Eclipse BIRT – Alternative zu Oracle Reports?

Henning von Bargen
Triestram & Partner GmbH
Bochum

Schlüsselworte
Eclipse, BIRT, Business Intelligence, Reporting, Open Source

Einleitung
Das bewährte Oracle Reports wird von Oracle offenbar nicht mehr weiter entwickelt und hat daher eine schlechte Zukunftsperspektive. Zum Glück gibt es Alternativen aus dem Open Source Bereich. Für das Produkt „lisa.lims“ hat T&P das Eclipse „Business Intelligence and Reporting Tool“ (BIRT) evaluiert und sich schließlich dafür entschieden. Der Vortrag bietet einen Vergleich von Oracle Reports und Eclipse BIRT im Hinblick auf verschiedene Kriterien und erläutert warum sich T&P für BIRT entschieden hat. Daneben wird ein Konzept gezeigt, wie in der Anwendung beide Produkte gemeinsam verwendet werden können.

Über den Autor
Henning von Bargen ist seit 1994 bei T&P beschäftigt und hat in dieser Zeit alle Versionen von Oracle Reports seit SQL*ReportWriter 1.1 in der Praxis kennengelernt.

Warum überhaupt über eine Alternative nachdenken?
Das Produkt lisa.lims von T&P ist ein typisches Beispiel für eine große Dialog- und Berichtsanwendung. Als lisa.lims entstand, war Oracle Forms mangels Alternativen „gesetzt“ - Alternativen zu Oracle Forms und Oracle RDBMS gab es höchstens von IBM. Für die Generierung von Berichten wurde zunächst ein hauseigenes Tool verwendet, das aber 1994 durch SQL*ReportWriter und noch im selben Jahr durch Oracle Reports abgelöst wurde.
Das Produkt ist bei vielen Kunden im Einsatz; die Größe der einzelnen Installationen variiert stark (von 4 bis ca. 800 Anwender).
Da inzwischen alle Welt über die Ablösung von Oracle Forms durch andere Techniken redet und Forms daher von Neukunden kaum noch akzeptiert wird, sollte auch die Oberfläche von lisa.lims auf eine andere technische Basis gestellt werden und bei dieser Gelegenheit auch gleich funktional erweitert werden. Bei einer solchen ohnehin radikalen Umstellung bietet sich die seltene Gelegenheit, den ganzen Schritt zu machen und auch für die Erstellung von Berichten ein anderes Werkzeug einzusetzen.
Durch den Verzicht auf Oracle Forms und Reports entfällt auch die Notwendigkeit des Oracle Application Servers. Dieser wurde mit jeder Version „gefühlt“ größer und komplizierter. Nebenbei sind dafür Lizenzen zu bezahlen und die Wartung ist nicht immer einfach.

Was sind die Alternativen?
Open Source: z.B. Pentaho Reporting, das Eclipse BIRT Project oder JasperReports.
Oracle: BI Publisher und APEX
Weitere kommerzielle Produkte anderer Hersteller

Anforderungen an ein Reporting Tool für lisa.lims
- Hohe Qualität der Ausgabe: Korrekte Ergebnisse, übersichtliche Darstellung, optisch ansprechend.
- Zuverlässigkeit/Stabilität, Nachvollziehbarkeit von Fehlern: da unbeaufsichtigte Erzeugung und Verteilung im Hintergrund.
- Durchsatz/Performance, da zwischenzeitlich hohes Druckaufkommen

- TrueType-Schriften, Barcodes, 2D-Codes
- Ausgabeformat PDF
- Entwicklungsaufwand
- Kosten, Support, Zukunftssicherheit
- Editierbare Ausgabe (Word-Format): Eher „nice-to-have“
- Interaktive Berichte spielen keine Rolle
- Ad-hoc-Berichte: dazu ist das komplexe Datenmodell weniger geeignet

Entscheidungkriterien

Die Gewichtung der Kriterien bezieht sich auf lisa.lims und kann bei anderen Projekten völlig anders sein. Erläuterungen im Vortrag.

