



Open Source Data Warehousing

Mark Madsen, TDWI Chapter Meeting – Jan. 23, 2008

www.ThirdNature.net



Attribution-NonCommercial-No Derivative
<http://creativecommons.org/licenses/by-nc-nd/3.0/us/>

New York, January 23, 2008

Open source software (OSS) has become a force in the commercial software industry. Data warehousing is not immune to the impact of open source, with developments in the past year affecting a range of different market segments. We're still in the early adoption phase for many of the open source business intelligence (BI) technologies, but some are mature enough to be considered. IT organizations are challenged with sorting through OSS to measure the risks, measure the rewards and decide what is worth evaluating.

This session will review some existing theory and research on technology adoption to help frame the open source discussion, then discuss the state of OSS projects that affect the BI and DW market. During this session we will cover:

Current thought on open source in IT

Open source product challenges and advantages

Reasons for considering or not considering open source for BI projects

Open source alternatives to proprietary software

Where We're Going



- Some definitions
- Some history
- Some theory
- Projects
- Adoption
- Practices and policies

Why are we talking about open source? It's a change in the global software markets, not just in data warehouse-related software, but everywhere.

BI is one of the first major business application categories to be affected (as opposed to system, infrastructure or developer technology categories).

Proprietary Software

Software under a license that provides limited usage rights only, provided in binary format.

Open Source Software (OSS)

Software under a license that allows acquisition, modification and redistribution.

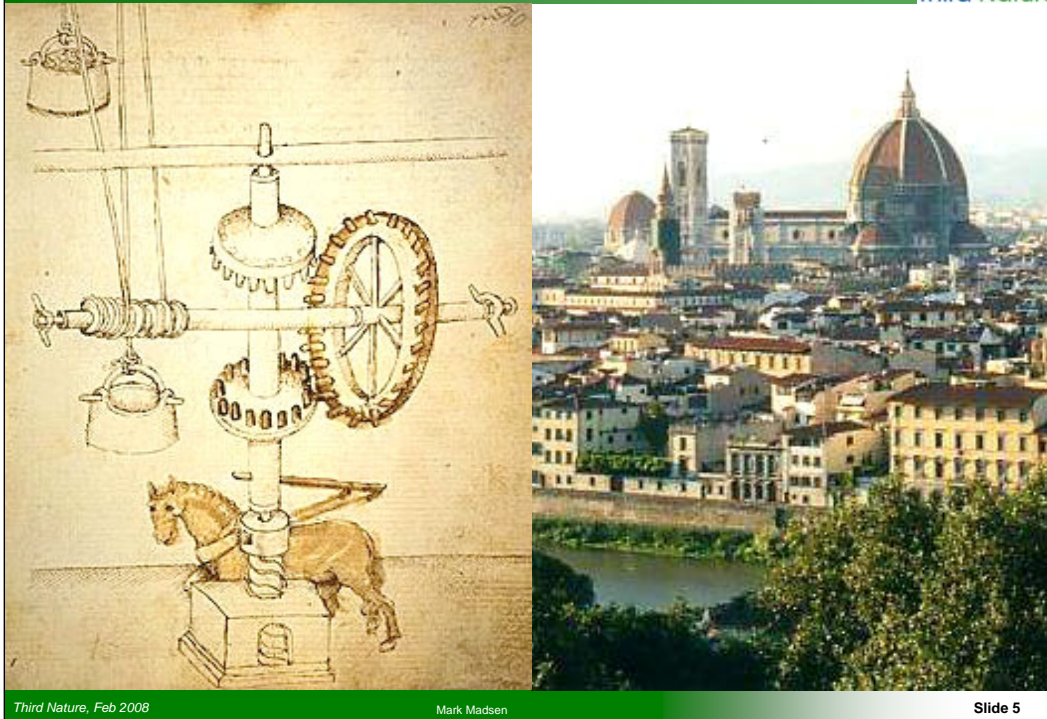
Freeware

Software that does not have licensing limitations, generally distributed in binary format. *Not* the same as open source.

Fauxpen Source

Something that's been appearing with greater frequency as open source has become more popular with proprietary vendors.

The First Recorded Patent



Third Nature, Feb 2008

Mark Madsen

Slide 5

A patent is really a grant by the government of a monopoly for some period of time. This first patent is not all that different from the intellectual property laws used for software now.

The Magnificent and Powerful Lords, Lords Magistrate, and Standard Bearer of Justice:

Considering that the admirable Filippo Brunelleschi, a man of the most perspicacious intellect, industry, and invention, citizen of Florence, has invented some machine or kind of ship, by means of which he thinks he can easily, at any time, bring in any merchandise and load on the river Arno and on any other river or water, for less money than usual, and with several other benefits to merchants and others, and that he refuses to make such machine available to the public, in order that the fruit of his genius and skill may not be reaped by another without his will and consent; and that, if he enjoyed some prerogative concerning this, he would open up what he is hiding and would disclose it to all;

And desiring that this matter, so withheld and hidden without fruit, shall be brought to light to be of profit to both said Filippo and our whole country and others, and that some privilege be created for said Filippo as hereinafter described, so that he may be animated more fervently to even higher pursuits and stimulated to more subtle investigations, they deliberated on 19 June 1421;

That no person alive, wherever born and of whatever status, dignity, quality, and grade, shall dare or presume, within three years next following from the day when the present provision has been approved in the Council of Florence, to commit any of the following acts on the river Arno, any other river, stagnant water, swamp, or water running or existing in the territory of Florence: to have, hold, or use in any manner, be it newly invented or made new in form, a machine or ship or other instrument designed to import or ship or transport on water any merchandise or any things or goods, except such ship or machine or instrument as they may have used until now for similar operations, or to ship or transport, or to have shipped or transported, any merchandise or goods on ships, machines, or instruments for water transport other than such as were familiar and usual until now, and further that any such new or newly shaped machine, etc. shall be burned;

Provided however that the foregoing shall not be held to cover, and shall not apply to, any newly invented or newly shaped machine, etc. designed to ship, transport or travel on water, which may be made by Filippo Brunelleschi or with his will and consent; also, than any merchandise, things, or goods which may be shipped with such newly invented ships, within three years next following, shall be free from imposition, requirement, or levy of any new tax not previously imposed.

The First Monopoly



Third Nature, Feb 2008

Mark Madsen

Slide 6

“Five hundred years ago, beset by spies, glassmakers on Murano, a small island in the Venetian lagoon, claimed that they had solved an ancient riddle: They perfected the process of manufacturing the world's first absolutely pure, clear, and uncolored glass. This was a bold statement in 1503. The glassmakers also stated that they could produce this glass in large, thin sheets free of imperfections. The announcement convulsed their competitors and began a 200-year monopoly that may still be the greatest monopoly on a luxury product that Europe has ever experienced. So it is no surprise that the Venetian doges immediately limited access to the island and implemented drastic countermeasures, declaring that anyone divulging the secret or defecting to a foreign glassworks would be hunted down and killed.”

The Origin of Copyright



- 1556: The Worshipful Company of Stationers and Newspaper Makers is granted a Royal Charter, giving it a monopoly over the publishing industry until ...
- 1710: “An Act for the Encouragement of Learning, by vesting the Copies of Printed Books in the Authors or purchasers of such Copies, during the Times therein mentioned”, otherwise known as the Statute of Anne, put the rights into the hands of authors

There's plenty more, but that summarizes it well enough. Authors own the copyright to their work. In the US, this is part of the constitution, and in general it means that authors are granted a limited monopoly over their works. With US and European copyright and patent laws getting out of hand, “limited” can now mean over a hundred years.

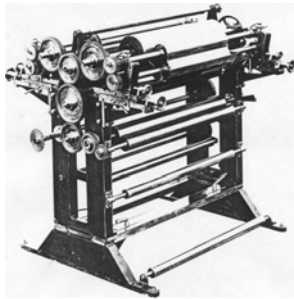
Software can be protected by both patents and copyright.

Copyright Continued to Evolve



Let's look at the one of the big areas that's driven copyright law.
This was the music industry in the 1700s, living largely on government aid.
Performance is controlled since you need resources to perform it.
The "industry" is made up of composers and performers.

Innovation Happened



5288

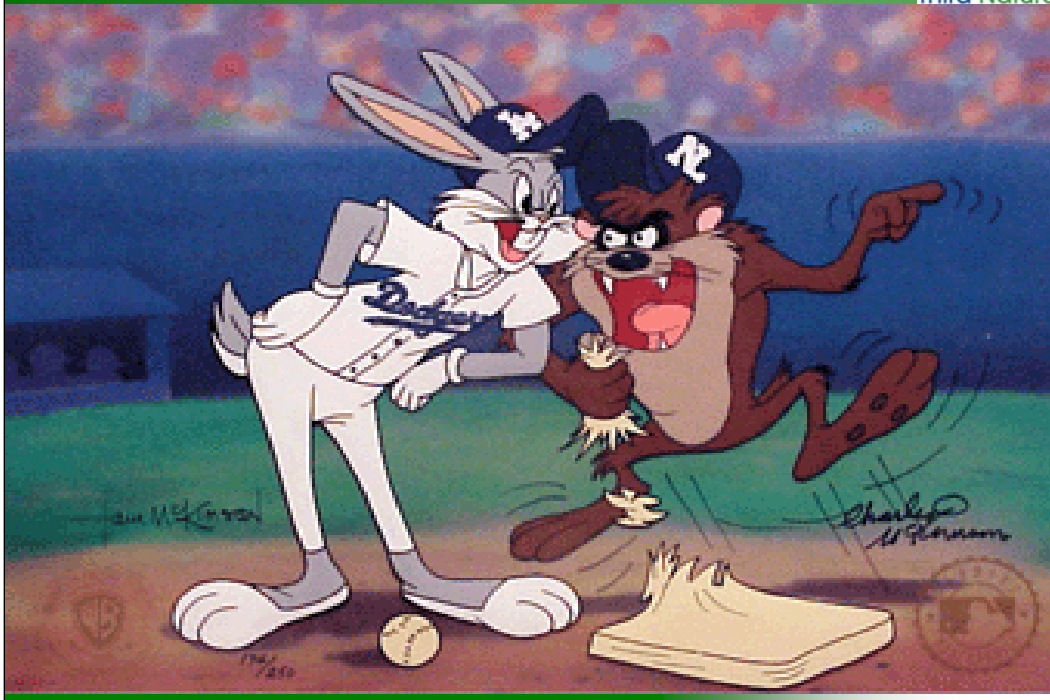


Innovation: mass printing, explosion of sheet music

What happened: sheet music publishers became the industry, sales of sheet music is the profit center, explosion of performers.

Only way to hear music was to go see a show or make your own, both of which profited the sheet music industry.

People Cried Foul



Third Nature, Feb 2008

Mark Madsen

Slide 10

Authors/composers wanted fair compensation, which they (mostly) got thanks to...

Lawmakers Intervened



Third Nature, Feb 2008

Mark Madsen

Slide 11

Lawmakers rewrote the laws to make them fairer and adjust the rules of the game, generally favoring the incumbent. Law always works like that except maybe in the Dosadi universe. If you don't get that reference then you need to read some dystopian fiction from the 70s.

Result



More Innovation



Computer + Code = Executable

This lasted until the phonograph came along, and with it came the devil's creation, records.

Buy the sheet music, record the record, sell many copies of the performance.

You can see where this will lead in the future.

You can also see why music and software have so much in common. Between looms and record players, this is where it all started.

More Complaint



Industry cried foul.

Sousa: they are stealing our music, they are killing the music industry

"them": but we bought the music

Was this legal? They had to figure this out, so...

The real Sousa quote is “

“If these infernal talking machines are allowed to continue we won’t have a voice box left in America. We will lose our voice boxes as we lost our tails when we came down out of the trees.” He was arguing against recording technology because it would move the focus of profit from the sheet music industry to the recording industry. RIAA used to be a bunch of pirates pillaging the sheet music industry. Now they’re the establishment fighting the same rearguard action. They had a good 100 year run (Sousa made that statement in 1908). Time to stop fighting creative destruction and evolve or die.

More Intervention



Third Nature, Feb 2008

Mark Madsen

Slide 15

Lawmakers convened.

Copyright law was changed to address the technological evolution and a new industry was born

More Results





Then along came radio.

Foul!



Industry cried foul.

Had an effect on killing the vaudeville theater industry as well

publishers: they are stealing our records, they are killing our industry

them: but we bought the records

the previous generation of pirates complained about the new pirates, tried to shut down radio

Intervention



Third Nature, Feb 2008

Mark Madsen

Slide 19

So copyright law was changed to address the technological evolution.

Result



An even more profitable industry evolves, and incidentally doesn't kill radio.

After Each Revolution, the Old Pirates Become the New Establishment



Then the VCR showed up. The blizzard of money around these guys wasn't enough.

