

# Open Source Reporting Solutions for Institutional Reporting: The BIRT Approach

*Peter Corbett and Dean Ward  
Edith Cowan University*

## Abstract

*Edith Cowan University (ECU) like many Higher Education Providers collects, manipulates and disseminates large quantities of information and data, both for internal as well as external use. Finding effective, cost-efficient methods of dissemination can be problematic.*

*Higher Education Providers invest in Business Intelligence Technology (BI) because it is widely believed it can enhance and improve performance, including dissemination, by using information to better understand interdependencies and relationships, as well as key variables and drivers that impact and underpin institutional performance. The knowledge gained from BI should provide leverage to pursue growth and profitability, as well as, identifying and minimizing risk and regulatory exposure.*

*Usually, investment in BI is a major capital expense with initial software, hardware and consulting costs. In addition there are expenses for licensing and consulting for software upgrades. In times of falling student numbers and tightening budgets, reducing initial and ongoing licensing costs is desirable.*

*The Eclipse Foundation produces a number of open-source software solutions one of which, the Business Intelligence and Reporting Tools Project (BIRT) may prove useful for institutional reporting. ECU Strategic Information Services embarked on a small proof of concept with BIRT to examine the viability of this type of solution.*

*The proof of concept brief being to:*

- · *Examine the pros and cons of traditional vs. open-source reporting solutions.*
- · *Determine potential open-source reporting solutions.*
- · *Investigate other Institutions using BIRT and their experiences, both successes and failures.*
- · *Determine usable BIRT Functionality.*
- · *Measure potential savings/costs.*
- · *Assess and recommend viability.*

*This paper discusses the results of the proof of concept and in particular some experiences in KPI reporting with BIRT.*

## Introduction

Business Intelligence (BI) in the broadest sense, refers to a class of software and technologies, developed with the purpose of enabling organizations to make better use of their information by gaining access to it, analyzing it and using it for decision making purposes.

In comparison, "Institutional Research involves the collection of data or the making of studies useful or necessary in (a) understanding and interpreting the institution; (b) making intelligent decisions about current operations or plans for the future; (c) improving the efficiency and effectiveness of the institution".<sup>1</sup>

Within the Institutional Research (IR) context, BI is a tool to that can be utilized to achieve the desired IR outcomes and is currently being used by Universities, to varying degrees.

---

<sup>1</sup> Paul L. Dressel, *The Nature of Institutional Research*, Michigan State University, 1966

BI can be achieved by many means and type or combinations of software. Products from companies such as, Microsoft, COGNOS, Business Objects and SAS are commonly used within Universities.

These products are commercially developed and build open proprietary technologies.

This paper considers an alternative model for BI, "Open Source" and in particular Edith Cowan University's experience with a test application of Business Intelligence and Reporting Tools (BIRT).

## The emergence of Open Source

Open Source (OS) is a recent phenomenon and describes an architecture of open standards, accessible software code and product developments based on alliances and group work. The success of OS has been spectacular:

- In competition to Microsoft Operating Systems, Linux has developed to the level of being a preferred standard for many<sup>2</sup>
- Apache Web servers is now a dominant player in the Web Server market<sup>3</sup>
- Perl and PHP are popular and commonly utilized programming languages

Officially "open source" is the trademark of the OS Initiative<sup>4</sup> which in its purest form is defined as a standard/license that has the following features:

- Free redistribution – there is no restriction in distributing the software, such as licensing fees
- Source Code – the program must include source code or the source code is available without charge on the Internet
- Derived Works – The product license must allow for modification and secondary products. These secondary products must adopt the same inherited standards
- Integrity of the Author's Code – The license may restrict source-code from being distributed and modified only where the license allows for patches to be made to the source code
- No Discrimination against Persons or Groups – No discrimination is allowable
- No Discrimination against Fields of Endeavour – there are no restrictions on which areas or how the license or software is used
- Distribution of License – The rights to license, when distributed, does not require an additional license
- License must not be Specific to a Product – The license can not be tied to a product
- License must not contaminate other Software - The license must not restrict other software that may be distributed with it.

As an outworking of the OS standard, the software may be perpetually available for development and improvement by the development community, where as proprietary software is exclusively under the control of and therefore can only be modified by the owner.<sup>5</sup>

---

<sup>2</sup> West, J., Dedrick, J., "Proprietary vs. open standards in the network era: an examination of the Linux phenomenon" System Sciences, 2001. *Proceedings of the 34th Annual Hawaii International Conference*, 3-6 January 2001, p. 10.

<sup>3</sup> Bellas, F. "Standards for second-generation portals" *Internet Computing IEEE*, March April 2004, p. 54.

<sup>4</sup> <http://www.opensource.org/>

<sup>5</sup> Conner K. R. & C.K. Prehalad "A resource-based theory of the firm: Knowledge versus opportunism" *Organizational Science* 1996, V7, 225

Other potentially differentiating features of the OS development model are:

- Greater involvement and ownership by developers – they are often motivated by interest rather than commercial reward
- The need for gatekeepers who check the integrity of code and incorporate changes into the official base product
- More adaptable and cost efficient when compared to new developments – Open Source is more adaptive to the users need
- More effective debugging and testing – There is a large dedicated development community
- Benefits from plurality – each person involved has a potentially different background such as industry, immediate needs from the software and different experience/expertise that can be contributed when developing the software

The label of “Open Source” is now being applied to a diverse range of applications, product and activities which diverge significantly from the purest form.

Extremely successful developments such as Wikipedia<sup>6</sup> have been described as OS and while using OS in their development, do not espouse the required characteristics.<sup>7</sup> OS in the context of Wikipedia reflects the content contributor model that it promotes. Therefore caution needs to be applied, wherever the OS label is applied and an assessment should be made to determine the real extent that OS, is actually an appropriate descriptor.

Indeed, a number of disputes and potentially litigious actions are becoming evident and relate to so called “Open Source” products, not in fact being such.<sup>8</sup>

One of the most effective methods of determining whether the product is OS, is to review the license. Generally there are six key license types and their specifications against some key OS requirements are summarized in the following table:

**Table 1. Comparing Open Source Software License Types**

License Type <sup>9</sup>	Code protected by copyright?	Can code be used in Closed Source Project?	Can project that uses code, be sold?	Must Source Code be released?	Provides for patents?
Public Domain	No	Yes	Yes	No	No
BSD/MIT	Yes	Yes	Yes	No	No
GPL (v2)	Yes	No	No	Yes	No
LGPL	Yes	Yes	Yes	Yes	No
MPL/CDDL	Yes	Yes	Yes	Yes	Yes
CPL/EPL	Yes	Yes	Yes	Yes	Yes

<sup>6</sup> <http://www.wikipedia.org/>

<sup>7</sup> Terdiman D. "Wikipedia's open-source label conundrum" *CNET News.com*, 9 December 2005.

<sup>8</sup> Byfield B. "Free Software Foundation and OpenOffice.org team up to escape Java trap" *NewForge* (<http://software.newsforge.com/software/05/05/16/1358227.shtml>) 16 May 2005.