Bedeutung	**Kriterium**
Hoch	Genaue Kontrolle über das Layout (genauer als 1 mm)
Hoch	TrueType-Schriften
Hoch	Barcodes und 2D-Codes
Hoch	Stabilität der Laufzeitumgebung
Hoch	Nachvollziehbarkeit von Fehlern
Hoch	Performance der Laufzeitumgebung
Hoch	Trigger-Code
Hoch	Dyn. Anpassung des Layouts
Hoch	Plattformunabhängigkeit
Hoch	Integrationsmöglichkeit
Hoch	Parallelverarbeitung
Normal	Leichte Änderbarkeit
Normal	Microsoft Word-Export
Normal	Microsoft Excel Export
Normal	Berichtsquelltextdateien im Textformat
Normal	Bilder
Normal	Zeichensatz
Normal	Komfort der Entwicklungsumgebung
Normal	Entwicklungsaufwand
Normal	Einarbeitung
Normal	Lizenzkosten
Normal	Wiederverwendung von Komponenten
Normal	Qualität des Supports
Normal	Weiterentwicklung
Normal	Einfache Verwendung von SQL
Niedrig	Stabilität der Entwicklungsumgebung
Niedrig	CSV-Export
Niedrig	Ausfallsicherheit und Lastverteilung der Laufzeitumgebung
Keine	Dialoggestützte Eingabe der Berichtsparameter zur Laufzeit
Keine	Interaktive Berichte (z.B. Drill-Down)
Keine	Smartphone-Unterstützung
Keine	Direktes Drucken von Berichten

Der Auswahlprozess

Schon aufgrund der notwendigen Plattformunabhängigkeit kamen einige kommerzielle Produkte nicht in Frage.

Oracle APEX: auf interaktive HTML-Berichte spezialisiert, für druckreifes PDF wird auf den BI Publisher zurückgegriffen.

Oracle BI Publisher: verfolgt einen anderen Ansatz. Zwar lassen sich damit auch gute PDF-Dateien erzeugen, aber bei den für Prüfberichte benötigten komplexen Datenmodellen und Layouts (Master-Multi-Detail über 5-6 Ebenen) stößt man schnell an die Grenzen: es muss dann mit XSLT / XSL:FO programmiert werden.

Open Source Tools: Internet-Recherche: Kandidaten Eclipse BIRT, JasperReports und Pentaho Reporting. Alle drei sind weit verbreitet und bieten einen ähnlichen Funktionsumfang
Mit BIRT und JasperReports wurden zunächst einige einfache Berichte umgesetzt. JasperReports war an einigen Stellen recht umständlich, der Designer (iReports) stürzte häufig ab. Insgesamt machte BIRT (2009/2010) einen runderen Eindruck als JasperReports.

Da auch das bisher Oracle Forms basierte User-Interface mit Eclipse (genauer: RCP/RAP) neu aufgesetzt wird, bot es sich natürlich an, zunächst BIRT auf Herz und Nieren zu prüfen. Vorteil: Entwickler finden sich schnell zurecht.
Eclipse BIRT ist Open Source und wird aktiv weiter entwickelt.

Vorgehensweise bei der Evaluierung

Die Erfahrung mit vielen Tools zeigt, dass „Scott/Tiger-Anwendungen“ aus mehr oder weniger trivialen Masken und Berichten mit jedem Tool einfach zu entwickeln sind. Um zu sehen ob ein Tool den Anforderungen von lisa.lims genügt wurde daher folgende Vorgehensweise beschlossen:

1. Erstellung von einigen einfachen Listenberichten (auch Master-Detail) basierend auf dem Datenmodell der Anwendung „zum Warmwerden“.
2. Erstellung eines einfachen Cross-Tab-Reports
3. Umsetzung eines komplexen Prüfberichts mit allen möglichen Besonderheiten wie Barcodes und Bilder aus der DB, komplexes Layout.
4. Integration der Berichte in das System zunächst prototypisch (z.B. für jeden einzelnen Bericht wird das Tool als eigener Prozess aufgerufen)
5. Vollständige Integration der Laufzeitumgebung in die Anwendung
6. Durchführung von Performance- und Lasttests

