

**Sanna Kotkaluoto, Juha Leino,  
Antti Oulasvirta, Peter Peltonen,  
Kari-Jouko Rähkä and Seppo Törmä**

# **Review of Service Composition Interfaces**



DEPARTMENT OF COMPUTER SCIENCES  
UNIVERSITY OF TAMPERE

D-2009-7

TAMPERE 2009

UNIVERSITY OF TAMPERE  
DEPARTMENT OF COMPUTER SCIENCES  
SERIES OF PUBLICATIONS D – NET PUBLICATIONS  
D-2009-7, OCTOBER 2009

**Sanna Kotkaluoto, Juha Leino,  
Antti Oulasvirta, Peter Peltonen,  
Kari-Jouko Rähkä and Seppo Törmä**

# **Review of Service Composition Interfaces**

DEPARTMENT OF COMPUTER SCIENCES  
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 978-951-44-7896-3  
ISSN 1795-4274

# Preface

This report was produced in the LUCRE project. LUCRE stands for Local and User-Created Services. The project is part of the Flexible Services research programme, one of the programmes of the Strategic Centre for Science, Technology and Innovation in the ICT field (TIVIT) and funded by Tekes (the Finnish Funding Agency for Technology and Innovation) and the participating organizations.

The Flexible Service Programme creates service business activity for global markets. The programme has the aim of creating a Web of Services. The programme creates new types of ecosystems, in which the producers of services, the people that convey the service and the users all work together in unison.

As part of such ecosystems, LUCRE will develop an easy-to-use, visual service creation platform to support the creation of context aware mobile services. The goal is to support user-driven open innovation: the end-users (people, local businesses, communities) will be provided with tools to compose new services or to modify existing ones. The service creation platform will build on the technology of existing mashup tools, widget frameworks, and publish/subscribe mechanisms. It will contain a range of common building blocks of context-aware services including people and groups, discussion channels, maps and location, and media.

This technical report is based on Deliverable D2.1 of LUCRE. Here we review available commercial solutions for service composition as well as research published in academic forums. The purpose of this review is to understand the main categories of solutions to this problem.

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Service composition</b>	<b>3</b>
2.1	Time of composition creation	3
2.2	Fields of research on service composition	4
2.3	Mashup patterns	5
2.4	End-user programming	5
<b>3</b>	<b>Use-time service composition</b>	<b>7</b>
3.1	Intel MashMaker	7
3.2	Mozilla Ubiquity	8
<b>4</b>	<b>Programming by example</b>	<b>10</b>
4.1	Programming by demonstration	10
4.1.1	Karma	10
4.1.2	Vegemite	11
4.2	Example modification	12
4.2.1	d.mix	12
<b>5</b>	<b>Visual programming</b>	<b>14</b>
5.1	Popfly	15
5.1.1	Mashup Creator	15
5.1.2	Service composition with Popfly Mashup Creator	18
5.1.3	Game Creator, Web Creator, and Popfly Space	19
5.1.4	Current state	20
5.2	Yahoo! Pipes	20
5.2.1	Service composition with Yahoo Pipes	21
5.2.2	Discovering and cloning pipes	25
5.2.3	The current state of Yahoo Pipes	26
<b>6</b>	<b>Form-based creation</b>	<b>28</b>
6.1	iGoogle	28
6.1.1	Discovering gadgets in iGoogle	32
6.1.2	Building gadgets	33
6.1.3	Popular gadget types in iGoogle	35
6.1.4	Creating themes	37
6.1.5	Social future	37
6.2	Netvibes	39
6.3	Yahoo! Widgets	43
<b>7</b>	<b>Script based creation</b>	<b>46</b>
7.1	Firefox extensions	46
7.1.1	Building Firefox extensions	50
7.1.2	Future directions	51
7.2	Facebook	52
7.2.1	Using Facebook applications	54
7.2.2	News feeds and requests	55
7.2.3	Application directory	56
7.2.4	Application initialization process	60
7.2.5	Security of Facebook applications	62
7.2.6	Facebook platform	64
7.2.7	The current state of Facebook	65
<b>8</b>	<b>Evaluation of mashup tools for end-user development</b>	<b>68</b>
<b>9</b>	<b>Summary and discussion</b>	<b>70</b>
<b>10</b>	<b>References</b>	<b>73</b>

# 1 Introduction

This report is a review of the state of the art in *service composition* by the *end-users*. We review available commercial services in service composition as well as research published in academic forums. The purpose of this review is to understand the main categories of solutions to this problem. What kinds of approaches, technologies and tools are available? How does a user interact with the tools? What kinds of services users end up composing?

The Web is full of services ranging from complex, globally used services, such as email, Amazon, Google Maps, Google Search, eBay, and Facebook, to smaller or more locally profiled services, such as currency converters, journey planners, eBanking, and so on. Popular services obviously address some important needs that are shared by many users. However, outside of the shared needs, there is a realm of all kinds of individual, evolving, and situational wants and needs. Service developers and providers cannot anticipate nor address them all. Indeed, the attempt to do so would lead into bloated software packages with decreasing performance and usability, and at the end there would be a lot of unnecessary duplication of functionality already existing in other services.

