

# Open Source and Standards as Catalysts for Innovation and New Business

Dr. Bob Sutor

VP, Open Source and Standards, IBM

[sutor.com/blog](http://sutor.com/blog)



# Open standards vs. open source

- Many people who have not done software development are confused between “open standards” and “open source”: they don’t know what code looks like and what you do with it.
- *A standard is like a blueprint*: it tells you what you must do if you actually get around to building something.
- An open standard is one that is developed and maintained in a particularly transparent way with community involvement, and is “freely” available and implementable.
- Open source is code, actual concrete software, and it may implement open standards.
- *Open source is built* and maintained in a particularly transparent way with community involvement, and is “freely” available.

# Innovation is ...

- Great new ideas that no one ever thought about before.
- Combining old ideas in new ways to provide new insight and possible paths forward.
- Breakthrough thinking.
- Possibly technical.
- More generally, possibly scientific.
- Possibly business-oriented or about economics.
- Possibly political.
- Possibly academic or even legal.
  
- In short, novel ways of thinking that disrupt the *status quo*, allow expanded thinking, new opportunities, changed landscapes, and growth.

# Example: Sahana



## SAHANA

Home of the Free and Open Source Disaster Management System

Sourceforge

Dev WIKI

**Sahana is a Free and Open Source Disaster Management system. It is a web based collaboration tool that addresses the common coordination problems during a disaster from finding missing people, managing aid, managing volunteers, tracking camps effectively between Government groups, the civil society (NGOs) and the victims themselves.**

- [Sahana Overview](#)
- [Screenshots](#)
- [Download and Install Sahana](#)



<http://www.sahana.lk/>

# Open source and standards: catalysts?

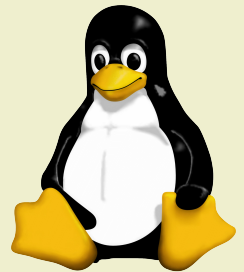
- Examples of open standards
  - HTML, HTTP, XML for web and information structure
  - SOAP, WSDL, WS-Security for web services and SOA
  - OpenDocument Format for office documents
- Examples of open source
  - Apache web server
  - GNU/Linux operating system for desktops & datacenters
  - mySQL and Apache Derby databases
  - OpenOffice desktop suite
  - Eclipse integrated software development environment
  - Firefox web browser

# Why does IBM consider open source important?

- OSS can be a major source of innovation
  - Innovation can happen any time, anywhere
  - Development through “open communities” leads to potentially broad ideas and creativity
- OSS is a good approach for developing emerging standards
  - Popular open source projects can become de facto / open standards
  - Wide distribution deployment
- OSS is a source of competition in marketplace
  - Office productivity applications (word processing, spreadsheets, presentation)
  - Operating systems (Linux for servers, desktops)
  - In some areas, perhaps only growing competitor to a single established vendor

# How does IBM use open source software?

- To run our business
- In our hardware
- In our software or as an alternative to commercial software
- As part of service engagements
- As an R&D collaboration vehicle
- As a way of influencing the direction of the IT industry
- As a competitive tool
- As a way of leveling the playing field
- As a way of invigorating stagnant market categories, giving us perhaps a second chance for business growth





Australian Government  
 Department of Finance and Administration  
 Australian Government Information Management Office

***A Guide to Open Source Software***  
 for Australian Government Agencies

*Developing and Executing an ICT Sourcing Strategy*

*Australian Government Information Management Office*  
 a Business Group of the Department of Finance and Administration