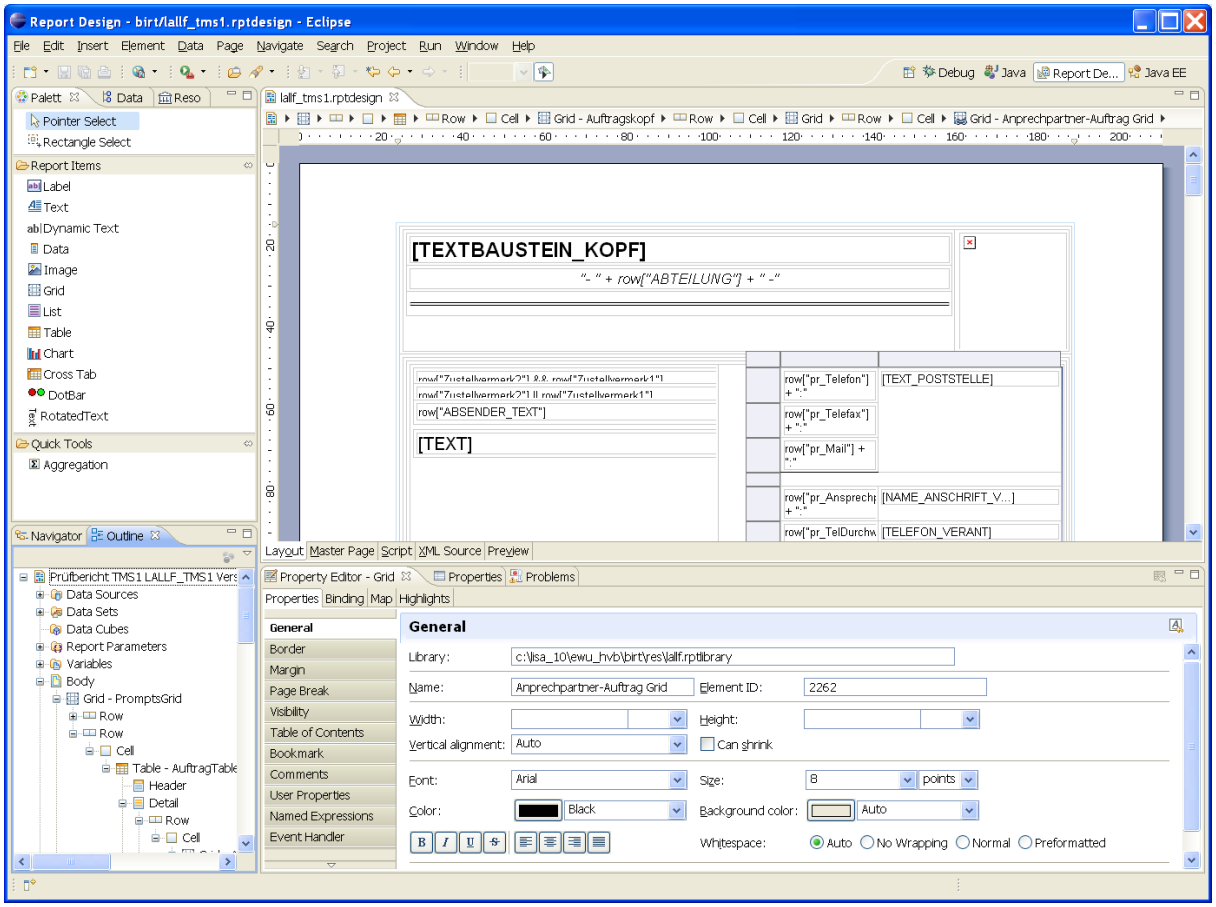
Bei jedem dieser Schritte kann es passieren, dass der untersuchte Berichtsgenerator an Kleinigkeiten scheitert. Wenn absehbar ist, dass sich das Problem nicht leicht lösen lässt, sollte zunächst ein anderer Berichtsgenerator untersucht werden.
Beispiele für solche Fallstricke in den einzelnen Phasen im Vortrag

In Phase 6, Performance- und Lasttests: Treten Fehler auf oder bricht gar das Gesamtsystem unter Last zusammen? Durchsatz? Umgang mit Resourcen? „Verbrauch“ von Resourcen (Memory Leaks, Handle Leaks)? Die Berichte sollten eine Mischung Berichten sein, wie sie auch später für die Anwendung typisch ist.