Also on the scene were the Sony Walkman, and later the portable CD player.

Each time a technology industry comes along with unanticipated and unknown revenue potential and business models, the established industry tries to maintain a monopoly.

Foul!



Each foul is made out to be bigger than the last.

The entertainment industry tried to prevent all these devices (and the audio and video cassettes) through legal action.

Lawsuits against consumer electronics firms, cassette manufacturers, VCR and media manufacturers.

The TV example:

cable: they are stealing our shows, we need to stop this

them: it's our TV, we'll decide when we want to watch a program or skip over adds

Choice quote: Jack Valenti - "VCRs are to the entertainment industry what the Boston strangler is to a woman at home alone."

The Supremes!



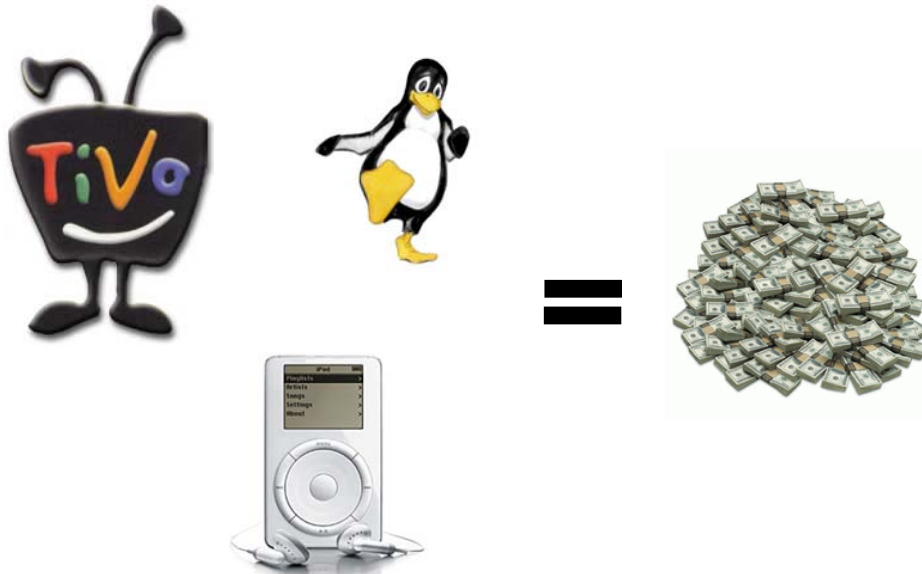
Lawmakers again involved. The US supreme court finally gets involved with the famous betamax decision, paving the way for the consumer electronics industry. International law largely follows US law to this point.

Re\$ult



The betamax supreme court ruling in 1984, paved the way for an entire multi-billion dollar industry that evolved from this technology change, so again the established industry was wrong (it always is).

What We Haven't Learned



But the old pirates are always replaced by new pirates.

Each time a technology industry comes along with unanticipated and unknown revenue potential and business models that threaten the established industry, the established industry tries to maintain a monopoly using patents and copyrights and whatever else they can buy.

We're going through this now with the latest audio and video formats and devices, only this time laws have been going in the wrong direction thanks to large sums of money and corrupt politicians with a short term viewpoint. The above are threatened, have been threatened, or have been co-opted.

Open source is about free market economics and what happens in to products that are perfect commodities.

Doomed industries with fat wallets are dangerous to all of us, and the current industry is worse than ever, largely because it is using money and politicians to force stasis.

John Philip Sousa was arguing against recording technology because it would move the focus of profit from the sheet music industry to the recording industry. He made this oddly familiar argument exactly 100 years ago in 1908.

RIAA used to be a bunch of pirates pillaging the sheet music industry. Now they're the establishment fighting the same rearguard action. They had a good 100 year run. Time to stop fighting creative destruction and evolve or die.

So, What is Open Source?



The media covering the software industry promotes the “free” aspect of open source software, touting it as the low cost alternative. Open source isn’t just software that costs nothing.

Even so, this is the biggest reason most companies try open source today.

What is Open Source?



Third Nature, Feb 2008

Mark Madsen

Slide 27

Think about software as research. Research is built on what came before, and makes what came before better (or invalidates it).

What is Open Source?



This is the IT analyst / late adopter view.

What is Open Source?



Freedom to use, to modify, to distribute. Freedom to tinker. Freedom from constraints.

What is Commercial Software, Really?



Third Nature, Feb 2008

Mark Madsen

Slide 30

What do you get when you buy commercial software?

The best definition: the Open Source Initiative,
<http://www.opensource.org/docs/definition.php>

What Makes Software Open Source?



**Academic
Licenses**

**Reciprocal
Licenses**

**Freeware
Licenses**

**Source Code
Licenses**

**Commercial
Licenses**

The fuzzy dividing
line between open
and closed source



More freedom

Less freedom

Open source software is no different than commercial software, so what makes open source software open source? One thing: the license.

Open Source is an evolution of a legal system that was founded 500 years ago, to meet software market demands and to address shortcomings in the system that prevent people (and businesses) from doing what they want to do.

It's also a rediscovery of the craft and guild model of collaborative development, the roots of software development.

A Little About Open Source Licenses



- Open Source licenses are about intent
 - Use the software for any purpose
 - Make and distribute royalty-free copies
 - Modify or extend the software and distribute it without payment of royalties
 - Access the source code
 - Combine the software with other software
- Academic Licenses
- Reciprocal Licenses



The key element to open source licenses is the intent. Unlike commercial license which restrict what you can do, OSS licenses intend for you to do what you want with the software. Most have the requirement that you contribute your modifications if you are making changes with the intention of distributing your own version.

Academic licenses give nearly complete freedom to distribute, modify or commercialize software. BSD, MIT Athena are examples.

Reciprocal licenses (the GPL is the most well-known) provide the same freedoms with one restriction: a derivative work must be licensed under the same terms as the original work.

OSI has a list of reviewed licenses. A company with one of these can be labeled "OSI Certified Open Source Software". Check opensource.org to see if the license for the software you are considering meets their requirements.

Complaints About Legal Issues



Open Source licenses are confusing

- Maybe if you have not read your commercial software license.

There are too many Open Source licenses

- Have you read your commercial software licenses?

Indemnification is a problem

- Are you sure you read your commercial software licenses?

The Open Software Initiative reads licenses so you don't have to.

This is more perception than reality. The SCO Linux lawsuit scared people, but then nothing happened. JBoss, Red Hat, Novell, IBM, HP all stepped in to indemnify their customers. SCO tried to sue some end-user companies but never won a case. In the US it's getting better, with a supreme court ruling that makes patent trolling more difficult.

If you are a user of software with no intention of distributing modifications or new products using OSS, don't worry too much about this as a risk. If you are planning to build and release software then you have a lot to think about.

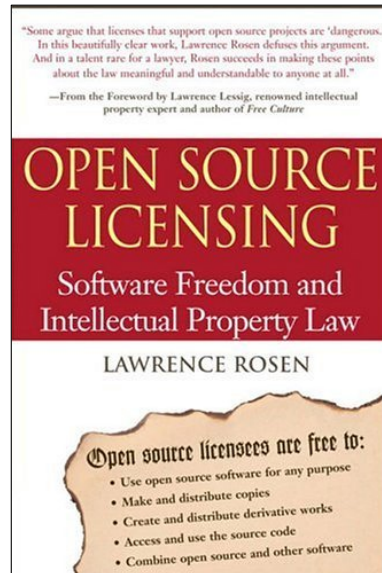
OSS licenses are often shorter than their commercial brethren. Makes it easier to read them.

Most contracts have limited indemnity clauses. Check the shrinkwrap licenses that often accompany enterprise software media some time.

Many commercial products embed OSS. Have you looked at what comes bundled with products like Business Objects. Or your cell phone?

The OSI reviews OSS licenses and certifies their goodness or badness, so you can check an OSS license against their list.

If You Want to Learn More



This is the book you want to read if you're going to be creating or modifying *and redistributing* open source software. If you're using OSS for your own purposes, you might consider glancing through the license that comes with the software. You're generally free to do what you want. The only time to take care is when dealing with some of the newer fauxopensource companies that claim to be open source but don't following the intentions described earlier.

Open Source Isn't Just Software



The innovation of open source isn't the software. The licensing is a legal hack with consequences to software markets.

- A non-proprietary product model
 - The license means you give all or most of it away
- Oriented more heavily around services
 - You make all or most of your money by providing services, and benefit from the enhancements and fixes provided by the users of the software
- Built and supported by a community of contributors
 - No community = no software

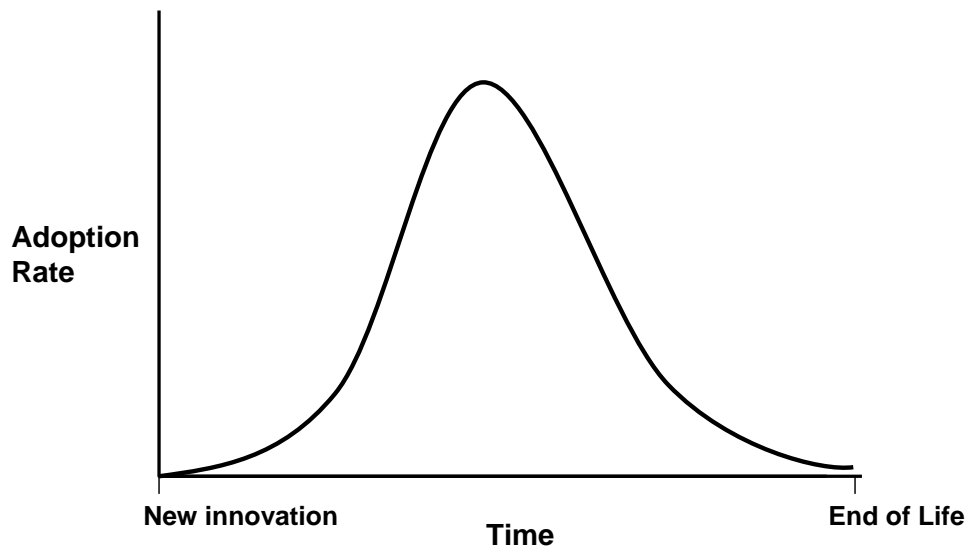
OSS is really about innovation and commoditization.

Open source isn't just about software. The licensing allows for and encourages different things than closed source, but the roots go back a long way into computing history and architectures.

In the end, open source is part of a process of commoditization and innovation which.

The key elements are social: collaboration in the community, and a licensing model requiring that everyone share.

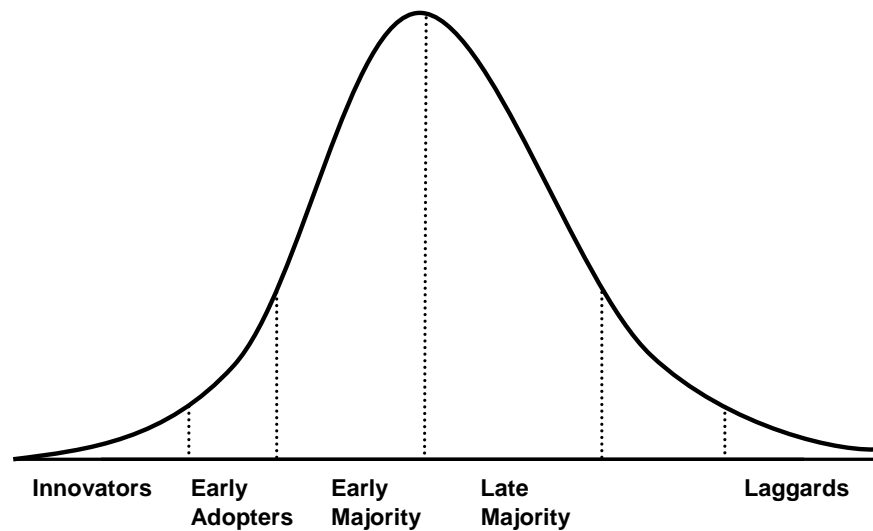
Innovation Adoption Theory



Diffusion of innovations theory was defined by Everett Rogers in a 1962 book which by an amazing coincidence was titled “Diffusion of Innovations.”.

He described the method by which people adopt new things, and showed that the rate at which adoption takes off and the rate at which later growth occurs define a bell curve for cumulative adoption of a particular innovation, with the slope of the head and tail dictated by the take off and later growth rates.

