren. Listing 7 zeigt diesen Teil des Codes.

Die Beschreibung des Cursors landet in der PL/SQL-Tabelle „P_CUR_DESC“ (Zeile 36ff), zudem müssen wir für jede Spalte eine Variable deklarieren, die den Wert der Spalte entgegennehmen kann. Diese Zuordnung geschieht in Zeile 43 ff. Ich verwende hier eine einzige Variable, weil ich eine Textersetzung machen möchte und daher ausschließlich mit Zeichenketten umgehen muss, woraus folgt, dass man in der „select“-Anweisung ein Casting nach Varchar2 vornehmen sollte. Eine angepasste Version unserer „BULK_REPLACE“-Methode übernimmt dann die Aufgabe, in der Zeichenkette der ersten Spalte alle Ersetzungsanker durch die Spaltenwerte der weiteren Spalten zu ersetzen (siehe Listing 8).

Wir übernehmen den Spaltenwert der ersten Spalte als Template und ersetzen in den weiteren Schleifen-Durchläufen

```

29 procedure describe_cursor(
30     p_cur in out nocopy integer,
31     p_cur_desc in out nocopy dbms_sql.desc_tab2)
32 as
33     l_column_count pls_integer;
34 begin
35
36     dbms_sql.describe_columns2(
37         c => p_cur,
38         col_cnt => l_column_count,
39         desc_t => p_cur_desc);
40
41     for i in 1 .. l_column_count loop
42         -- Registriere Variable als Ausgabevariable dieser Spalte
43         dbms_sql.define_column(
44             c => p_cur,
45             position => i,
46             column => g_value,
47             column_size => 4000);
48     end loop;
49
50 end describe_cursor;
```

Listing 7

```

function bulk_replace(
    p_cur in integer,
    p_cur_desc in dbms_sql.desc_tab2)
    return varchar2
as
    l_result varchar2(32767);
    l_anchor varchar2(32 char);
begin
    for i in p_cur_desc.first .. p_cur_desc.last loop
        if i = 1 then
            dbms_sql.column_value(p_cur, i, l_result);
        else
            l_anchor := '#' || p_cur_desc(i).col_name || '#';
            dbms_sql.column_value(p_cur, i, g_value);
            l_result := replace(l_result, l_anchor, g_value);
        end if;
    end loop;
    return l_result;
end bulk_replace;
```

Listing 8

```

SQL> declare
2     l_result varchar2(32767);
3     l_stmt varchar2(32767) := q'^
4     select kua_text_template tmpl,
5           kun_titel titel,
6           kun_vorname vorname,
7           kun_nachname nachname
8     from kunden kun
9     join kunden_art kua
10      on kun_kua_id = kua_id^';
11     c_delimiter constant char(1 byte) := chr(10);
12 begin
13     code_generator.generate_text(
14         p_stmt => l_stmt,
15         p_result => l_result,
16         p_delimiter => c_delimiter);
17     dbms_output.put_line('Ergebnis:' || c_delimiter || l_result);
18 end;
19 /
Ergebnis:
Sehr geehrte Frau Dr. med. Müller
Sehr geehrter Herr Schmitz
Sehr geehrte Damen und Herren

```

Listing 9

```

code_generator.generate_text(
    p_stmt => l_stmt,
    p_result => l_result,
    p_delimiter => c_delimiter);

l_stmt := q'^
select next_template tmpl,
       :result kunden_liste
from ...^';
open l_cur for l_stmt using l_result;
code_generator.generate_text(
    p_cur => l_cur,
    p_result => l_result,
    p_delimiter => c_delimiter);

```

Listing 10

die Ersetzungsanker gemäß unserer Konvention. Der Aufruf dieser Methode kann nun zum Beispiel wie in *Listing 9* erfolgen.

Ich möchte einen wesentlichen Punkt herausarbeiten: Wir haben konditionale Logik, die hier erforderlich gewesen wäre, wenn wir auf herkömmliche Art und Weise programmiert hätten, durch Daten ersetzt. Durch den Join der Kunden zur Kunden-Art ist diese konditionale Logik überflüssig geworden, denn diese Logik hätte doch nur anhand des Typs des Kunden entscheiden müssen, welches Template verwendet werden soll, und eventuell noch, welche Ersetzungen nun durchgeführt werden sollen und welche nicht. Diese Fallunterscheidungen sind nun nicht mehr erforderlich, wir stellen die Ersetzungsanker als Spalten in SQL zusammen, weisen die passen-

den Templates zu und überlassen SQL den Rest. Der Vorteil wird offensichtlich, wenn wir weitere Kundentypen hinzufügen, denn diese haben keinerlei Änderung am Code zur Folge, der Code funktioniert einfach. Lediglich, wenn weitere Ersetzungsanker erforderlich werden, müssen diese der „select“-Anweisung hinzugefügt werden.