As an example consider a simple task of finding a new apartment. There are many services that provide real estate listings. They typically include a standard set of information about each apartment, the building, the neighborhood, and the terms of sale. However, an individual buyer may have important other criteria for the purchase, such as distance to day care centers, schools, playgrounds, grocery stores, specialty shops, and bus stops. Some of these criteria may be relevant to subgroups of customers – such as the average air quality or walkability<sup>1</sup> of the neighborhood – and some completely specific to an individual buyer – such as the time it takes to drive to his or her job, hobbies, or friends. It is quite clear that the service provider cannot include such information in the listings. Moreover, it may not be in the interest of the seller even to volunteer all generally interesting information, in case it is unfavorable to the apartment. The buyer, however, ends up using a number of different web sources and services to find out the issues critical to him or her about each potential apartment. This requires copying and pasting information between services and applications, with possible clean up and transformation of the data in between. Multiple browser windows would need to be opened and accessed. The combination of information would mostly happen in the head of the buyer. When the number of potential apartment grows larger, the evaluation requires disciplined bookkeeping. After a while the buyer may become overburdened and frustrated, and find it difficult to keep track of the big picture. He or she would simply like to have a service that looks for apartment listing and evaluates them according to his or her individual interests, and in case an apartment rates favorably, would send a notification. How could users be better supported in this kind of tasks?

Ultimately, users would probably wish to simply state *what* they want a composite service to do. For example, “I would like to have a notification of apartments that cost less than 200000 € located so that air-quality is good, and within half an hour driving from my job.” To satisfy such requests would require detailed semantic descriptions of services, together with either automatic reasoning capabilities or elaborate crowdsourcing arrangements. Unfortunately, there are not any realistic and practical solutions available in those areas yet.

However, technology to enable users to tell *how* they want a composite service to work is developing fast. There are tools that make it possible to compose, for example, a following kind of service: “take apartments from craigslist.org, filter apartments with price less than 200000 €, evaluate the address in airquality.com, filter apartments where the result is good, compute the driving time from the apartment to the job in maps.google.com, filter apartments where the result is less than half an hour“. These tools use technologies developed in the area of end-user programming, or more generally *end-user development*. Indeed, Lieberman et al. (2005) have predicted that the goal of human-computer interaction will evolve from just making systems *easy to use* to making systems that are *easy to develop*, as it is impossible to design systems that are appropriate for all users and all situations (MacLean et al., 1990).

---

<sup>1</sup> Walkscore (<http://walkscore.com>) provides this information for neighbourhoods of largest cities in US

This report reviews service composition tools that enable assembling existing sources and services into new combinations of desired content and functionality. If such tools were accessible and easy-to-use, there would ultimately be numerous small-scale services most of which would satisfy some personalized, situational, and evolving needs of individuals. These services would belong to the *long tail* of software: the many small applications that collectively have a large impact (Anderson, 2006). However, the ability to easily compose services would also have the potential to uncover hidden needs of people; when shared with others, their ideas and innovations could boost the evolution of the whole service ecosystem.

Spreadsheets present a major success story in the area of end-user programming (Nardi, 1993). Many office workers without any studies or previous experience in programming have been able to create non-trivial spreadsheets for real uses. Other areas where end-users have been able to participate directly in development are web authoring and business process automation. End-user development and consequently service composition seem thus feasible ideas, but to what extent? It is commonly understood that programming—and development of reliable software in general—is difficult. While it is evident that end-users are able to create simple services, it also seems obvious that they will not be able to solving many software engineering problems, such as those related to performance, scalability, distribution, concurrency, and interoperability, that are difficult even for developers.

The goal of LUCRE project is to gain understanding of the potential of user-created services by developing and studying service composition tools and technologies. There are many interesting questions in the area. Are there reasonable compromises between ease-of-use and power of the tools? Is it possible to have a smooth progression from simpler methods to powerful ones? Are there problems or dangers inherent in service composition by end-users? What is a reasonable division of work between end-users and developers and how could fruitful collaboration between them be organized?

In the next chapter we will discuss the service composition problem to focus the study to relevant technologies and tools. In the rest of the document we will cover some of the tools in depth.

## 2 Service composition

*Service* is a term that has many meanings. In this report it is used in the following, relatively technical meaning:

*Service is a software component made available through a network.*

In contrast to an *application* that is executed in the client, a *service* is associated with a server that hosts it and at least to some extent supports its execution. The implications are that services are typically independent of the client platform, require no installation or updating from users, can utilize background data potentially shared among all the users, and their execution can be monitored.

Services can naturally be created by ordinary programming but in this report we are interested in higher-level conceptualizations of service creation. The emphasis is specifically in service *composition*, not in service development in general. Composition means the creation of a service as a relatively simple combination of available services. For example, a geographical location pointed by the user on a map could be connected to a service that tells road conditions there. The focus is thus not in complex control flows that involve conditional execution (if-statements), iteration (loops), or state maintenance (variable assignments).

By *end-user* we mean anyone who uses a service, is not a developer, and has not studied software development. Being an end-user does not imply that a person is generally uneducated, unskilled, or cognitively challenged. Many end-users of spreadsheets have been office workers who are well educated, although not in software development. Conversely, the lack of software-related education does not make a developer an end-user.

### 2.1 Time of composition creation

Different tools conceptualize and approach service composition in different ways. One difference concerns the time when the composition is created. There are essentially two alternatives:

1. *Use-time composition*. When a user is browsing the Web, he or she can combine services in an *ad hoc* manner. These are usually one-time, throwaway compositions for the particular purpose at hand.
2. *Design-time composition*. A user can create a service composition in advance of using it. The result is a *service artifact* that is typically reusable in many similar situations later on.

In use time composition, the system typically suggests—or otherwise makes it possible to discover—other services or data that can be combined with the data being examined. For instance, if the user is looking at a list of restaurants that includes the addresses, the system can propose to display the locations of the restaurants on a map.

Design time composition can be used multiple times afterwards. For example, the user can build from a map service, a restaurant list, and the current location of the mobile device a composite service that displays ten closest restaurants on the map centered on the current location of the mobile device. He or she can then use the service multiple times in similar situations and even publish it so that other people can also use it.

The potential for reuse is valuable but it brings with it many difficulties inherent in programming. The service should work in a range of future situations, instead of just in the particular situation at hand as in use-time composition. This requires some way to abstract the relevant properties of the situation as well as use of some notation in which the abstract service can be expressed.<sup>2</sup>

---

<sup>2</sup> Below we will discuss a technique called programming-by-demonstration that can form a bridge from use-time composition to reusable services by allowing the system to infer reusable services from use-time examples.

