Working with others

By admin
Published: 10/04/2007 - 08:32

Phil Chandler , September 2007

It is of course a truism, often repeated, [1] [2] [3] that the Internet has been the basis for a revolution in (remote) interpersonal communications, collaboration and data sharing. It is probably safe to say that there would be very few of the Free/Libre and Open Source (FLOSS) projects that exist today without the collaboration technologies the Internet supports. One of the many effects of the powerful tools FLOSS has put in to the hands of creative people is that it has potentially made them more independent. No longer are they reliant on specialists with access to expensive software and hardware to carry out aspects of their projects for them. Their limitations are now time and knowledge, not the lack of access. It is in fact precisely this issue that the Digital Artists' Handbook seeks to address, by providing authoritative information to guide practitioners in to new fields of endeavour.

The downside of this independence is that many artists find themselves more isolated, working alone at home rather than interacting with others at shared studios or where shared resources were previously found.

The Internet, being fundamentally a communications medium, offers potential solutions to this isolation, but the solutions themselves have, to date, largely dictated that collaboration happens in new ways, shaped by the technology. For some, the thousands of FLOSS coders for example, the tools have made possible projects that would otherwise be virtually inconceivable, but for other artists looking to enhance their existing practice with new digital methods the situation is perhaps more double-edged.

It maybe be useful to step back for a moment and consider what we mean when we talk about working, or collaborating with others. For a start it could be divided in to five broad types of collaboration:

Communication

This is the everyday backwards and forwards of discussion:

email: To most people the original and still the fundamental communication channel on the Internet. Although under constant assault by junk (spam) email, the simplicity, cheapness, robustness and asynchronous nature of the email system are it's fundamental advantages. There are probably few Internet users who are not already familiar with using email, and all internet service providers offer email services to users.

chat/instant messaging/IRC: where synchronous, near real time communication is required these text-based media have proved popular. The common theme is that users type what they want to say in to a chat window, and their words will appear almost instantly in the corresponding windows of those they are connected to. In all cases there are servers out on the Internet that act as hubs for this communication, and a level of complexity is introduced by needing to know which system the person you wish to converse with is using. Chat may be on a one to one basis or in a group situation. Client software is available for all common platforms, sometimes bundled with the operating system. These days systems tend to offer additional services such as the ability to send a file to someone you're chatting to, or even to use voice and video communication. In other words technologies such as chat, VoIP and video conferencing (see below) are converging. There are even websites that allow you to set up a chat session without even installing any software. [4]

Voice over IP (VoIP) / 'Internet telephony': This describes the technologies that use the basic protocols of the Internet to transport voice communications between users. This has allowed computers to be used as telephony devices, and for users to turn their high speed Internet connections in to cheap international telephone lines. Due to the way the Internet presently works it is often hard to achieve equal quality of service compared to a traditional circuit-based telephone connection, however the low cost has encouraged a boom in VoIP's use. It has been popularised by the proprietary Skype software [5] but many open source equivalents exist [6] [7] . As with chat services, you need to create an account on the VoIP service you wish to use, and install the relevant software. Whilst computer to computer connections are generally free, connecting to a traditional phone is usually a paid for service.

Content management systems (CMS), wikis and discussion forums: Although these equally fall in to the area of websites and self-publishing, I mention them here due to their use for community and project websites, where registered users can discuss, publicise and share work. One example is the Lancashire Artists' Network site [8] , based on the open source Drupal content management system [9] . CMS' provide a framework where non-technical users can create web content of various types via their browser without needing to understand how to code. Some CMS's support, in addition to text and images, audio, video, as well as specialised areas such as discussion forums. Wikis fit in to a similar model, however tend to be simpler, more focussed tools which are designed to make adding text and linking information together particularly easy. The most famous example of a wiki is probably Wikipedia [10] , however many collaborative projects have a wiki for members to discuss and develop project documentation [11] . Forums are webpages that are designed around the idea of threaded discussions, where users post topics and replies to those topics.

mobilephones/SMS: Of course mobilephone networks are not strictly speaking part of the Internet since they do not use the Internet's protocols, however they are in many countries a ubiquitous communication system. The wide availability of the system is somewhat offset as a collaboration medium by the present general restriction to one-to-one communication and mainly voice and text messages. As bandwidth capacity increases and costs fall it may well become more viable to expand in to video use. A number of projects, for example Twitter [12] , are exploring how to integrate mobile and web-based systems. Whether such projects are serious collaborative tools or toys for the under-employed is still open to debate.

social networking websites: These are essentially highly branded content management systems, allowing users to present their content within fairly tightly controlled home pages. As the market matures various networks are competing on the functionality they offer in terms of managing content and connections to other members. [13] [14] [15] The challenge facing the large networks at present is how to convert their large memberships in to a profit without alienating them. [16]

virtual environments: the natural extension of social networking and online gaming is the 3D virtual environment, the most high profile one at the time of writing being Second Life [17] . Although in essence a relatively immersive chat client, (with voice features now appearing), from a collaboration point of view virtual environments offer some additional features such as the ability to embed, e.g. video content, in virtual objects, to allow sharing and performance of works. As with nearly all online environments, the tools available for use online are prescribed by the environments themselves.

video conferencing: This can cover a wide range of technologies and qualities. At one end there is the low resolution, somewhat jumpy output of cheap webcams combined with domestic broadband connections. As mentioned above such functionality is increasingly being integrated with text based chat and VoIP software. At the other end of the spectrum there are the high quality, high bandwidth options most commonly found in corporations and universities, who have access to the expensive Internet connections required. [18]

Sharing work

Although there is often no clear demarcation, since you can for example send files by email, this refers to technology more explicitly designed around sharing data files:

project websites: Aside from the discursive uses of project websites and wikis, these provide some of the simplest means of widely disseminating digital works. Content management systems often have built in functionality to manage file repositories and control access to the contents, for example restricting downloads to registered users.

file transfer protocol (FTP) sites: The traditional method of disseminating large files over the Internet, this method is perhaps falling from favour compared to the various web-based options. It is nevertheless useful to have a good FTP client [19] in the tool kit to access such sites. Incidentally if you are managing your own web space and need to upload files to your web server, if your provider supports it, a much more secure alternative to FTP is SFTP [20] [21] . SFTP is supported 'out the box' in most Linux file managers.

code management tools such as CVS: For more complex projects, especially where it is important to track what changes have been made to work by multiple collaborators, a code management system is vital. Simon Yuill explores this in detail in his article, however from a new user point of view the first requirement is probably a friendly graphical client to access such systems. You need of course to have the right client for the particular system you're using but examples are [22] [23] [24].

peer-to-peer filesharing: If media and record industry hype was to be believed [25] [26] then peer-to-peer filesharing networks are the font of all evil. However for collaborative projects that need to disseminate very large files, such as major software projects or video sharing, the ability to share the bandwidth load across many users has proven very useful. In essence peer-to-peer networks work by some users downloading files from an original 'seed' server, and then being willing to allow others to download the same files from them. As time goes on more copies proliferate across the Internet meaning that no single server or user takes the bulk of the demand. A technologically advanced example is Bittorrent [27] .

Sharing Ideas

Obviously an idea can be shared by describing it in words, sending a code example or other sample of work. Sometimes though more structured methods aid the collaboration process. One is the idea of design patterns as Simon Yuill discusses in his article, however a more general and widely popular method is that of mindmapping with some good free tools available [28] . Mindmapping is an example of a visualistion tool to allow individuals and groups to share and refine their ideas. These can be coupled with feedback mechanisms such as annotation, voting and messaging [29] .

Project Management

For larger or longer term projects more formal management tools may be appropriate. Some are built in to code sharing sites such as Sourceforge [30] , while others provide a specialist approach, such as dotProject [31] which can be hosted on a website and thus accessible to all members of the project. For many, projects wikis provide the necessary level of co-ordination. The term groupware is sometimes applied to integrated packages of collaboration software including email, instant messaging, and document management functions. [32] [33]

Working in real time

The methods mentioned so far are either a) asynchronous and/or b) essentially for talking about the work, or for making some version of the work available for others to see, try, or perhaps alter themselves. There are simple online creative tools [34] , and even some that allow more than one person to work at the same time [35] . These are obviously not comparable with the tool sets provided by such software as GIMP [36] and Inkscape [37] . It is also noticeable that many of these tools tend to be basic drawing tools. What is still at an early stage are ways for artists to actually work together, remote from each other, on the same piece of work, at the same time. There are interesting experiments going on, such as Furtherfield's Visitors' Studio [38] , however most have been orientated towards developing new tools specially for use on the Internet. But why should we leave the tools we are used to using 'at home' when we go online? There is a new frontier to be explored as to how we make the tools we are comfortable with available to us when we start working with others remotely. Perhaps the area where the most progress has been made so far is in the audio world with technologies such as streaming, Pure:Data and Open Sound Control (OSC) [39] . Going further one idea might be to have an underlying framework that can share sessions between any appropriately adapted software. This is perhaps somewhat analogous to the role JACK [40] plays in the free audio software world, where any piece of software that has JACK support built in can exchange audio data with other similarly equipped applications. In this case what would be shared between two remote instances of a particular application would be information about their state, key presses and mouse clicks etc.

Whatever technical solution is found it is to be hoped that it won't be too long before a new level of working with others becomes available, whereby we can work interactively on the same piece of work, using the full power of the tools we've invested time learning. Perhaps one day we will be able to take our favourite tools in to virtual 3D studios and work closely alongside our chosen collaborators wherever in the (real) world they might be.

Notes

[1] “UN Fights US over internet's future” The New Zealand Herald 2005

http://www.nzherald.co.nz/section/2/story.cfm?c_id=2&amp;ObjectID=10354409 [Accessed 12/12/2007]

[2] “Internet and e-mail policies” ACAS 2006 http://www.acas.org.uk/index.aspx?articleid=808 [Accessed 12/12/2007]

[3] “How the Internet has Changed Our Lives” Neilsen/Netratings 2003 http://www.nielsennetratings.com/pr/pr_031229_uk.pdf [Accessed 12/12/2007]

[4] http://www.gabbly.com

[5] http://www.skype.com

[6] http://www.ekiga.org

[7] http://www.linphone.org/index.php/eng

[8] http://www.lanarts.com

[9] http://drupal.org

[10] http://www.wikipedia.org

[11] http://puredyne.goto10.org

[12] http://twitter.com

[13] http://www.bebo.com

[14] http://www.myspace.com

[15] http://www.facebook.com

[16] http://news.bbc.co.uk/1/hi/technology/7130349.stm [Accessed 12/12/2007]

[17] http://secondlife.com

[18] http://www.videocentric.co.uk/videoconferencing/videoconferencing.shtml#...

[19] http://filezilla-project.org

[20] http://www.winscp.com

[21] http://cyberduck.ch

[22] http://www.tortoisecvs.org/ and http://tortoisesvn.tigris.org

[23] http://sente.epfl.ch/software/cvl

[24] http://www.twobarleycorns.net/tkcvs.html

[25] http://www.theregister.co.uk/2007/12/12/copyrights_and_wrongs

[26] http://www.riaa.com/physicalpiracy.php?content_selector=piracy_details_o...

[27] http://www.bittorrent.com/what-is-bittorrent

[28] http://freemind.sourceforge.net/wiki/index.php/Main_Page

[29] http://en.wikibooks.org/wiki/Collaborative_Networked_Learning:_A_Guide/S...

[30] http://sourceforge.net

[31] http://www.dotproject.net/index.php

[32] http://www.icthubknowledgebase.org.uk/opensourcegroupware

[33] http://www.opengroupware.org/en/applications/index.html

[34] http://selfportraitchallenge.net/2007/02/27/march-online-tools

[35] http://www.imaginationcubed.com/Imagine

[36] http://gimp.org

[37] http://www.inkscape.org

[38] http://www.furtherstudio.org/live

[39] http://www.yourmachines.org/tutorials/collab_stream.html

[40] http://jackaudio.org

" >

span-->

- Collaborative development: theory, method and tools
- Social networking

‹ Pure Dataflow - Diving into Pd up Collaborative development: theory, method and tools ›

Collaborative development: theory, method and tools

By dummy
Published: 09/06/2007 - 23:00

Simon Yuill , September 2007

One of the most significant features of the development of Free Software projects is the immense numbers of people involved. It has been estimated that in the year 2006, almost 2,000 individuals contributed to the coding of the Linux kernel alone, which is just one, admittedly very important, software project amongst the many thousands that are currently in development worldwide [1] . Alongside such numbers, other Free Software projects may be the work of just one or two individuals, or changing sizes of groups who come and go throughout the lifetime of a particular project [2] . One of the reasons for the success of Free Software has been the creation of tools and working practises to support such levels and varieties of involvement. Whilst often tailored for the needs of programming work, many of these tools and practises have spread into other areas and uses, with tools such as the wiki, which provides the technical infrastructure for the online encyclopedia Wikipedia, being one of the best known and most widely used [3] .

Collaborative development emphasises the idea of people working together and this is a key aspect of the Free Software ethos. In writing and talking about Free Software, Richard Stallman repeatedly speaks about the 'Free Software community' and the idea of software as something created by a community and belonging to a community, rather than it being exclusively the work of one individual or corporation [4] . 'Collaboration' in itself, however, does not fully cover the full implications of a Free Software practise. Free Software is about making the work you create available for others to also create from. It is a form of production that enables and encourages others to produce. Although it is often associated with the idea of 'appropriative' art and remix culture (in the sense of re-using material that others have created), what Free Software proposes is quite different. This goes beyond the idea of the simple re-use of materials and is better described as a form of 'distributive' practise [5] . Free Software is distributive because in sharing the source code along with the software, it distributes the knowledge of how that software was made along with the product, thereby enabling others to learn and produce for themselves from that. This is something which neither appropriative art or remix culture do. Free Software is something more than just collaboration, or just sharing. It recognises that invention and creation are inherently social acts that produce their own forms of sociability through this process [6] . The collaborative potential of Free Software therefore arises out of this basic principle of distributive production. Given this basis, it is no surprise that many of the actual tools created by the Free Software community for its own use are inherently sociable in their design and are geared towards a process in which people can build and learn together [7] .