<sup>9</sup> For more details on these acronyms and the specifics of the licenses, visit <http://www.opensource.org/>

The numbers of license type variants are constantly changing and reflect legal and other developments. The most pervasive license is GPL (General Purpose License) and it is expected that GPL (v3) will be released in 2006 and will reflect a revised definition of source code and consider issues related to digital management/infringement.

A key success factor in OS is individual involvement. Why individuals participate in OS developments has been the subject of research. Raymond<sup>10</sup> suggested three reasons:

- direct benefit
- enjoy programming
- enhanced reputation and recognition by peers

Subsequent research supports these reasons.<sup>11</sup>

As one OS commentator stated:

*Open source is a way of life...*<sup>12</sup>

The OS "Model" has many variants, however the following can be indicative of the "general" model:<sup>13</sup>

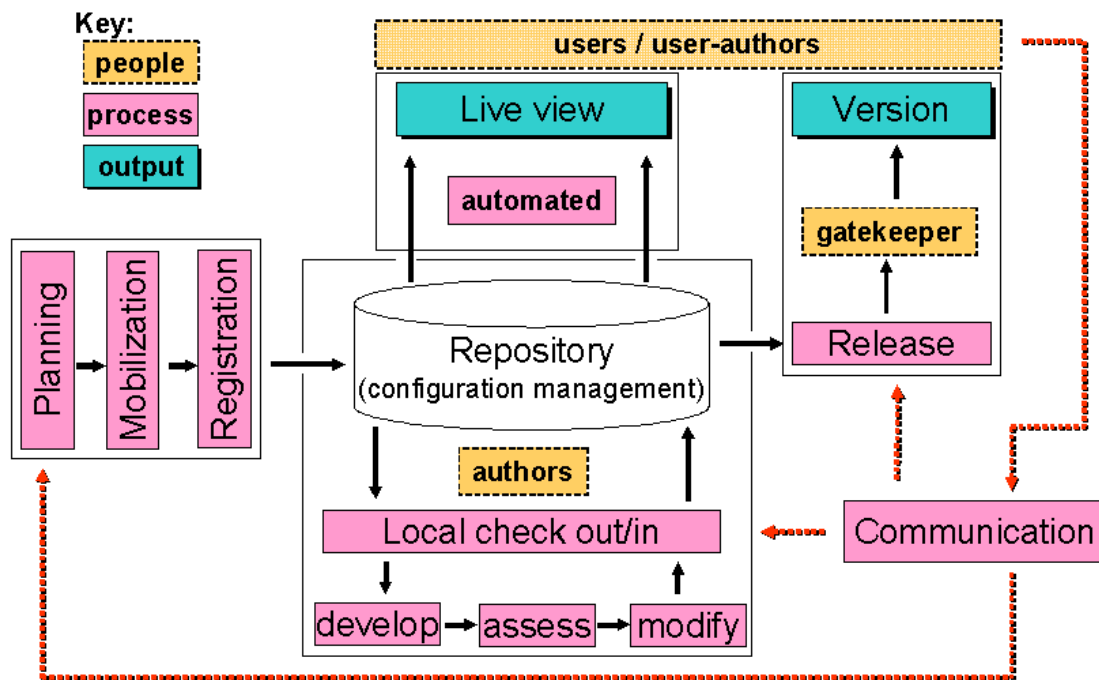


Figure 1: People and processes involved in open content development.

Within an OS Model, the participants involvement can be along a hierarchy:<sup>14</sup>

- Project Leaders
- Core Members

<sup>10</sup> Raymond E. *The Cathedral and the Bazaar: Musings on Linux and Open Source from an Accidental Revolutionary*, 1999, O'Reilly and Associates.

<sup>11</sup> Lakhani K.R., Wolf R. *BCG Hacker Survey*, 2002. Results are at [www.osdn.com/bcg](http://www.osdn.com/bcg). The research was undertaken in conjunction with the Boston Consulting Group.

<sup>12</sup> O'Reilly T. "Lessons from Open Source Software Development" *Communication of the ACM*, 1999, V42, No. 4, 33

<sup>13</sup> Keats D. "Collaborative development of open contents: A process model to unlock the potential for African universities" *FirstMonday*, 2003, V8, 2.

<sup>14</sup> Nakakoji K., Yamamoto Y., Nishinaka Y., Kishida K. and Ye Y. "Yoshiyuki "Evolution Patterns and Models" *Proceedings of the International Workshop on Principles of Software Evolution*, International Conference on Software Engineering, 2002, Orlando, Florida, pp. 76-85.

- Active Developers
- Peripheral Developers
- Bug Fixers
- Bug Reporters
- Readers
- Passive Users

With OS products maturing, that is having progressed well beyond the developmental phase that attracted initial involvement from interested developers and contributors, a number of issues have arisen. In the maturation phase, there is a requirement for some one to undertake the routine support needed. This support includes providing assistance where questions arise or problems are identified. One approach has been the emergence of users group and newsgroups where questions or issues can be posted. This approach is extremely effective where a mass of developers and other users exist. However, as no guarantee of support exists and in the case of OS Products entering the corporate world, this is a significant issue requiring consideration. Similarly, the individual involvement of staff, who may well be instrumental in having an OS product adapted within a organization, is a risk in itself, should that individual leave. Other issues that have been noted in the success of mature OS products include lack of:<sup>15</sup>

- User acceptance, due to relatively poor graphical user interfaces (GUI)
- Version control and fragmentation occurring – different versions being used in the same organization
- Understanding of the full lifecycle costs

As *The Economist* stated:

*The open-source method has vulnerabilities that must be overcome if it is to live up to its promise. For example, it lacks ways of ensuring quality and it is still working out better ways to handle intellectual property.*

*But the biggest worry is that the great benefit of the open-source approach is also its great undoing. Its advantage is that anyone can contribute; the drawback is that sometimes just about anyone does. This leaves projects open to abuse, either by well-meaning dilettantes or intentional disrupters. Constant self-policing is required to ensure its quality.<sup>16</sup>*

In the Australian context, adoption of OS has been conservative and arguably behind leading countries. Goode in a survey of 500 “top” Australia Firms<sup>17</sup> found that OS was rejected because:

- They could not identify any relevance to their businesses
- Perceptions of lack of reliable ongoing support
- High changeover and learning costs
- Beliefs of incompatibility with current technologies/software

In recognition of the risks associated with OS, organizations have entered the market on commercial terms ie undertaking customization for organizations or providing help desk type support.<sup>18</sup>

An even more pronounced trend relate to organizations establishing what they consider to be sustainable business models, the majority of which are based on promoting OS products, for free.<sup>19</sup> Whether this is in

---

<sup>15</sup> Kenwood C.A. "A Business Case Study of Open Source Software" The MITRE Corporation, a paper sponsored by US Army, MP 01B0000048, July 2001.

<sup>16</sup> *The Economist* "Open, but not as usual", 16 March 2006.