## 2.2 Fields of research on service composition

Services can be composed—as any programs or software systems—through ordinary programming. However, there are at least the following active areas of research and development specifically targeted to the creation of service compositions:

1. *Service-oriented architecture (SOA)*. In the world of Service-oriented architecture, the concept of Web Service is used to mean services specified to allow machine-to-machine use. There are numerous XML-based standards developed by organizations such as W3C and OASIS to govern the specification and use of Web Services. According to the standards, service interfaces are described using Web Services Description Language (WSDL), messages between services with SOAP, and service discovery using UDDI registry. For service composition there are two concepts: orchestration and choreography. *Orchestration* means the specification of a hierarchical, process-like composite service using the Web Services Business Process Execution Language (WS-BPEL) notation. This approach is mainly targeted to business processes and it expects the component processes to provide structured XML data through well-specified interfaces. *Choreography* refers to a conversational, non-hierarchical approach for composition of Web Services. It specifies a protocol for peer-to-peer interactions, defining, for example, the possible sequences of messages exchanged to guarantee interoperability. It is not directly executable, as it allows many different realizations, either through ordinary programming or through a suitable orchestration from each peer. Choreography can be expressed with Web Services Choreography Description Language (WS-CDL). Orchestration and choreography cannot be characterized as proper end-user methods since WS-BPEL and WS-CDL are relatively complex specifications that include many programming concepts: variables, loops, conditional execution, synchronization, exceptions, and so on. There are also visual WS-BPEL tools available from providers of SOA technologies, such as IBM, SAP, Oracle, Microsoft, and Apache.
2. *Mashups*. Mashups are services that combine functionality and data from multiple web sources to help users solve tasks not originally envisioned by the authors of those sources (Lin 2009). The mashup approach takes as the starting point all the services available in the Web, including all kinds of user-directed services and sources in addition to the Web Services with well-defined APIs mentioned above. There is a broad range of different kinds of approaches, technologies, and tools developed in the mashup area; active innovation takes place both with regard to the tools and to the actual creation of composite services. Mashups aim to utilize interesting services (Google Maps, Craigslist, weather information, public statistics, etc.) even when the data provided by services is unstructured—i.e., in HTML—and requires the use of information extraction techniques. When compared to the process-oriented focus of orchestration, mashup research addresses a different spectrum of problems ranging from information extraction to information visualization. On the other hand, many mashups are experimental and not created for any mission-critical purposes, as might well be the case with Web Service orchestrations. It should be noted that the field of enterprise mashups (Hoyer, 2008) is a combination of service-oriented architecture and mashups. The aim is to give end-users within enterprises a possibility to easily compose services for their situational needs.
3. *Semantic Web Services*. In the field of Semantic Web Services, the service composition problem is seen as an automatic reasoning task. The user specifies the goals and requirements (“show closest ethnic restaurants on a map”), and the system automatically infers a composite service that fulfils the request using the services available in the Web. The most obvious use case for automatic methods is mobile and ubiquitous computation, where the set of available services changes as the result of physical mobility and composition should happen mostly without human intervention, in an opportunistic and transparent manner. Technology for automatic composition of services comes from the research on knowledge representation and planning algorithms within the field of artificial intelligence. The approach requires detailed semantic descriptions of available services, either using some ontologies or with folksonomies. Unfortunately, the problems of semantic interoperability are difficult and they are still, to a large extent, unsolved today. Furthermore, the algorithms to combine services with each other tend to be intractable although there has been a lot of progress in them and small composition problems could certainly be solved.



The review in this report will be limited to the mashup approach, as the role of an end-user is unclear in the other approaches. The field of mashups contains many interesting and relatively practical methods based on the ideas generated in the field of end-user programming. In addition, neither the service-oriented architecture nor the semantic web services approach has not been able to adapt to the fast development of the Web nor to the social and ecosystem aspects of service composition.

## 2.3 Mashup patterns

Wong and Hong (2008) studied a sample of mashups to uncover patterns that might indicate requirements for future mashup tools. The study was preliminary but nevertheless interesting. It showed that there were many mashups that did not combine several services but merely provided a new kind of interface to an existing service. For instance, Leaflets ([getleaflets.com](http://getleaflets.com)) are specialized versions of common web sites to run on Apple phone, and oSkope ([oskope.com](http://oskope.com)) provides a visual search interface to Amazon and eBay.

The authors identified the following mashup patterns that they considered interesting with respect to future mashup tools:

- *Aggregation*: Aggregate multiple web sites together or summarize sets of data.
- *Alternate or in-situ use*: Support new methods of interacting with data from a website or support specialized use outside of the typical use case.
- *Personalization*: Use the personal information about the user or allow the user to create personalized information, for example, lists of items of interest.
- *Focused view of data*: Provide indexing or categorization for the contents of a website.
- *Tracking*: Monitoring changes to the underlying data, e.g., new apartments in real estate listings.

## 2.4 End-user programming

In the field of end-user programming, one central question is what makes programming hard. Blackwell (2002) gives three answers:

1. *Loss of direct manipulation*. Direct manipulation allows cognitively simple interaction with a system: the current status of the system is visible, a single action has a single visible effect, and the system can be restored to previous situation. In programming the situation is almost opposite to this. The program will be executed in the future and applied to unknown data. It needs to work in different situations. There is thus no single situation to inspect. Operations can have multiple effects that can furthermore be temporally and spatially distributed.
2. *Use of notation*. The artifact that the programmer creates for further execution must be specified using some notation. All notational systems are artificial, as there is no ideal or natural programming notation. In any notation, the design choices require tradeoffs between which cognitive tasks are facilitated and which are inhibited. In the research on end-user programming, the need to use a textual notation with strict syntax and semantics is considered one concrete problem. Visual notations can alleviate some of these issues but many problems still remain.
3. *Abstraction as a tool for complexity*. Need for abstraction is a result of loss of concrete situation. The expression requires using some notation. According to Blackwell (2002), abstraction is a particularly difficult aspect in programming for end-users.