Eclipse BIRT

Um es vorwegzunehmen: Der erste Kandidat Eclipse BIRT hat in vollem Umfang überzeugt. Auf weitere detaillierte Tests mit anderen Berichtsgeneratoren wurde daher verzichtet.
Die Performance und Stabilität der BIRT Berichte ist vergleichbar mit Oracle Reports, eher noch etwas besser.
Fast alles was mit Oracle Reports geht, ist auch mit BIRT möglich; umgekehrt kann BIRT vieles was Oracle Reports nicht kann.

Wie sieht BIRT aus?



Screenshot der BIRT Entwicklungsumgebung

Die Entwicklungsumgebung verwendet Eclipse; das Fentsterlayout kann beliebig angepasst werden.
Es gibt vier wesentliche Bereiche:
Oben links: **Palette** (zum Einfügen neuer Layout Items) oder **Data Explorer** oder **Resource Explorer** (jeweils zum Drag&Drop von bestimmten Elementen in den Bericht).
Unten links: **Outline** (vgl. Objekt Navigator in Oracle Reports) oder **Navigator** (Dateisystem)
Hauptbereich: **Layout** (vgl. Layout Editor in Oracle Reports).
Unten: Properties (Eigenschaften) des ausgewählten Elements
Für viele Aufgaben gibt es außerdem eigene Dialoge, z.B. zum Anlegen/Bearbeiten von Data Sets.

Begriffswelt: Oracle Reports – Eclipse BIRT

Oracle Reports	Eclipse BIRT
.rdf-Datei (Entwicklung) .rep-Datei (Kompilierte Datei) .xml/.rex-Datei (Definition in Textform)	.rptdesign-Datei (XML-Format) „Geheime“ Funktionen in Berichten sind also nicht möglich, es sei denn sie werden in Java implementiert.
.pll/.plx/.pld-Datei	.js-Datei (Javascript), eigene Plugins, eigene Java-Klassen
.tdf-Datei	.rpttemplate-Datei (XML-Format) Themes und Styles
-	.rptlibrary Datei (vergleichbar mit .olb in Forms)