This chapter provides an overview of some of the development tools currently in use in Free Software projects. Each tool is described in terms of how it works and how it might be applied and integrated into different types of project. In some cases it also discusses the background to the tools, how they came into existence and some of the conceptual underpinnings of how they handle their task. This does not cover all of the tools you would use to create a software project however, only the ones that are necessary for people to work together. For that reason there is no discussion about different code editors, compilers or debugging tools. Nor does it cover issues such as designing and planning a software project from scratch or the typical development cycle involved in that. These are all substantial issues in their own right and are covered elsewhere [8] . The focus here is specifically on tools that help you work with other people, and how to make what you do available to others.

Communication

Any project involving a number of people working together relies on efficient communication between its members. Email and internet chat (or messaging) are the two most widely used forms of online communication and play an important role in any collaborative project. Email is best suited for sending documents and detailed information amongst people, whilst chat is best for realtime discussion. When working in a group it is obviously important to keep everybody in the communication loop, so most projects will make use of a mailing list to coordinate emails between people and a designated chat channel for discussion. As it is also useful to keep a record of communication between people, systems which provide built in storage of messages are a good idea.

Mailman is a mailing list system created as part of the GNU project [9] . It is easy to setup and use, and is provided with a web-based interface that allows people to configure and administer multiple lists through a conventional web-browser. One of its best features is a web-based archive system which stores copies of all messages on the list and makes them available online. The archive allows messages to be searched and ordered in different ways, for example, by date, sender or subject, as well as arranged into topic threads. Each thread starts with a new message subject and follows the responses in a structured order that enables you to see how the discussion around it developed. The archive can be made open to the public, so anyone can read it, or protected so that only people who are subscribed to the list can view it.

It is common for a large project to have multiple mailing lists, each one dedicated to a particular kind of discussion area or group of people. For example there may be one for developers who are writing the code, and another for people writing documentation and instruction materials. There may also be one for users who can ask and discuss problems in using the software, and sometimes additional lists for reporting bugs to the developers and putting in requests for new features in the code. Given all this activity, mailing list archives provide an important information resource about the development of a project, how to use the software created in it, or different approaches to programming problems. They can be a valuable resource, therefore, for someone learning to program, or for someone joining a long-running project who wants to understand its background and personal dynamics better. More recently, a number of academics making studies of FLOSS development have also analysed mailing lists as a way of understanding some of the social structures and interplay involved in such projects [10] .

There are many forms of chat and messaging system in use today. IRC (Internet Relay Chat) is one of the oldest systems dating back to the late 1980's and is one of the most widely used. IRC can support both group discussions (known as conferencing) or one-to-one communication, and can also provide a infrastructure for file-sharing [11] . Channels are used as a way of defining distinct chat groups or topics in a way that is analogous with dedicated radio channels. Each channel can support multiple chatrooms which may be created and disposed off on the fly. A chat can be completely open to anyone to join or may require password access. For extra security IRC chats can also be encrypted. IRC is not itself a tool, but rather a protocol that enables online discussions. To use IRC requires a server for hosting the chats and clients for each of the participants. There are many public IRC servers, such as freenode.net, which provide free IRC channels for people, and these are often sufficient for most small-scale projects. Larger projects may wish to have their own dedicated IRC servers.

When working together it is common for programmers to have an IRC client running on their machine where they can ask quick questions to other developers or handle discussions in a more immediate and flexible manner than email permits. IRC can also be an excellent way of taking someone through a process step-by-step remotely. Whilst having a public archive of a mailing list is common, it is less usual to store IRC discussions. As IRC is a sociable medium there can be a lot of ephemeral or trivial content exchanged between people (as in everyday chat between friends) and storing all of this for public consumption is probably not worth the disc space. All good IRC clients, however, have built-in logging capabilities which allow the user to store the discussion in a text file for later reference.

Logo of BitchX, a free IRC client

There are a number of IRC clients for GNU/Linux. ircII is the oldest IRC client still in use, and one of the first to be created. BitchX is a highly popular offshoot of ircII originally written by "Trench" and "HappyCrappy" and has a certain style of its own synonymous with aspects of hacker and l33t culture. irssi is a more recent client written from the ground up to be highly modular and extensible. All of these clients can be scripted to handle various forms of automated tasks, such as chat bots, or integrate with other processes [12] .

Sharing Code

When Richard Stallman first made EMACS available to the world, and Linus Torvalds released Linux, people generally made feedback and contributions to the code via newsgroups and email communication [13]. This is fine for small-scale projects, but as the number of people and amount of work involved in a project grows, this kind of interaction between the code and developers becomes quickly unmanageable. In the late 1980's, Dick Grune, a programmer working on the Amsterdam Compiler Kit (ACK) project, wrote a set of scripts for making changes to source code files in a way that ensured the code could be changed at different times by different people without accidentally losing one person's input [14]. These scripts were called the Concurrent Versions System (CVS), as they enabled different versions of the code to exist side by side as well as be integrated into one another. They were later developed into a self-contained tool that is still one of the most widely used 'version control' or Source Code Management (SCM) systems.

Alongside CVS, a number of other SCMs have become popular, these include Subversion (SVN), Git and darcs (David's Advanced Revision Control System) [15]. Each has its own particular features and the design of a particular SCM system often carries an implicit model of code development practises within it. There are, however, various common features and tasks that are standard across different SCMs. SCMs generally work on the basis that the source code for a given software project is organised into a set of distinct files. The files are stored in a 'repository' which is often simply a normal file directory on a computer containing both the source files and meta-data (background information) about the changes that have been made to them - sometimes the meta-data is held separately in a database. Programmers can work on the source files independently, submitting their changes and additions to the repository and obtaining copies of the files as they are changed by other programmers. The SCM manages this process and enables the different contributions to be combined into a coherent whole.

The main tasks normally performed by a SCM are:

1. to store each new version of a source file without erasing previous versions, and to enable earlier versions to be retrieved. These tasks are often known as 'committing' changes and 'checking out' revisions.

2. to associate specific changes in a source file with a specific programmer. This is usually done by requiring programmers to log into the SCM under a specific username each time they 'commit' their changes to the code.

3. to provide a history of changes in a given source file and enable files to be compared to one another. A comparison is often called a 'diff' after the UNIX commandline tool for displaying differences between two files.

4. to warn or indicate when two or more programmers have attempted to change the same lines of code, known as a 'conflict'.

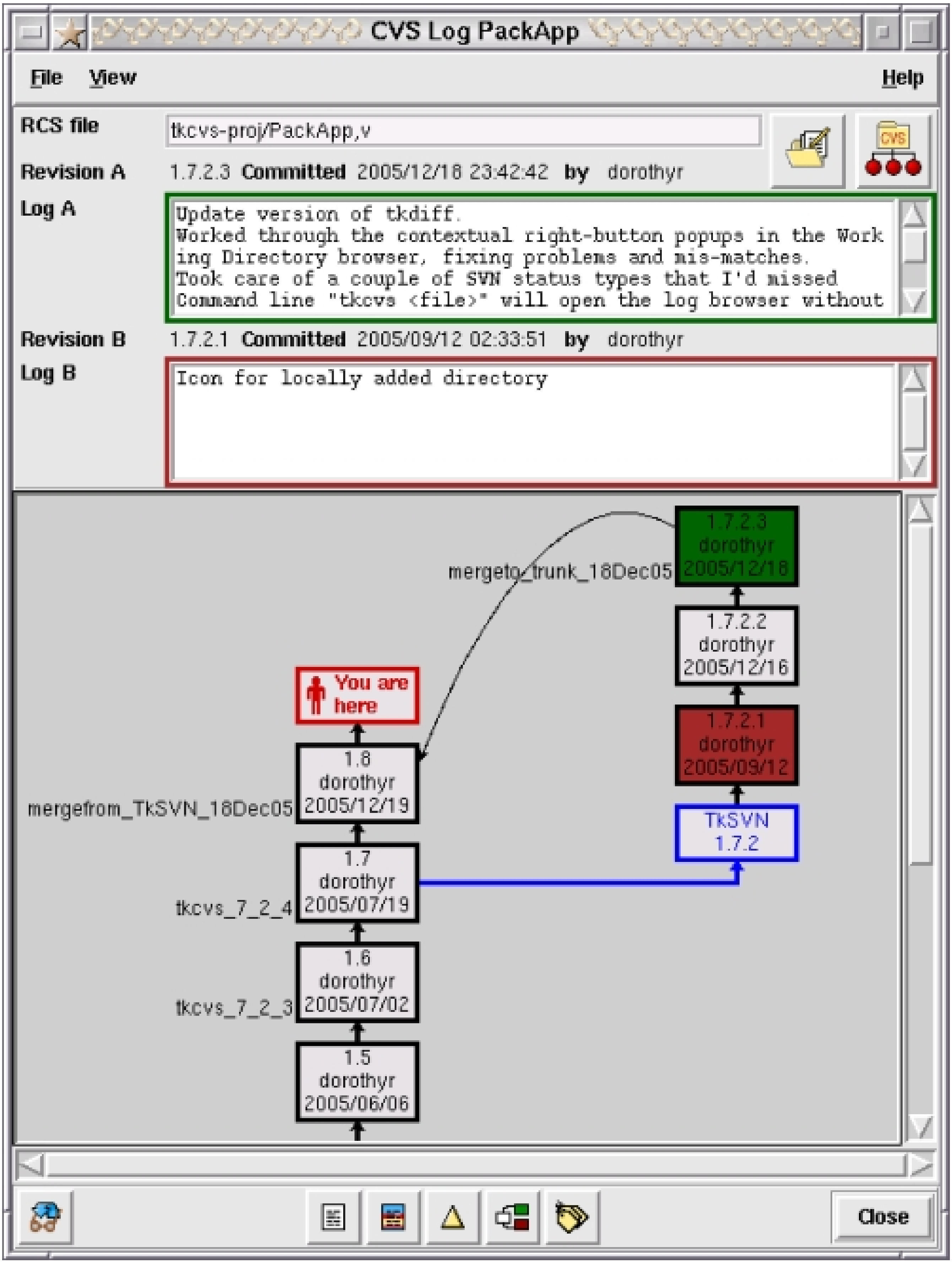
5. to enable parallel versions of the source files to be maintained alongside each other. This is known as 'branching', with each parallel version being a separate branch.

6. to combine different parallel versions of a source file into one final version, known as 'merging'.

One of the most significant differences between different types of SCM lies in the kind of repository system that they use. The key distinction being between those that use a centralised repository and those that are decentralised. CVS and SVN both use centralised repositories. In this case there is only one copy of the repository which exists on a specific server. All the programmers working on the project must commit their changes and checkout revisions from the same repository. Git and darcs both use decentralised repositories. With this approach, every programmer has a complete copy of the repository on their own machine, and when updating code commits their changes to that local version. The different repositories are then periodically updated to bring them into sync with one another through use of a networking mechanism. This communicates between the different repositories and merges their changes into one - a process known as 'pulling' and 'pushing' the changes between repositories. The decentralised approach is, in some ways, similar to a situation on a centralised server in which each programmer is working in their own branch. This may seem like a counter-intuitive approach towards maintaining coherence between the different programmers' work, and, perhaps for this reason, centralised systems such as CVS and SVN are often the more popular. Decentralised SCM systems have, however, shown themselves to be extremely effective when dealing with very large projects with many contributors. The Linux kernel, with nearly 2,000 programmers and several million lines of code, is one such large project, and the Git system was specifically written by Linus Torvalds for managing its development. The reason for the effectiveness of decentralised repositories for large projects may be that with the greater the number of people making changes to the code, the greater the amount of 'noise' this activity produces within the system. Within a centralised system this noise would affect everybody and, in doing so, amplify itself in a kind of feedback loop. Within a decentralised system, however, the activity, and resultant 'noise', is spread over many smaller areas thereby making it more contained and less likely to effect others. With a large project furthermore, in which many small groups may be focusing on different areas of its development, having multiple repositories also enables each group to focus on the key issues that it is addressing. In this sense each repository articulates a different 'perception' of the project.

Darcs has a very particular approach to updating code, called the "Theory of Patches", which derives from author David Roundy's background as a physicist [16] . This adopts concepts from Quantum mechanics and applies them to a model of code management in which any kind of change to the source code is abstracted as a 'patch' [17] . A variety of different operations are implemented to apply a patch to a repository. The advantage of Roundy's patch system is that it enables a variety of tightly tuned forms of merging to be performed that other SCM systems do not support, such as only updating replacements of a specific word token within different versions of source files (most SCM systems would change each entire line of code in which the token is found), or applying changes in a given order or combination (called sequential and composite patches).

For the programmer, interacting with a SCM system is generally done through a simple commandline tool. This enables the programmer to more easily integrate the SCM interface with whatever code editor they prefer to work with, and many code editors, such as EMACS, provide a mechanism through which the SCM can be directly accessed from the editor [18] . Various graphical SCM clients have also been developed however. The TkCVS and TkSVN tools provide GUI based interaction with the CVS and Subversion repository systems [19] . In addition they can display visual charts showing the structure of the source files within the repository, and mappings of branches, revisions and merges within a project.



RCS file tkcvs-proj/PackApp,v

Revision A 1.7.2.3 Committed 2005/12/18 23:42:42 by dorothyr

Log A

```
Update version of tkdiff.
Worked through the contextual right-button popups in the Work
ing Directory browser, fixing problems and mis-matches.
Took care of a couple of SVN status types that I'd missed
Command line "tkcvs <file>" will open the log browser without
```

Revision B 1.7.2.1 Committed 2005/09/12 02:33:51 by dorothyr

Log B

```
Icon for locally added directory
```

Screenshot of Branch Browser in TkCVS

As with mailing lists, code repositories also provide a great deal of information on how a software project evolves. A number of tools have been developed to generate statistics and visualisations of different aspects of a repository, such as showing which files the most activity is focused around and the varying degrees in which different programmers contribute to a project [20] .