These problems have been addressed with different kinds of end-user programming techniques. Nardi (1993) divided the discussion into following areas: task-specific programming languages, interaction techniques in end-user application development, application frameworks, and collaborative work practices.

In service composition approaches based on mashups, most of these are used in one form or another. In the following we will discuss the following service composition approaches:

1. *Programming-by-example*: Using a particular instance of execution, input-output relations, or existing programs as basis for creating new programs.
2. *Visual programming*: Replacing the textual programming notation with graphical notation. Graphical notations usually consist of blocks and connectors.
3. *Forms-based creation*: Using forms, tables, assembly canvases, or other special structures to simplify service composition. Spreadsheets belong to this category.
4. *Script-based creation*: Making the programming easier and more natural, e.g. through scripting.

In the rest of the report we will review several different mashup tools, each of which uses one or more of these approaches. We have classified the tools to above-mentioned categories based on the dominant or most visible approach. However, we first start by looking at some tools for use-time service composition, as they do not naturally belong to any of the categories above.

## 3 Use-time service composition

In this section we will review two systems for use-time—or runtime—service composition. It should be noted that in this context, the term “use” means the same as “browsing the Web”. Other potential use contexts, such as many of those encountered with mobile devices, are not necessarily as natural for on-the-fly service composition. The tools reviewed below are Intel MashMaker and Mozilla Ubiquity, both experimental tools running as an extension of Firefox browser (see subsection 7.1).

### 3.1 Intel MashMaker

*“Intel® Mash Maker takes mashups to a whole new level with an innovative and radical new way of browsing the Internet. ... Instead of browsing through websites, Mash Maker's built-in technology enables you to browse through contexts and semantics, as well as easily view information the way you want, and presented the way you want.”* – Intel MashMaker introduction on AMO<sup>3</sup>

Intel MashMaker is a web browser extension—originally for Firefox but now also available for IE—that allows one to create mashups live as part of browsing. This is possible because the mashing up takes place in the browser, and not on a server as is usual. For example, a user could construct a mash-up where the favourite food listed on a friend's MySpace page would be accompanied with a button that enables a reservation at a nearby restaurant selling that food<sup>4</sup>.

MashMaker installs as a toolbar on the browser (Figure 1). Since MashMaker suggests mashups that it can apply to the current page as one browses it, no programming ability is necessary<sup>5</sup>. The suggestions are based on the site content and structure in addition to using a structure/semantic model for specific websites generated by users and stored on the server. When encountering a website for which no structure or semantic model exists, a user can create a new one that will enable others to create mashups with that site. It is also possible to modify the current model by adding new capabilities.<sup>6</sup> In that way, MashMaker harnesses the wisdom of the crowds and relies on the community to teach it about the structure and semantics of web sites<sup>7</sup>.

MashMaker maintains a simple social network of its users. It allows users to share data, widgets, and widget suggestions. Data can be published to friends as writable or read-only. This allows creating ad-hoc social-networking applications (Ennals, 2007). MashMaker has user accounts, and while there is a guest user account, it does not allow creating new mashups, rate mashups and widgets, or bookmark or showing mashup URLs<sup>8</sup>.

Intel MashMaker is still marked as Experimental extension of Firefox. Although it offers interesting possibilities, it has not found its audience as it has been downloaded only 264 times thus far (2/2009). It is possible that it does not offer the functionality that people want or is not easy enough for the majority to use. The last time Intel MashMaker extension was updated was 10/2008.

---

<sup>3</sup> <https://addons.mozilla.org/en-US/firefox/addon/9395>

<sup>4</sup> <http://www.crn.com.au/News/61645,idf-intel-releases-mashmaker-preview.aspx>

<sup>5</sup> <http://software.intel.com/en-us/articles/intel-mash-maker-mashups-for-the-masses>

<sup>6</sup> <http://freegeographytools.com/2007/intels-mash-maker-another-mashup-maker-but-different>

<sup>7</sup> <http://software.intel.com/en-us/articles/intel-mash-maker-mashups-for-the-masses/>

<sup>8</sup> <http://mashmaker.intel.com/web/informationguest.html>

Be that as it may, MashMaker does represent “a bit of a departure for the company,” as Jeff Klaus, marketing director for Intel Mash Maker, admitted. While it enriches the browsing experience, it hardly sells any more hardware.<sup>9</sup>



Figure 1 – Intel MashMaker

### 3.2 Mozilla Ubiquity

Mozilla Ubiquity<sup>10</sup> is an experimental environment for use-time composition of services (Figure 2). It is essentially a collection of easy textual commands that allow the user to look for information and apply that information to current web page as well as other web sources.

The Ubiquity commands, for example *map*, *look*, *add calendar*, *translate*, *email this to*, and so on, resemble natural language and are typed into the command interpreter at the top of the browser window. Ubiquity is not meant just for information integration but for the whole range of tasks possible with a Web browser, such as sending email or managing calendar. Textual commands may at first sound like an inconvenient idea and a step back in the evolution of user interfaces. However, if we consider the idea that people are willing to write text to move to a Web address or to make a search in the Web using a search engine, it seems natural to think they might wish simply to tell the computer what they want it to *do*: show a location on a map, send a message to someone, and so on.

An example use scenario of Ubiquity is to select a portion of the text in a normal browser window, press `alt-space` to get to the Ubiquity command interpreter, and type the command “email this to alyssa”. Ubiquity would suggest possible completions to the command, for example, “email this to alyssa.p.hacker@mit.edu.” When the user accepts the suggestion, Ubiquity brings up a web email program and creates a message pre-filled with the recipient address and the selected content.