Report Parameter (nur skalare Werte)	Report Parameter (auch Listen mit mehreren Werten)
Datenmodell	-
Trigger (mit PLSQL-Code)	Event (mit Javascript Code)
-	Datenquelle (Data Source), auch mehrere
Abrage (Query) im Datenmodell	Data Set mit JDBC Datenquelle
-	Scripted Data Set
-	Filter für Data Sets
Gruppe (Group) im Datenmodell	Table Group / List Group im Layout
Filter für Gruppen (PL/SQL)	Filter für Gruppen (Wizard oder Javascript)
Formelspalte im Datenmodell	Computed Output Values bei Data Sets und beim Binding an Data Sets im Layout
Summenspalten im Datenmodell (z.B. Summe, Maximum)	Aggregation Columns bei Data Sets und beim Binding von Table und List Items
Platzhalterspalte (Placeholder) im Datenmodell	Computed Output Columns ohne Wert, die erst im onFetch-Event versorgt werden, Report Level Variables
Wizards für verschiedene Berichtsformen	.rpttemplate-Dateien mit „Cheat Sheet“ bzw. für einfache Tabelle einfach Data Set ins Layout ziehen
Matrixbericht	Cross Tab Report
-	Data Cube mit Data Sets aus verschiedenen Datenquellen
Layout-Trigger für Formatierung und Sichtbarkeit	Visibility Expression, Events onCreate, onRender (Javascript), Filter für Table und List Items
Wiederholungsrahmen (Repeating Frame) Gebunden an Gruppe aus dem Datenmodell (Werte für Bindevariablen implizit)	List Item oder Table Item Gebunden an Data Set Werte für Data Set Parameters explizit festgelegt
Rahmen (Frame)	Grid Item der Größe 1x1
-	Grid Item (tabellenförmige Anordnung)
Explizite Anker	nicht benötigt
Randbereich (Margin)	Master Page
Titelbereich, Hauptbereich, Schlussbereich	Wechsel der Master Page vor oder nach bestimmten Layout Items
Boilerplate Text	Label Item, Text Item oder Dynamic Text Item
Datenfeld (Field) im Layout, gebunden an Feld aus dem Datenmodell	Data Item, Text Item oder Dynamic Text Item
Verstecktes Feld Verwendung um Texte zusammenzusetzen	- Dynamic Text Item
Bild im Layout	Image Item
Grafische Formen (Linien, Kreise, Rechtecke)	Nur waagerechte und senkrechte Linien möglich.
-	Einbinden von SVG-Grafiken
Diagramm (Chart)	Chart
-	Dynamisch generierte Bitmaps
Barcode mit TrueType Schriftart	Barcode mit TrueType Schriftart
Barcodes / 2D-Codes mit z.B. ID Automation Library	Kostenloses Plugin oder z.B. ID Automation Library
Um 90° gedrehter Text	Kostenloses Plugin

Haken und Ösen bei der Entwicklung von Berichten

Bei den folgenden Punkten gab es zunächst Schwierigkeiten, die aber beseitigt werden konnten:

Stabilität des BIRT Report Designers

Der Designer wird bei jedem Speichern des Berichts etwas langsamer und stürzt evtl. nach einigen Stunden Arbeit ab – ähnlich wie frühere Oracle Reports Versionen. In der Praxis kein Problem.

Barcodes

Für reine Barcodes wie Code 39 ist es natürlich möglich, die Ausgabe mit Hilfe einer entsprechenden TrueType-Schriftart zu erzeugen. Aus Oracle Reports Zeiten steht noch eine Lizenz für die bewährte Java Barcode-Library von ID Automation zur Verfügung. Diese wird über einen kleinen Javascript-Wrapper angesprochen, so dass z.B. für einen DataMatrix-Code nur ein Image Item und eine Zeile Code notwendig sind. Seit kurzem gibt es auch ein kostenloses Plugin (nicht getestet)

Javascript

Diese Sprache muss man nicht unbedingt mögen und man wird gelegentlich über die Laxheit der Sprache fluchen, die leicht zu Fehlern zur Laufzeit führen kann. Mit PL/SQL sind viele solcher Fehler ausgeschlossen. Andererseits ist Javascript deutlich mächtiger als PL/SQL.

Fehlermeldungen

Eine Schwäche von BIRT ist, dass Fehlermeldungen leicht untergehen. Das lässt sich allerdings ändern (Details im Vortrag).

SQL in Events/Triggern

Das ist sicher der Punkt, den viele am meisten vermissen werden. Mit Hilfe eines eigenen Plugins konnte BIRT dazu überredet werden, dass aus Javascript fast überall SQL (inkl. Bind-Variablen) aufgerufen werden kann. Auch DML ist möglich und der Aufruf von Prozeduren/Funktionen mit IN- und OUT-Parametern (Beispiel im Vortrag).

Bilder aus der Datenbank

Der Zugriff auf Bilder ist nur über BLOBs möglich, nicht über BFILEs => mit Stored Function lösbar.

Layout

Die Voreinstellungen von BIRT (für Seitenumbrüche und Texte) sind für PDF-Ausgabe ungünstig. Das meiste lässt sich per Style Sheet global ändern.

Gruppenfunktionen

BIRT arbeitet in dieser Beziehung einfach anders als Oracle Reports. Während bei Oracle Reports Gruppierungen und entsprechende Gruppenfunktionen im Datenmodell definiert werden, geschieht das bei BIRT im Layout (bei Table und List Items). Das hat Vor- und Nachteile, vor allem aber muss der Entwickler etwas umdenken.