Adopter Categories



He divided the people who adopt innovations into categories, and determined the average population percentages to be:

innovators 2% - venturesome, educated, multiple sources of information, greater propensity to take risk

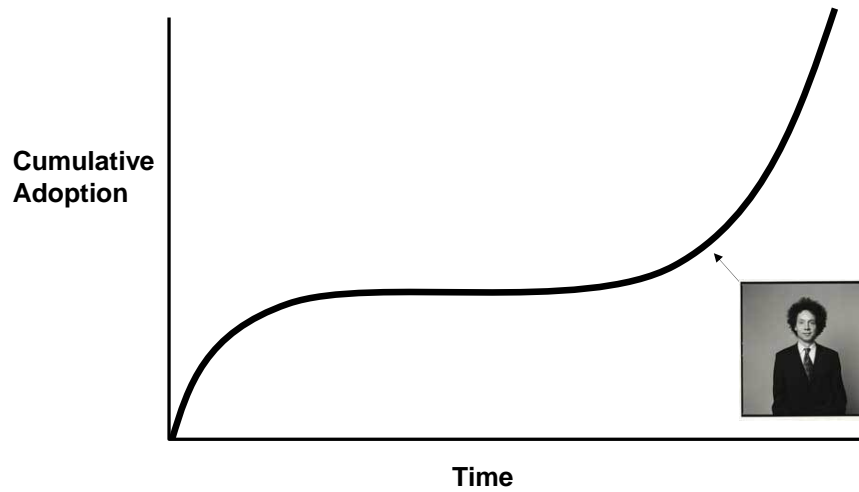
early adopters 15% - social leaders, popular, educated

early majority 34% - deliberate, many informal social contacts

late majority 34% - skeptical, traditional, lower socio-economic status

laggards 15% - neighbors and friends are main info sources, fear of debt

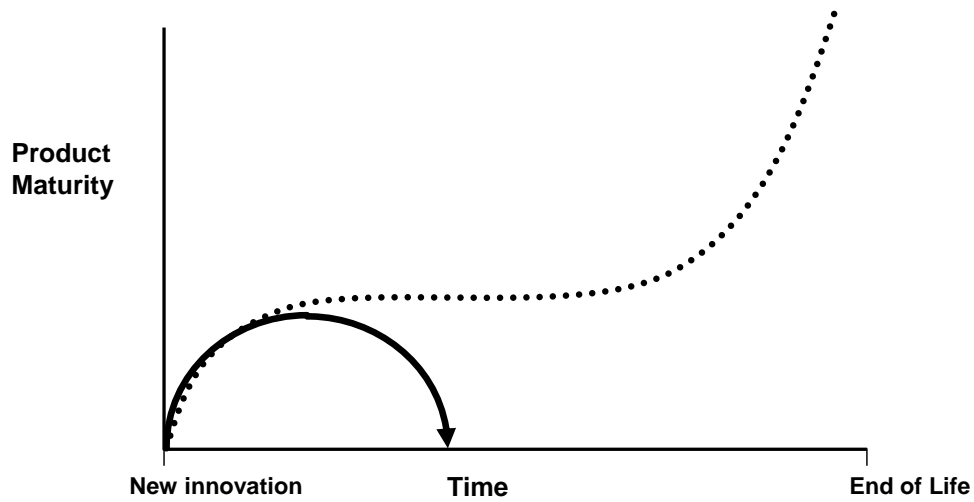
Market Adoption



Another useful graph is the market curve, showing cumulative adoption over time until the market is saturated.

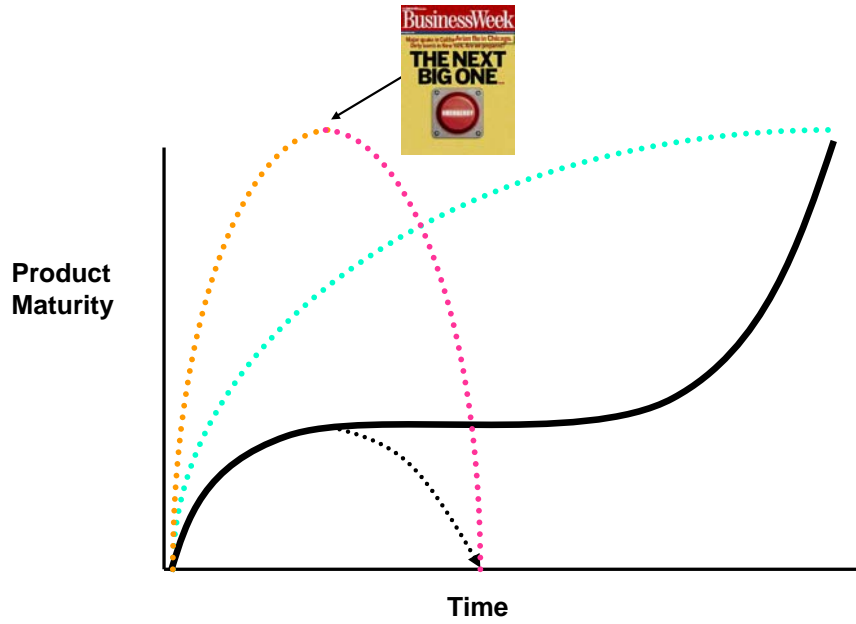
An interesting thing is the leveling off of the rate, before it hits the asymptotic part of the adoption curve. Moore has a lot to say about this.

Some Ideas Aren't That Good



If we look at the curve as not just cumulative adoption, but product maturity, it becomes more interesting. It's really the curve of *successful* technologies that reach maturity.

Curves Can Explain a Lot

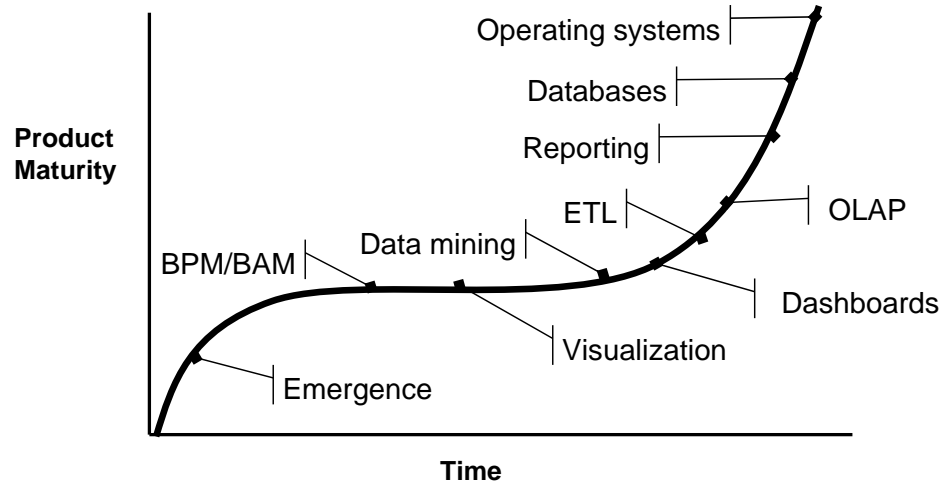


There are some other curves that can be superimposed on this curve.

Describing Technology Markets



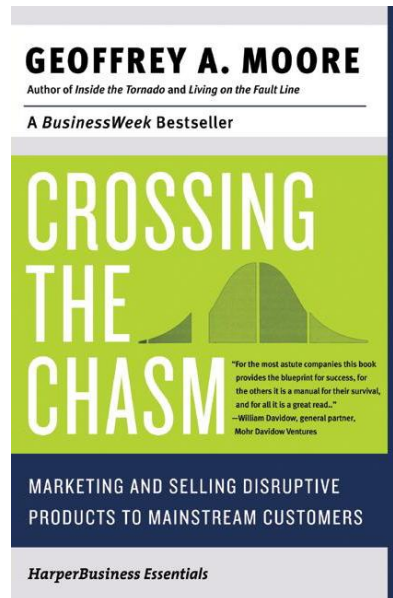
The data warehousing and business intelligence market can be described by the same curve, with different component technologies at different points along that curve.



The adoption/maturity curve applies to a single technology. The flat part is the chasm, where technologies may disappear, or languish, or cross and lift off.

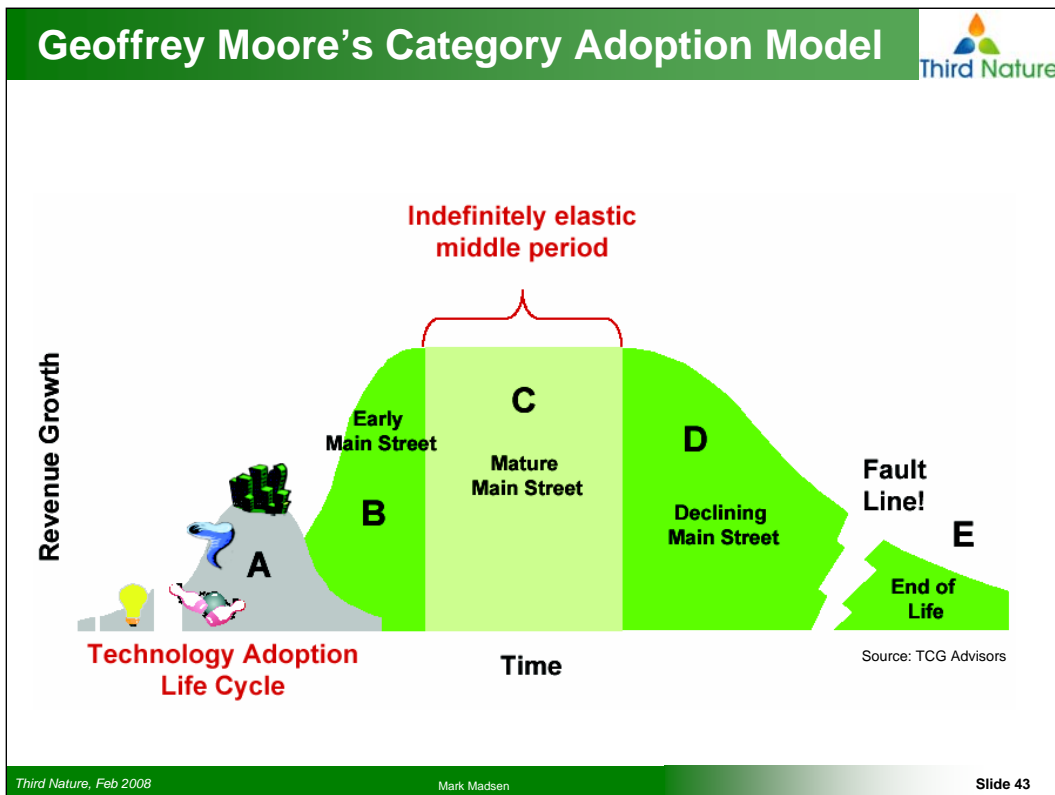
Data warehousing and business intelligence isn't a single technology, it's a collection of different technologies at different levels of maturity. Many of the basic technologies are mature, providing entry for OSS alternatives.

Crossing the Chasm



Moore studied innovation in the technology industries extensively and extended the model. His first book, “Crossing the Chasm”, was published in 1991 and became the bible of technology marketing strategy.

Moore's key insights are that the groups adopt innovations for different reasons, and these adopters require different segments and strategies in order to make a technology successful, thus dictating how companies providing technology should manage themselves at different points in the lifecycle of their markets.



Two models here: the chasm model and the category model.

He gave new names to the adopters:

Technology enthusiasts – committed to the technology assuming it will improve their lives; the drawback is that techies typically have no money. With open source that's not a problem.

Visionaries – people who want to use discontinuous innovations to make significant changes and gain an advantage over other organizations

Pragmatists – neutral on technology and look for a proven track record of productivity or reliability, generally from the market leader

Conservatives – cautious, price-sensitive, only undertake changes when required

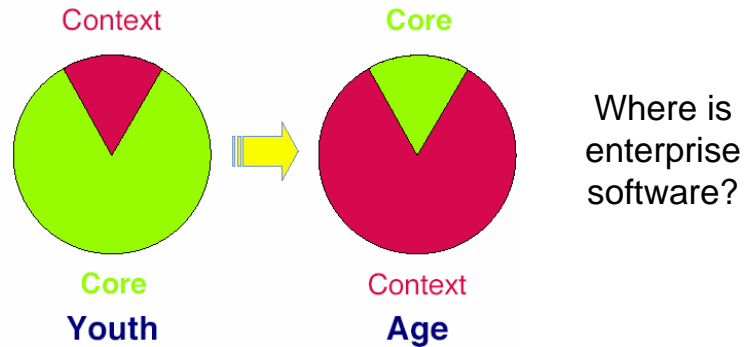
Skeptics – only change when it's absolutely required

Something Else Moore Talks About



Core: Any process that contributes directly to sustainable differentiation leading to competitive advantage in target markets.

Context: All other processes required to fulfill commitments to one or more stakeholders.



Source: TCG Advisors

Based on these concepts, you could conclude that:

Since OSS adoption is reliant on/part of the commoditization of software, and looking back at Christensen's theories, it is commoditizing based on maturing architectures, that OSS is really working to replace the software that makes up a large part of your context.