An advanced user can also create new commands. There is a community forum called The Ubiquity Herd where these commands can be shared with others. As of 3/2009, close to 800 user-contributed commands have been published in the forum, but that number naturally includes lots of duplicates.

<sup>9</sup> [http://news.cnet.com/8301-10784\\_3-9921313-7.html?tag=blog.promos](http://news.cnet.com/8301-10784_3-9921313-7.html?tag=blog.promos)

<sup>10</sup> <https://wiki.mozilla.org/Labs/Ubiquity/>

How does Ubiquity help the user in composing services? What is the added value compared to the base-line scenario where the user copies and pastes content from one service to another? Basically Ubiquity simplifies user's interaction with services. With a couple of key presses the user can in a flexible manner apply other services into the data at current page. For example, the user can translate a piece of text, show location on a map, email the content to a friend, and so on, all without invoking translation, mapping, or email services and without the need for copy and paste. It is easy to believe that there are users who can learn to use Ubiquity fluently but who would find it too difficult to carry out same actions without it.

Ubiquity achieves this by hiding service invocations behind the commands so that the user does not need to remember Web addresses or to search for available services. The user just tells the system in an abstract level what he or she wants to do, not how to do it. Since users can create new commands and share them in the Ubiquity community, it is possible that the coverage of commands becomes pretty extensive. However, that possibility also creates the problem of discovering and identifying desired commands. Even the auto-completion functionality of Ubiquity would not work well if there are numerous similar commands available.

An obvious limitation of the approach of Ubiquity is that while it supports a composition of the current page with another service, it is not clear at all how that could be generalized to composition of multiple services, unless all the complexities are hidden behind the implementation of a command.

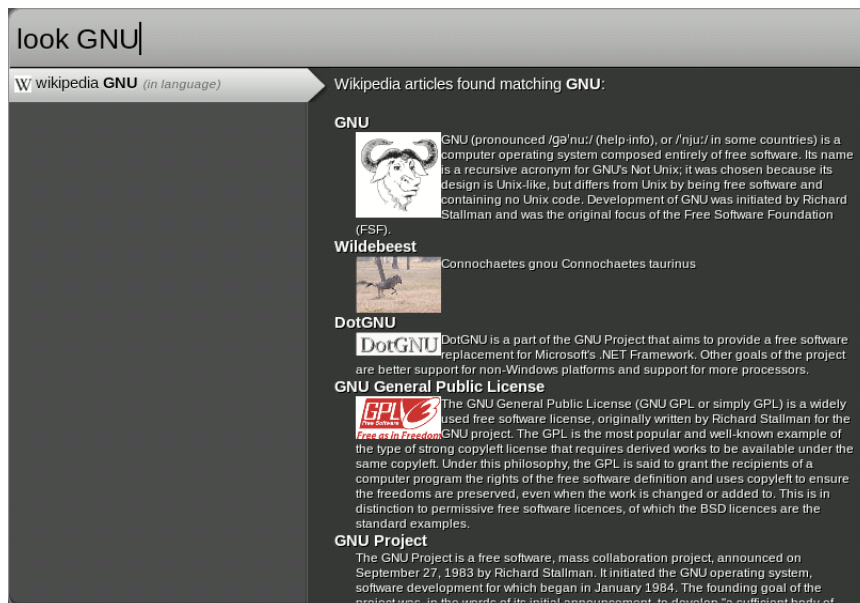


Figure 2 – Mozilla Ubiquity

## 4 Programming by example

*Programming by example* is also known as *programming by demonstration*. Halbert (1984) defined it as programming a software system in its own user interface. The user of the system writes a program by demonstrating what the program should do on a concrete example. The system records the sequence of actions, possibly infers a generalization of the sequence, and can perform it again or re-apply it to other similar cases. The first system working like this was PYGMALION, written by David C. Smith (1975).

Another related meaning for the programming by example is—in contrast to demonstration of a novel example—the versioning of an existing example. This approach can best be called *example modification*, and it has become a ubiquitous strategy in end-user development (Nardi, 1998). Modification of a working example speeds up development as it provides stronger scaffolding than when writing code from scratch (Hartmann, 2007).

### 4.1 Programming by demonstration

From the perspective of the user, programming by demonstration does not require the user to edit a program in any notation. The user works with the system in the similar way as when he or she is ordinarily performing a task. The system is in a recording mode that captures the actions of the user, and suggests generalizations for the actions. The mashup program is created in an implicit manner in the background, although some systems can additionally allow the user to view and edit the program.

#### 4.1.1 Karma

Karma (Tuchinda, 2008) is a mashup tool that enables the integration of data from multiple web sources into a data table through demonstration. According to the model behind Karma, information integration can be divided into the following subtasks:

1. *Data retrieval*. Extracting unstructured data from web pages (in HTML) into a structured format such as a table or XML. The original data may be distributed on the page or span multiple pages. When user clicks a piece of data to extract, Karma uses an XPath generalization scheme to find other data items in similar role, especially data in similar position in a list (in html lists, table rows, or behind item specific links).
2. *Source modelling*. Assigning attribute names for each data column to enable the detection of relationships between the new data source and existing ones. Karma keeps repositories of previously extracted data tables and it can suggest column names based on the values of data. For example, if the value is an address and there is a previous data table containing addresses, Karma can recognize the similarity and suggest the column name “Address” from the previous table also to the new data.
3. *Data cleaning*. Transformation of data values into appropriate, canonical format. Cleaning can also include fixing misspellings. For example, a name written as “Hacker, Alysssa P.” may need to be corrected and transformed into the form used in previously extracted data tables such as “Alyssa P Hacker”. While the correct form to use is up to the user, it is essential that all data that must be combined to use the same form. The user specifies for one or more cells what the end result of the cleaned data should look like. Karma automatically deduces a cleaning transformation that can then be applied to all cells in the same column.
4. *Data integration*. Combination of two or more data sources together. This is basically a join operation between tables that is possible when the column names match (achieved by source modelling) and data values are the same (achieved by data cleaning). The user fills out the empty cells in a table by picking values or attributes based on Karma’s suggestions. Karma then infers the correct table to join with the original one.