Formatierung von Seitenzahlen

Kleines Problem: Die Seitenzahl im Format „Seite X von Y“ kann nicht rechtsbündig ausgegeben werden. Workaround: Kleiner Postprocessor (auf Basis von iText).

Was kann BIRT besser als Oracle Reports?

Berichte (Abfragen, Layoutobjekte o.ä) zur Laufzeit skriptgesteuert anpassen

Man stelle sich das so ähnlich vor wie den Forms API Master, aber zur Laufzeit.

Zur Laufzeit des Berichts lassen sich beliebige Objekte – seien es Datenquellen, Abfragen oder Layoutelemente *hinzufügen*, löschen oder anpassen.
Sehr einfach ist es, Abfragen zur Laufzeit dynamisch zu ändern (ähnlich wie in Oracle Reports mit lexikalischen Parametern).

Tabellenspalten ausblenden, den zusätzlichen Platz für die übrigen Spalten verwenden

Möglich, da BIRT ein *echtes Tabellenmodell* im Layout hat (wie HTML).

Wiederverwendung von Komponenten

Dies ist vielleicht der größte Vorteil gegenüber Oracle Reports. Ähnlich wie in Oracle Forms mit den Objektbibliotheken können bei BIRT Report-Libraries angelegt werden. Eine solche Library enthält beliebige Elemente wie Datenquellen, Abfragen oder Layoutstrukturen.
Elemente aus der Library können per Drag&Drop in den Bericht hereingezogen werden, der Bericht referenziert dann zur Laufzeit das Element aus der Bibliothek. Änderungen in der Bibliothek werden zur Laufzeit automatisch berücksichtigt.
Anwendungsbeispiel: Ein komplettes Adressfenster, Master Pages, sowie sonstige immer wieder verwendete Elemente z.B. Kopfzeilen, Unterschriftenbereich, Textbausteine usw.
Einfaches Beispiel „Textbaustein“: In der DB gibt es eine Tabelle mit Textbausteinen. Aus der Library wird ein Layoutelement „Textbaustein“ in den Bericht gezogen und dann nur noch im Dataset Parameter Binding der Name des gewünschten Textbausteins übergeben.
Durch die konsequente Verwendung von Libraries bleiben die einzelnen Berichte überschaubar und Fehlerbehebungen oder Änderungen erfordern meist weniger Aufwand, da nur an *einer* Stelle etwas geändert werden muss.
Neben den Libraries gibt es Templates, das sind aber im Prinzip nur Kopiervorlagen für neue Berichte.

Scripted Data Sets

Bei einem Scripted Data Set programmiert der Entwickler selbst die open/fetch/close Methoden. Auf diese Weise können beliebige Daten die Basis für ein Data Set bilden.

Mehrfache Verwendung von Data Sets innerhalb eines Berichts

Caching von Ergebnissen

Verknüpfung von Daten aus verschiedenen Datenquellen

(mehr im Vortrag)

Weitere Gesichtspunkte

Migrationsaufwand

Es gibt kein fertiges Tool für die Konvertierung von Oracle Reports nach BIRT.
Daher muss ein vorhandener Bericht, der in BIRT laufen soll, (abgesehen von den SQL-Statements) komplett neu entwickelt werden.
Der zeitliche Aufwand dafür ist als Faustregel ungefähr so groß wie er für die Entwicklung des Berichts mit Oracle Reports war. Eine gewisse Ersparnis ergibt sich daraus, dass die SQL-Abfragen mit minimalen Anpassungen in BIRT weiter verwendet werden können.

Zukunftssicherheit

BIRT ist (laut Aussage auf www.birt-exchange.org) führend bei den Open Source BI Tools.
Da das Produkt Open Source ist, gilt dasselbe wie für alle anderen Open Source Tools. Wichtig ist, dass mit IBM und Actuate auch kommerzielle Unternehmen dahinter stehen. Man kann davon

ausgehen, dass das Produkt über viele Jahre weiter gepflegt wird. Für BIRT spricht auch, dass BIRT-Berichte als XML gespeichert werden.