**Table of contents**

**Glossary ..... 6**  
**Foreword ..... 7**  
**Introduction ..... 8**  
 Sourcing open source software ..... 8  
 A definition of open source software ..... 8  
 What is source code? ..... 9  
 Open source software and commercial software ..... 9  
 History and development of OSS ..... 10  
 OSS usage within government ..... 10  
 Overview of the OSS industry ..... 11  
 Larger vendors ..... 12  
 Established SME vendors ..... 12  
 Boutique consultancies and SME vendors ..... 13  
 The Australian open source industry ..... 13  
**Typical concerns about open source software ..... 14**  
 Cost of licences ..... 14  
 Availability of support ..... 14  
 In-house support ..... 15  
 External support ..... 15  
 Maximising flexibility in support options ..... 16  
 Software reliability ..... 16  
 Maturity and longevity ..... 16  
**Sourcing open source software ..... 18**  
 Sourcing scenarios ..... 18  
 In-house sourcing ..... 19  
 In-house procurement checklist ..... 20  
 External sourcing ..... 20  
 Open source products and single sourcing agreements ..... 21  
 Incidental sourcing ..... 22  
 Custom software development ..... 22  
**Preparing a procurement plan ..... 23**  
 Evaluating the business case ..... 23  
 Defining business requirements and priorities ..... 24  
 Sourcing issues to consider ..... 25  
 Writing inclusive RFTs ..... 25  
 Defining selection criteria ..... 26  
 Assessing the value of OSS solutions ..... 26  
 Meeting Australian Government requirements ..... 27

Contents

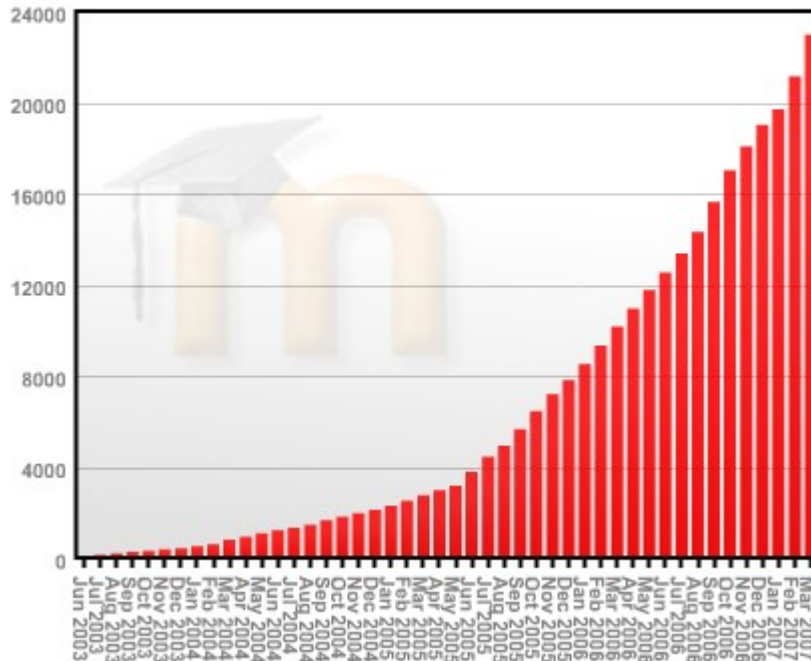
[http://www.sourceit.gov.au/\\_data/assets/pdf\\_file/42065/A\\_Guide\\_to\\_Open\\_Source\\_Software.pdf](http://www.sourceit.gov.au/_data/assets/pdf_file/42065/A_Guide_to_Open_Source_Software.pdf)