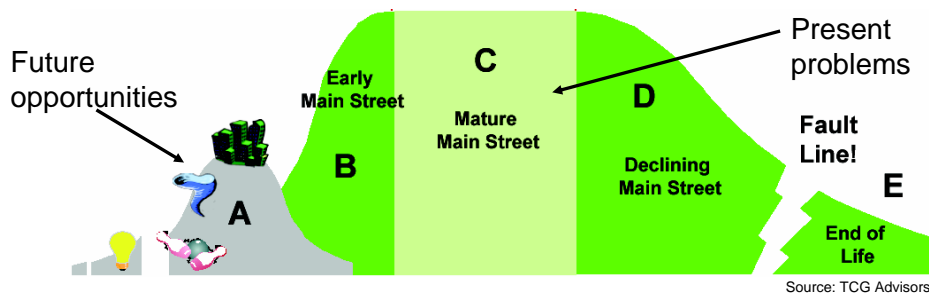
So probably the first place to look at OSS is as the "cheaper" rather than the "better". But due to its inherent customizability, it's still a good possibility for "unavailable in the commercial market".

OSS Isn't a Single Technology



Like data warehousing, OSS isn't a single technology. It's a category of software that crosses many software markets.

- Where in the adoption life cycle are BI/DW and open source technologies you want to consider?
- Where do you fit on the adopter scale?



Source: TCG Advisors

Third Nature, Feb 2008

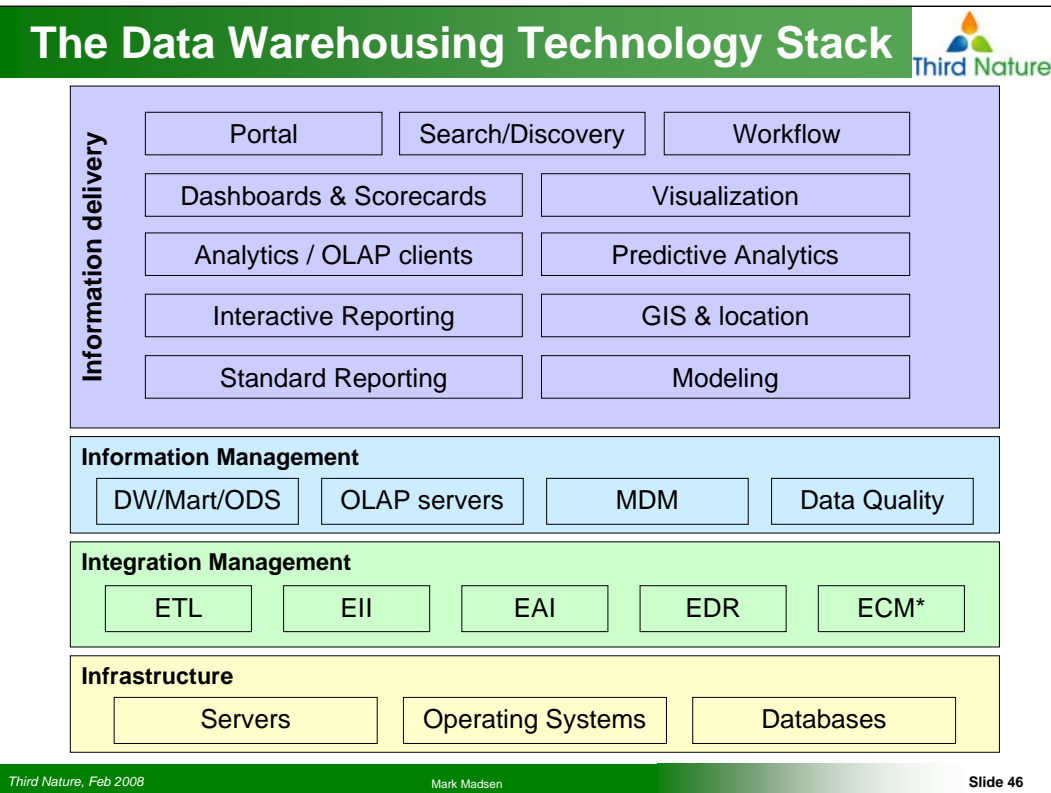
Mark Madsen

Slide 45

Open source software isn't a single technology. It's not even a market. It's a category of software and a method of production.

The big difference between the early people (enthusiasts and innovators) is that they are motivated more by future opportunities than by present problems. This should help you determine where you fall in the spectrum.

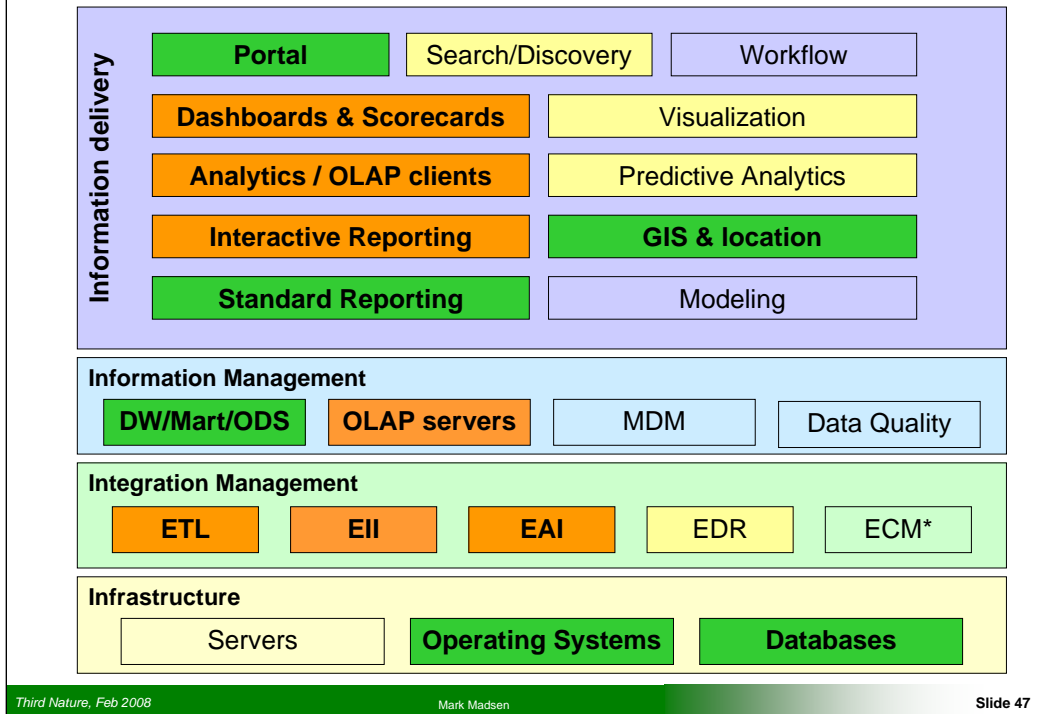
Early adopters are technology enthusiasts looking for a radical shift, where the early majority want a "productivity improvement". The latter group want a whole product, where the earlier group only needs the core product and make the rest themselves.



From an architect's perspective, data warehouse implementations can be put into four distinct layers of technology focused on different purposes.

From a market perspective, the areas with the most emphasis are the bottom and top layers of the stack. Partly it's due to the evolution of software, and partly it's due to the fact that the middle layers are inherently difficult or less self-contained.

Maturity for OSS Components of the Stack



Third Nature, Feb 2008

Mark Madsen

Slide 47

This is looking at maturity from a commercial tool perspective, not from a base code or technology perspective.

Green = mature enough to challenge commercial offerings

Orange = maturing, may or may not be suitable depending on environment

Yellow = still early, niche tools, not yet at the standard for commercial user-driven tools but may be suitable for technical users/developers; one exception is statistics tools and some focused data mining tools that are robust enough to be commercial challengers

No shading = very early or not yet appropriate in a DW/BI context

From an architect's perspective, data warehouse implementations can be put into four distinct layers of technology focused on different purposes.

Open Source Alternatives: Infrastructure



Platform options

- Less open: Windows, Unix, IBM
- More open: Linux, BSD
- Mixed: proprietary appliances built with commodity hardware, some engineering and open source



With the stack in mind, let's look at each layer and some of the leading alternatives.

It's hard to separate the hardware from the operating system since OSS operating systems were originally designed for commodity boxes but have now moved to cover everything from supercomputers to embedded devices.

Together they make up the infrastructure you are running on. There is no open source hardware, so this lists commodity hardware vendors - any will do.

Linux comes in "distributions" – versions of the core Linux operating system that have been packaged and optimized to meet different market needs. For commercial use, Red Hat and SUSE are the most commonly supported for third party software. Debian and its variants are popular, and CentOS is aimed at serving commercial environments but is less established than SUSE and Red Hat.

You can also use Plan 9 or the FreeBSD Unix distribution, which is the most mature of all the Unix-type operating systems. More vendors and projects support Linux, so you may have more compiling to do if you want to run code on FreeBSD or Plan 9 that wasn't built there.

One interesting element is the combination of commodity hardware with open source software to provide a data warehouse appliance that addresses the high cost of performance on large quantities of data.

Appliances aren't really open source, but I include them for completeness.

Market Maturity: Linux Adoption



Table 1: Global Server Operating System Market Share

Platform	2000	2003	2006
Windows NT/200 X Server	14.0 mil (58%)	16.0 mil (53%)	18.0 mil (50%)
NetWare	3.5 mil (14.6%)	1 .6 mil (5.3%)	1.0 mil (2.7%)
UNIX (all)	2.8 mil (11.7%)	2.3 mil (7.7%)	2.0 mil (5.6%)
Linux (Servers)	1.5 mil (6.3%)	5.2 mil (17.3%)	11.0 mil (31%)
Total	24 million	30 million	36 million

“For competitors and companies still on the sidelines (end customers, ISVs, channel partners), this forecast should provide additional justification to the market. Linux is no longer a fringe player. Linux is now mainstream.”

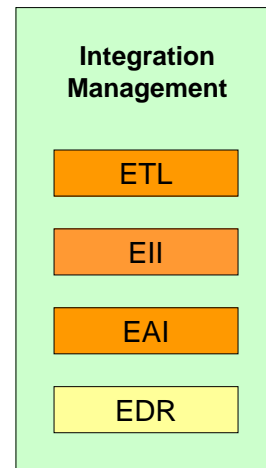
Source: IDC research

Some areas are mature. Linux is one. Databases are another.

Some areas are still evolving, like OLAP and interactive reporting.

Several ETL alternatives

Apartar
CloverETL
Enhydra Octopus
JitterBit
KETL
Kettle (*Pentaho Data Integration*)
SnapLogic (*sort of*)
Talend



Integration management is a rapidly maturing area for open source projects. There are only a handful of projects. Over time, this area will develop as more developers become aware of the need and utility of data integration software over hand-coded integration.

Apartar is a corporate-supported ETL offering that's still in the early stages of release.

Octopus is more usable in java-only environments and

CloverETL can be run standalone or embedded in java applications. It's a transformation framework and engine more than a robust ETL tool.

Jitterbit is yet another in a long list of community/free professional/cost projects.

KETL is second on the list. They have some additional interesting features like clickstream processing, data profiling and multi-server (MPP) support.

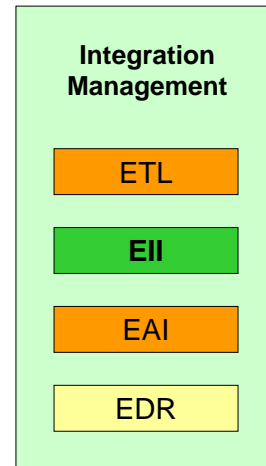
Kettle (now part of the Pentaho project/company) is the most robust. Metadata driven, has a GUI, looks and is most like a commercial ETL tool. Large list of database sources, including an SAP adapter.

Talend is a venture-backed European company focusing on the ETL space.

Open Source Alternatives: Integration



- EII / Data Federation
 - Red Hat (via MetaMatrix acquisition)
 - MySQL Federated storage engine
 - Saga.M31 federation servlet
- EAI
 - Jboss Messaging
 - ActiveMQ
 - OpenAdaptor & elemenope
 - Many more
- EDR
 - Only replication with databases, no heterogeneous support



EII and federated access are limited in choices and in functionality. The database vendors can handle their own, but not generally others. The federated servers act more like EII software, but they are limited in functionality, generally designed to be embedded rather than standalone.

Red Hat made news this year when they bought MetaMatrix, an EII vendor and released it as open source.

EAI support comes in different types. Full messaging including transport is addressed in ActiveMQ and Jboss Messaging. There are also interface layers that help to abstract interfaces across multiple transports, including both open and proprietary software.