5. *Data visualization.* Displaying of the final data, for example, on a map, or as a table or graph. This task is outside the functionalities provided by Karma.

The programming by demonstration is supported in a different manner in each of the steps. In data retrieval step, Karma proposes generalizations based on *lists* in which data pointed to by the user resides. In source modelling, Karma proposes column names based on previously extracted data. In data cleaning step, Karma uses deduction to find out a correct transformation based on the examples that user has provided. Data integration, on the other hand, is based on reasoning about the user operations to narrow the choice of which tables to join. The user does not need to search for data sources to integrate nor know database operations. The integration based on selection of existing data values guarantees that such exist.

Karma puts these separate techniques into a unified framework that enables end-users to build information integration mashups in a data-centric manner without encountering code at any point. This is in contrast to visual programming approach where the user often needs to be familiar with programming concepts, even if presented in a visual language.

Karma's strength is the extraction and combination of data from existing web sources. It does not, however, support arbitrary operations with tables such as executing queries based on a table or collecting data based on computation performed on the data (Lin, 2009).

#### 4.1.2 Vegemite

Vegemite (Lin, 2009) is a mashup tool based on programming by demonstration, iterative and interactive manipulation of data by the user, and mixed initiative interaction. It has been designed to support the use-time creation of ad hoc mashups but users can also store and share the mashups created. Vegemite consists of two main parts:

- *VegeTables*, a spreadsheet like tables for storing data, and
- *CoScripter* engine (Little, 2007) for recoding and playing back actions on Web pages.

The user interface of Vegemite is shown in Figure 3. There are three areas in the interface: current web page on top-right, script panel (CoScripter) on the left, and data table (VegeTable) at the bottom.

The user extracts data from a web page to the VegeTable. The data extraction works by demonstration in a manner similar to in Karma. In the extraction mode Vegemite recognizes the elements that the user is clicking on the web page. It identifies all the possible parent elements of the clicked elements in the DOM tree and computes a partial XPath from each parent to the clicked elements. Then it identifies for each parent the possible other elements that could be reached with a suitable generalization of the XPath. Vegemite uses the parent that creates most siblings to the clicked nodes as the current hypotheses for generalization. It highlights the new elements to the user as an extraction suggestion. If the user accepts the suggestion, the elements are copied to the current VegeTable. The suggestion is recomputed after each new click by the user.

The data extracted to the VegeTable can then further be processed using scripts. Vegemite is not limited only to information integration but the mashups created can perform any action recordable by CoScripter on the data. The CoScripter can be used to record most actions by a user on a web browser. CoScripter produces a user understandable script of the operations executed and this script is then presented in the script panel where it can be further edited. Mashups can infer additional columns to the table, send email messages, and make calendar entries, and other activities that can happen inside a web browser. The derived information can be computed by different services on the web. For example, in a mashup that computes distances from a possible apartment to important places like job, schools, and hobbies, the script can visit a map web site and insert information to compute the driving distance.

A small user study reported in Lin (2009) is based on three scenarios: determine the walkability of apartments in a real estate listings, compute the driving distance to nearby yogurt places, and extract a certain category of archived visa bulletins. The first two required combining two existing services and the last one selecting data from one service. There were 8 users in the study. The results were encouraging. Of the 24

tasks the users were able to complete 18, failed 4, and skipped 2 due to time restrictions. There was evidently a steep learning curve, since two of the four failures happened in the first task, and the subjects generally struggled in that task. Three of the four failures happened to the two non-programmers participating in the study. Several users found the process of composing a mashup overly complex, and some reported that they could not have been able to use the tool without instruction. Obviously, the usability of the tool needs to be improved.

The main advantage of Vegemite over Karma is its ability to derive new information based on extracted information and use the services available on the Web to do this, for instance, using currency converters to create a column to calculate prices in other currencies. It is also possible to make other actions that are not possible in information integration tools such as Karma.

Vegemite is implemented as a Firefox extension. CoScripter is also available separately as a Firefox extension.