A number of free public code repositories are available on the internet, such as the GNU Project's Savannah repository. Alongside the SCM system, these often provide a packaged set of tools for managing a project. These are discussed in more detail below.

Sharing Ideas

A good piece of software is more than just so many pages of code. A good piece of software is a good set of ideas expressed in code. Sometimes those ideas are expressed clearly enough in the code itself, and there are those who argue that a good programmer is one who can express herself or himself clearly in this way [21] . Sometimes, and perhaps more often with larger projects, the ideas are not so easily contained in a single page of code or are implicit more within its bigger structure or how particular components within a program interact with one another.

One approach to this issue has been to provide comments and documentation notes directly within the text of the program code itself, and most programming languages include a special syntax for doing this. Some languages, such as Forth and Python, have taken this idea a step further and enable the documentation notes to be accessible from within the code whilst the program using it is running [22] . For almost all programming languages, however, a set of tools known as 'documentation generators' are available which can analyse a set of source files and create a formatted output displaying both the structure of the code and its accompanying comments and notes. This output can be in the form of a set of web pages which directly enable the documentation to be published online, or in print formats such as LaTeX and PDF. 'Document generators' are often dedicated to a particular programming language or set of languages. PyDoc, EpyDoc and HappyDoc are all systems designed specifically for Python [23] . Doxygen is one of the most widely used document generators and supports a number of languages including C++, Java, and Python [24] . When used in conjunction with Graphviz (a graph visualisation tool), Doxygen can also create visual diagrams of code structures that are particularly useful for understanding the object-orientated languages that it works with [25].

Another, quite different, approach lies in the use of Design Patterns. Design Patterns are a concept adapted from the work of architect Christopher Alexander. Alexander was interested in developing a means through which architects and non-architects could communicate ideas of structure and form, so that non-architects, in particular, could envisage and describe various forms of domestic and urban building systems and thereby have greater control over the design of their built environment [26]. As applied to programming, Design Patterns provide a means of articulating common programming paradigms and structures in ways that are separate from actual code and can be explained in ordinary language [27]. They focus on the structure and concepts, the ideas, behind the code, rather than the specifics of how a particular piece of code is written. This allows ideas to be transferred more easily from one project to another even if they do not share the same programming language. In 1995, the Portland Pattern Repository was set up to collect and develop programming Design Patterns through a SCM-like system. This system was the first ever wiki [28]. A wiki is basically a web-based text editing system that provides a simplified form of version control like that used in CVS. The wiki created a simple yet robust way of enabling the content of web-pages to be updated in a collaborative fashion. It was quickly realised that this could be applied to all manner of different topics that could benefit from collaborative input, and the wiki spread to become one of the most widely used content management systems on the internet, with the most famous example being the Wikipedia encyclopedia [29].

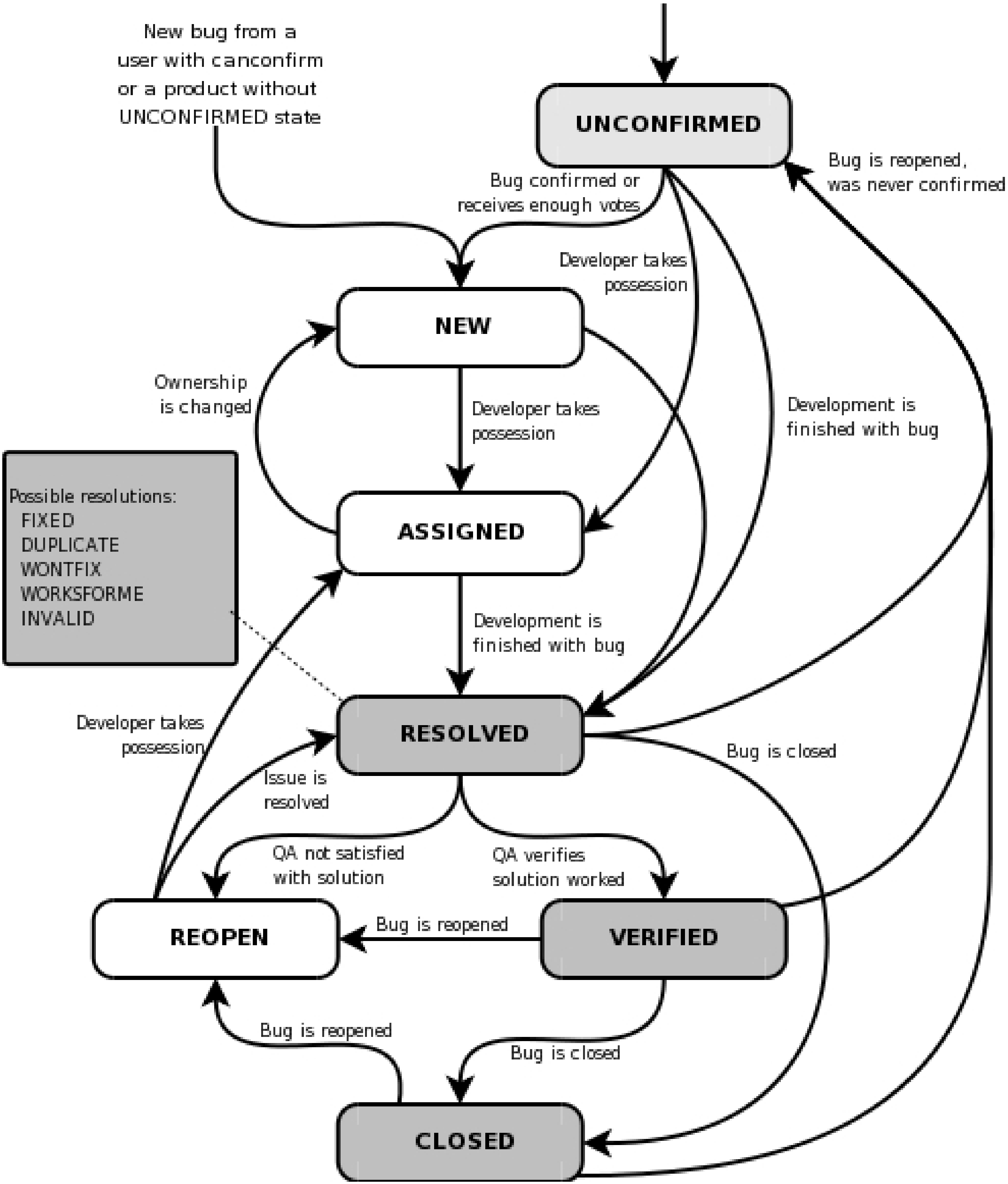
Nowadays, a whole range of different wikis are available, ranging from simple text-only systems to more sophisticated ones supporting images and other media content [30]. Within a programming project wikis provide a simple and effective publishing system for updating information and documentation related to the project. They can augment the primarily technical information produced through a documentation generator to provide more descriptive or discursive information such as user manuals and tutorials.

Project Management

The various tools described so far can all be combined into more comprehensive packages that provide a complete framework for managing a software development project. Often these are available for free through public development repositories such as Savannah and BerliOS, although it is also possible to set up your own system independently. Both Savannah and BerliOS are specifically geared towards supporting fully Free and Open Source projects. Savannah is run by the GNU Project and BerliOS is run by the German group FOKUS, both are not-for-profit organisations [31] . SourceForge provides a similar free service but is run as a commercial enterprise [32] . These sites are often described as 'developer platforms', a platform in this case meaning a web-based service that facilitates or provides an infrastructure for other people's projects [33] .

To use one of these platforms you must first register as a member and then submit a description of your proposed project, each site will provide details of what kind of information they wish you to provide in this description. The project will be reviewed by a selection panel within the organisation or user community that runs the platform and, if accepted, you will be given an account under which your project will be hosted. In addition to resources such as a code repository and mailing lists, all of these platforms also provide tools such as bug tracking systems and web-site facilities for publishing development news and documentation.

Bug tracking systems are tools that enable users and developers working on a project to report bugs they have discovered in the software. These reports can then be parcelled out amongst the developers, who will report back to the tracking system when the bug has been fixed. The status of each bug report is normally published through an online interface. This is particularly useful for any medium-to-large scale project and provides one useful development mechanism that non-programmers can contribute to. One well-known and widely used bug tracking tool is Bugzilla which was originally created for the development of the Mozilla web-browser [34] .


New bug from a user with canconfirm or a product without UNCONFIRMED state
UNCONFIRMED
Bug is reopened, was never confirmed
Bug confirmed or receives enough votes
Developer takes possession
NEW
Ownership is changed
Developer takes possession
Development is finished with bug
Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
ASSIGNED
Development is finished with bug
Developer takes possession
RESOLVED
Bug is closed
Issue is resolved
QA not satisfied with solution
QA verifies solution worked
REOPEN
Bug is reopened
VERIFIED
Bug is reopened
Bug is closed
CLOSED

Bugzilla Lifecycle diagram

A similar service is a 'feature request' system. This enables users and developers of a software project to submit requests and ideas for features that are not currently part of the project but could be implemented at a later stage. As with the bug tracker it provides a good way for non-programmers to contribute to a project. The Python project is one example of a project that has made good use of this facility and has its own system called PEP, short for Python Enhancement

Proposal [35] .

If you wish to set up your own development platform a number of packages are available. Savane is the system used by Savannah and BerliOS [36] . Trac is another system more widely used in smaller projects. It includes a wiki system and is much easier to customise in order to create a self-contained project site with its own look [37] .

Sharing spaces

Whilst online development has been a feature of Free Software, it would be wrong to assume that everything happens exclusively online. The online world supports a rich variety of social groupings many of which have equivalents within the offline world. These often provide strong social contexts that link Free Software practise to local situations, events and issues. Linux Users Groups (LUGs) are one of the most widespread social groups related to Free Software. These groups normally include a mix of people from IT professionals and dedicated hackers, to hobbyists and other people who use GNU/Linux and other forms of Free Software in their daily lives. LUGs are usually formed on a citywide basis or across local geographic areas, and there will most likely be a LUG for the town or area in which you live [38] . They combine a mailing list with regular meetings at which people can share skills, discuss problems and often also give talks and demonstration of projects they are working on or are interested in. Similar groups with a more of an artistic angle to them include Dorkbots, which include all forms of experimental tinkering and exploration of technology, and groups such as Openlab which operate more like an artists collective for people working with Free Software-based performance and installation [39] .

Nerdcore Central, hackmeet

Hacklabs and free media labs are actual spaces, setup and run on a, mostly, voluntary basis. A typical lab will make use of recycled computer hardware running Free Software and providing free public access to its resources. As spaces they can provide a base for developing projects, whether that involves actual coding or a place for the developers to meet. Many labs also provide support for other local groups, such as computers for not-for-profit organisations. Hacklabs generally have a strong political basis to them and will often be linked with local Indymedia and grassroot political groups [40] . They are frequently, though not necessarily, run from squatted buildings. Such action makes a conscious link between the distributive principles of Free Software and the distributive principles of housing allocation promoted by the squatting movement [41] . Hacklabs have developed primarily in Italy and Spain, such as LOA in Milan and Hackitectura in Barcelona, but the Netherlands and Germany also have many active hacklabs, such as ASCII in Amsterdam and C-Base in Berlin [42] . Historically, these have had close associations with the Autonomist and left-wing anarchist groups. Free media labs are usually less overtly political, often with an emphasis on broader community access. Historically they can be seen to have grown from the ethos of the community media centres of the 1970s and 80s, with Access Space, set up in Sheffield in 2000, being the longest-running free media lab in the UK [43] . Free media labs are more likely to use rented or loaned spaces and may even employ a small number of staff. It would be wrong, however, to suggest that there are sharp divisions between hacklabs and free media labs, rather, they present different models of practise that are reflective of the social and economic contexts in which they have emerged. The government-sponsored Telecentros in Brazil offer a different example of such spaces, varying from providing low-cost public computer access to supporting more radical projects such as MetaReciclagem who focus on training people to become self-sufficient with technology [44] . An important offshoot of MetaReciclagem were the Autolabs setup by the Midiatactica group in the favelas of Sau Paulo [45] .

A different use of space to bring people together is found in the many forms of meetings and collective events that are characteristic of the Free Software world. Again these range from self-organised events such as hackmeets to more official festivals and conferences. These may either act as general meeting points for people to come together and exchange contacts, skills and ideas, or as more focused short-term projects that aim to achieve particular tasks such as pulling resources to complete a software project. Examples at the more commercial end of the spectrum include the Debian conference, DebConf, and Google's Summer of Code [46] . The Chaos Computer Club meetings and TransHack are examples from the more grassroots end of things, whilst Piksel and MAKEART are two examples of artist-led events that combine aspects of a hackmeet with an arts festival [47] .

Starting your own project

If you are starting up a Free Software project of your own for the first time, the best thing to do is take a look around at how other projects are run and the tools that they use. Take a bit of time to familiarise yourself with the different tools available and pick the ones that feel best suited to your needs and way of working. A good way to do this is to create a little dummy project on your own machine and test out the tools and processes with it. Another way of building up experience is to join an existing project for a short while and contribute to that, or even just join the developers mailing list to see what kind of issues and ideas come up there. One of the best ways you will find support is through the Free Software community, so make contact with your nearest LUG, see if there are any hacklabs and free media labs in your area and take a trip to the next hackmeet, conference or festival that comes your way.

If you are working with other people, see what the more experienced ones within the group can tell you about how they work and what tools they use. If your entire development team is new then it is probably best to come to a collective decision about what tools you are working with and choose the same set of tools for everyone so that you are learning the same things together. There are certain tools which you will have to share anyway, for example, you will all have to use the same SCM whether it is CVS, SVN, darcs or something else. Other tools can be different among different developers without causing conflicts. You should be able to use different code editors and different IRC clients for example without any problems.