OSS Alternatives: Information Management



Data quality / data profiling: OSDQ (profiling)

MDM and related technologies: *nothing*

Metadata repositories: *nothing*

Databases: *almost* as good as commercial vendors

ROLAP/OLAP: Mondrian, Palo



Information Management

DW/Mart/ODS

OLAP servers

MDM/CDI

Data Quality

Databases are here since we want to talk about them in relation to data warehouse use, more than just as basic storage infrastructure.

MySQL is the most popular, with Postgres a close second. Ingres is really a superior database (now) but it doesn't have the community support behind it. This could change, but it depends more on CA than anything else. There's also Bizgres, a Postgres derivative with data warehousing feature support, but it's offered with some strings attached that make it not as attractive.

Mondrian is the only ROLAP server, and Palo is an early-stage OLAP server with an unknown future and specifically designed with Excel in mind for interfacing.

Appliances like Greenplum, Datallegro and Netezza replace the database portion of the stack as well as the underlying hardware and operating system and so fit in this layer as well. As mentioned before, these are commercial offerings leveraging open source to commoditize this layer, but are not open source offerings themselves.

There are now several companies working on engines for MySQL that address BI/DW needs.

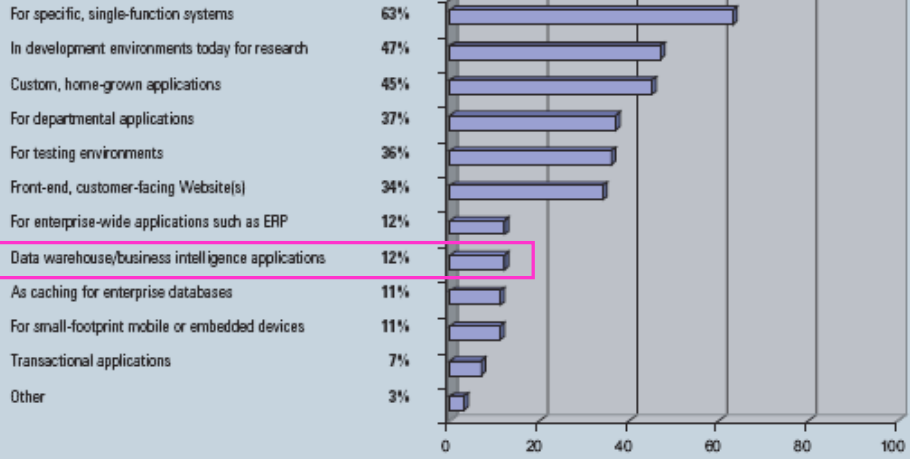
There are also a few open source columnar database variants out there. LucidBD (<http://www.luciddb.org/>) among them and being actively used in a large scale environment.

Open Source Database Use for BI/DW



FIGURE 19: How Open Source Databases are Used

(Among Respondents Using Open Source Databases)



Source: IOUG Open Source in the Enterprise survey

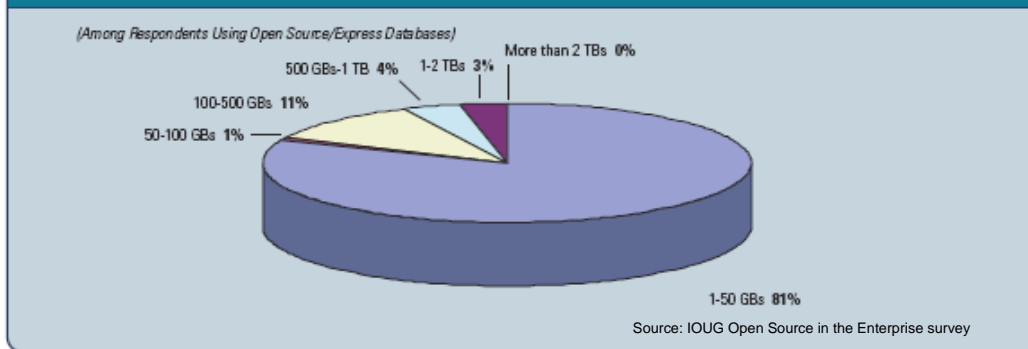
Data Volume is Still a Concern



There are two axes to performance: number of queries and volume of data

- Only 3% of open source databases in this survey were larger than one terabyte
- 23% of Oracle databases in the survey were larger than 1 TB

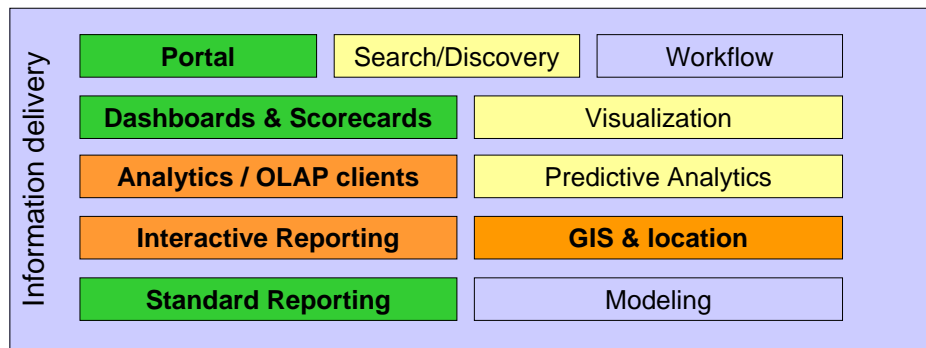
FIGURE 16: Amount of Data is stored within Respondents' largest open source database



OSS Alternatives: Information Delivery



Too many functional areas to cover, so we'll focus on some of the more mature or interesting options related to BI/DW.



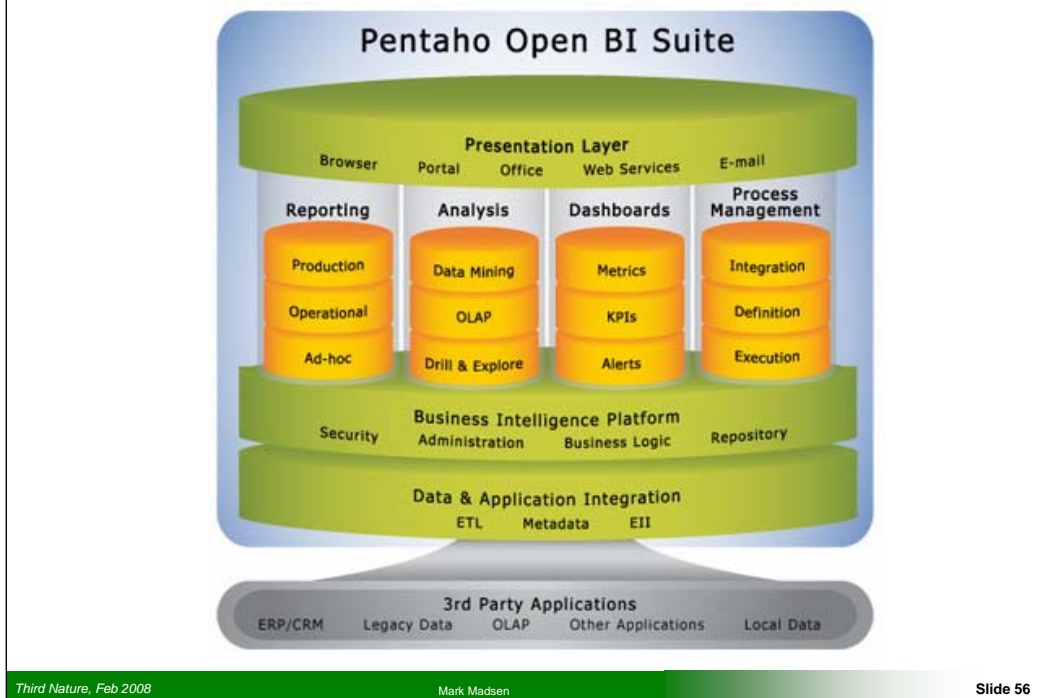
Green = mature enough to challenge commercial offerings

Orange = maturing, may or may not be suitable depending on environment

Yellow = still early, niche tools, not yet at the standard for commercial user-driven tools but probably suitable for technical users/developers; one exception is statistics tools and some focused data mining tools that are robust enough to be commercial challengers

No shading = early, or not yet appropriate in a DW/BI context without programming

BI Suites: Pentaho



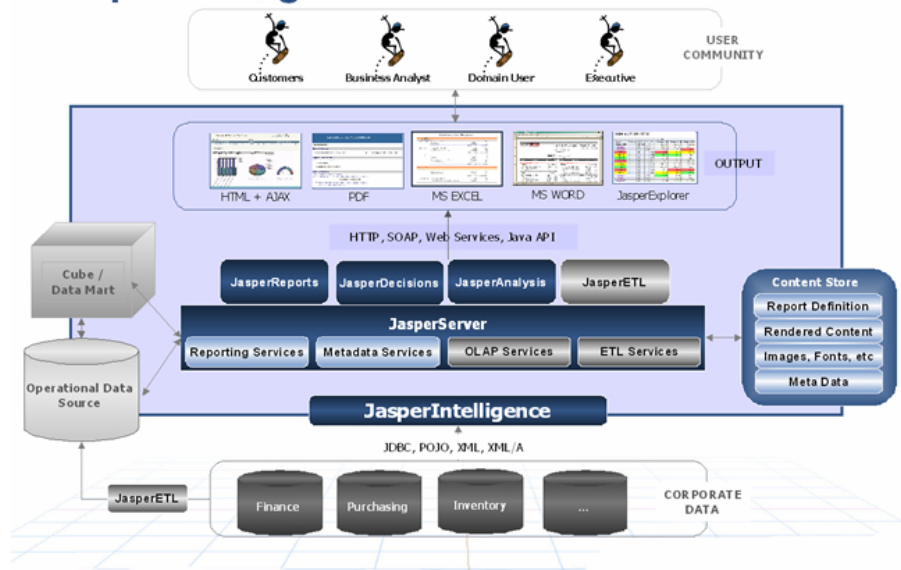
One of the OSS BI companies getting the most press of late. Mondrian, JFreeReport and Kettle – all major category players in the OSS BI space – have joined forces with Pentaho. Their goal is to be the Business Objects of the OSS space, covering the complete range of DW/BI functionality.

Offer the OSS/free version and a proprietary extended “pro” version. Goal of the company is to make revenue off of pro licenses in addition to services.

BI Suites: Jasper Intelligence

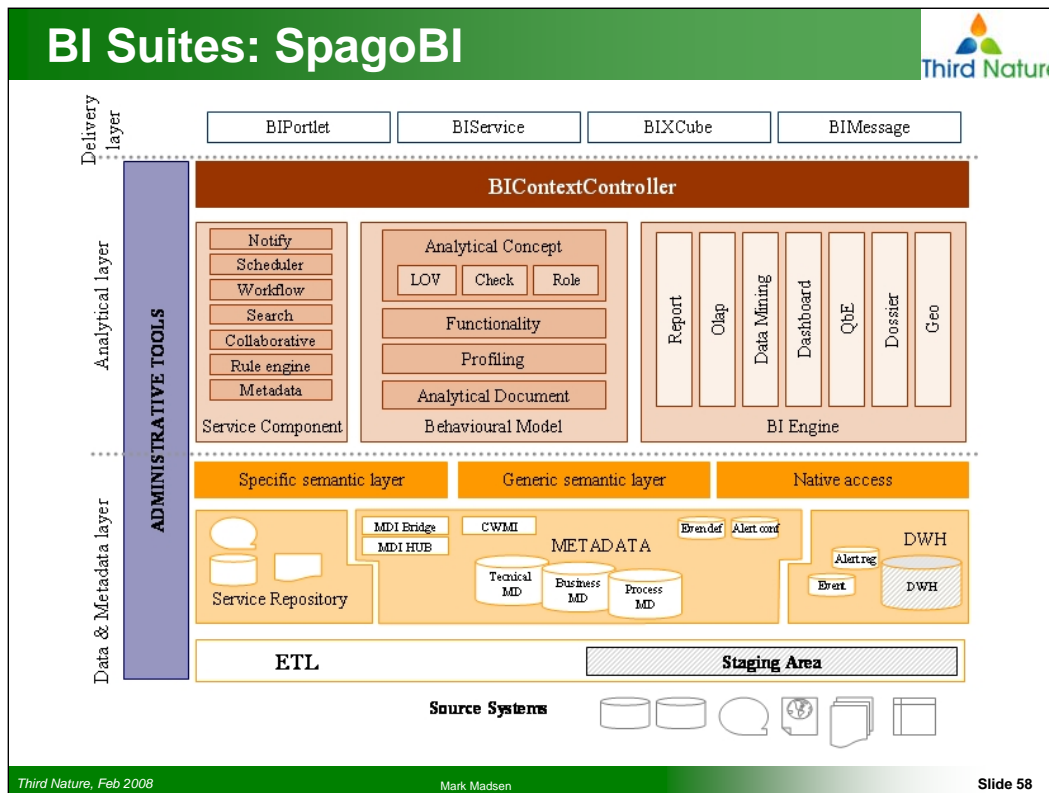


JasperIntelligence Architecture



Jasper doesn't have quite the breadth of offering that Pentaho is pushing forward, but they have the same vision and much tighter component integration. They partner with Talend for ETL.