<sup>17</sup> Goode S. "Something for nothing: management rejection of open source software in Australia's top firms" *Information and Management*, July 2005, Volume 42, p. 669.

<sup>18</sup> For example, [http://software-support.gbdirect.co.uk/open\\_source\\_support\\_types.html](http://software-support.gbdirect.co.uk/open_source_support_types.html).

<sup>19</sup> See Keonig J., "Seven open source strategies for competitive advantage", *IT Managers' Journal*, [www.itmanagersjournal.com/management/04/05/10/2052216.shtml](http://www.itmanagersjournal.com/management/04/05/10/2052216.shtml), 14 May 2004.

fact “sustainable”, is a concern with some commentators, suggesting an “OS crash”,<sup>20</sup> whereas others suggest that the OS product development approach is the long term business model of the future.<sup>21</sup>

## Open Source Business Intelligence

In the last three years there has been an explosion of activity in OS BI products.

To assist in classifying the particular facets of BI, that are now available via OS products, the BI “stack” can be viewed at the following levels:

- Databases
- Extract, Transform, Load (ETL) – the process of creating the metalayer or data warehouse needed to support BI
- Online Analytical Processing (OLAP) – holding data for immediate processing where dimensional relationships exist
- Relational Reporting
- Data Mining
- Data Warehouse
- Integrated products combining several of the above levels

While the following is not definitive, it is indicative of the growth of Open Source BI products:

**Table 2. Open Source BI products<sup>22</sup>**

BI Stack	Products
Databases	MySQL – <a href="http://www.mysql.com">www.mysql.com</a> Bizgres – <a href="http://www.bizgres.org">www.bizgres.org</a> ExtenDB – <a href="http://www.extendb.com">www.extendb.com</a> DBDesigner – <a href="http://fabforce.net/dbdesigner4">fabforce.net/dbdesigner4</a> MonetDB - <a href="http://monetdb.cwi.nl">monetdb.cwi.nl</a> Postgresql – <a href="http://www.postgresql.org">www.postgresql.org</a>
ETL	Enhydra Octopus – <a href="http://octopus.objectweb.org">octopus.objectweb.org</a> Bee - <a href="http://bee.insightstrategy.cz/en/index.html">bee.insightstrategy.cz/en/index.html</a> Clover ETL – <a href="http://cloveret1.berlois.de">cloveret1.berlois.de</a> DataSift - <a href="http://www.datasift.org">www.datasift.org</a> KETTLE – <a href="http://www.kettle.be">www.kettle.be</a>
OLAP/Reporting	Mondrian – <a href="http://mondrian.sourceforge.net">mondrian.sourceforge.net</a> Jpivot – <a href="http://jpivot.sourceforge.net">jpivot.sourceforge.net</a> Palo – <a href="http://www.palo.net">www.palo.net</a> Jasper – <a href="http://jasperreports.sourceforge.net">jasperreports.sourceforge.net</a> Art – <a href="http://art.sourceforge.net">art.sourceforge.net</a> Open Reports – <a href="http://oreports.com">oreports.com</a> BIRT – <a href="http://eclipse.org/birt/">eclipse.org/birt/</a> XMLA – <a href="http://reportportal.com/download.asp">reportportal.com/download.asp</a>
Data Mining	Weka – <a href="http://www.cs.waikato.ac.nz/ml/weka">www.cs.waikato.ac.nz/ml/weka</a> YALE – <a href="http://rapid-i.com">rapid-i.com</a> R-Project – <a href="http://www.r-project.org">www.r-project.org</a>
Data Warehouse	Bizgres – <a href="http://www.bizgres.org">www.bizgres.org</a>
Integrated Products	Pentaho – <a href="http://www.pentaho.org">www.pentaho.org</a> GreenPlum – <a href="http://www.greenplum.com">www.greenplum.com</a> Breadboard – <a href="http://www.breadboardbi.com">www.breadboardbi.com</a>

<sup>20</sup> La Monica M., "Is open source a bubble ready to burst" *CNETNews.com*, 7 November 2005.

<sup>21</sup> Kawamoto D. & Olsen S. " Tech VIP's say future belongs to open source" *CNET News.com*, 21 July,2005.

<sup>22</sup> At the time of writing a number of the Products were growing and transforming into other areas of the BI Stack with some planning to become Integrated Products.

MarvelIT – [www.marvelit.com](http://www.marvelit.com)  
Noetix - [noetix.com](http://noetix.com)  
ObjectWeb – [www.objectweb.org](http://www.objectweb.org)  
DecisionStudio-Professional – [www.decisionstudio.com/products](http://www.decisionstudio.com/products)  
Megallanes – [jmagallanes.sourceforge.net](http://jmagallanes.sourceforge.net)  
OpenI – [www.openi.org](http://www.openi.org)  
Sprago BI – [spagobi.objectweb.org](http://spagobi.objectweb.org)

In November 2003, Hyperion Systems surveyed the availability of OS for BI – Data Warehousing and 7416 downloads of the only available OLAP server and associated reporting programs were noted.<sup>23</sup>

In a later survey between November 2005 and January 2006, Ventana Research surveyed 522 organizations concerning their OS BI plans.<sup>24</sup>

Of the 61% who responded, approximately half had revenue of more than \$100M (USD).

The key findings of the survey included:

- 62% were developing plans to implement OS BI, to varying degrees
- 20% has already implemented BI to some extent
- 18% had purchased support for their BI
- 12% had engaged consultants
- Deployment of OS BI to organisations with more than 1,000, were expected to increase by 200% to 300%, in the following 12 months
- Cost was the primary factor driving consideration of OS BI. The largest segment (48 percent) of respondents expected OS BI to be half the price of equivalent commercial BI software; 25% expected it to cost the same as commercial BI software.
- More than 50% of respondents that OS BI was sufficiently feature-rich to satisfy their requirements.
- Most of the respondents who were considering or using OS BI software already licensed commercial BI software.

Ventana's overall conclusions are that OS BI:

*adoption may be led by agile organizations that have fewer resources than larger companies and that open source BI is used for different purposes than is commercial BI, has lower cost and potentially fewer features, but is no less adequate for users.*<sup>25</sup>

A further indication of the OS BI momentum is the number of downloads. In May 2006, there were more than 40,000 OS OLAP downloads.<sup>26</sup>

In a recent update, Ventana surveyed 320 organisations and concluded that OS BI was gaining more momentum than previously estimated:

*Eight-three percent of organisations [of 320 surveyed] are considering, are in the process of deploying, or have already implemented open source BI" and "Only four percent of respondents said they would not deploy more open source BI in the future based on their current experience.*<sup>27</sup>

Even though OS BI has gained acceptance, there exists significant difference between Proprietary BI and OP BI:

---

<sup>23</sup> Poole J., Hyperion Systems, *Enterprise Data Forum*, 3-6 November 2003, Cherry Hill, Philadelphia.

<sup>24</sup> <http://www.ventanaresearch.com/oseabi/oseabi.aspx?id=1327>.