Einarbeitungszeit für Entwickler

Die Einarbeitungszeit sollte nicht unterschätzt werden und ist vergleichbar mit Oracle Reports. Erste Erfolge werden schnell erzielt, wenn es aber darum geht, komplexe Berichte umzusetzen und das Tool wirklich zu beherrschen, sollte man von mehreren Wochen oder besser Monaten ausgehen –das ist bei Oracle Reports ähnlich.

Integrationskonzept

Frühzeitig sollte überlegt werden, wie BIRT Berichte in die eigene Anwendung eingebaut werden können. Es gibt viele Möglichkeiten.

Konzept des lisa.lims Reporting

Der Bericht bekommt eine Job-ID übergeben und holt sich alle weiteren Informationen anhand dieser Job ID selbst aus dem Repository.
Die Reporting-Komponente der Anwendung verwendet entweder rwservlet (per http) um Oracle Reports Berichte zu erzeugen oder einen selbst entwickelten Multihreaded BIRT Server (per Socketverbindung) um BIRT Berichte zu erzeugen – jeweils immer nur als Datei. Die Verteilung der Datei übernimmt die Anwendung selbst. BIRT und Oracle Reports können problemlos parallel verwendet werden.
Mehr dazu im Vortrag.

Entwicklungsaufwand

Der Entwicklungsaufwand ist im direkten Vergleich zu Oracle Reports *geringfügig höher*. Die Ursachen dafür:
Javascript ist als Sprache für Events fehleranfälliger als PL/SQL, insbesondere da der BIRT Designer bei Javascript-Fehlern versucht einfach weiter zu machen.
Die etwas *umständlichere Verwendung von SQL* in Triggern über Javascript.
Data Sets: Parameter müssen *explizit übergeben* werden (in Oracle Reports alle selektierten Spalten als Bind-Variablen verfügbar). Das erfordert für jeden Parameter ein paar zusätzliche Sekunden.
Der etwas höhere zeitliche Aufwand wird m.E. dadurch wettgemacht, dass dem Entwickler mit BIRT deutlich mehr Möglichkeiten zur Verfügung stehen.
In jedem Fall ist es ratsam (gilt genauso für Oracle Reports) sich ein *Framework* mit Hilfsfunktionen und sonstigen wiederverwendbaren Objekten zu schaffen.

Performance

Die Performance der Laufzeitumgebung ist sehr gut, subjektiv sogar etwas besser als bei Oracle Reports. In konkreten Zahlen:
Beim Lasttest liefen maximal 5 BIRT Threads parallel. Pro Bericht muss mit einem Overhead von etwa 1-2 Sekunden Wartezeit durch das lisa.lims Framework gerechnet werden, der also unabhängig von der verwendeten Report Engine entsteht (egal ob Oracle Reports oder BIRT). Aufgerufen wurde eine Mischung aus relativ komplexen Prüfberichten (viele SQL-Statements, relativ komplexes Layout, mit DataMatrix-Codes) und einfachen Listenberichten, wobei teilweise mehrere hundert Seiten lange Listen erzeugt wurden.
In diesem Szenario schaffte der BIRT-Prozess (mit wie gesagt max. 5 BIRT Threads) einen Durchsatz von 1000 Berichten in 20 Minuten auf einem handelsüblichen Server (DELL Server PE2950, Intel Xeon CPU E5430 @ 2.66 GHz, 4 GB Ram, Windows 2003 R2).
Die CPU-Last auf dem Application Server blieb dabei moderat (unter 20%). Da sehr viele Debug-Meldungen ausgegeben wurden, könnte der Durchsatz noch weiter gesteigert werden. Der Java-

Prozess hat während der Zeit nie mehr als 300 MB Hauptspeicher benötigt und es wurden keine Windows-Handles „verbraucht“.
Der Durchsatz war vorher mit Oracle Reports ähnlich.