Jasper is, according to a MySQL survey, the leading reporting tool used with MySQL.



SpagoBI is part of OW2 (formerly ObjectWeb), a big open source middleware initiative with many different projects.

The project is maintained by [Engineering Ingegneria Informatica](#), a large Italian services firm.

Similar to the other suites, they integrate multiple projects together into a full solution. The Spago BI project uses Jasper, BIRT, Weka and a host of other things.

An interesting aspect to their implementation is the choice to use a portal as the primary interface. This allows many different BI objects to be used in both commercial and open source implementations, since they are based on JSR 168 portlet specs. If you go with their stack then you would install the eXo portal.

Reporting

BIRT

JFreeReport, JFreeChart

OpenI

OpenReports

BEE

OLAP

JPivot & Mondrian (*Pentaho OLAP*)

BEE

Palo



There are more. Some that I've checked are abandoned, others are losing community to the major players, some are continuing.

Jasper – leading reporting tool on MySQL

BIRT – specifically designed for embedded / operational reporting

OpenI – hard to get good information on their success

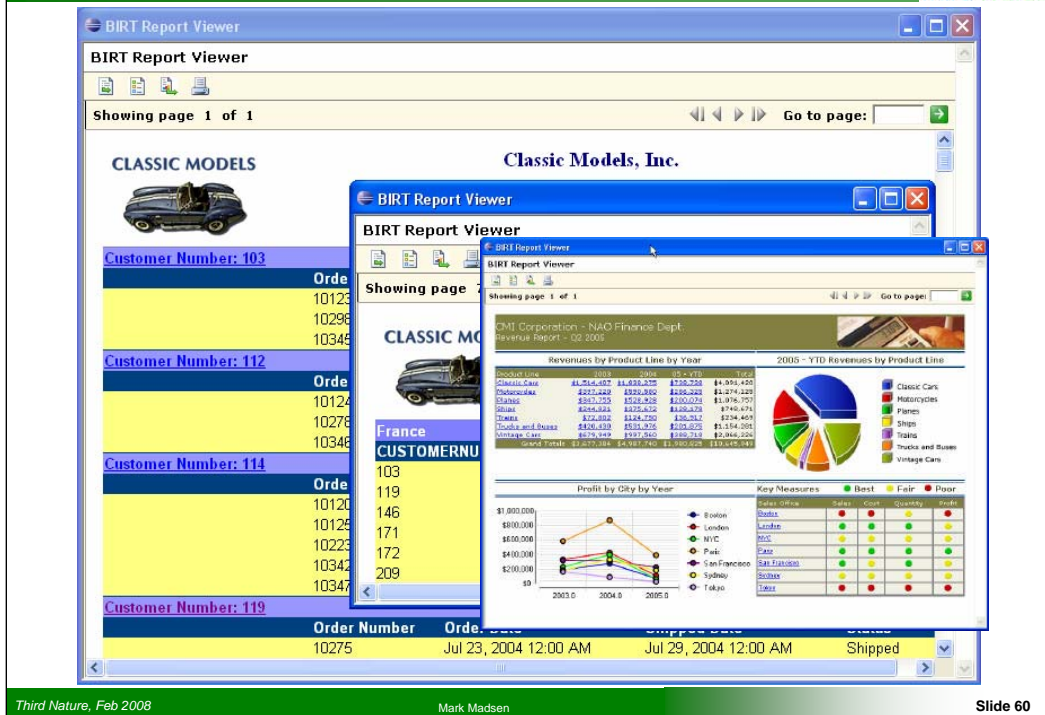
OpenReports – single-person project with small community, aimed at making reporting easier to use

BEE – European based project, interesting in that it's based on perl, TK, etc. and not other reporting.

OLAP: not many choices

Most often JPivot and Mondrian together, but they can be used independently. Mondrian is the ROLAP server.

Palo is an early stage MOLAP tool aimed at use with Excel.



The screenshot displays the BIRT Report Viewer interface. The main window shows a report for "Classic Models, Inc." with a table of orders and a sidebar for customer numbers. Overlaid windows show detailed financial reports, including "Revenues by Product Line by Year" and "Profit by City by Year".

Revenues by Product Line by Year

Product Line	2003	2004	03 + YTD	100%
Classic Cars	\$1,514,487	\$1,638,335	\$3,152,822	\$1,955,426
Motorcycles	\$272,448	\$329,890	\$602,338	\$1,379,420
Planes	\$347,255	\$228,428	\$575,683	\$1,076,757
Trains	\$264,813	\$322,512	\$587,325	\$746,494
Trucks	\$22,882	\$124,730	\$147,612	\$234,469
Trucks and Buses	\$108,828	\$531,816	\$640,644	\$1,154,091
Vintage Cars	\$879,149	\$937,840	\$1,816,989	\$2,946,226
Grand Total	\$2,777,764	\$4,937,760	11,700,028	19,025,547

Profit by City by Year

City	2003.0	2004.0	2005.0
Boston	~\$1,000,000	~\$1,200,000	~\$1,500,000
London	~\$800,000	~\$1,000,000	~\$1,200,000
NYC	~\$600,000	~\$800,000	~\$1,000,000
San Francisco	~\$400,000	~\$600,000	~\$800,000
Sydney	~\$200,000	~\$400,000	~\$600,000
Tokyo	~\$100,000	~\$200,000	~\$400,000

Key Measures

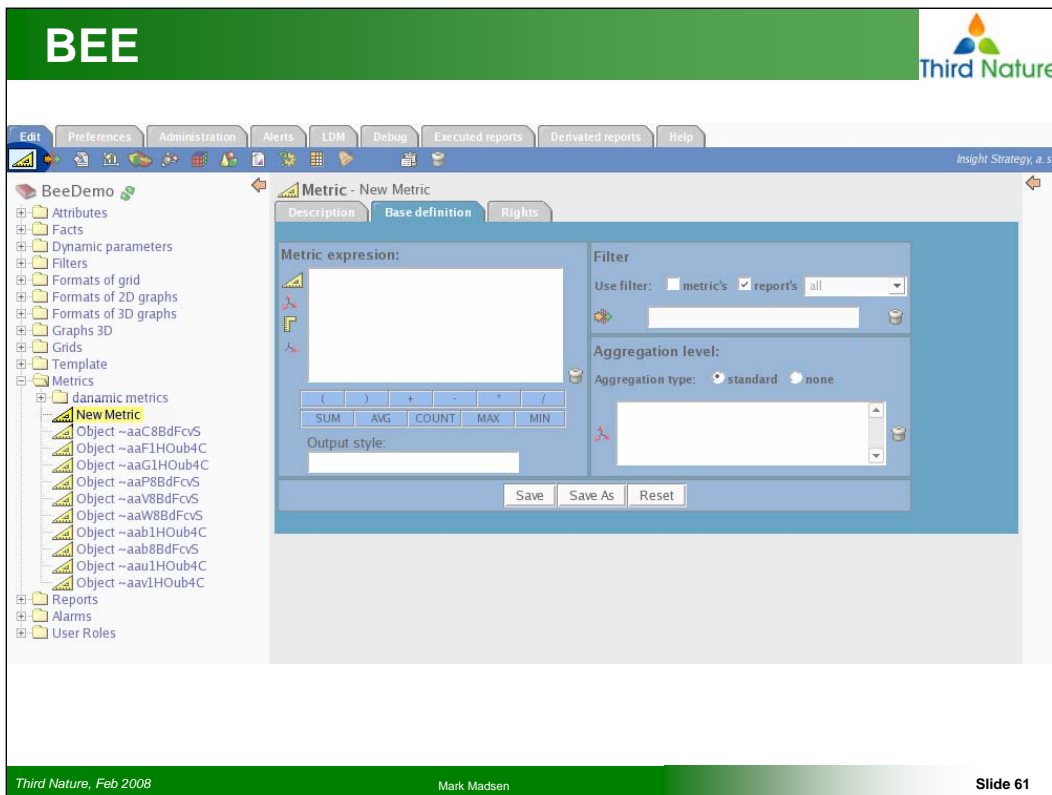
Measure	Best	Fair	Poor
Sales Online	●	●	●
Sales	●	●	●
Cost	●	●	●
Quantity	●	●	●
Profit	●	●	●

Good article on deploying:

<http://www.onjava.com/pub/a/onjava/2006/07/26/deploying-birt.html>

The APIs are the *Design Engine API* (DE API), the *Report Engine API* (RE API), and the *Chart Engine API* (CE API). The DE API is responsible for creating and modifying the XML report design format. This API is what the Eclipse BIRT Report Designer uses to create the report design (*rptdesign*), library (*rptlibrary*), and template (*rpttemplate*) files. The RE API is responsible for consuming these files and producing the report output. The Report Designer Preview and Web Viewer servlet use this API to generate reports. The CE API can be used to create and render charts standalone or through the DE and RE APIs.

The BIRT Web Viewer is a web application (servlet-based), comprised of servlets and JSPs, that encapsulates the RE API to generate reports. In addition to generating reports, it supports HTML pagination, PDF, Table of Contents (TOC) functionality, and export to CSV.



BEE is a European project.

Not that much activity based on downloads.

Interesting in that it isn't based on other reporting/OLAP projects for the basics. Built in perl, TK, etc.

Integrates with R for statistics, which is interesting since most others haven't done this yet.

Navigation menu: Edit, Preferences, Administration, Alerts, LDM, Debug, Executed reports, Derived reports, Help

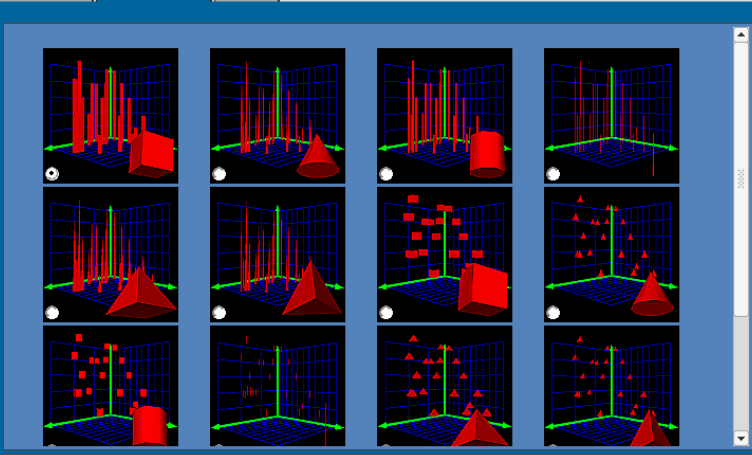
Insight Strategy, a.s.

BeeDemo

- Attributes
- Facts
- Dynamic parameters
- Filters
- Formats of grid
- Formats of 2D graphs
- Formats of 3D graphs
 - New format 3D**
 - Graphs 3D
 - Grids
 - Template
- Metrics
- Reports
- Alarms
- User Roles

3D graph format - New format 3D

Tabbed interface: Description, Main definition, Rights



Buttons: Save, Save As, Reset

OLAP: JPivot & Mondrian



Products		Region	Measures		
			Measures[0]	Measures[1]	Measures[2]
-All Products[0]	-All Region[0]		1,015.22	780.70	890.02
	All Region[0]				
	+Region[0]		873.77	971.56	1,102.23
	-Region[1]		1,074.22	945.91	823.55
	Region[1]				
	+City[0]		876.48	923.64	1,026.13
	+City[1]		1,136.24	825.63	1,067.35
	+City[2]		955.63	1,071.25	1,159.65
	+City[3]		1,163.37	1,015.91	949.34
	+City[4]		909.86	1,136.97	984.30
	+City[5]		1,116.37	1,063.09	985.76
	+City[6]		1,089.23	1,063.38	1,006.19
	+City[7]		1,172.89	992.38	1,050.60
	+Region[2]		983.33	834.35	1,131.19
+Region[3]		1,231.73	1,041.51	1,026.49	
+Region[4]		1,043.67	1,167.71	903.53	
All Products[0]	+Category[0]	-All Region[0]	890.73	1,170.83	955.32
		All Region[0]			
		+Region[0]	1,121.95	957.97	915.73
		+Region[1]	894.19	1,214.22	1,089.54
		+Region[2]	1,124.23	1,042.52	925.80
	+Region[3]	904.35	1,106.97	911.88	
	+Region[4]	1,086.60	1,031.78	860.41	
	+Category[1]	+All Region[0]	939.83	1,082.63	977.79
	+Category[2]	+All Region[0]	988.25	850.40	959.15

Dashboards

- Pentaho has their own dashboard product
- Palo can be used for dashboards as well as OLAP
- BEE Project (reporting and dashboards)
- VitalSigns
- MarvellIT Dash

Portals

- JBoss Portal
- Liferay Portal
- Apache Jetspeed
- Plone
- eXo
- Over 100 others...