<sup>25</sup> [http://www.dmreview.com/article\\_sub.cfm?articleId=1061024](http://www.dmreview.com/article_sub.cfm?articleId=1061024)

<sup>26</sup> North K. *Open Source BI: The Great Leap Forward*, <http://www.sqlsummit.com/Trends/OpenSourceBI.htm>

<sup>27</sup> Gardner D.W. *Spending on business intelligence in IT departments is picking up steam, two market research reports say*, [Networkcomputing.com](http://www.networkcomputing.com) 4 May 2006

<http://www.networkcomputing.com/channels/enterpriseapps/showArticle.jhtml?articleID=187200358>

**Table 3. Comparing Proprietary and Open Source BI Software**

Area of Comparison	Proprietary BI	Open Source BI – (Supported Products)
License Costs (% of Project)	80%	20%
Implementation Cost (% of Project)	20%	80%
“Bells and Whistles”	Extensive	Basic to Moderate
Support	Standard level provided	Varies on the type of licence adopted
Full BI Stack?	Varies – some BI stacks, some best of breed	Varies – some BI stacks, some best of breed
Generalised view of Industry Commentators	Well developed and evolving	Work in Progress
Flexibility	Move towards consolidation BI Stacks	More accommodative to single BI stack products
Industry trend	Consolidation and move towards full BI Stacks	Expand and improve products
Integration with Enterprise Systems	Moderate to Complex, high costs	Relatively easy and can be done in house

Nevertheless, as an indication of the mainstream adoption and scalability of planned OS BI developments, the largest publicly declared not for profit project is the European Internet Accessibility Observatory (EIAO). EIAO is a planned data warehouse of Internet activity aimed at providing a basis for benchmarking, policymaking, research and actions to develop accessibility to Internet.<sup>28</sup> As such, it represents collaboration within the European Union.

## Overview of Business Intelligence and Reporting Tools Project (BIRT) and the Eclipse Foundation

To understand the genesis of BIRT is necessary to understand the Eclipse Foundation.

The Eclipse Foundation was effectively founded by IBM as a successor to its proprietary OS products WorkBench and VisualAge software. This software was effectively “donated” to the not for profit Eclipse Project, which now has companies such as IBM, Borland, Rational, Red Hat, Sybase, Fujitsu, Hitachi, Oracle, SAP, OMG, and Intel on its Board of trustees<sup>29</sup> and is now North American dominant JAVA development environment with around 45% - 65% of the market. Also currently:

- Around 20 Universities are actively developing the platform and many are supported via research grants from the Project<sup>30</sup>
- The active developer community is 1200 in over 60 countries

The stated aim of the Eclipse Foundation is:

*The Eclipse Foundation is a non-profit corporation formed to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services*

with the following key bylaw:

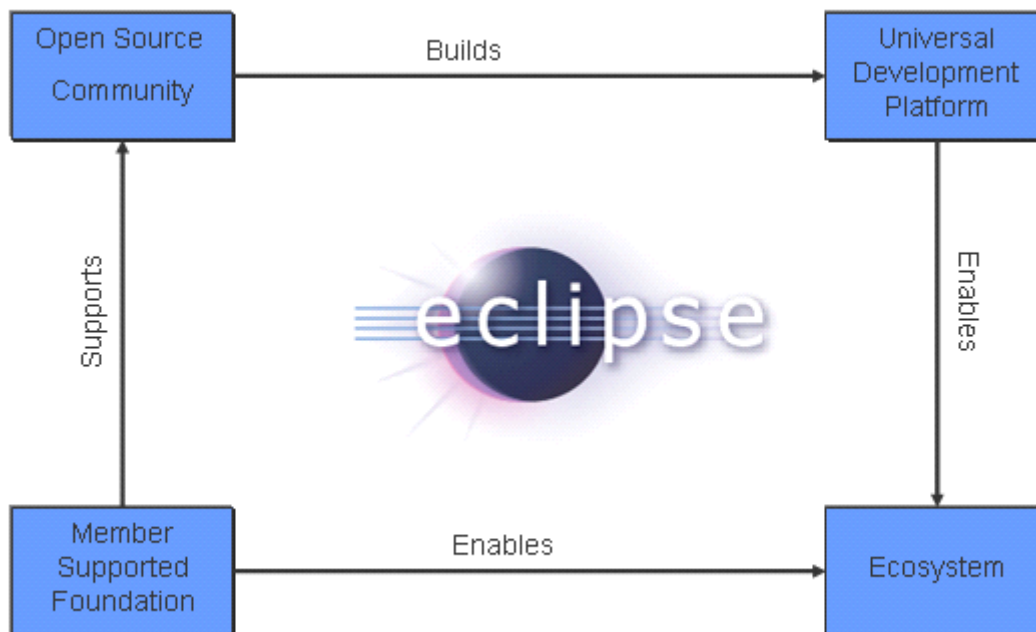
<sup>28</sup> <http://www.eiao.net/>

<sup>29</sup> Currently, there are 18 Board Members, who besides contributing a membership fee of \$250,000USD, are required to ensure/support on a scheduled roadmap, and contributing eight developers full time to work on open-source projects

<sup>30</sup> Universities include Oregon Health & Science University (USA), University of Aarhus (Denmark), Northeastern University (USA), Carleton University (Canada), University of British Columbia (Canada), University of Washington (USA), Ecoloes des Mines de Nates (France), Queensland University of Technology and Monash University (Australia).

The Eclipse technology is a vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools (the "Eclipse Platform"). Eclipse Platform tools are exemplary in that they verify the utility of the Eclipse frameworks, illustrate the appropriate use of those frameworks, and support the development and maintenance of the Eclipse Platform itself; Eclipse Platform tools are extensible in that their functionality is accessible via documented programmatic interfaces. The purpose of Eclipse Foundation Inc., (the "Eclipse Foundation"), is to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services."<sup>31</sup>

Diagrammatically, the Eclipse Foundation is structured<sup>32</sup> as follows:



Within this context, BIRT is one of many open source developments that have used the Eclipse Platform and is part of the "Ecosystem".

On 30 June 2006 the Eclipse Foundation released 10 open-source projects which sought to synchronize versions. In so doing, around seven million lines of code were released under the parent projects title of "Callisto." These include upgrades for the modelling framework, Web tools platform, visual editor, test and performance tools, and C and C ++ integrated development environment tools. Included in the release was a new version of BIRT.<sup>33</sup>

## BIRT

BIRT is a top-level Eclipse project that provides a reporting system for web applications, in particular but not limited to those based on Java and J2EE. The Project Management Committee that oversees BIRT comprises representatives from Actuate Corporation ("Actuate"), IBM, Innovent Solutions and Inetsoft Technology.

Actuate is an enterprise and BI reporting applications provider based in San Francisco, California provided the initial code contribution to the project and provides ongoing support in the form of dedicated project managers and developers.