Sollte es Anwendungen geben, bei denen man auf unterschiedlichem Aggregationsniveau Daten zusammenstellen muss, lässt sich dies dadurch erreichen, dass man das Ergebnis einer ersten Konvertierung mittels Code-Generator in eine Variable speichert, die man über die „select“-Anweisung des nachfolgenden Aufrufs dem Code-Generator als Ersetzungszeichenfolge zur Verfügung stellt (*siehe Listing 10*).

Natürlich ist dieses Beispiel grob vereinfacht, indem es einer Kundenart ein festes Template zuordnet. Wir können jedoch den Gedanken weiterspinnen und eine Tabelle mit Templates einrichten, die einer Kundenart für verschiedene Anwendungszwecke zugeordnet würden. Zudem können wir weitere Fähigkeiten in den Code-Generator integrieren, wie etwa die Möglichkeit, mit „NULL“-Werten umzugehen.

Wir haben diesen Ansatz in einem Projekt verwendet, in dem aus Metadaten Tabellen, Views, Indizes und Trigger erstellt werden mussten. Die Tabellen lagen in unterschiedlichen Ausprägungen vor (externe Tabelle, partitionierte Tabelle etc.), sodass für gleiche Spalten-Typen unterschiedliche DDL-Klauseln erstellt werden mussten. Die verschiedenen Templates für die Spalten-Typen, die Erzeugung externer Tabellen, Indizes und so weiter wurden in einer Template-Tabelle gepflegt und durch mehrfachen Aufruf des Code-Generators zu komplexen DDL-Anweisungen zusammengestellt. Die gesamte Anwendungslogik kam dabei ohne hartkodierte Fallunterscheidungen aus, neue Tabellen-Typen konnten durch Erweiterung der Templates einfach integriert werden, die Codemenge nahm drastisch ab.

Was sind also die entscheidenden Ideen dieser Problemlösung? Zum einen die Idee, die Ersetzungsanker zu benennen, wie die Spalten einer „select“-Abfrage, und diese für die Ersetzungen zu verwenden. Zusätzlich noch die Entscheidung, Text-Templates an die Kunden-Typen zu binden und in der Datenbank zu speichern. Erst dadurch ist dieser Ansatz möglich: Die Templates liegen nicht als Konstanten in irgendeinem Package, womit sie für SQL nicht mehr greifbar wären, sondern in einer Tabelle.

Fazit

Das Besondere an dieser Art der Programmierung ist, dass wir uns bemühen, SQL so viel wie möglich für uns erledigen zu lassen. Wir integrieren SQL so in unsere Programmierung, dass wir von der Flexibilität und den Möglichkeiten dieser Sprache profitieren, und betrachten SQL nicht nur als Mechanismus, um Daten der Datenbank für die Programmierung verfügbar zu machen.

Der Punkt, den ich klarmachen möchte, ist, dass die Ausrichtung der Programmierung von PL/SQL immer auf Datenmengen, auf Relationen und ihre Verarbeitung gerichtet sein sollte. Was kann die Datenbank für uns tun? Die entscheidende Antwort ist, dass sie viel mehr tun kann, als wir als prozedurale Programmierer denken.

Wenn wir ein Teilproblem in SQL ausdrücken können, sollten wir dies auch

tun. Dieser Gedanke ändert die Blickrichtung auf ein Problem und ich habe in vielen Projekten erfahren, dass die elegantesten Problemlösungen erst durch diesen geänderten Blickwinkel möglich wurden.