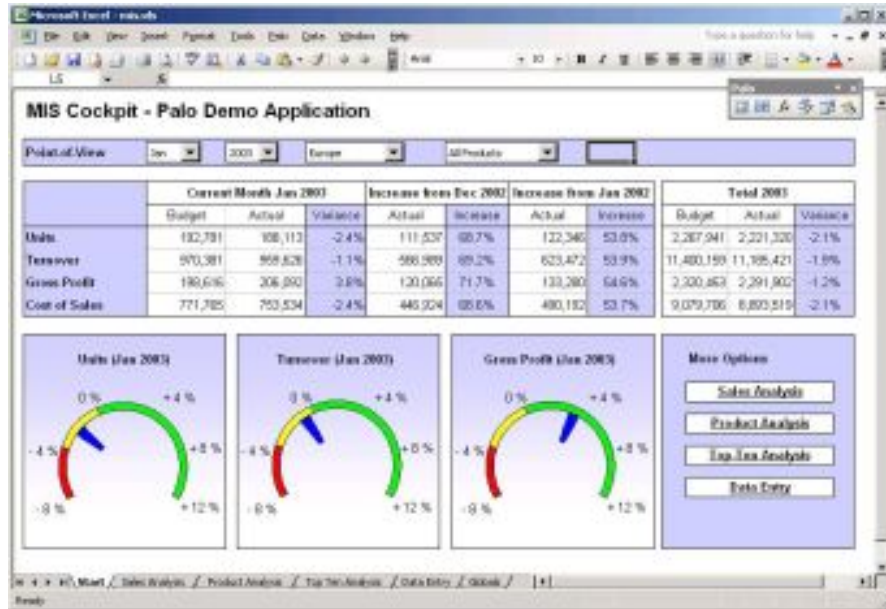


Dashboard offerings are somewhat limited but the portal and content management market has over a hundred open source offerings. Many are CMS-related, aimed mostly at provisioning external or internal web sites. Some of the more popular portals are listed.

Search: new apache donated project (from cnet) <http://incubator.apache.org/solr/> that handles basics like word stemming, stop words, etc. but is mostly for indexing and search, without more robust text analysis capabilities.

Good CMS resources: <http://www.cmswatch.com/> and <http://www.opensourcecms.com/>

OLAP Dashboard: Palo Interface



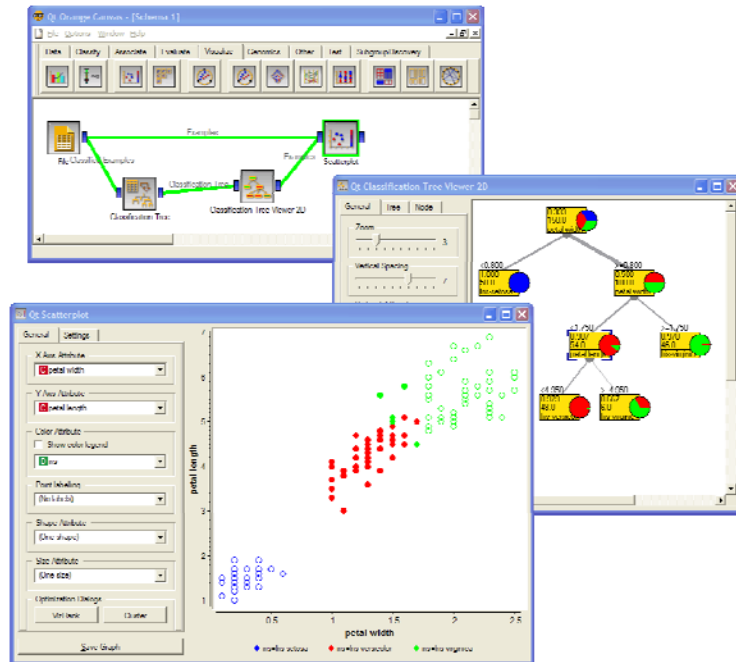
A screen shot of a demo application created using data pulled from the open source Palo Multi-Dimensional database. The cockpit was created using the free excel addin and uses only Palo and Excel functions.

OSS Alternatives: Predictive Analytics



Key projects:

- Weka
- R
- Orange



There are a lot of predictive analytics projects with a focus on different aspects and techniques for PA and data mining. Some are simple one-purpose tools, others are embeddable libraries, and some are complete suites that offer sophisticated interfaces and operations.

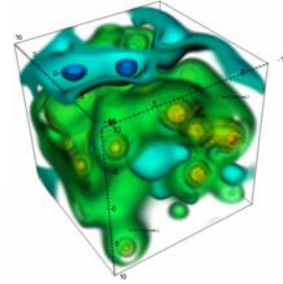
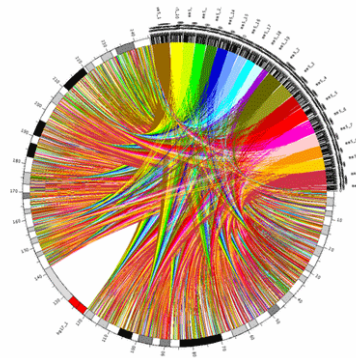
OSS Alternatives: Visualization



Visualization: many, many offerings

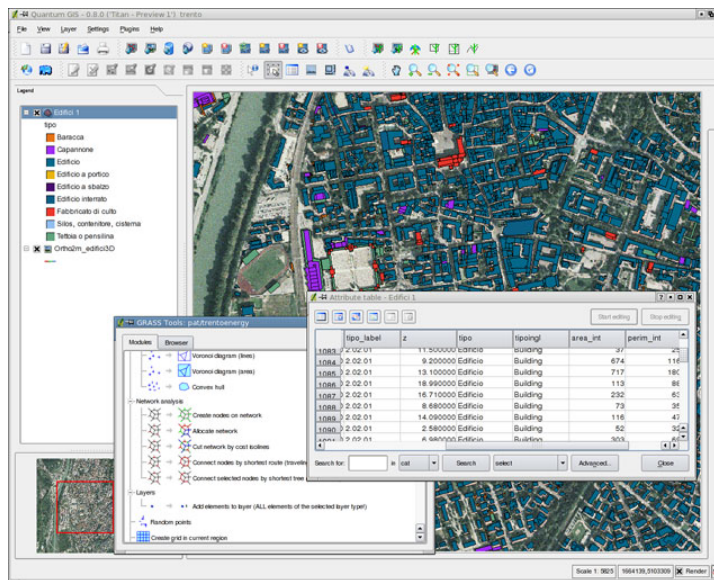
Most are libraries, a few are tools.

- VisIt
- Prefuse
- Processing
- Circos



The visualization space has some interesting projects. In general, there are tools for visualization aimed at data analysts, or embeddable libraries that allow you to build visualizations into your application, or tools for building dynamic interactive visualizations. You can also duplicate some of the interesting things with a little bit of Flash programming.

Open source is overrunning commercial GIS



Visit FreeGIS.org for a huge list of software, data and projects.

Some notable projects:

Grass, SAGA - analysis

MapServer – render spatial data via web

uDig (desktop app, eclipse based), Geotools, Geoserver (web server platform), PostGIS

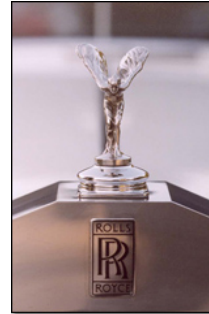
OSSIM for advanced image processing

OGC – open geospatial consortium (standards)

OSGeo – open source geospatial foundation, provides MapGuide (<http://mapguide.osgeo.org/>)

Why Consider Open Source?

IT is after one of three things:



As we just saw, there are many alternatives in almost every category of DW/BI software. Some are better than others, some categories are not close enough to commercial alternatives to consider. So why are people looking at open source?

Better alternatives

Equivalent alternatives that are cheaper to acquire or own

Items that are unavailable

Items that are unavailable – this also includes features missing from a product. OSS provides the full software or gap filler.

The Top Stated Reason: Cost Savings



~70% of companies surveyed stated lower costs as the reason for OSS deployments

Source: CIO Insight survey

Edition	Servers	CPUs per Server	License	Maintenance	Total 1 yr
Oracle SE	4	2	\$120,000	\$26,400	\$146,400
Oracle SE	4	4	\$240,000	\$52,800	\$292,800
MySQL Network Platinum	4	2	\$0	\$19,980	\$19,980
MySQL Network Platinum	4	4	\$0	\$19,980	\$19,980
Ingres r3 Premium	4	2	\$0	\$15,960	\$15,960
Ingres r3 Premium	4	4	\$0	\$31,920	\$31,920

Source: Meta Group

What if: you took 50% of that savings and applied it towards a new hire? How much value would you get over money spent on support contracts?

Third Nature, Feb 2008

Mark Madsen

Slide 70

A compelling argument, particularly when you consider how much more you get for your support cost.

From Meta:

“MySQL Network includes many more services than typical software support offerings. Included in the cost is a set of technical advisors and certified configurations tested within a software stack. It also includes the commercially licensed version of MySQL, and consulting services for help with schema review, performance tuning, and even code reviews of server-side, user-defined functions.”

Another statistic from the same survey:

86% of companies said open source meets or exceeds the expectations they had for cost savings for Linux, with the primary area being license fees.

Customization



Third Nature, Feb 2008

Mark Madsen

Slide 71

If you don't customize, you are by definition doing what everyone else is doing. Where is the value in keeping pace with the herd?

Commercial vendors hate it when you want to do this.

But if want to use BI tools in an operational context (operational BI being one of the big technology drivers in BI right now), you need the ability to easily fit them into the operational environment. That often means customizing them in some way. Commercial tools are often hard to embed in a transparent fashion.

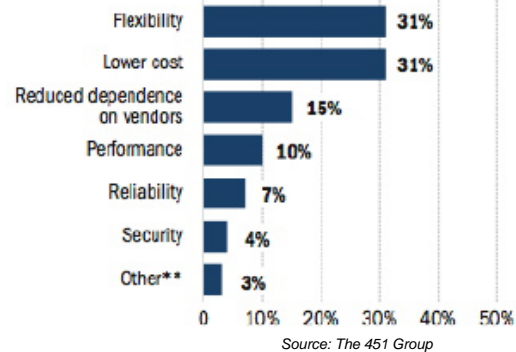
Ability to Customize Provides Value



- 65% of companies surveyed said OSS sparked innovation in their IT departments.
- 71% of companies deploying believe OSS provides them a business advantage
- Among these companies, customization, functionality, and scalability were the top reasons to use open source.

Source: CIO Insight

After your organization adopted open source software, what was the primary benefit of its use?



An interesting note on the stats: the companies who saw an advantage believed customization was important. This seems to be a trait of early adopters and not representative of more mainstream adopters, particularly if you go with Moore's perspective.

The 451 Groups survey is interesting because it shows that costs actually ranked as the #2 benefit, not the #1 as anticipated.



Avoid vendor imposed upgrade cycles

The two big problems:

- Major releases, ready or not!
- End of life and de-supported versions

Sales and marketing schedules dictate releases of software that isn't ready yet. OSS isn't under that kind of pressure.

Likewise, the vendors often follow a product release schedule that can be too fast for customers, particularly with infrastructure products.

They also drive the termination of support and maintenance for older versions of their products. If vendors had their way, there would only be two versions: the latest version, and the one they are working on.

Avoid technology lock-in



- Sometimes the vendor's core technology is good, but it takes you away from the direction the commodity market is moving.
- Modularized architectures and technology stacks provide options to change at different layers. Proprietary alternatives remove options.

Open source evolves boundaries with open standards, unlike proprietary software where even the standards are subject to arbitrary change.

Sometimes technology imposition is worse than a single product because it influences larger parts of the IT infrastructure. For example, using some of Microsoft's products requires many other products in the Microsoft stack which duplicate parts of your existing infrastructure. Open source software tends to be more modular. Where it isn't modular, at least it won't cost you a lot to use those components.