# Example: Explosive growth of open source in education

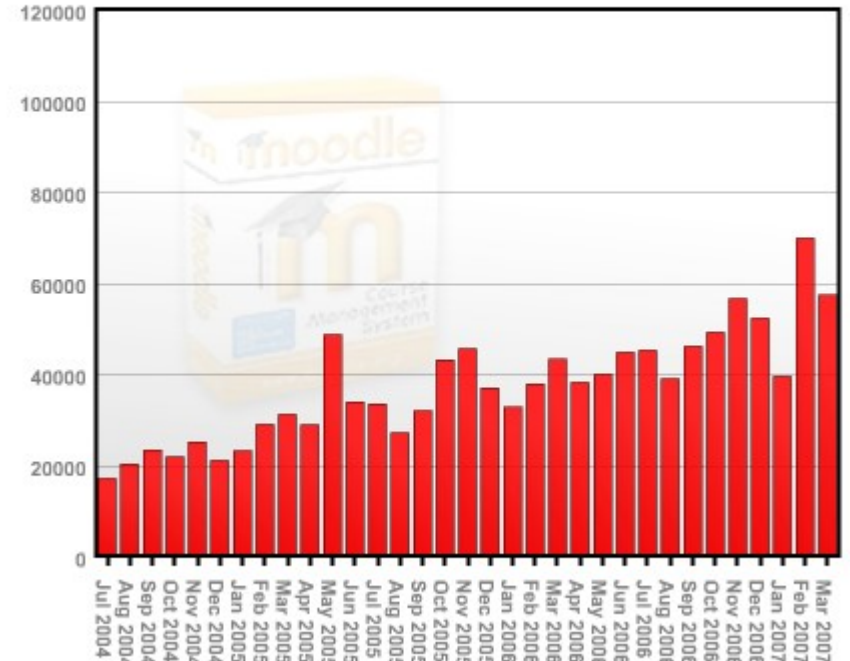
## Moodle sites

Total known sites



## Moodle downloads

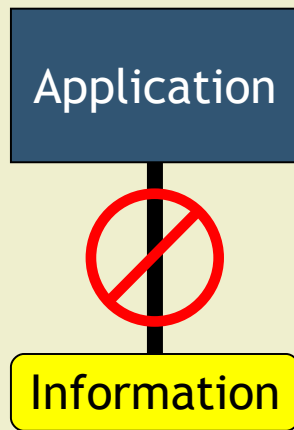
Downloads per month



# So what if I can't charge for the software?

- There are several common open source business models
  - Embedding open source in proprietary products
  - Using open source for hosted applications or services
  - Software subscriptions
  - Consulting, integration, maintenance, education
  - Dual licensing with an open source and commercial license
  - Layering proprietary software over open source (e.g. Linux)
  - Extending and enhancing
  - Patronage (e.g. support open source to sell hardware)
- There will be more

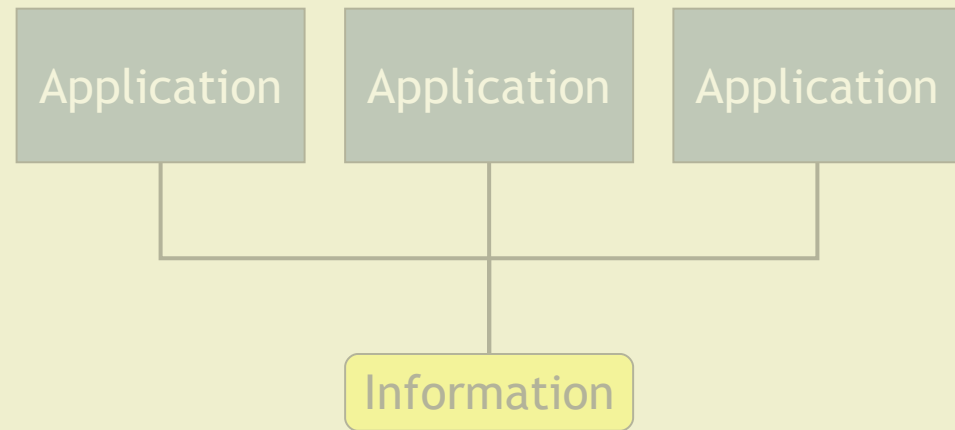
# In the old days, SW providers controlled the data



## Old Style

Information is closely linked to the application that created it.

Control is with the software developer *not* the customer.

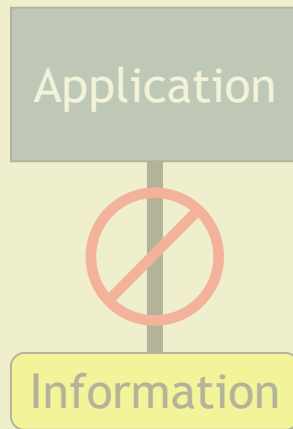


## New Style

Information is represented using open standards not under the control of a single vendor, and multiple applications can create and access it interchangeably.