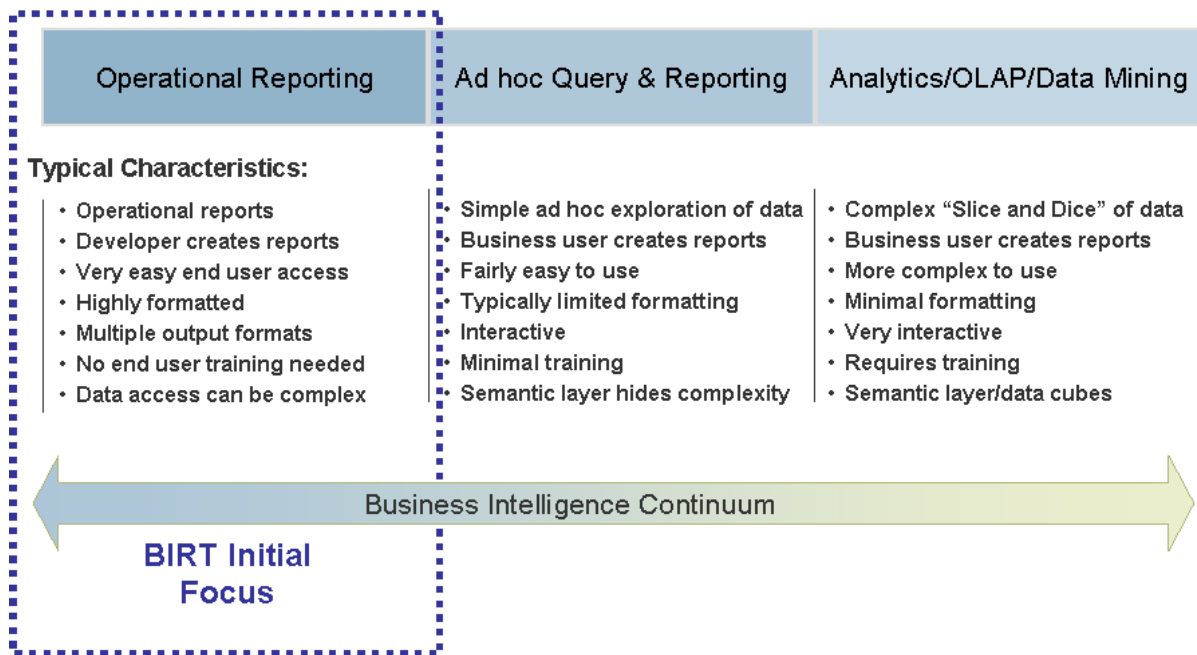
BIRT is 100% Java-based and Actuate's strategy is to appeal to the community of Java developers by providing an embeddable product for Java applications.

<sup>31</sup> [http://www.eclipse.org/org/councils/roadmap\\_v2\\_0/index.php](http://www.eclipse.org/org/councils/roadmap_v2_0/index.php)

<sup>32</sup> Actuate Corporation, 2006

<sup>33</sup> Sullivan L. "Eclipse Keeps Open-Source Timeline" *TechWeb*, 23 June 2006.

While the project plan for BIRT provides for a full BI tool set, the initial focus is on operational reporting as detailed in the diagram below.<sup>34</sup>



Ventana's market analysis concludes:

*BIRT and other open source BI projects (JasperReports, Pentaho) will likely make custom report generation for Java-based application development obsolete within the next few years. These technologies also will displace non-Java reporting technologies (Crystal) where 100 percent Java is a priority. For Java BI products that are not open source but for which commercial licenses (Jreports, Oracle Reports) are sold, there may be some negative impact. Whether BIRT supersedes JasperReports as the open source reporting technology leader remains to be seen...Open Source BI software is not yet broad enough in functionality to impact use of leading BI platforms (Actuate, Business Objects, Cognos, Hyperion, Information Builders, Microsoft, MicroStrategy, SAP) for non-Java-based analytic applications and for enterprise BI standardization initiatives.<sup>35</sup>*

Outside of the initial scope, but anticipated components that will be added to BIRT include Executive Information Systems (EIS), statistical analysis, modeling capabilities (what-if analysis), Data Mining Tools, Data Warehouse Modeling Tools, Extract Transform and Load (ETL) tools and Data Quality Tools.

It is the intention of the BIRT project to target three types of users application developers creating applications that need to embed data retrieval and presentation in their applications, report developers that are less skilled and use a GUI to create reports and business users that want to create or customize existing reports via a web interface.

The BIRT is extensible and supports extension points allowing developers to extend the product to meet additional needs.

The current BIRT release (v2.1.1.) comprises two main components:

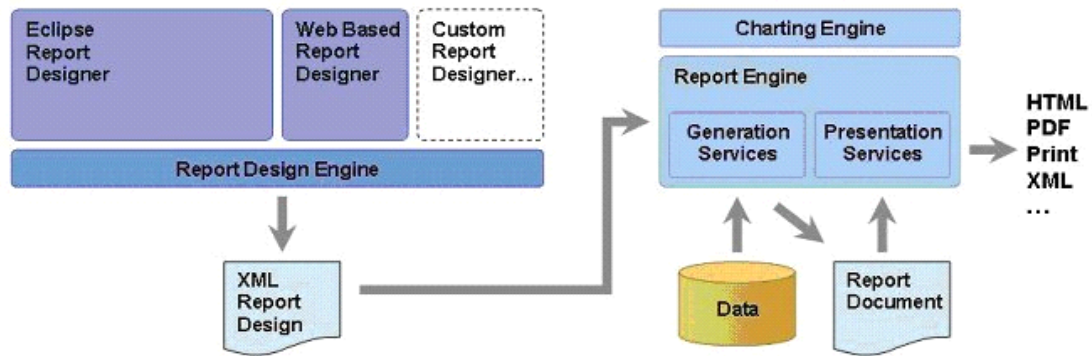
1. The report designer – for the development and testing of data sets, reports and templates
2. The runtime component – for implementation on a J2EE application server.

In addition BIRT offers a charting engine that allows charts to be added to an application. The diagram below represents the basic architecture of BIRT.<sup>36</sup>

<sup>34</sup> Actuate Corporation, 2006.

<sup>35</sup> Rogge E. "Review: Actuate's Open-Source Reporting Tool", *TechWeb*.

<sup>36</sup> Actuate Corporation., 2006.



## The BIRT Experience

The following section details ECU's experiences with BIRT from installation through production of a basic proof of concept report. The proof of concept was limited to the following scope:

- Installation
- Data source connection
- Report Design
- Customization of Report to University Standards
- Paramterisation of Reports
- Deployment and Execution on Web Server

### Installation

The standard method of installation requires the user to download the separate required Eclipse components from the web site and install each separately and configure the PC to run the Eclipse platform.

There are several components required:

- Eclipse Platform
- Graphic Editor Framework (GEF)
- Eclipse Modelling Framework (EMF)
- BIRT
- iText (jar files)
- JDBC Drivers (if required)
- Language Packs (if required)
- Java Run Edition (JRE)

When BIRT was initially released in 2005, Actuate offered an installer that packaged all components and automatically configured the users system for BIRT. At the time Actuate only offered this package for a fee (nominal) but has since changed its position and provides the package free of charge.