Adoption: Dealing With the Risks



16% of respondents to a Ventana survey said “adoption by large enterprises” would influence their decision to use Open Source

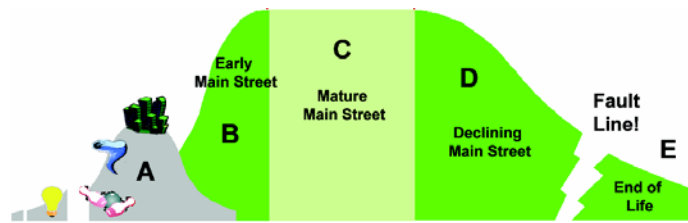
According to this survey, the early majority is looking to the innovators before jumping in.

What is your position on Moore's adopter scale?

Emerging Tech: The IT Analyst Paradox



- “Open Source BI isn’t ready.”
- “It’s not comparable.”
- “It is not ready for production use today. Open source BI is in its infancy, and will not be ready for a few years.”
- “Open source BI is a work in progress.”



But where do the analysts live?

What do the IT analyst firms say?

Most of the IT analyst firms are wrong.

The analyst paradox: now is precisely the time to get into open source BI because firms like Gartner say it isn't. When analyst firms acknowledge the existence of a technology and say “not yet”, this means that the innovators are already using it and early adopters have been working out the problems. The market is seeded for the early majority.

Historically, analyst firms flip-flop suddenly when the early main street companies say they're considering the technology. If you want early value and advantage, this is the time to be in the market, but also to have realistic expectations.

A common criticism of open source is that “it's not comparable” to commercial products. Neither were GNU compilers, Linux, Tomcat or Jboss yet they all displaced or are displacing commercial software. The missing word in the second quote is “yet” when comparing to commercial products.

Common Traits of OSS Adopters



Early adopter profile (more risk, focus on differentiation)

- Already use Linux or have operational experience with Unix
- Use scripting languages (Python, PHP, Perl) and / or Java for internal development
- Believe internal labor provides more value than large capital outlays for software



Third Nature, Feb 2008

Mark Madsen

Slide 77

The first few items address some of the risks outlined earlier.

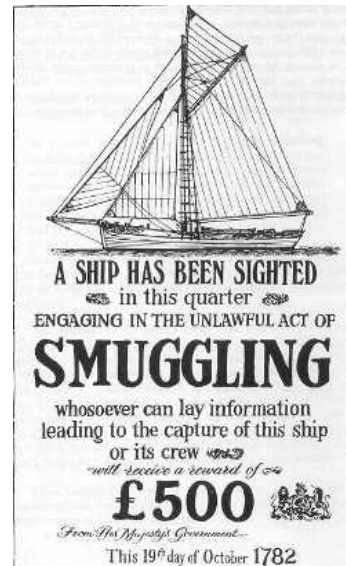
Follow A Structured Evaluation Process



Open source bypasses the normal IT software discovery process: it's bottom up

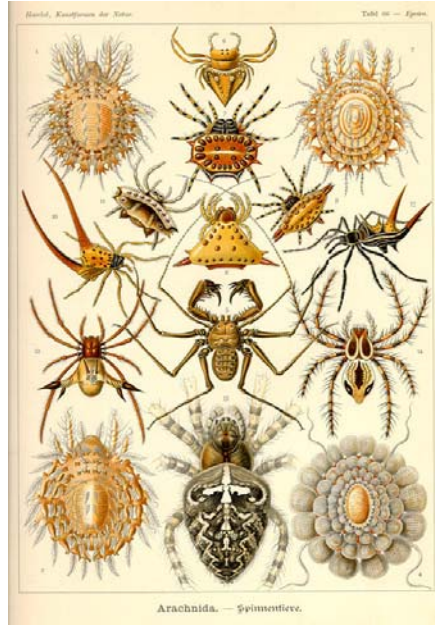
- How you learn about projects
- Where you find them
- How you evaluate them
- How you acquire them

Need to follow a structured process, but one that differs from the standard IT process



You find open source software on your own, they don't generally come to you.

Some Evaluation Criteria Will Change



Different evaluation criteria are needed for open source

- Community is key
- Focused use more than broad-ranging tools
- Interoperability
- Customizability
- Need to review licenses

Some organizations can help

- Open Solutions Alliance
- Open Source Initiative
- Business Readiness Rating

It's a different animal. Some different rules apply.

Resources to help:

OSA - www.opensolutionsalliance.org

Open Source Initiative – OpenSource.org

Business Readiness Rating - OpenBRR.org

Two other organizations, both of whom settled on the same name for their models and have some useful advice:

Open Source Maturity Model – Navicasoft (so-so model)

Open Source Maturity Model – Cap Gemini (they seem to not “get it” as much based on all the copyright notices splashed all over their web site and model)

Estimating Project Viability and Maturity



- Harder to research than most commercial products and companies.
- Need different metrics since “revenue” and “market share” metrics are meaningless.
- Should look at:
 - Usage (type and volume)
 - Community activity (forums, bug reports, fixes)
 - Key contributors
 - Project longevity and stability

Most IT analyst firms don't track open source because it doesn't fit their company evaluation focus or commercial payment model.

Market share is very hard to work out because there's no reliable way to track licenses, customers, and of course there's no revenue.

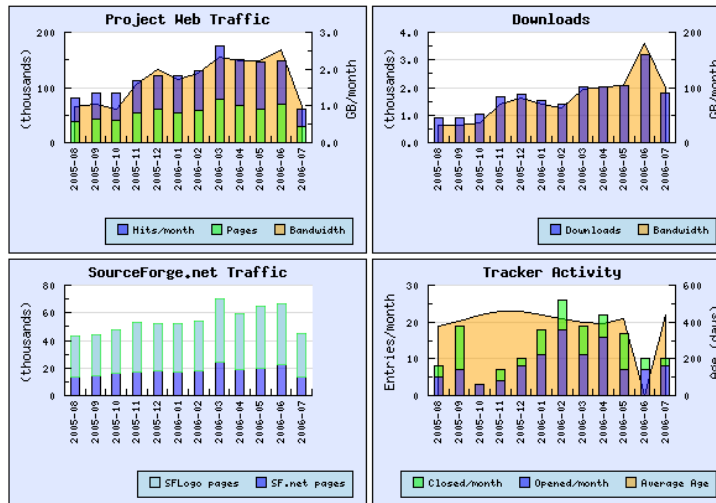
Some things can be tracked by web-crawling stats, e.g. web server or app server share.

Estimating Project Viability and Maturity



Sourceforge (for projects hosted there) offers some useful statistics to help evaluate projects.

Usage Statistics For Mondrian



Next best thing: download stats, project activity stats (like bugs reported, fixed, community discussions on forums, etc.)

Download stats are often used as a proxy in the absence of customer counts you might get for commercial software. Trying to find the number of active users and references is worthwhile.

One problem with these stats is that some projects have moved off SourceForge, so the stats are either absent or misleading.

Sourceforge.net (for projects hosted there) is a good place to start.

Change the Software Acquisition Process



Normal IT controls for software acquisition don't address Open Source

- Internal project-based acquisition is not repeatable, can cause trouble without larger scope IT planning
- Unless paying for support, bypasses both procurement and legal processes
- No control of evaluation process.

Most of this is missing on the open source side, and it's easier to use inappropriate software that needs to be changed out later.

RFPs, checklists, purchasing departments are there to ensure quality and avoid fraud (even though it seems they are more often in the way), and open source makes a lot of these irrelevant or puts the onus on you.

Address the Maintenance Process



Processes are different:

- How do you decide when to move to a new release?
- Who keeps track of critical fixes, and how do you deal with more frequent fixes?

Choices for maintenance:

- Manage the maintenance on a project-specific basis
- Centralize OSS maintenance processes
- Third-party or commercial OSS management support

Aside from the problem of multiple projects independently using open source, there is the problem of internal maintenance processes.

Distributed maintenance is generally not a good idea in the long term.

Centralized has problems too: many disparate packages on disparate platforms means hard for one group to do.

Address Your Support Processes



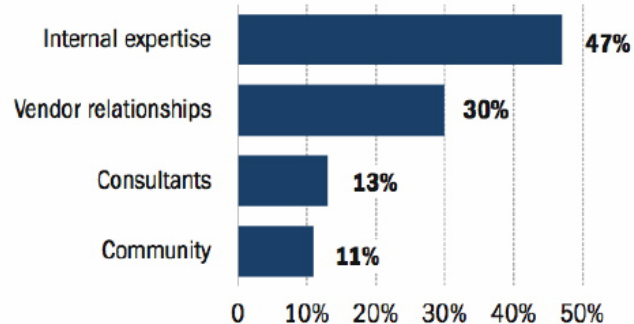
OSS unbundles software licensing and support, The four models for Open Source:

- Unsupported
- Community
- Vendor
- Third-party

Your choices:

- Buy support
- Self-support

How does your organization support open source software (e.g., internal expertise, consultants, vendor relationships, etc.)?



Source: The 451 Group

The reality with enterprise software today is that you usually end up fixing your own problems before the vendor does.

Buying support – just like the commercial model today, but probably cheaper. This is the route most IT shops go with larger projects or more complex software.

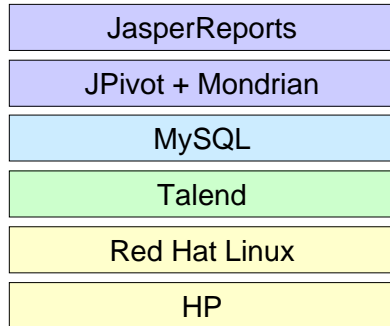
Self support is usually done at the start, but often companies want support later. The project team relies on community when there are problems they can't handle, not too different from enterprise software today. Forums are often better than the vendor.

Design for a Mixed Environment

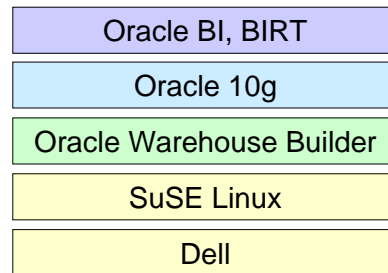


Modularity is the way of open source

You *can* build an entire data warehouse stack on OSS, but it may not be practical to do so.



Even if your IT department tries to be a single-vendor shop, you can still consider mature infrastructure technologies.



You will be using multiple technologies, some will be OSS and some will be commercial. It isn't practical to do it all for most businesses.

Where there are integration points, hybrid environments do have some challenges. E.g. commercial vendor support of specific distributions or versions of OSS projects. Because of this, some commercial vendors provide distributions along with their software so you don't run the risk of mixing incorrect distros/versions.

You are riding the innovation-commodification curve, which is largely a result of modular architectures and standards.

Futures: Software Utopia



Third Nature, Feb 2008

Mark Madsen

Slide 86

Open source as a concept is still early in its life. The GPL was created in 1985, a very short time when compared to hundreds of years of contract law.

There are many issues surrounding intellectual property (patents, copyrights), commercial software sales and support in a world of perfect commodities, and changes in software architecture and communications driving companies to more outsourcing.

Perfect commodities leads to everything being open source and we all live in happily in low-G space colonies.

Questions?



Open Source BI/DW Projects



BI and Analytics

BEE - bee.insightstrategy.cz/en/index.html
BIRT - www.eclipse.org/birt
JasperSoft – www.jaspersoft.com
MarvellIT - www.marvelit.com/dash.html
OpenI – openi.sourceforge.net
OpenReports – oreports.com
Orange - www.ailab.si/orange
Palo – www.palo.net
Pentaho - www.pentaho.com
R - www.r-project.org
SpagoBI – spagobi.eng.it
Weka - www.cs.waikato.ac.nz/~ml/index.html
VitalSigns - <http://vitalsigns.sourceforge.net/>

• Databases

www.greenplum.com (bizgres)
www.ingres.com
www.mysql.com
www.postgresql.org
www.enterprisedb.com

Integration

Apatar - www.apatar.com
CloverETL - cloveretl.berlios.de/
JitterBit - <http://www.jitterbit.com/>
KETL - www.ketl.org
Octopus - www.enhydra.org/tech/octopus/index.html
OSDQ - sourceforge.net/projects/dataquality
Pentaho - www.pentaho.com
Red Hat – www.redhat.com
Saga.M31 Galaxy - galaxy.sagadc.com
Talend - www.talend.com
SnapLogic – www.snaplogic.com

This is a list of some of the more relevant projects in the BI/DW space. You can hunt up more at SourceForge.net and FreshMeat.net as well as through simple online searches of “open source” and the type of tool you’re looking for.

Creative Commons



Thanks to the people who made their images available via creative commons:

veldt - http://flickr.com/photo_zoom.gne?id=185538767&size=|

canal - <http://flickr.com/photos/mcsixth/150749007/>

glassblower - <http://flickr.com/photos/cazasco/261229878/>

porthole - <http://flickr.com/photos/lwr/24925322/>

lock - <http://flickr.com/photos/tremeglan/400428163/>

Creative Commons



This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/us/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