Control is with the customer *not* the software provider.

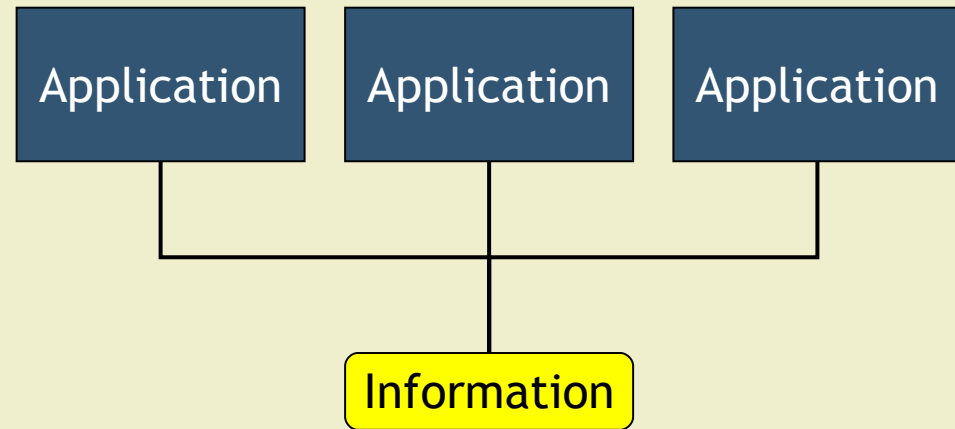
# But customers are demanding something else



## Old Style

Information is closely linked to the application that created it.

Control is with the software developer *not* the customer.

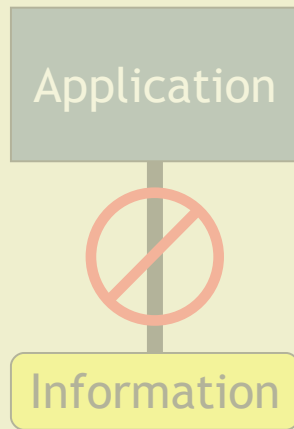


## New Style

Information is represented using open standards not under the control of a single vendor, and multiple applications can create and access it interchangeably.

Control is with the customer *not* the software provider.

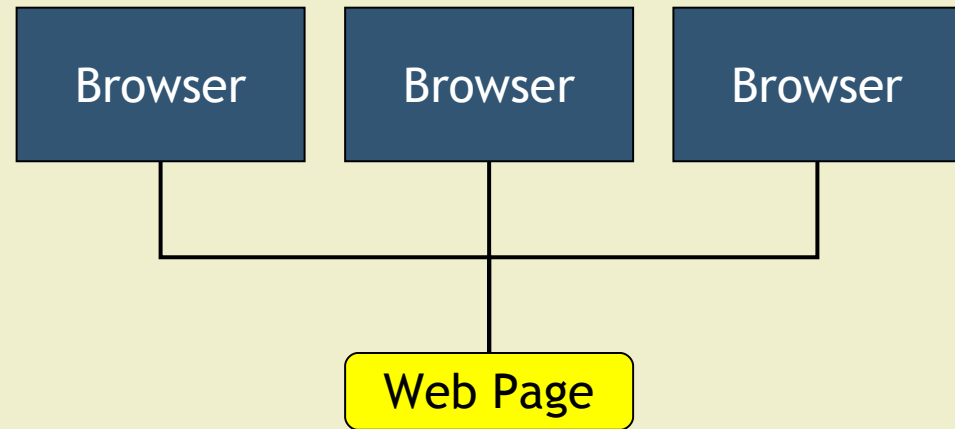
# This is a continuation of an unstoppable trend



## Old Style

Information is closely linked to the application that created it.