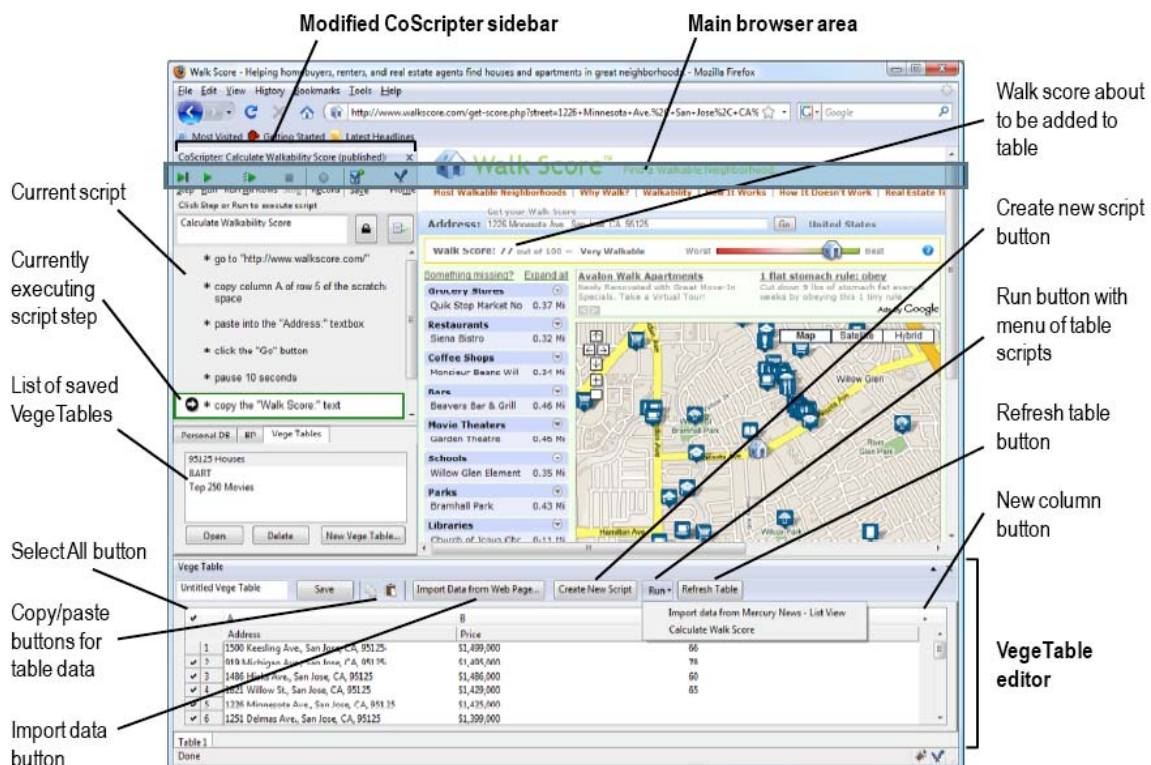


Figure 3 – Vegemite user interface

## 4.2 Example modification

Example modification is a development strategy that users have been using in script-based systems for ages. In this subsection we review one system based on that approach, called d.mix (Hartmann, 2007).

### 4.2.1 d.mix

d.mix is a tool for design-time creation of mashup artefacts. It is based on the concept of example modification, meaning that a user can take an existing service and modify it slightly to create a new service with similar structure.



The key insight in d.mix is that many complex services available on the web today provide an API that can be used to access their content and functionality. The APIs can, however, be complex and difficult to understand and learn. When a user finds a web site that performs in a way that could also be used in another service he or she envisions, it is difficult to connect what he or she sees and what API calls need to be made to produce it.

d.mix uses two ways to help users take advantage of the APIs. First, it provides so-called *site-to-service maps* that simply present in a visual manner which parts of the user interface have been created through which API calls. In the Figure 4 there is an example of such map. The map allows using the surface structure of the web as a mean to facilitate application development.

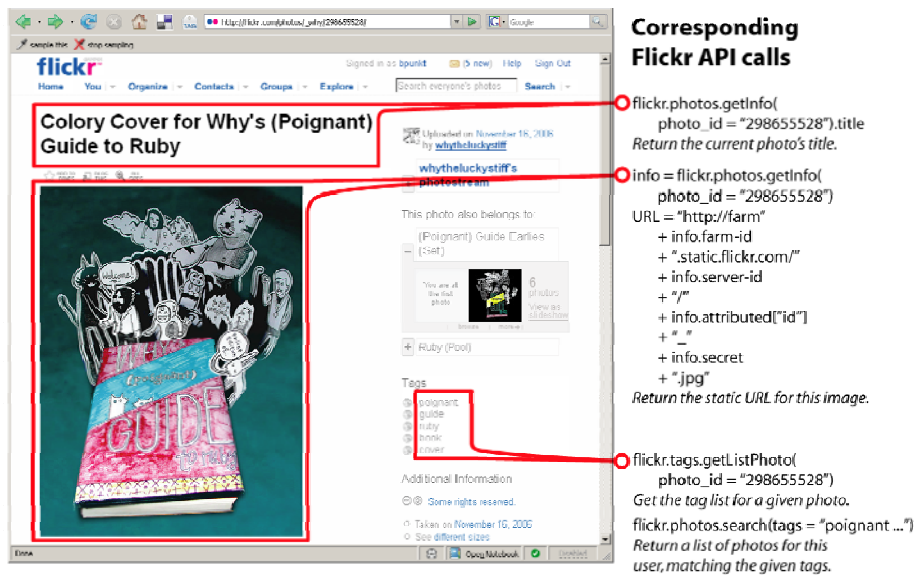


Figure 4 – Site-to-service map in d.mix

Secondly, d.mix has a server-side *active wiki* that hosts the scripts, and provides an environment for authoring and sharing of source code and actual services. The active wiki uses the social structure of the Web to facilitate service development. The ecosystem of developers and end-users can in complementary roles create solutions and use scenarios that benefit all the participants.

The original web site does not need to provide any support for the d.mix. The active wiki maintains a collection of site-to-service maps that have been contributed by developers. The use of d.mix takes place through a programmable http proxy that rewrites the current web page and adds JavaScript annotations to indicate services that can be sampled from a page—based on the site-to-service maps—and to visually augment the services with dashed borders.

The user can add d.mix buttons—*sample this* and *stop sampling*—to the browser's bookmark bar. Implemented as a bookmarklet—a bookmark containing JavaScript instead of an address—the pressing of sample this button sends the current browser location to the d.mix active wiki. This invokes the d.mix proxy that annotates the page using its site-to-service map. The user can click on the areas with dashed borders to select elements and send them to the active wiki. The selected elements can be parameterized and combined into a new page.

Hartmann (2007) reports on a successful evaluation study with d.mix. The eight subjects all had previous Web development experience and knowledge of the Ruby programming language used in d.mix. The subjects managed to create working services in 21 out of 24 tasks.