Assuming that your first project will probably not be too ambitious, you may not need to use tools such as bug trackers, however, there is no harm in integrating these into your working process in order to build up a sense of how they operate. If you want to utilise the full range of development tools it is probably best to opt for an account with a public developer platform such as Savannah or BerliOS, or to use an all-in-one package such as Trac.

Not just code - Free Software as artistic method

The early years of internet art were characterised by a number of projects based around large-scale public collaborations often in the form of texts or images that could be added to and altered by different people [48] . These projects emerged at a time when Free Software was only just beginning to gather momentum and so they did not make use of the approaches it offers. There is no reason why a collaborative project that is producing something other than computer code could not make use of the tools and practises described here, however, and it has been suggested that a CVS system can be understood as a form of 'collaborative text' in its own right [49] .

None of these early projects, however, fully realises the idea of a Free Software practise as an artistic method. For all the collaborative nature of their construction, these projects often focus on a self-contained artefact as their sole end. They are collaborative but not distributive. More recently, however, a number of projects and approaches have been developed which not only use Free Software tools to create the work, but also realise a form of the Free Software ethos in how the projects are engaged with. The Openlab group in London have been developing small software systems for supporting improvised group performances which are themselves open to be reprogrammed during performance [50] . Social Versioning System (SVS) is a project which combines reprogrammable gaming and simulation systems with an integrated code repository. As with the Openlab tools, reprogramming is one of the key modes of engagement, but in linking this to a repository in which the changes and contributions of players can be examined, it aims to present coding itself as a form of dialogue, with the gaming projects critically exploring forms of social systematisation and urban development [51] . Plenum and Life-Coding are two performance projects which have adopted a similar approach [52] . All of these projects have, in different ways, adapted aspects of Free Software development tools and practises. Whilst less consciously designed as an artwork in itself, the pure:dyne project also demonstrates such an approach. Here, artists have collaborated in the development of a version of the GNU/Linux operating system specifically geared towards using and supporting Free Software based artistic tools and projects [53] . Pure:dyne is built on top of dyne:bolic, a version of GNU/Linux for artists and activists focusing on streaming and media tools [54] . Dyne:bolic was made partly possible through the Linux From Scratch project, which publishes guides for people wishing to create their own version of the GNU/Linux operating system [55] . This chain of development through successive projects is an excellent example of the distributive principle in action.

It would be wrong, however, to see these projects as a development from the earlier experiments with networked collaborative artworks. A much stronger analogy lies with the collective improvisation groups of the 1960s such as the Black Arts Group of St Louis (BAG) or the London-based Scratch Orchestra. In a move which in many ways pre-empts the hacklabs of today, the Free Jazz group BAG converted an abandoned warehouse into an inner-city 'training centre' jointly run by the group and local community. The space supported rehearsals and performances by BAG combined with various forms of public classes, workshops and discussions groups focused around issues affecting the local community. Here the collaborative musical practise of the jazz ensemble is structured upon, and reflective of, the wider distributive principle of the arts space in which they operate [56] . Another analogy is evident in the Scratch Orchestra who, like BAG, were an improvised music collective. Participation was open to anyone who wished to join regardless of musical background or experience. One of the key aspects of its practise were the development of 'scratch books' in which Orchestra members collated their own performance scores using whatever forms of notation they wished - from actual musical notation to abstract diagrams, written instructions and collaged found materials [57] . The scratch books were exchanged between different members who could use, alter and adapt from each other's work. Some versions of the scratch books were published under early forms of copyleft licences in which people were not only free to copy, but also encouraged to submit changes and additions that would be incorporated into later versions [58] . This was the embodiment of an idea that Cornelius Cardew, one of the founding members of the Scratch Orchestra, summarised in the statement: "the problems of notation will be solved by the masses." [59] Programs are notations just as music scores are, and we can clearly see how Cardew's sentiment applies as much to the practise of Free Software as it did to the music of the Scratch Orchestra.

Like the Free Software projects of today, these groups sought to distribute the knowledge and skills of production as an integral part of what they produced, but they also sought to explore and expose the social relations through which people worked. They deliberately challenged the notion of artistic authorship as the exclusive act of a single, socially disengaged, individual - one which still dominates the artworld today - as well as being critically aware of issues of power and hierarchy within collective practise. It is not just the distribution of knowledge, but also the distribution of access and power that these groups engaged with and faced up to. After several years of productive activity, the Scratch Orchestra fell apart due to internal tensions. Cardew put this down to a lack of sufficient self-criticism amongst its members, while others attributed it to one group seeking to exert its influence unfairly over the Orchestra as a whole [60] . Ultimately these are issues which any form of collaborative development needs to address and be critically aware of in some form, for, as the Scratch Orchestra discovered, working with others is not always an easy or inherently egalitarian process [61] . The success of any such project rests, not only in its ability to create something through collaboration, but in developing an awareness of what is at stake in collaboration itself.

Notes

1. see Jonathan Corbet, 2007, "Who wrote 2.6.20?", http://lwn.net/Articles/222773/ and David A. Wheeler, c.2006,"Counting Source Lines of Code", http://www.dwheeler.com/sloc/

2. there have been a number of studies on the structure of FLOSS development teams, see, for example: Rishab Aiyer Ghosh, 2003, "Clustering and dependencies in free/open source software development: Methodology and tools", http://www.firstmonday.org/issues/issue8_4/ghosh/index.html

3. Wikipedia is an online encyclopedia that anyone can contribute to: http://www.wikipedia.org

4. Richard Stallman, "The GNU Project", http://www.gnu.org/gnu/thegnuproject.html , also in the book Richard Stallman, 2002, Free Software, Free Society: Selected Essays of Richard M. Stallman, GNU Press; Boston.

5. for a more detailed account see Simon Yuill, 2007, "Free Software as Distributive Practise", forthcoming

[6] this should not be confused with the idea of 'social media' which has become prominent in an approach to online commerce known as Web 2.0, Free Software foregrounds the social basis of production, whereas, Web 2.0 seeks to commodify social relations, see Dmytri Kleiner and Brian Wyrick, 2007, "InfoEnclosure 2.0", in MUTE, vol 2 issue 4, also online: http://www.metamute.org/en/InfoEnclosure-2.0

[7] people can also build and learn apart, a common example being when one software project makes use of a library or codebase developed by a different group of people without any direct communication or collaboration between the people involved.

[8] there are numerous online guides for programming with Free Software tools, such as Mark Burgess and Ron Hale Evans, 2002, "The GNU C Tutorial", http://www.crasseux.com/books/ctutorial/ , Mark Burgess, 2001, "The Unix Programming Environment", http://www.iu.hio.no/~mark/unix/unix_toc.html , Richard Stallman, 2007, "GNU Emacs manual", http://www.gnu.org/software/emacs/manual/emacs.html , and Eric Raymond, 2003, "The Art of Unix Programming", http://www.catb.org/~esr/writings/taoup/html/

[9] http://www.gnu.org/software/mailman . For a discussion of the ideas behind mailman see: Barry Warsaw, 2000, "Mailman, the GNU Mailing List Manager", http://www.linuxjournal.com/article/3844

10. a good example is Nicolas Ducheneaut's OSS Browser, http://www2.parc.com/csl/members/nicolas/browser.html

11. for general information on IRC see http://www.irc.org and http://www.irchelp.org and Jarkko Oikarinen, undated, "IRC History", http://www.irc.org/history_docs/jarkko.html

12. ircII: http://www.eterna.com.au/ircii , irssi: http://irssi.org , BitchX: http://www.bitchx.org

13. this is described in Richard Stallman, "The GNU Project", ibid., and Eben Moglen, 1991, "Anarchism Triumphant: Free Software and the Death of Copyright", http://emoglen.law.columbia.edu/publications/anarchism.html

14. Dick Grune, Concurrent Versions System CVS. http://www.cs.vu.nl/~dick/CVS.html ACK was Stallman's original choice for a compiler in the GNU Project, when he approached Grune about using ACK however, Grune refused and, as a result, Stallman had to create his own compiler, which became GCC. ACK has since become obsolete whereas GCC is one of the most powerful and popular compilers currently in use today.

15. CVS: http://www.cvshome.org , SVN: http://subversion.tigris.org , Git: http://git.or.cz , darcs: http://darcs.net . For a detailed discussion of how Git works see: Jonathan Corbet, 2005, "The guts of git", http://lwn.net/Articles/131657/

16. David Roundy, undated, "Theory of patches", http://darcs.net/manual/node8.html#Patch

17. 'patches' are a common name for small changes to a code file, the repository is called a tree in Roundy's theory - all repositories have a tree-like structure.

18. the EMACS code editor was originally written by Richard Stallman, http://www.gnu.org/software/emacs

19. TkCVS and TkSVN: http://www.twobarleycorns.net/tkcvs.html

20. one example is Gitstat, which produces online visualisations of activity within Git repositories:

http://tree.celinuxforum.org/gitstat . Stats for the Linux kernel are available at: http://kernel.org

21. This idea is best represented in Donald Knuth's notion of 'literate programming', see: Donald E. Knuth, Literate Programming, Stanford, California: Center for the Study of Language and Information, 1992, CSLI Lecture Notes, no. 27.

22. Forth: http://thinking-forth.sourceforge.net , Python: http://www.python.org

23. PyDoc: http://docs.python.org/lib/module-pydoc.html , EpyDoc: http://epydoc.sourceforge.net , HappyDoc: http://happydoc.sourceforge.net

24. http://www.stack.nl/~dimitri/doxygen

25. http://www.graphviz.org

26. Christopher Alexander, et al., A Pattern Language: Towns, Buildings, Construction, New York: Oxford University Press, 1977, for a discussion of Alexander's ideas applied to software development see: Richard Gabriel, 1996, "Patterns of Software: tales from the Software Community", Oxford University Press; New York, PDF available online:
http://www.dreamsongs.com/NewFiles/PatternsOfSoftware.pdf

27. the classic text on software design patterns is: Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1995, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley. Design patterns are not specific to Free Software practise, and did not originate out of it, but have become popular amongst many programmers working in this way.

28. http://c2.com/ppr

29. see note 3 above.

30. For a comparison of different wiki systems see:
http://en.wikipedia.org/wiki/Comparison_of_wiki_software

31. Savannah: http://savannah.gnu.org , BerliOS: http://www.berlios.de

32. http://sourceforge.net

33. for a discussion of web platforms in relation to artistic practise, see: Olga Guriunova, 2007, "Swarm Forms: On Platforms and Creativity", in MUTE, vol 2 issue 4, also available online: http://www.metamute.org/en/Swarm-Forms-On-Platforms-and-Creativity

34. http://www.bugzilla.org

35. http://www.python.org/dev/peps

36. https://gna.org/projects/savane

37. http://trac.edgewall.org

38. there is a listing of LUGs organised by country available at: http://www.linux.org/groups , see also: http://www.gnu.org/gnu/gnu-user-groups.html

39. Dorkbots are worldwide, the main site from which all Dorkbot groups can be accessed is: http://www.dorkbot.org . Openlab groups currently exist in London, Glasgow, Amsterdam and Brussels, the two most active are in London: http://www.pawfal.org/openlab , and Glasgow: http://www.openlabglasgow.org

40. Indymedia is a worldwide movement to create independently produced news coverage. There are many local Indymedia groups across the world, the main site is at: http://www.indymedia.org

41. for background and discussion about the squatting movement see Anders Corr, 2000, No Trespassing!: Squatting, Rent Strikes and Land Struggles Worldwide, South End Press, and Advisory Service for Squatters, 2005, The Squatters Handbook, Freedom Press: London

42. Hackitectura: http://hackitectura.net , LOA: http://www1.autistici.org/loa/web/main.html , ASCII: http://www.scii.nl , C-Base: http://c-base.org . For a listing of different hacklabs internationally see: http://www.hacklabs.org . For a good overview of typical hacklab activities see: hydrarchist, 2002, Hacklabs - A Space of Construction and Deconstruction, http://info.interactivist.net/article.pl?sid=02/07/23/1218226&amp;mode=neste...

43. http://www.access-space.org Access Space grew out of the Lowtech project which started earlier in the mid-1990s.

44. http://www.metareciclagem.org

45. http://autolabs.midiatatica.org , see also: Ricardo Rosas, 2002, "The Revenge of Lowtech: Autolabs, Telecentros and Tactical Media in Sao Paulo", in Sarai Reader 04: Crisis / Media, Sarai Media Lab; Bangalore, also available online: http://www.sarai.net/publications/readers/04-crisis-media/55ricardo.pdf

46. DebConf: http://www.debconf.org , Summer of Code: http://code.google.com/soc

47. Chaos Computer Club: http://www.ccc.de , TransHack: http://trans.hackmeeting.org , Piksel: http://www.piksel.no , MAKEART: http://makeart.goto10.org

48. two of the earliest such projects include Roy Ascott, 1983, The Pleating of the Text: A Planetary Fairy Tail, (see Frank Popper, 1993, Art of the Electronic Age, Thames and Hudson: London, p. 124) and SITO: http://www.sito.org , an online collaborative image bank started in 1993, for an overview of SITO's history see: http://en.wikipedia.org/wiki/SITO

49. Florian Cramer, 2000, "Free Software as Collaborative Text", http://plaintext.cc:70/essays/free_software_as_text/free_software_as_tex...