Stabilität

Inzwischen läuft die Anwendung mit BIRT bei mehreren Kunden produktiv und es gab bisher keinerlei Probleme die auf BIRT zurückzuführen sind.
Bei diversen Oracle Reports Installationen gibt es dagegen gelegentlich Fehler der Art „REP-51002: Verbindung zum Reports Server nicht möglich“ oder bei Reports 11 „REP-300: OCI_INVALID_HANDLE“ (für diesen 2. Fehler gibt es inzwischen einen Patch!). In solchen Fällen sind stets manuelle Eingriffe notwendig
In der Hinsicht Performance liegen beide Produkte etwa gleichauf und bei der Stabilität ist nach unseren Erfahrungen Eclipse BIRT deutlich überlegen, obwohl auch Oracle Reports schon durchaus solide läuft.

Für wirklich sehr große Berichte (mit zigtausenden Seiten in einem Bericht) ist Oracle Reports vermutlich besser geeignet, da BIRT die Ergebnisse und auch das resultierende Layout komplett im Hauptspeicher aufbereitet. Oracle Reports kommt evtl. mit weniger Hauptspeicher aus.
Auf der anderen Seite benötigt Oracle Reports im Vergleich zu BIRT mehr Arbeitsspeicher wenn JPEG-Bilder in Berichte eingebunden werden und auch die PDF-Ausgabedateien sind dann sehr groß.

Komplexität der Installation und Konfiguration

Der Oracle Application Server wird von Version zu Version komplexer. Beim Oracle WebLogic Server 11 nimmt die Installation und Konfiguration des AS für Forms & Reports beim Kunden oft mehrere Stunden in Anspruch. BIRT dagegen ist einfach nur eine Sammlung von JAR-Dateien, d.h. für BIRT müssen nur einige Dateien kopiert bzw. entpackt werden und es muss der Classpath entsprechend angepasst werden. Je nach Betriebssystem müssen dann vielleicht noch TrueType-Schriften kopiert werden und die Datei fontsConfig.xml angepasst werden – das war’s dann auch schon.

Support

Bei Problemen und Fragen hilft die Community in der Regel schnell und fundiert. Falls das einmal nicht reichen sollte (was bei uns bisher nicht vorgekommen ist), kann kommerzielle Unterstützung bei verschiedenen Firmen (auch in Deutschland) gekauft werden.
Erweiterungen und neue Features kann man sich wünschen, sie werden aber wohl nur umgesetzt wenn jemand dafür zu zahlen bereit ist oder freundlicherweise selbst einen Patch bereitstellt. Bei Fehlern ist es ähnlich: Man kann dafür abstimmen, einen Fehler möglichst schnell zu beheben, sollte aber nicht davon ausgehen dass jeder Fehler umgehend behoben wird, es sei denn man ist bereit das selbst zu tun oder dafür zu bezahlen. Das ist im Grunde auch nicht viel anders als bei kommerziellen Produkten.

Fazit

Der Einsatz von BIRT hat sich in unserem Fall bisher absolut bewährt. Der geringfügig höhere Entwicklungsaufwand wird durch die einfachere Installation und den niedrigeren Installations- und Wartungsaufwand ausgeglichen.
Die Ergebnisse sind jedoch nicht 1:1 übertragbar, da der Anwendungsfall in mancherlei Hinsicht speziell ist –insbesondere da interaktive Berichte keine Rolle spielen und für die Parametereingabe sowie die Berichtsplanung und -verteilung bereits fertige Komponenten existierten. Für die Integration wurde ein einfacher Multithreaded-BIRT-Server entwickelt, der ähnlich wie das Oracle rwsverlet Aufgaben entgegen nimmt und Berichte als Dateien generiert.

Kontaktadresse:
Henning von Bargen
Triestram & Partner GmbH
Kohlenstraße 55
D-44795 Bochum

Telefon:	+49 (0) 234-94375-0
Fax:	+49 (0) 234-452206
E-Mail	h.vonbargen@t-p.com, info@t-p.com
Internet:	www.t-p.com