### Standalone

The standalone installation will install the report designer (RD) product that enables the production of reports and templates. RD is supported on recent versions of MS Windows (2000 on) and Linux (Redhat and SUSE). All Eclipse products including RD require the JRE.

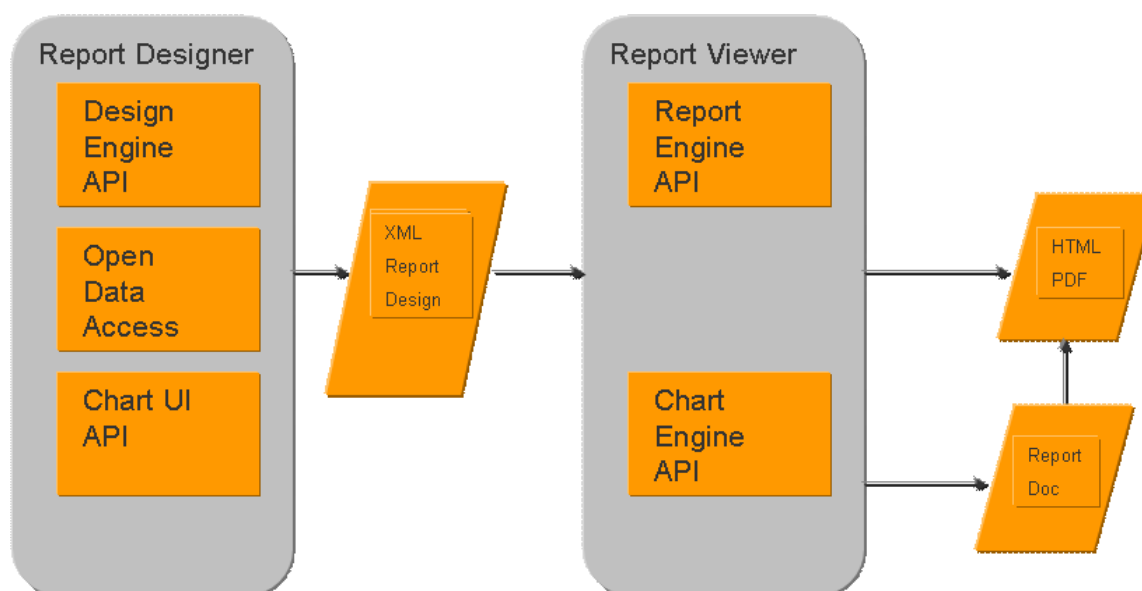
The installation of each individual component is time-consuming and requires better than basic computing skills and therefore restricts unskilled users. The best option is the installer however some configuration may still be required depending on the platform.

Beels observed with suspicion that:

BIRT 2.0 has a bit of a tricky install process, which makes one wonder if it's harder than needs be to "advertise" for the commercial Actuate version whose only additional virtue, other than support, is a standard complete installer.<sup>37</sup>

JDBC/ODBC drivers are included as standard and allow connection to ODBC sources. Support for text files is also standard. JDBC support for other data sources – MySQL and Derby require separate installation.

A stand alone report engine with test server is installed that generates reports for testing or distribution if needed. It was observed that the results from the test server did not always accurately reflect the results from the report engine running on an application server as detailed in the diagram below.<sup>38</sup>



The RD has an automated updates manager that checks for newer versions of all Eclipse components at start up and can be initiated manually. The update manager can prove somewhat confusing when asking the user about updates to Eclipse components not directly related to BIRT.

### *Application Server*

The application server installation installs the report engine on a J2EE server. This installation is beyond the scope of the basic user and requires some knowledge of application server principles.

The report engine is supported on J2EE application servers Tomcat v4.1.x and 5.5.x and JBOSS. ECU utilised Tomcat V5.5.17 during the proof of concept project.

### *Report Components*

The main components of a BIRT report are:

#### *Data*

BIRT provides support for access to data from databases, web services and Java objects. The most recent release added support for JDBC and XML support and the facility for a designer to write custom interfaces to data sources using the Open Data Access (ODA) framework. This allows the user to add runtime support for tabular data.

Multiple data sources are supported in a single report.

<sup>37</sup> Beels, C. "Java Feature - Business Intelligence and Reporting with BIRT A first experience account". *Java Developers Journal*. <http://www.amazon.com/BIRT-Guide-Reporting-Diana-Peh/dp/0321442598> , 17 April 2006.

<sup>38</sup> Actuate Corporation, 2006.

### *Data Transforms*

Transformation of data into the shape and order required for presentation. Data can be sorted, summarized, filtered and grouped as required. The report designer can choose to let the database (where applicable) do some of the transformation but BIRT performs transformations for non-database sources e.g. flat files and Java objects.

BIRT supports standard, assignment, comparison, computational and logical operators and has a reasonably broad set of built-in functions that provide aggregation, time and finance operations.

### *Business Logic*

The application of business rules to manipulate data into a form that meets business needs is the application of business logic. Report developers can build conditional logic into BIRT reports to change the data and presentation of a report.

BIRT supports JavaScript for the application of business logic and allows developers to call existing Java code if available.

### *Presentation*

BIRT supports some useful presentation facilities that enable sophisticated reports to be presented to the user. Beyond the simple presentation techniques (lists, tables etc.), BIRT supports interactive charts and images, page navigation, parameterization and report table of contents.

### ***Report Designer Interface***

The RD opens in the Eclipse Software Development Kit (SDK) and prompts the user to create or open an existing project. BIRT report designs are project based with the report design stored in XML.

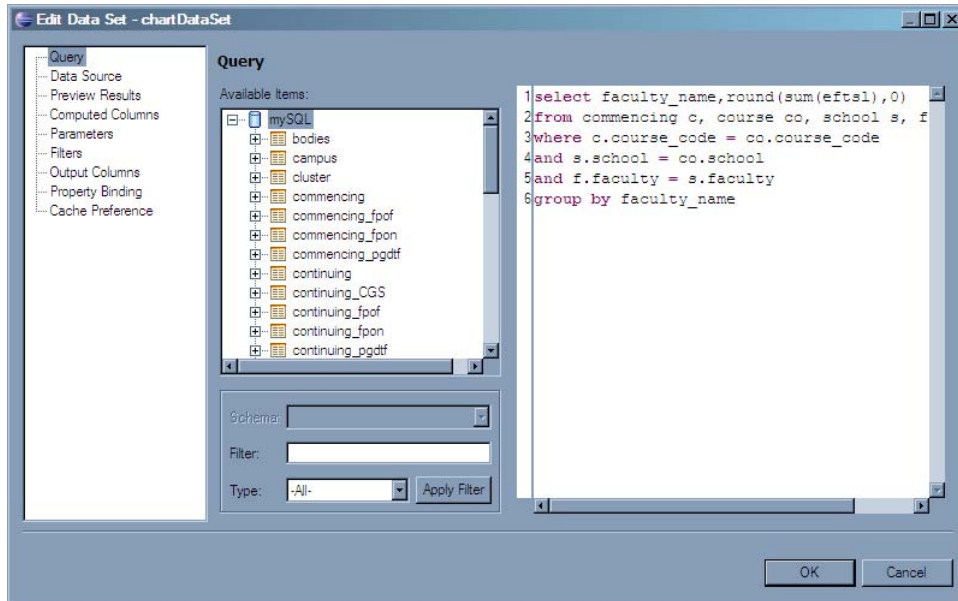
A project can contain multiple reports, templates and libraries. The benefit of the project based approach being that report components can be shared between reports in the same project. Libraries are a repository of reusable report components.