50. http://www.pawfal.org/openlab/index.php?page=LaptopDrummingCircle

51. http://www.spring-alpha.org , and http://www.spring-alpha.org/svs

52. Plenum was part of Node.London, 2005, Life-Coding was presented at Piksel in 2007.

53. http://puredyne.goto10.org

54. http://www.dyne:bolic.org

55. http://www.linuxfromscratch.org

56. Benjamin Looker, 2004, Point From Which Creation Begins: The Black Artists' Group of St. Louis, St. Louis: Missouri Historical Society Press. For BAG this was an approach that was shared with the wider Black Arts Movement: "Seeing their art as a communal creation, national leaders of the Black Arts Movement had rejected Romantic and post-Romantic notions of the individual artist working in isolation or estrangement from his social context. Instead, they stressed art's functional roles, urging that it be created in a communitarian and socially engaged stance.", Looker, p. 66

57. for an overview of the Scratch Orchestra see Cornelius Cardew, 1974, Stockhausen Serves Imperialism and Other Articles, Latimer New Dimensions: London, PDF available online: http://www.ubu.com/historical/cardew/cardew_stockhausen.pdf . Extracts from the scratch books were collated and published in Cornelius Cardew (editor), 1974, Scratch Music, MIT Press: Massachusetts, MA.

58. see for example Cornelius Cardew (editor), 1969, Nature Study Notes, Scratch Orchestra: London: "No rights are reserved in this book of rites. They may be reproduced and performed freely. Anyone wishing to send contributions for a second set should address them to the editor: C.Cardew, 112 Elm Grove Road, London SW13.", p. 2

59. Cardew, 1974, p. 88.

60. different perspectives on this are provided in Cardew, ibid, Stefan Szczelkun, 2002, The Scratch Orchestra, in "Exploding Cinema 1992 - 1999, culture and democracy", available online: http://www.stefan-szczelkun.org.uk/phd102.htm , and in Luke Fowler's film documentary about the

Scratch Orchestra, "Pilgrimage From Scattered Points" (2006).

61. For a more critical perspective on colloborative working see Ned Rossiter, 2006, Organized Networks: Media Theory, Creative Labour, New Institutions, NAi Publshers: Rotterdam, and Martin Hardie, The Factory Without Walls, http://openflows.org/%7Eauskadi/factorywoutwalls.pdf

Images

[1] Logo of BitchX, a free IRC client.

[2] Screenshot of Branch Browser in TkCVS.

[3] Bugzilla Lifecycle diagram. Included in the source package for Bugzilla and available under the Mozilla.org tri-licence (MPL, GPL, LGPL). For use in the Bugzilla article.

[4] Nerdcore Central, hackmeet, photo by Armin Medosch, from: www.thenextlayer.org/node/74

" >

span-->

‹ Working with others up Social networking ›

Social networking

By marloes
Published: 10/04/2007 - 09:19

Iman Moradi , October 2007

Before the rise of the worldwide web, the Internet was already a successful model for allowing communities to form and share information. Newsgroups, mailing lists and countless forums and multi-user environments existed which allowed us to express our views and practice being social. It has been fittingly said that the Internet augmented traditional forms of communication and added to people’s social ties. (Irina Shklovski, 2004) [1]

Despite the opportunities and new modes of social communication that these augmentations allowed us, it's safe to say that what was on offer then was never as pervasive, inviting, interactive or facilitating as what we can enjoy today. As the web has developed, so too have the skills of countless specialist internet programmers and our insatiable appetite to communicate, consume and play with information in new and varied ways.

As a result, the web is now teeming with internet applications which are user friendly, rich in functionality and crucially provide features which allow us to form and maintain social ties in a manageable fashion. We can now exchange knowledge, benefit from aggregated views, share our personal media, access, manipulate, describe and generate items of data with unparalleled ease.

What is known in the industry as a perpetual beta, [2] refers primarily to how new, often experimental and cutting edge features are continuously integrated to such online services and software. Many of these features, usually offered freely, easily rival and surpass the potential benefits of locally installed paid for and free desktop software as they are developed quickly with a steady eye on direct user feedback and usage patterns.

However with the great potential that these tools have, there are some serious tradeoffs as well. The very mechanisms which allow us to enjoy social experiences and get things done quicker and more conveniently can potentially be put to other end uses as well which we may or may not find agreeable.

Of these concerns, there are the anticipated issues associated with implicitly entrusting our content with the service provider and third parties which ‘they’ trust, but more disconcertingly our mere interactions with any of these tools inevitably leave a rich information trail as well which can be stored and analysed without our knowledge or direct permission.

The broadly worded user license agreements or terms of service documents for these tools and services basically allow the companies which operate them great liberties to utilise any information we share with them, or any content we create, which may include tracking our usage even at times when we are not using their services! [3]

Whether the end purpose of any of these activities is to deliver a better more customised service, target us with advertising, sell our usage data to third parties, indemnify the providers from any responsibility when something goes wrong, or grant them rights to freely own and use our content as they please, it is all possible and indeed nonetheless questionable. [4]

This article aims to introduce a select few of the most well known tools for collaboration. It does so critically, highlighting their potential to enhance existing methods of communication and the way we work, as well as mentioning some of their drawbacks and pitfalls.

This article is aimed primarily at new users who want a starting point or introduction to an interesting and rapidly growing area of development.

Web 2.0 and Social Software

The Internet is awash with buzzwords and hype. You may read or hear some terminology which never existed until a couple of years ago. Some of this terminology is very specific, Blogs and Wiki’s are examples of such, and they are clearly defined things which have, needless to say, been around for a while. A Blog, short for weblog, or a Wiki both allow you to edit and update content which others can see, they are names for two generic types of product, with several variants and configurations and mutations of each available to use.

Blogs are like online diaries or journals, with each entry having a date and an archive of previous entries. Wiki's, such as the Wikimedia Foundation's suite of wikis and their online encyclopaedia "Wikipedia" referenced in this article, are non-linear flexible repositories which allow specific users or anyone to create pages, edit and interlink content while keeping a record of revisions and changes. Both are viewable within a web browser without any additional software being required. Open Source Blogging systems such as Wordpress [5] have also brought ease to blogging and many commercially hosted systems exist to allow users to maintain blogs without any technical know-how [6] .

These days however, other terms can be quite nebulous and can cover a number of different things. Web 2.0 (usually expressed two-point-zero, or two-point-o) is such a term. Web 2.0 is not the name of a new technology, but rather a way of referring to group of common technologies and methods which websites and associated services are developing and supporting to facilitate 'greater collaboration and sharing between users'. [7]

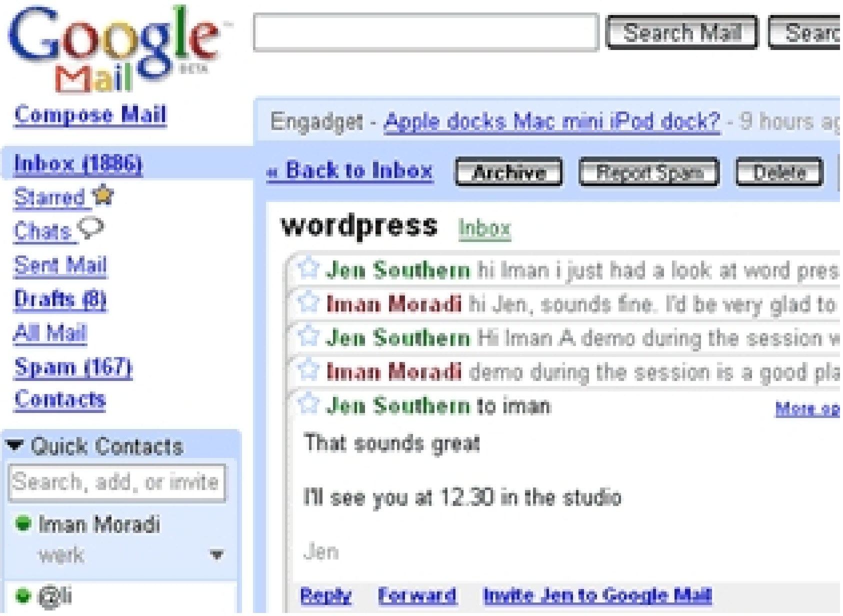
Social Software itself is the blanket term used to categorise tools which allow us to communicate and interact with other users, and although that in itself is nothing new, today this term is widely being used to describe web based software which adheres to some basic Web 2.0 principles.

The Wikipedia article on Social Software, gives a comprehensive list [8] of tools which are used for online communication. It has descriptions of instant messaging, text based chat technologies (IRC), Internet forums, Blogs, Wikis, as well as social bookmarking, virtual worlds and massively multiplayer games.

To better understand the area, it's important to know what these tools are doing and what they are enabling us, and the companies who offer these tools, to do differently.

Websites as applications

One of the first web based tools which led the way in terms of providing a very robust and capable service which behaved like a desktop application was Google's Gmail web based email service which has been around since the summer of 2004. Although it wasn't alone in doing so, it presented a shift in the way we view web based services and their sophistication.



Engadget - Apple docks Mac mini iPod dock? - 9 hours ag

wordpress

Jen Southern hi Iman i just had a look at word pres

Iman Moradi hi Jen, sounds fine. I'd be very glad to

Jen Southern Hi Iman A demo during the session w

Iman Moradi demo during the session is a good pla

Jen Southern to iman

That sounds great

I'll see you at 12.30 in the studio

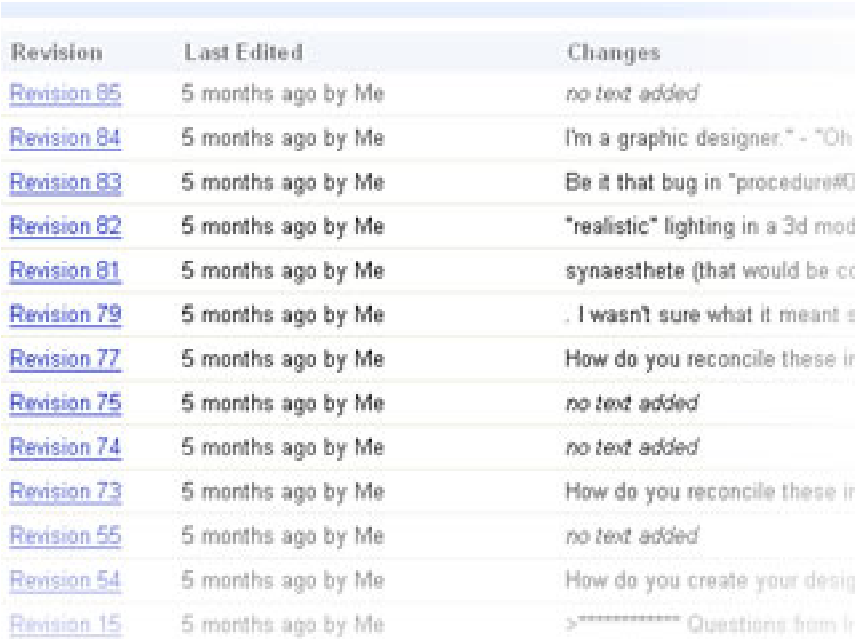
Jen

Feed reader, conversation view and instant messaging list.

With Gmail, you can simply search for emails instead of sorting them, and unlike other large web based email providers of a couple of years ago, the whole page doesn't refresh needlessly page by page in order to display your email. In fact in terms of functionality and user experience, when Gmail arrived, it was closer to a desktop based application than anything else on offer at the time. Other large web based email providers such as Yahoo mail and Microsoft Hotmail have since followed suit.

Currently, Gmail's main interface combines a feed reader (for showing you news items from your favourite sites) an instant messaging application (allowing you to converse with contacts in real-time), and it also has a very useful email clustering tool which groups related emails and their replies together.   Although these features are available in separate products and have existed for some time in other places, Gmail found a way of packaging them all together (by generating enough revenue from targeted advertising) and provided the service for free on the web. [9]

Google's Gmail now integrates with a suite of other tools such as Google Calendar and Google Docs and Spreadsheets, each other product allows you to share your data with other users with relative ease. For example you can share a calendar with other users, share documents and edit them collaboratively in one place to eliminate the need for excessive emailing and the chore of revision tracking. [10]



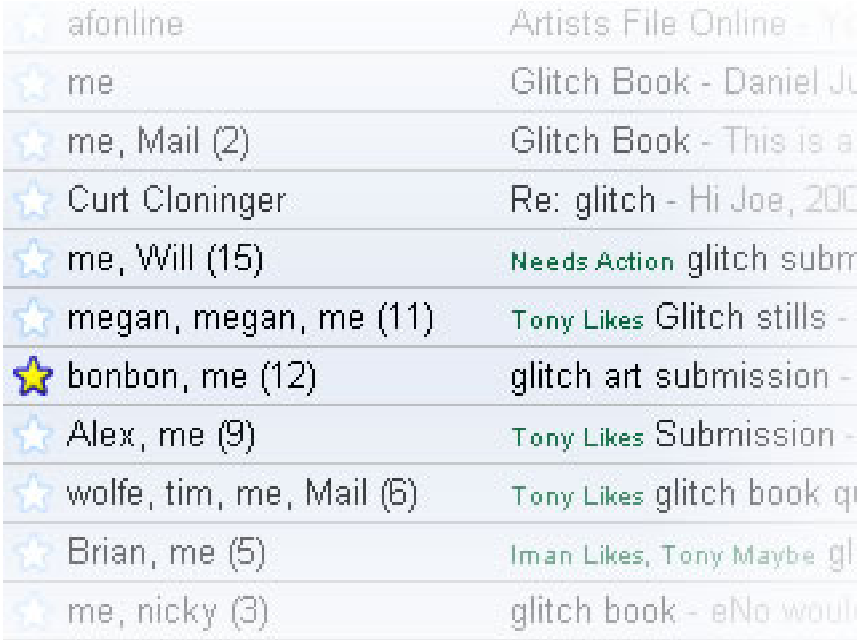
Revision	Last Edited	Changes
Revision 85	5 months ago by Me	*no text added*
Revision 84	5 months ago by Me	I'm a graphic designer." - "Oh
Revision 83	5 months ago by Me	Be it that bug in "procedure#0
Revision 82	5 months ago by Me	"realistic" lighting in a 3d mod
Revision 81	5 months ago by Me	synaesthete (that would be co
Revision 79	5 months ago by Me	. I wasn't sure what it meant s
Revision 77	5 months ago by Me	How do you reconcile these ir
Revision 75	5 months ago by Me	*no text added*
Revision 74	5 months ago by Me	*no text added*
Revision 73	5 months ago by Me	How do you reconcile these ir
Revision 55	5 months ago by Me	*no text added*
Revision 54	5 months ago by Me	How do you create your desig
Revision 15	5 months ago by Me	>************* Questions from ir

Google docs revision tracking

Recently, myself and other collaborators decided to use a shared Gmail account extensively over the course of three years to track submissions for art and design book collaboration. Without it, it would have been very difficult to keep track of the 700 or so submissions we received from over 200 people. Additionally, we used Google Spreadsheets to comment on these submissions and work together on editing essays and interviews. Using social software greatly simplified the task at hand, in that we mainly worked on it during weekends away from regular work.