Control is with the software developer *not* the customer.



## New Style

Information is represented using open standards not under the control of a single vendor, and multiple applications can create and access it interchangeably.

Control is with the customer *not* the software provider.

# The effect of open standards

- Standardization drives software interoperability and interchangeability.
- This breaks the old-style dependence on proprietary methods, trade secrets, and single providers.
- It builds a strong foundation on which others can quickly build and innovate.
- When the playing field gets levelled in this way, we get increased competition with new people and ideas entering the marketplace.
- Standardization is scary to those who are afraid of losing marketshare, product cash cows, or cannot execute well in such an open world.

# Factors driving the shifts to open

- Feature exhaustion of applications
- Availability of at least partial open source solutions
- A shift to online applications like salesforce.com and Google Docs
- Consumer comfort with the web and distributed, interactive computing
- Young, technology-savvy people who don't accept the proprietary software business as the steady state
- Social networking software and sharing
- The rise of “mashups,” web services, Web 2.0, and virtual worlds



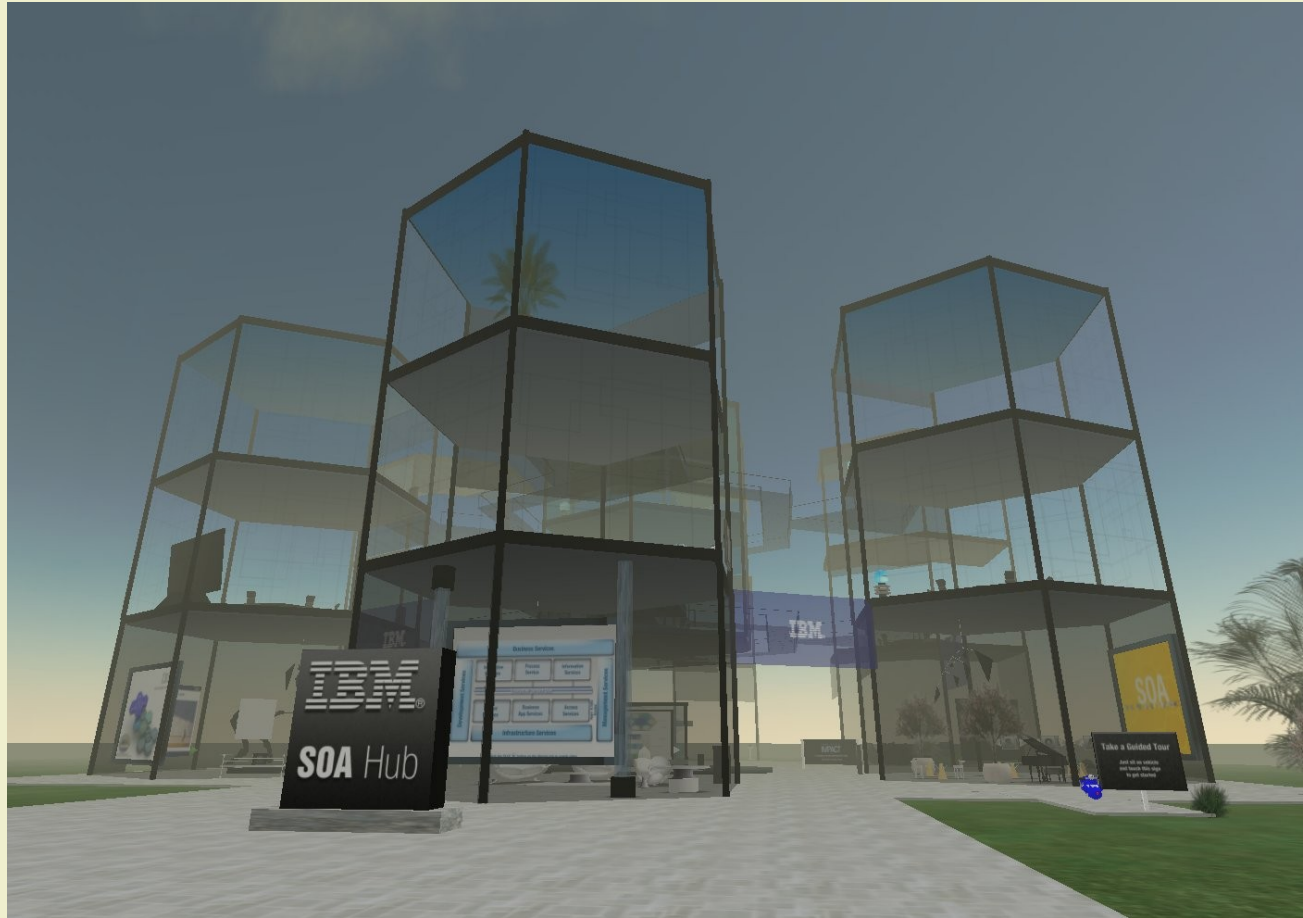
# SOA accommodates both open and proprietary