A report developer can use a library explorer to retrieve a component from a library or save a report component to the library. Library objects include visual report items, data sources, data sets, parameters, and translations. Multiple reports designs from multiple projects can be open concurrently in RD.

There are wizards to assist the user in creating basic reports and create default layouts. Wizards also exist for creating data sources and data sets. A BIRT data source represents a connection profile for a given database, text file or other data source and would normally contain locations, usernames and passwords if required.

A data set represents a view of the data returned from a given data source and can be retrieved via a stored procedure or a SQL query. The data source creation wizard walks the user through the processes involved in creating the data set. The data set editor invoked by the wizard allows the user to create a SQL query using a drag and drop interface and has a preview pane allowing confirmation of the data returned.

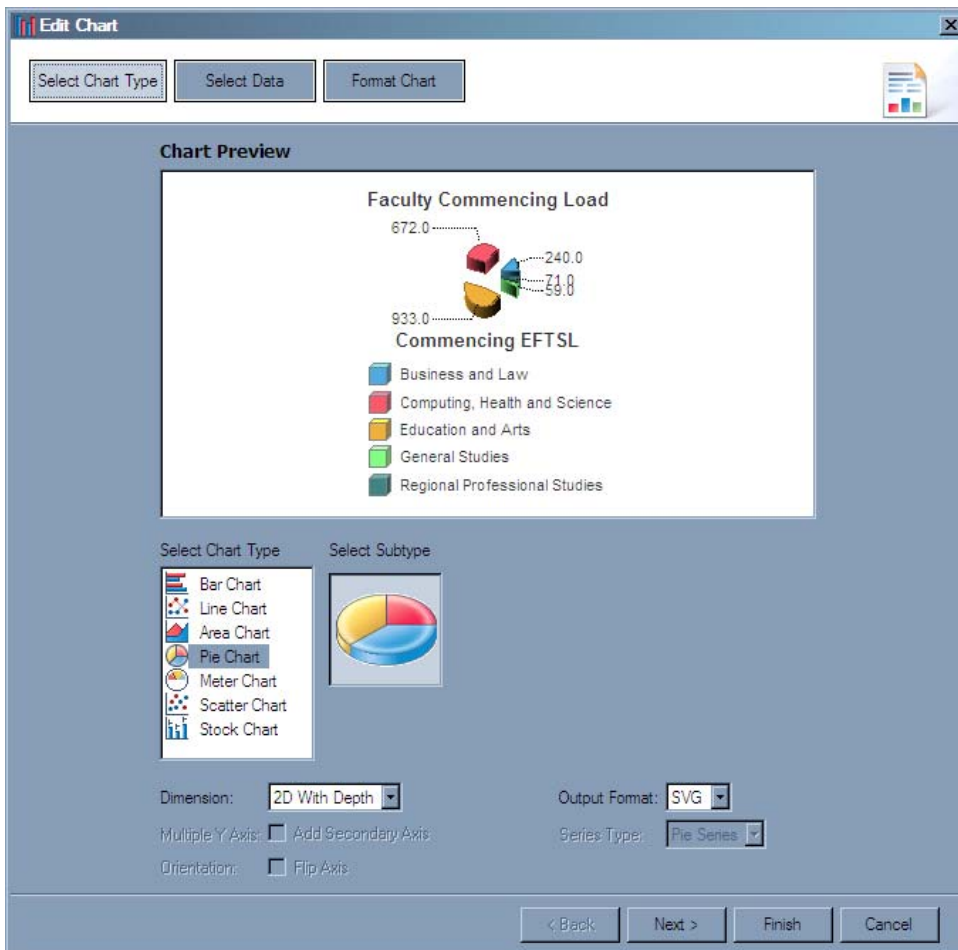
## BIRT Data Set Editor



A palette of report objects both structural and informational are available for the user to drag and drop into a report design. Each time an object from the palette is invoked a wizard is initiated to assist the user.

The chart editor which is a three stage process (style, data, formatting) is comprehensive but may be overwhelming for an inexperienced report designer due to the sheer number of variables. Dual y-axis graphs are now available after a recent enhancement.

## BIRT Chart Editor



The main frame of the RD interface contains the report design that can be viewed in either layout view (graphic structure) or preview. The user creates a report using the preferred view and can test the output. Data from the created data sets can be dragged and dropped or an expression editor can perform the same task.

Each report object structural or content can have events associated that are triggered based on report conditions and the actual report and the report scripts can be edited to manage report events.

The XML code that defines the report can be viewed and edited in the same pane if required.

### ***BIRT Report Types***

The BIRT project provides for the following report types:

- Lists – Creates a simple but flexible presentation of a set of data rows. This is the most basic of report types and is composed of a series of bands. Each band can contain any number of sections that print sequentially and if the report designer requires, conditionally.
- Charts – Chart types include bar, line, area, pie, scatter, stock and the now popular meter chart. Charts can be rendered all web compatible formats including scalable vector graphics (SVG) and support events allowing user interaction (e.g. drill-throughs).
- Crosstabs - Cross-tabulation or matrix displaying data in two dimensions. (Not yet implemented in the current release, but planned for a future release.)

Recent correspondence from Actuate indicated:

*The BIRT PMC (Project Management Committee) released the draft project plan for BIRT 2.2 last week [October 7, 2006]. A Dynamic Crosstab Report Item is listed as one of the projects. This is an often requested feature and it is great to see that it is slated for the next revision of BIRT. That said, static crosstab reports can certainly be built using the current version of BIRT. Additionally, a fully dynamic crosstab can be built with the help of the Design Engine (DE) API, although the DE API approach requires a little more work and entails building or modifying a report design on the fly.<sup>39</sup>*

- Letters & Documents – Formatted document or notices that can include text, formatting, lists, charts, etc.
- Compound Reports – A combination of all of the above.
- Dashboards – A variation on a compound report that provides a generally graphical view of the status of an organization. Usually measures performance against indicators.

### ***Proof of Concept Environment***

BIRT Report Designer

- Microsoft Windows XP SP2
- Pentium 4 3GHz
- 1 Gb RAM

Application Server (RD and Application Server Co-existing on Test Machine)

- Apache Tomcat v5.5.17

---

<sup>39</sup> Weathersby, J. "Building Cross-tabs in BIRT", *BIRT WORLD*, 12 October 2006.

## **BIRT Strengths**

### ***Price***

The Eclipse Common Public License allows for the use of BIRT (and Eclipse) free of charge.