With all the functionality and interactivity which Gmail offers, there are some other issues associated with the use of Gmail which have made privacy advocates quite worried. They primarily relate to Google’s policies regarding the use, analysis and retention of user data.

It's noteworthy to mention that although as editors of the book, we consented to have our sent and received emails scanned by Google’s software to show targeted advertising (as the cost of having a free service) the people sending us email did not explicitly enter into any agreement to do so, to essentially have the contents of their emails to us scanned [11] . Neither did they agree to Google's vague data retention policies, which can keep emails (even deleted ones) on the system indefinitely. [12]



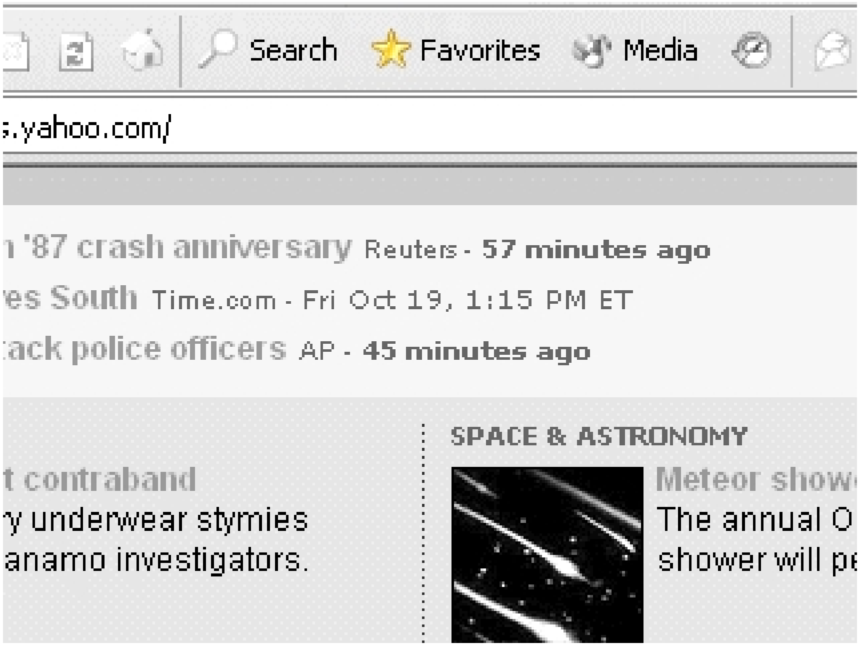
☆	afonline	Artists File Online - Y
☆	me	Glitch Book - Daniel Ju
☆	me, Mail (2)	Glitch Book - This is a
☆	Curt Cloninger	Re: glitch - Hi Joe, 200
☆	me, Will (15)	Needs Action glitch subm
☆	megan, megan, me (11)	Tony Likes Glitch stills -
★	bonbon, me (12)	glitch art submission -
☆	Alex, me (9)	Tony Likes Submission -
☆	wolfe, tim, me, Mail (6)	Tony Likes glitch book qu
☆	Brian, me (5)	Iman Likes, Tony Maybe gl
☆	me, nicky (3)	glitch book - eNo woul

Cropped portion of Gmail inbox interface, with correspondence clustering and labelling for messages.

Tagging / Folksonomies

Another great example to illustrate some of these key differences and familiarise you with common facets of many Web 2.0 applications is del.icio.us [13] . Delicious, which is now owned by Yahoo, is a social bookmarking tool. It’s a website and service which can integrate with your internet browser using a number of easy to install plug-in’s. It basically allows you to bookmark web pages which you find interesting with two clicks. You are then able to access these links from any web browser, see what others are linking to and perhaps share your own links with specific users by forming networks.

On surface there is nothing new to the storage of bookmarks aspect. For years, web browsers have had Favourites or Bookmarking functionality in-built, so the web pages which you frequently visit or find interesting can be retrieved later so long as you bookmark them or add them as a Favourite.



s.yahoo.com/

n '87 crash anniversary Reuters - 57 minutes ago

es South Time.com - Fri Oct 19, 1:15 PM ET

ack police officers AP - 45 minutes ago

t contraband
y underwear stymies
anamo investigators.

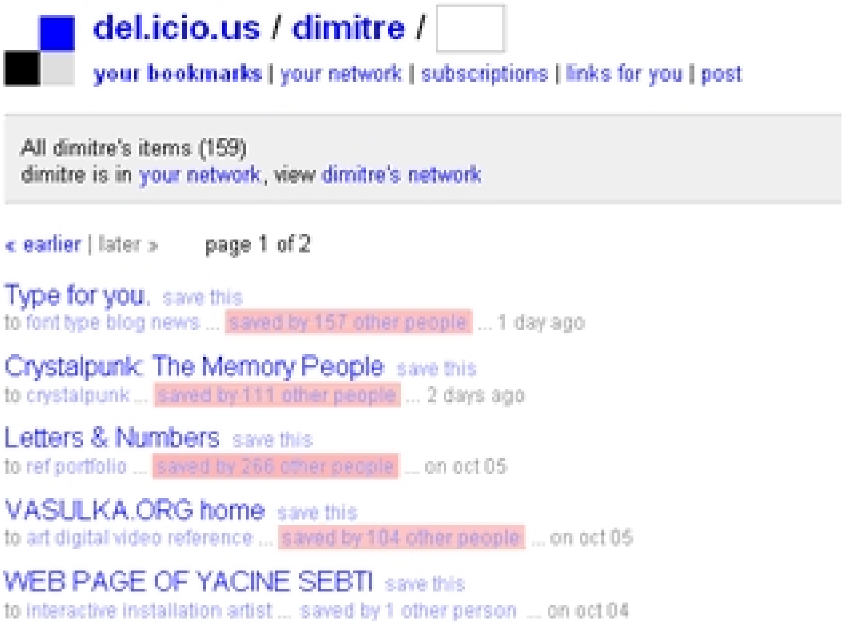
SPACE & ASTRONOMY

Meteor show
The annual O
shower will pe

Bookmark feature in Internet Explorer 6.

But anyone who has used Favourites knows too well the problem of having many links to different sites, with non-descriptive names and the pain of not being able to find a particular site again. Also, you may end up changing computers without backing up your links or you wish to retrieve your links at someone else’s computer or at work and suddenly the non portable Favourites situation inside your browser seems quite limiting.

With Delicious, which is one of the largest social bookmarking sites, instead of storing your links on your computer, you save them onto a designated area on the Delicious website, and these links are associated with your username.



del.icio.us / dimitre /

your bookmarks | your network | subscriptions | links for you | post

All dimitre's items (159)
dimitre is in your network, view dimitre's network

« earlier | later »   page 1 of 2

Type for you. save this
to font type blog news ... saved by 157 other people ... 1 day ago

Crystalpunk: The Memory People save this
to crystalpunk ... saved by 111 other people ... 2 days ago

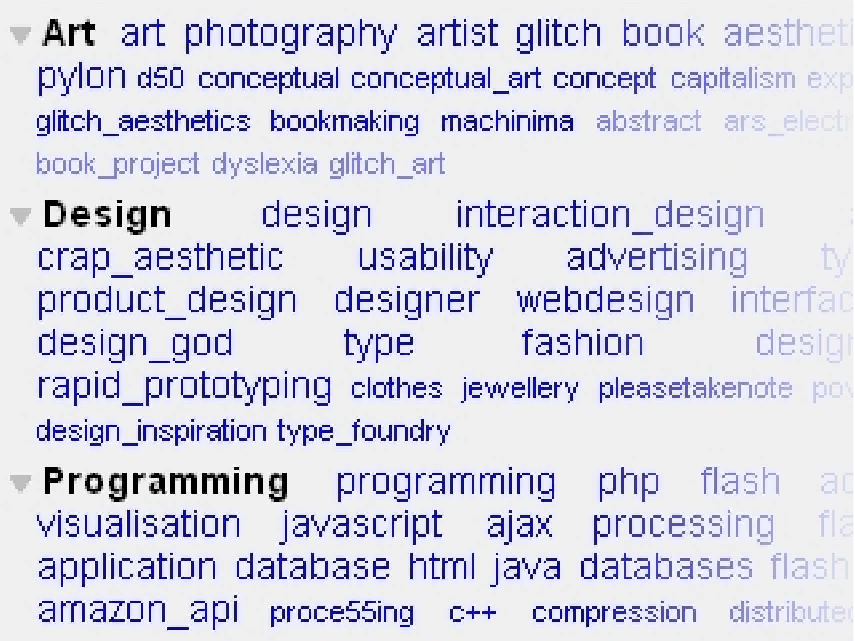
Letters & Numbers save this
to ref portfolio ... saved by 266 other people ... on oct 05

VASULKA.ORG home save this
to art digital video reference ... saved by 104 other people ... on oct 05

WEB PAGE OF YACINE SEBTI save this
to interactive installation artist ... saved by 1 other person ... on oct 04

Screenshot of a users delicious bookmarks feed.

As well as allowing other people to view your links and allowing you to view their links, Delicious offers an interesting way to search for and organise these links.   You assign your own ‘tags’ or descriptive keywords to each link you save and in effect this assists in retrieving them later.



▼ Art art photography artist glitch book aestheti
pylon d50 conceptual conceptual_art concept capitalism exp
glitch_aesthetics bookmaking machinima abstract ars_electr
book_project dyslexia glitch_art

▼ Design design interaction_design
crap_aesthetic usability advertising ty
product_design designer webdesign interfac
design_god type fashion desig
rapid_prototyping clothes jewellery pleasetakenote po
design_inspiration type_foundry

▼ Programming programming php flash ac
visualisation javascript ajax processing fla
application database html java databases flash
amazon_api proce55ing c++ compression distribute

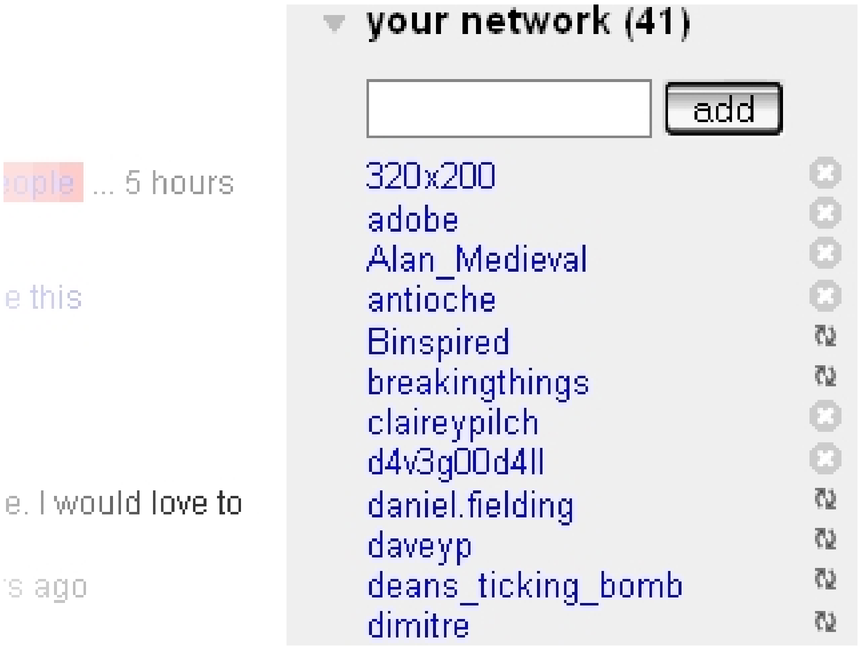
Crop of a clustered tag cloud. Showing frequency of tags visually.

Being able to tag items using a specialist vocabulary and the practice of doing so collaboratively, referred to as a Folksonomy [14] , clearly has its benefits over taxonomies and rigid information hierarchies [15] , and it has some drawbacks too.

Firstly, consider how a Dewey organised physical library can be so inherently confusing without knowing what the numeric shelf designations represent or without the benefit of having a library catalogue search tool or librarian at hand. Anyone can effectively become a subject specialist or categoriser within folksonomies, and therein lies a potential benefit and incentive, especially to categorise niche and newly emerging phenomenon.