Code-Referenz

- <http://github.com/j-sieben/codegenerator>



Jürgen Sieben
j.sieben@condes.de

Termine

Januar

12.01.2018
DOAG Datenbank Webinar: Data Guard Best Practices
Online

15.01.2018
Regionaltreffen Thüringen
Jörg Hildebrandt

18.01.2018
Regionaltreffen Stuttgart
Jens-Uwe Petersen & Anja Stollberg

18.01. - 19.01.2018
DOAG 2018 Noon2Noon
Martin Klier, Robert Marz & Axel vom Stein

30.01. - 31.01.2018
DevCamp 2018
Hamburg

Februar

01.02.2018
Regionaltreffen München/Südbayern
Andreas Ströbel

01.02.2018
Regionaltreffen Bremen
Ralf Kölling

09.02.2018
DOAG Datenbank Webinar
Online

20.02.2018
Regionaltreffen Halle/Leipzig
Matthias Reimann

21.02.2018
DOAG Forms Day 2018
Jürgen Menge
Berlin

März

05.03.2018
Regionaltreffen München/Südbayern
Andreas Ströbel

09.03.2018
DOAG Datenbank Webinar: Erfahrungen mit Oracle In-Memory @Mobiliar
Online

12.03.2018
Regionaltreffen Osnabrück/Bielefeld/Münster
Andreas Kother & Klaus Günther

13.03. - 15.03.2018
Javaland 2018
Phantasialand in Brühl bei Köln

14.03. - 15.03.2018
11. Primavera Community Day
Sebastian Hunke & Sebastian Schweinle
Wien

15.03.2018
Regionaltreffen Stuttgart
Jens-Uwe Petersen & Anja Stollberg

April

09.04.2018
Regionaltreffen München/Südbayern
Andreas Ströbel

13.04.2018
DOAG Datenbank Webinar:
Online

19.04.2018
DOAG-Leitungskräfteforum 2018
Berlin

21.04.2018
DOAG ordentliche Delegiertenversammlung 2018
Berlin

25.04. - 26.04.2018
APEX Connect 2018
Düsseldorf

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- Frank Wagner
- Michael Hey
- Theodora Chatziapostolou
- Thomas Wiese
- Thomas Sührer
- Jacqueline Zühlke
- Benjamin Gnichwitz
- Patrick Weiden
- Thomas Betz
- Herbert Riess
- Oliver Braun

Firmenmitglieder DOAG

- innogy SE, Matthias Radtke
- Hamburg Netz GmbH, Andrea Buhmann

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Stefan Kinnen. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
Chefredakteur (ViSdP): Wolfgang Taschner
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Mylène Diacquenod, Marina Fischer, Yann Neuhaus, Fried Saacke

Titel, Gestaltung und Satz:

Alexander Kermas, DOAG Dienstleistungen GmbH (Anschrift s.o.)

Fotonachweis:

Titel: © aurielaki/123RF
S. 17: © bagiuiani/123RF
S. 21: © Chaowalit Koetchuea/123RF
S. 26: © Sebastien Decoret/123RF
S. 35: © rido/123RF
S. 39: © Natthawut Panyosaeng/123RF
S. 43: © Igor Zhuravlov/123RF
S. 51: © neyro2008/123RF
S. 60: © alphaspirt/123RF

Anzeigen:

Simone Fischer, DOAG Dienstleistungen GmbH (verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org
Mediadaten und Preise unter: www.doag.org/go/mediadaten

Druck:

adame Advertising and Media GmbH, www.adame.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags. Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

dbi services ag www.dbi-services.com	S. 11	E-3 Magazin www.e-3.de	S. 25	Oracle www.oracle.com	U 2
DOAG e.V. www.doag.org	S. 50, S. 59, U 3	MuniQsoft GmbH www.muniqsoft.de	S. 3	Trivadis AG www.trivadis.com	U 4

< DevCamp /> 2018

30. - 31. Januar in Hamburg





Wir leben Digitalisierung.



Zu allen Storys per QR-Code oder unter:
m.trivadis.com/digitalisierung

Die Digitale Transformation macht vor keiner Branche halt und verändert Wertschöpfungsketten und Strukturen auch Ihres Unternehmens. Sie müssen sich neuen Herausforderungen stellen, in andere Richtungen denken, Ihr Geschäftsmodell anpassen und weiterentwickeln, damit Sie auch in Zukunft wettbewerbsfähig bleiben. Sprechen Sie mit uns.

m.trivadis.com/digitalisierung | info@trivadis.com



BASEL ■ BERN ■ BRUGG ■ DÜSSELDORF ■ FRANKFURT A.M. ■ FREIBURG I.B.R. ■ GENÈVE
HAMBURG ■ KOPENHAGEN ■ LAUSANNE ■ MÜNCHEN ■ STUTTGART ■ WIEN ■ ZÜRICH

trivadis
makes IT easier. ■ ■ ■