*"The Eclipse Foundation makes available software, documentation, information and/or other materials for open source projects (collectively "content")."*<sup>40</sup>

It has been observed by Gonsalves that some sceptically view this apparent BIRT strength as a potential weakness.

*But companies planning to invest in open-source BI systems should be cautious. Critics say that vendors could use the license-free technology to attract customers, and then gradually lock them in with proprietary modules built on top of the platforms. That, however, may not be a bad thing, since the lower initial cost could give corporate customers a good opportunity to kick the tires before making a commitment.*

*But no matter how companies choose to approach open-source BI, the technology is making an impact that will be felt in the market for years to come.*<sup>41</sup>

### ***Flexibility***

The ability to add code triggered by events to change report behavior is a flexible and sophisticated facility. The availability of the software code allows for the integration of BIRT into existing systems and for commercial use if applicable.

The ease at which a corporate style sheet can be imported and applied makes report customization flexible. A RD user can quickly apply corporate styles to a new report.

### ***Ease of Deployment***

On the assumption that the user has a reasonable degree of IT skills, deployment of the RD is a relatively simple task. The deployment of the report viewer on a J2EE Application Server while non-trivial is not overly complicated. It was possible to deploy the report viewer to an application server and produce a custom report in less than half a working day.

### ***Parameter Driven Reporting***

Parameter driven reporting allows designers to create one report that satisfies the needs of many users. BIRT allows the creation of report parameters and when published on a web server.

List based parameters can be statically or dynamically populated. Dynamically populated list reduce report maintenance times and costs.

## **BIRT Weaknesses**

### ***Documentation***

This commentary could quite easily appear under the strengths heading however in the context of the proof-of-concept the following remarks relate to the 'field reference guide' or what a RD user would normally consider user documentation.

---

<sup>40</sup> Eclipse Foundation Software User Agreement, March 17, 2005.

<sup>41</sup> Gonsalves, A. "BI Gets A Boost From Open-Source Community". *TechWeb*, [http://www.intelligententerprise.com/blog/archives/2006/06/bi\\_gets\\_a\\_boost.html](http://www.intelligententerprise.com/blog/archives/2006/06/bi_gets_a_boost.html), 29 June 2006.

Actuate has integrated help documentation for BIRT that will assist the novice user in creating basic reports. Documentation on complex tasks is a little thin. Context sensitive “Cheat Sheets” assist with getting started with new tasks but consume some the work area. Nonetheless the cheat sheets are useful.

Actuate advises that two reference books relating to BIRT are in the final stages of production and should be available soon. Both books will be published under the Eclipse banner.

Outside of the scope of the proof of concept, technical documentation for the developer wishing to use or extend the BIRT API is comprehensive and can assist a technically oriented user in understanding the processed behind some of the RD functionality. This documentation appears to be implemented with JavaDoc.

### *Skill Set*

To be productive in creating report designs the BIRT user will require at least some working knowledge of SQL. In addition, with the report output being HTML based some knowledge of HTML and CSS would also be well-placed.

### *Consulting and Support Costs*

Actuate’s business model for BIRT deviates from the norm by not relying on licensing fees for revenue generation. In the case of BIRT, the model relies upon a groundswell of support resulting in wider-spreading adoption and revenue sourced from increasing consulting and support fees.

Actuate’s web pages relating to fees have changed several times since the release of BIRT but the most recent lists three types of support subscriptions, basic, silver and gold offering increasing levels of support respectively. Only the Gold subscription offers ongoing telephone/email support with a service level agreement (SLA).

### *Report Management*

There appears to be no mechanism for publishing reports to a web server other than manually copying to the required location on the server. Publishing and version control would be useful. This functionality is standard equivalent on commercial products.

### *“Bursting”*

“Bursting” refers to distribution of customized reports to different users based on user characteristics such as user organizational role or level. The content of the same report would vary when “bursted” based on the user characteristics.

Recent releases of equivalent reporting products both commercial and open source have report “bursting” capabilities. BIRT in its native form does not have this functionality.

### *Security*

Neither BIRT RD nor the Web Server runtime have native report security management. The user will need to implement security requirements as needed.

## **BIRT experiences at other Higher Education Institutions**

Higher education institutions are reasonably well represented in terms of activity in the BIRT project. Tsinghua University (Beijing, China) is on the official list of BIRT project contributors and provides feedback on requirements, design or testing.

A recent BIRT presentation by Actuate listed The University of Michigan and the University of Missouri as adopters of BIRT although no contact details for these institutions could be obtained.

A search of activity on BIRT newsgroups revealed several other US-based Higher Education Institutions active to varying degrees.

Contact was made with several the Institutions revealing that some institutions were merely testing BIRT's usefulness while others were implementing and integrating BIRT into production environments.

Of note is advice from the University of Indiana. The University of Indiana's Decisions Support Services business unit advised that BIRT will be implemented with its current reporting environment for its Decision Support System on 29 October 2006.

The University of Indiana advised that BIRT offered some excellent features and functions that it could not currently provide with its existing reporting environment. The University of Indiana was most impressed with BIRT's the ability to drill through to other reports with parameters, the ability to apply cascading style sheets, and robust charting and graphing.

No evidence of the use of BIRT in Australian Higher Education Institutions was observed.

## Potential Savings and Costs

The potential savings in ongoing license fees make BIRT very attractive to any organization considering implementing a BI or database reporting solution. Depending on the scope of the requirements, BIRT may represent an out-of-the-box solution.

The actual savings are difficult to quantify without comparing commercial contracts for an equivalent product and measuring the total cost of ownership (TCO) including internal and external administrative overheads. These costs vary drastically from site to site and based on the method of deployment.

In the worst case, if it was necessary to employ consulting for implementation and take up a support subscription the TCO must still be significantly lower than an equivalent commercial product.

The risk of course is that BIRT does not satisfy requirements and the opportunity cost related to an unsuccessful implementation. A robust proof of concept and testing project should be undertaken to mitigate this risk.

Support contracts with commercial software vendors are generally in the range of 20 – 25% of license fees and allow for support and software upgrades. If in-house expertise exists then savings in this area will be significant.

## Conclusions

It's hard to ignore BIRT when considering a database reporting solution however, in its current form its functionality leans heavily toward that of a reporting solutions rather than a true BI tool. The planned enhancements when implemented should change this and then BIRT will become a true BI tool.

Nonetheless, BIRT is still great value for developers and report producers who are experiencing limitations with their current software solutions. Potential users should not be discouraged by the Java implementation of BIRT. During the proof-of-concept it was sociable and no unexpected failures were encountered.

Beels sums up BIRT's potentials:

*"BIRT isn't quite at the same level as a commercial solution like Business Objects, Microsoft Reporting Services, or Actuate, but it's encouraging how much one can do with this initial version. With sufficient grassroots support from loyal users, BIRT should become one of the most important and visible elements of the Eclipse suite, and rightly so."<sup>42</sup>*

---

<sup>42</sup> Beels, C. "Java Feature - Business Intelligence and Reporting with BIRT. A first experience account". *Java Developers Journal*. 17 April 2006.