Secondly, we could consider that freely allowing categorisation could also create Metanoise [16] which relates to inaccurate, erroneous, incomplete, or irrelevant data which is associated with any described item. [17]

Either way due to the large numbers using such services, any form of democratisation of the categorisation and description process can potentially be beneficial especially to minority users. (Scott Bateman 2006) [18]



your network (41)

add

320x200
adobe
Alan_Medieval
antioche
Binspired
breakingthings
claireypilch
d4v3g00d4ll
daniel.fielding
daveyp
deans_ticking_bomb
dimitre

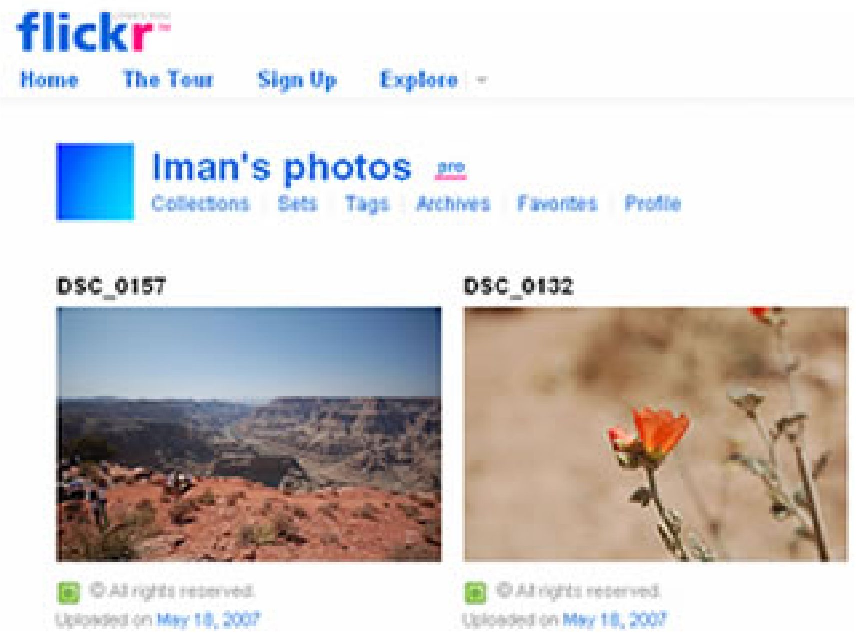
A user's personal network of other delicious users.

According to Thomas Van Der Wall (2005), who coined the phrase Folksonomy, “Who is doing the tagging is important to understand”. Using Delicious, as well as simply retrieving previously bookmarked content, we can actively share it with others too.   Delicious also allows you to form your own network and permits you to send any link to specific people in your network.

Some of the key features of Delicious are replicated across other systems. Collaborative content or knowledge filtering happens because we can see what others have shared and how they’ve described it. Searches for items unknown by name can instantly become quite fruitful if we know or can guess what others may categorise them as.

Additionally, when you bookmark an item you can discover who else has bookmarked it too; a quick visit to their bookmarks may uncover other hidden gems.

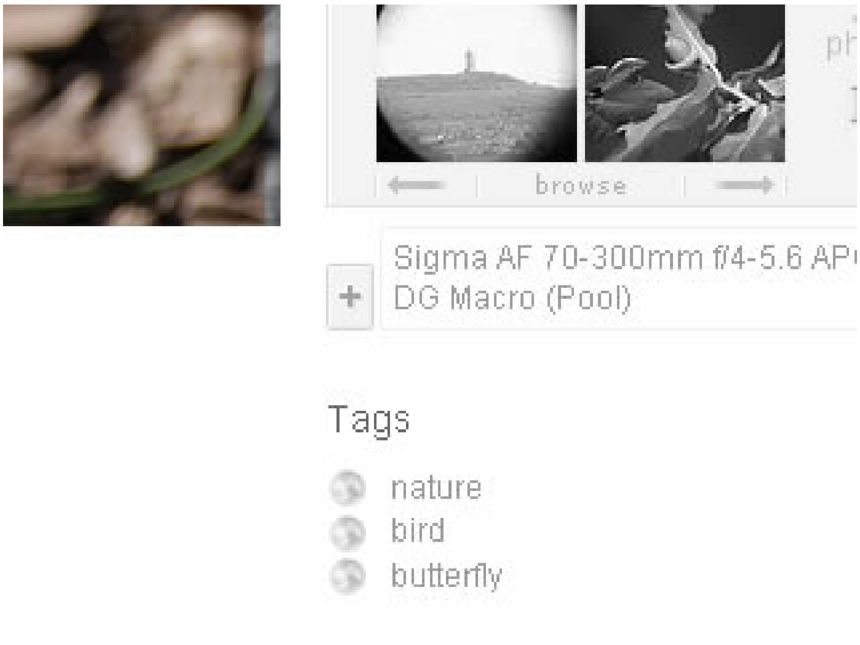
“Folksonomy tagging can provide connections across cultures and disciplines (a knowledge management consultant can find valuable information from an information architect because one object is tagged by both communities using their own differing terms of practice)” (Thomas Van Der Wall, 2005) [19]



Screenshot showing a crop of a personal Flickr page.

Flickr, which is a photo sharing website, allows the same affordances for personal photos and builds on the social network aspect quite extensively, while enriching photos with user provided tags.

According to Stewart Butterfield, one of Flickr's co-founders. "The job of tags isn't to organize all the world's information into tidy categories, It's to add value to the giant piles of data that are already out there." [20] Photo sharing websites have existed on the web but none have succeeded in adding as much value and use to our personal photos.



Sigma AF 70-300mm f/4-5.6 AP
DG Macro (Pool)

Tags

nature
bird
butterfly

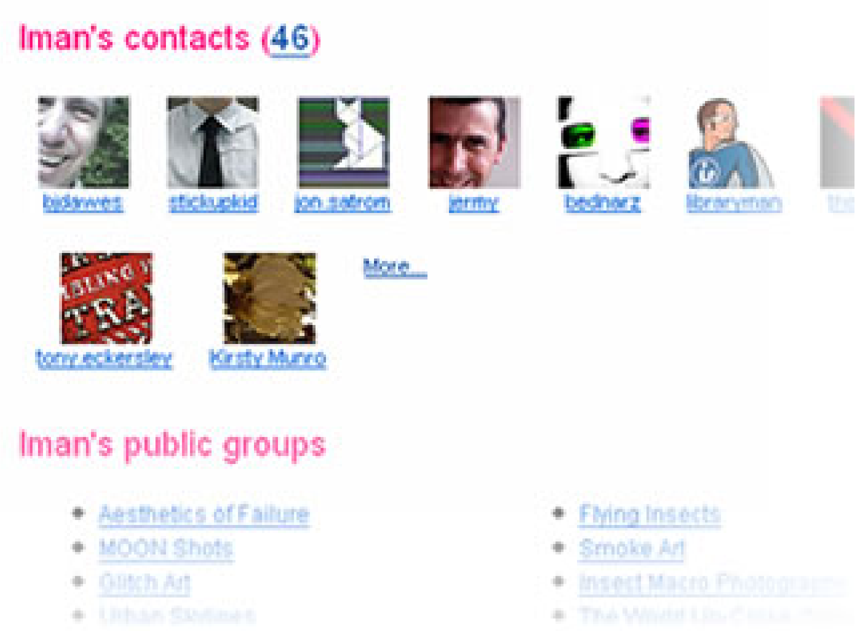
A crop of the photo detail page on Flickr showing three tags and a group (pool) which the photo belongs to.

While both these tools are successful and used by many, it's important to also state that there is a wealth of information which has yet to be fully utilised by the company which now owns both products, Yahoo!

In 2006, Yahoo’s former Chief Operating officer was quoted referring to the monetization of Flickr in due course. [21] “We have products like Flickr that are not yet really monetized right now.” and exactly how is something which will no doubt be informed and facilitated in part by “…user data about who they are and what they have done,…” although this wasn’t a wildly outrageous statement, it can be seen to go against the grain of the founding spirit of some of these tools, which weren’t initially designed, based and led by plans for monetary profit.

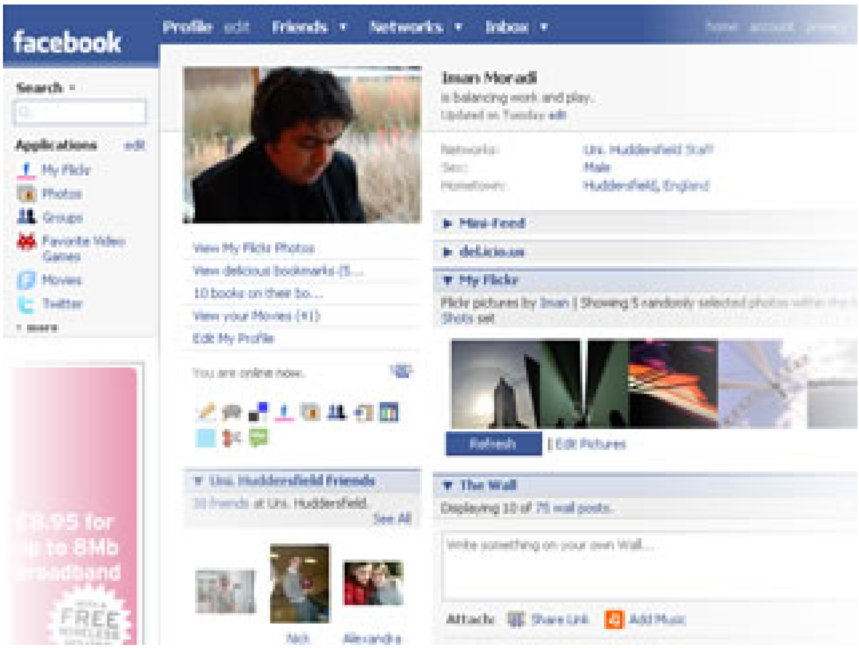
Social Networks

If you have a blog you are said to contribute to the blogosphere [22] which is the idea that blogs exist within a community. If you upload your photos to Flickr and add your friends to your contacts list you have formed a social network. The term friend can sometimes denote a looser designation in internet mediated communication and especially web 2.0 applications, than it does offline, and the term social network signifies the numerous types of connection between users of a website.



view of contacts page on Flickr.

The phenomenon of social networking sites [23] however, actually refers to websites which literally focus on showing, verifying and facilitating links between people sharing similar interests or members of particular groups.   Wikipedia maintains a growing list of these sites [24] . I’ve taken Facebook, currently the seventh most popularly visited site on the web as an example. [25]



Iman Moradi
is balancing work and play.
Updated on Tuesday edit

Networks: Uni. Huddersfield Staff
Sex: Male
Hometown: Huddersfield, England

Flickr pictures by Iman | Showing 5 randomly selected photos within the Shots set

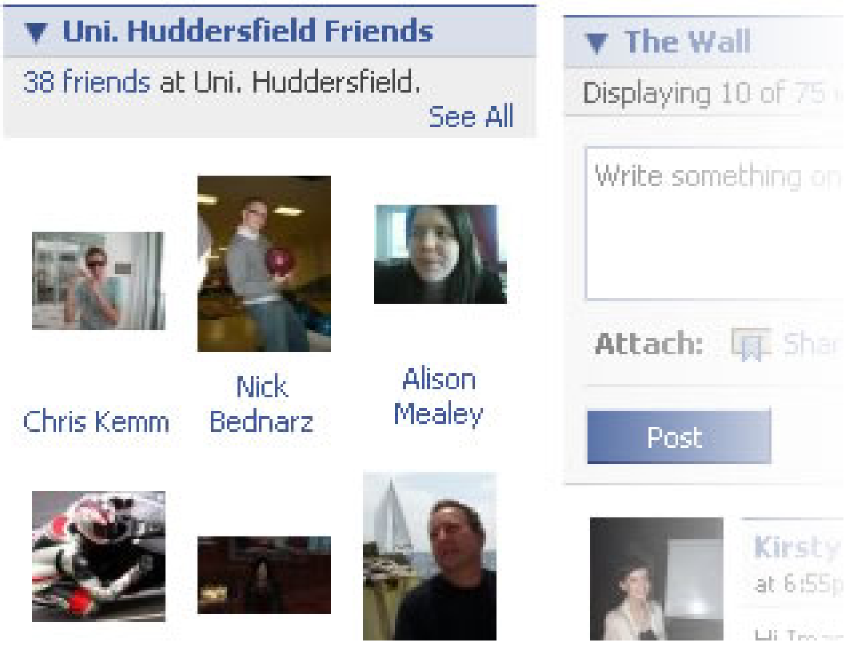
You are online now.

38 friends at Uni. Huddersfield.

Displaying 10 of 75 wall posts.

Write something on your own Wall...

A typical facebook profile page.



38 friends at Uni. Huddersfield.

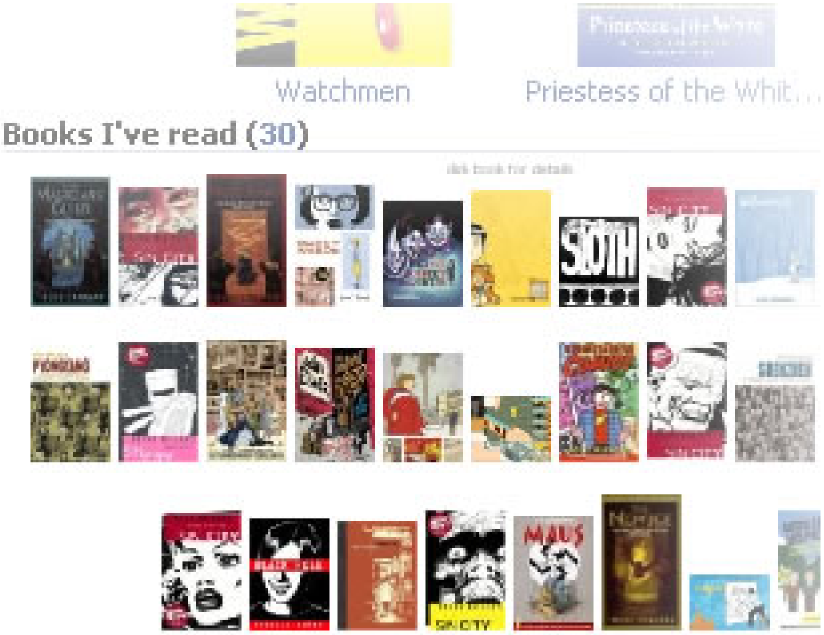
Displaying 10 of 75

Write something on

Facebook Friends list.

Facebook is a closed community which means you have to register or be invited before accessing many of the functions on the site, it allows you to keep a list of friends across various networks which you belong to. It incorporates messaging functionality, photo hosting, and status notifications as well as groups within its highly interactive profile pages.