- ... but only in the right places
- Service Oriented Architecture is the leading trend in building flexible, efficient, distributed computing systems.
- Open standards exist that allow services to communicate in common ways.
- The services themselves have to be fast enough, secure enough, scalable enough, and interchangeable enough, *however they are implemented*.
- Thus open implementations may get the job done, but proprietary ones may get it done better.
- However, price considerations will factor into how the software solution will ultimately get implemented.

# Adjusting to open source

- Move to SOA and open standards.
- Put pressure on software vendors to open up APIs, protocols, and data formats.
- Accept that open source is not going away.
- Begin to use Linux in datacenter implementations.
- Create an internal cross-unit group to examine, manage, and educate about open source.
- Answer the question: what happens if customers demand all open source software in ten years? Five years?
- Be prepared to shift your business models to heavily make use of open source software.

# Example: IBM SOA in Second Life



# Example: IBM and retail in Second Life



# SL starts to go open source



## Second Life Viewer Goes Open Source

Have a hand in shaping the future of Second Life! Linden Lab announces the release of the Second Life viewer source code.

- Read the [Linden Lab blog post](#)
- Read the [press release](#)
- Check out our [Open Source FAQ and more](#)

### Linden Lab To Open Source Second Life Software

#### Creator of Leading 3D Virtual World Releases Source Code To Viewer Software

San Francisco, CA – January 8, 2007 – Linden Lab®, creator of 3D virtual world Second Life® ([www.secondlife.com](http://www.secondlife.com)), is releasing the code of its Viewer application to the open source software development community. Developers can now access the source code to the Second Life end-user software in order to make modifications, enhancements and to add new features. The move marks Linden Lab's continued commitment to building the Second Life Grid as an open, extensible platform for development, rather than a closed proprietary system.

Total Residents:	<b>5,678,694</b>
Logged In Last 60 Days:	<b>1,683,013</b>
Online Now:	<b>34,141</b>
US\$ Spent Last 24h:	<b>\$1,534,556</b>
LindeX Activity Last 24h:	<b>\$203,696</b>
<a href="#">SEE MORE economic statistics here!</a>	

# Implications

- Client goes first, so will lead to better user interface, faster graphics, better tools for developers, and better accessibility on more platforms.
- Server will go next, will yield improvements there but also possible different “worlds.”
- Shift will be from proprietary protocols to open ones.
- Money will be made from providing the service, both on the Internet and behind firewalls.
- They will stay competitive by providing a better quality of service, inworld conveniences, and better communities.
- Just as the web provided a catalyst for innovation and new business, so can this.
- No guarantees, but don't ignore it!

# Some trends and predictions

- Software interoperability standards (interfaces, protocols, and information formats) will not be proprietary in the long run.
- New software design will separate “that which must be open” and “that which might be proprietary.”
- Some common software application market categories will have their proprietary products collapse in the next decade.
- Watch what the kids are doing.