It also allows developers to create applications which extend the core functionality of Facebook and utilise the power of existing networks to spread.



Watchmen

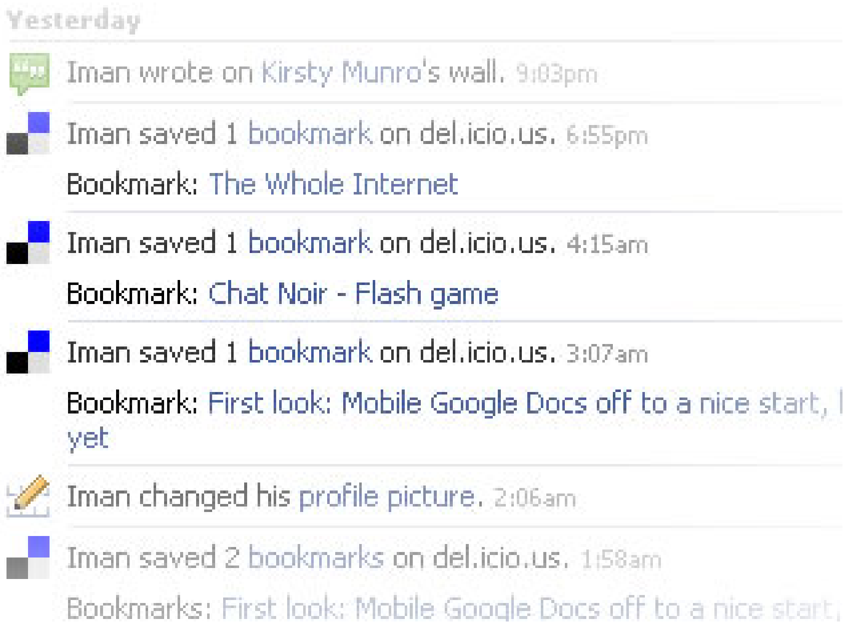
Priestess of the Whit...

Books I've read (30)

click book for details

Facebook Visual Bookshelf Third Party Application.

Facebook also demonstrates the power for information to be replicated and viewed within different frames on different sites. When I bookmark something on delicious I've opted to have my recent bookmarks displayed within my Facebook profile as well.



Yesterday

Iman wrote on Kirsty Munro's wall. 9:03pm

Iman saved 1 bookmark on del.icio.us. 6:55pm

Bookmark: The Whole Internet

Iman saved 1 bookmark on del.icio.us. 4:15am

Bookmark: Chat Noir - Flash game

Iman saved 1 bookmark on del.icio.us. 3:07am

Bookmark: First look: Mobile Google Docs off to a nice start,
yet

Iman changed his profile picture. 2:06am

Iman saved 2 bookmarks on del.icio.us. 1:58am

Bookmarks: First look: Mobile Google Docs off to a nice start,

Facebook Feed, displaying Delicious bookmarks as they happen.

Likewise, I can have my Flickr photos displayed on there too. Some of this has been made possible using RSS or really simple syndication and other agreed protocols and methods for sites to inter-operate. In the case of RSS feeds, essentially my Flickr photos, my Delicious bookmarks or my Wordpress blog posts have their own automatically generated file which other sites, or feed readers can utilise to link to my freshly updated content. Facebook literally grabs the most recent information you allow it to (links and photos) and redisplays it within your profile page.

Although all of this aggregation is currently by choice, there are many aspects to the Facebook experience which have come under fire from critics as being far from the in users control or deliberately confusing. [26] Facebook doesn’t offer the same amount of granular control which Flickr offers for individual items of data on your profile being accessible or locked away. Joining a large network, by default, allows members of that network to view your page without restriction.

Privacy advocates and ordinary users who have lost their basic privacy due to the illusion of privacy which Facebook fosters are naturally outraged. There have been concerns over colleges in the US taking action against their students based on specific Facebook activity [27], and a notable case with MySpace [28] where a 27-year-old student teacher was allegedly denied teaching credentials as her photo labeled drunken pirate, depicting her holding a plastic cup (possibly containing alcohol), was deemed "unprofessional" and encouraged underage drinking.

A critical view of recent developments

We are experiencing a time of rapid change and development, where many possibilities exist in the use of the current offering of technologies and tools, to great benefit. Up to this point in this article, a number of these tools have been profiled while mentioning some of the potential implications and concerns over misuse of the data we are willingly sharing. There is very little debate that these tools are simplifying previously complex tasks and enabling us to do things quicker, with less hassle, but what else are they enabling? Are we discovering these features and the full extent of their reach only when it’s too late?

“information collected by many Web sites — browsing histories, search histories, wish lists, preferences, purchase histories — is amassed and manipulated in ways consumers never know about.” [29]

On Social networking sites, we can profile our musical interest, our reading lists, political affiliations and educational history, we can form groups with other people, and have discussions within those groups. Essentially new forms of expressing our interests and telling people what we are doing every minute of every day are being developed and popularly accepted. But given the clandestine nature of what goes on behind the scenes and those broadly worded terms of service agreements, it does seem strange that we are voluntarily submitting quite so much personal information, for other individuals and establishments to peruse with such ease.

With something as simple as advertising that is being displayed on Gmail pages, there was initially a reasonable amount of controversy which surrounded Google’s fabled content extraction technology. Essentially this feature delivers highly targeted, relevant advertising in relation to the contents of your email. Information concerning the geographic origins and local time of your email correspondent is allegedly amongst the things which content extraction uses and analyses. One could argue that these businesses are legitimately driven by profit, and therefore need to utilise ingenious methods to conduct business, but how far are we letting them go? Is our data and reputation more valuable to us than the price they place on it?

Literally speaking, “MySpace was sold in 2005 for a price that corresponded approximately to 35 US$ per user profile. In 2006 Facebook sources suggested a valuation for their network of 2 billion US$ which would translate to 286 US$ per user profile” (Giles Hogben, 2007), yet the damage that could be done by an invasion of privacy comes at much higher costs.

“The consequences of misusing Facebook can damage a student's chances for their career and hinder their social life.” (Brandon J. Mendelson 2005) [30]

Privacy International which are a human rights group, in a recent survey of several web based services and companies, deemed Google to be hostile to user privacy. They also identified breaches in several other services and their offerings. [31] Primarily, it's the lack of transparency and clarity regarding some of the issues raised in this article which are worrying, as they can set irreversible trends and lower the expectations of the levels of privacy by future users if we become complacent and accepting of the current methodologies for handling our privacy matters.

While the companies themselves aren't necessarily out to get us, their attitudes and responsibilities towards protecting our data are questionable. Facebook's relationship with third party developers is an example of such. Third parties who develop applications for Facebook aren't vetted by Facebook and their seemingly innocuous applications can access our user data. Or as some users have found, we are not in full control of our membership, [32] to the point that it's very difficult to leave such sites without considerable effort.

As other users of these communities have found out, the slightest criticism of the service itself is sometimes met with very heavy handed resistance and censorship [33] . There have also been well documented cases of data retention and privacy concerns of companies such as Yahoo which have affected the lives of political activists in countries such as China [34] .

With all the affordances web mediated communication and specifically web 2.0 services offer us, and all of their benefits, it's good to bear in mind that there is after all no such thing as a free lunch, and to constantly assume that privacy is very fragile and never guaranteed in the hands of fickle multinational corporations.

Other Services not mentioned in this article

Wordpress.com allows anyone to create and maintain their own weblogs. http://wordpress.com/

Writeboard is a collaborative word processor which makes keeping note of revisions and collaborating with others on writing a document simple to manage. http://writeboard.com

Tadalist facilitates the creation of easy to use to-do lists. http://tadalist.com

Upcoming is a service which allows you to create, manage & share listings for events.
http://upcoming.org

Meebo is an in-browser instant messaging program, which supports multiple IM services, including Yahoo! Messenger, Windows Live Messenger, Google Talk, AIM, ICQ, and Jabber. Meebo also makes it easy to create instant rooms for mini conferences just by sharing a web address. http://meebo.com

Last.fm is an online radio and music based profiles site which allows you to profile your musical taste, and discover new music. http://last.fm

Librarything – allows you to catalogue & share your personal library. http://librarything.com

ELGG - As used by http://community.brighton.ac.uk is an open source social platform and content management system. http://elgg.org/

Bibsonomy, CiteULike, Connotea are further examples of social bookmarking tools.

http://www.bibsonomy.org , http://www.citeulike.org , http://www.connotea.org

Notable Project / Organisation

Attention Trust is an organisation aimed at Empowering people to exert greater control over their "attention data", leading efforts to drive developers and business leaders towards greater transparency and dialogue with users and lawmakers over privacy issues. http://attentiontrust.org/

Services mentioned in this article

Flickr allows you to upload and share your pictures online. Its one of the most successful web 2.0 services. http://flickr.com

Delicious is a fully featured social web-bookmark manager. http://del.icio.us

Google Documents, Spreadsheets and   Presentations.

Document editing with publishing and collaboration features. http://docs.google.com

Google calendar system with sharing/sms notification functionality. http://calendar.google.com

References

SCOTT BATEMAN, CHRISTOPHER BROOKS, GORD MCCALLA (2006)

Collaborative Tagging Approaches for Ontological Metadata in Adaptive E-Learning Systems

GILES HOGBEN (2007), Security Issues and Recommendations for Online Social Networks

European Network and Information Security Agency, ENISA

BRANDON J MENDELSON (2005) Misuse of Facebook. Blogcritics website, [accessed October 2007]

IRINA SHKLOVSKI, S. K., ROBERT KRAUT (2004) The Internet and Social Interaction: A Meta-analysis and Critique of Studies, 1995-2003

THOMAS, VAN DER WALL (2005) Explaining and Showing Broad and Narrow Folksonomies. [accessed October 2007]

Footnotes

[1] Katz & Aspden in The Internet and Social Interaction: A Meta-analysis and Critique of Studies. http://www.cs.cmu.edu/~kraut/RKraut.site.files/articles/Shklovski04-Inte...

[2] http://en.wikipedia.org/wiki/Perpetual_beta

[3] "We may use information about you that we collect from other sources, including but not limited to newspapers and Internet sources such as blogs, instant messaging services and other users of Facebook, to supplement your profile." Facebook Privacy Policy. http://www.facebook.com/policy.php

[4] Blog post outlining and highlighting the vaguely worded / veiled threats present in Terms of Service documents which the majority of ordinary users tend to ignore. http://scholarsandrogues.wordpress.com/2007/09/02/dont-be-evil-unless-yo...

[5] http://wordpress.org Open Source Blogging Software

[6] Wordpress.com, Tumblr, Typepad, Vox, Blogger, Edublogs are all examples of hosted blogging services.

[7] http://en.wikipedia.org/wiki/Web_2.0

[8] http://en.wikipedia.org/wiki/Social_software

[9] This possibility for linking search records, browsing history, emails, photos and instant messaging conversations is seen by many as being very invasive of ones privacy. Specially in the case of linking search queries to individual users via the unified Google account. Moreover its important to note, Gmail is offering these tools for free because it can make a profit from advertising revenue, as advertising revenues fall one has to be weary of other ways in which they could potentially make money by analysing and selling our data.

[10] Googles Promotional Video for Google Docs: http://www.youtube.com/watch?v=eRqUE6IHTEA

[11] http://www.epic.org/privacy/gmail/faq.html

[12] Letter written in 2004, by thirty-One Privacy and Civil Liberties Organizations to urge Google to suspend Gmail, over concerns that they are lowering privacy expectations.
http://www.privacyrights.org/ar/GmailLetter.htm

[13] http://del.icio.us

[14] http://en.wikipedia.org/wiki/Folksonomy

[15] http://www.vanderwal.net/random/entrysel.php?blog=1635

[16] http://en.wikipedia.org/wiki/Meta_noise

[17] In Metacrap, Cory Doctorow refers to seven insurmountable obstacles with regards to user categorisation or objects. http://www.well.com/~doctorow/metacrap.htm

[18] Online PDF version of “Collaborative Tagging Approaches for Ontological Metadata in Adaptive E-Learning Systems”
http://www.win.tue.nl/SW-EL/2006/camera-ready/02-bateman_brooks_mccalla_...

[19] Thomas Vander Wall’s, “Explaining and Showing Broad and Narrow Folksonomies”
http://www.personalinfocloud.com/2005/02/explaining_and_.html

[20] Daniel Terdiman, “Folksonomies Tap People Power.”
http://www.wired.com/science/discoveries/news/2005/02/66456

[21] Yahoo! Q3 2006 Earnings Call Transcript.
http://seekingalpha.com/article/18639-yahoo-q3-2006-earnings-call-transc...

[22] http://en.wikipedia.org/wiki/Blogosphere

[23] http://en.wikipedia.org/wiki/Social_network_service

[24] http://en.wikipedia.org/wiki/List_of_social_networking_websites

[25] Alexa ranking November 2007.

[26] Danah Boyd, Confused by Facebook
http://www.zephoria.org/thoughts/archives/2007/09/06/confused_by_fac.htm...

[27] Misues of Facebook may cost you. http://blogcritics.org/archives/2005/12/11/213844.php

[28] http://www.bloggingstocks.com/2007/04/30/myspaces-drunken-pirate-sues-co...

http://www.thesmokinggun.com/archive/years/2007/0426072pirate1.html

[29] Online Ads vs. Privacy. NY Times
http://www.nytimes.com/2007/05/12/technology/12online.html?ei=5090

[30] See 27.

[31] http://www.privacyinternational.org/article.shtml?cmd[347]=x-347-553961

[32] http://siyavash2005.googlepages.com/facebook

[33] http://www.theregister.co.uk/2007/09/14/facebook_group_deleted

[34] http://www.nytimes.com/2005/09/08/technology/08yahoo.html

Images

All images courtesy of the author.

" >

span-->

‹ Collaborative development: theory, method and tools up Publishing your work ›