## 5 Visual programming

Another way to give the users the freedom to create their own applications and tune them as they like is to use an editor that has graphical components to do visual composing. Well-known visual composing software is Repenning's (1993) AgentSheets, which is "a substrate for building domain-oriented, visual, dynamic programming environments that do not require traditional programming skills." The idea behind AgentSheets was to support the perception of programming as problem solving by having mechanisms to incrementally create and modify spatial and temporal representations. In AgentSheets (Figure 5) this is implemented by extending the object oriented approach with an "agents in a sheet" paradigm that consists of a large number of autonomous, communicating agents organized in a grid, and whose look and behavior is controlled by designers using the system to create dynamic visual environments with domain-oriented spatial-temporal metaphors.

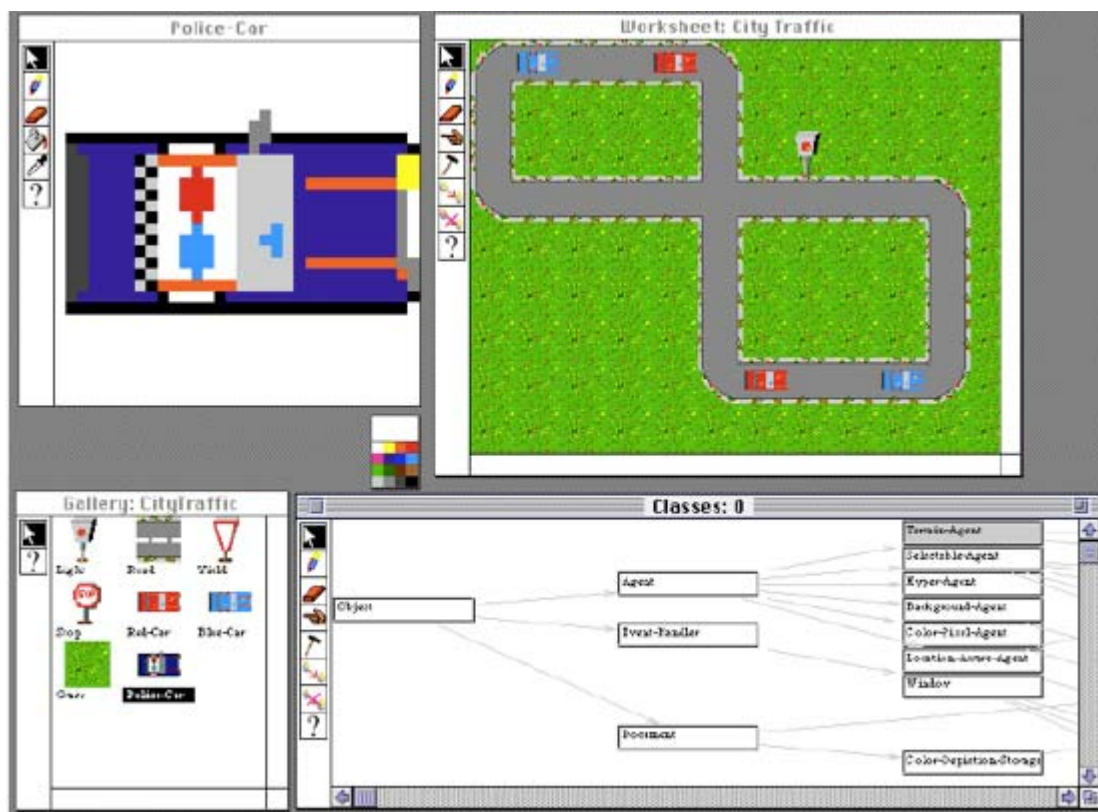


Figure 5 – AgentSheets User Interface.

A user study (Repenning 1991) of a real world application using AgentSheets showed that visual composing can provide efficient means for creating and extending intelligible applications. Smith's (1994) KidSim application also used agents, persistent software entities dedicated to a specific purpose, in visual composing of a game for children, but it also used example-based programming techniques for recording certain rules in the game, for example how the character that the player controlled could move. When tested with children, this combination worked well and the kids were able to create different kinds of games easily.

With the introduction of Web 2.0, application building has been shifting from native desktop applications to Internet-based application that have a web interface. There are many tools that can create so called "mashups," collections of information or services found from the Internet. Barret et al. (1997) introduced the Web Browser Intelligence (WBI) that could can annotate hyperlinks with network speed information, record pages viewed for later access, and provide shortcut links for common paths, thus personalizing the user's web experience by joining personal information with global information to effectively tailor what the user sees.

We will now discuss in more detail how two systems that use visual composing kind of methods for creating web based applications, namely Popfly and Yahoo Pipes!, work .

## 5.1 Popfly

*“Popfly is designed to enable non-technical users to be able to create without code, then share creations with friends by embedding them everywhere. You shouldn’t have to write code to be able to customize services on the Web; it should be as easy as visually ‘snapping’ together existing components. You can think of Popfly as the ‘YouTube for applications’ where you can discover, rate, comment, and remix user-generated applications and samples.”<sup>11</sup>*

Microsoft’s Popfly, a free webpage, mashup, and online game creation and sharing environment, requires installing Microsoft Silverlight browser plugin that enables such features as animation, vector graphics, and audio-video playback. In addition, Windows Live ID is required. Currently in public Beta stage, Popfly consists of Game Creator, Mashup Creator, Web Creator, and Popfly Space that has some social networking features.

Gadgets created with Popfly can be embedded on one’s own web page(s) or in other web services that support gadgets from outside, such as Windows Live Spaces (MSN Spaces), Facebook, Twitter, or Dapper—or dozens of others<sup>12</sup>. In addition, these gadgets can function as Windows Vista sidebar gadget.

In addition to being used as gadgets in the Vista sidebar or being embedded on web pages and services, the mashups, web pages and games created with Popfly are saved in each user’s Popfly Space. Public projects that are shared can be “ripped” by other users for understanding how they function and to use as templates of their own work.

It is permitted to create commercial applications with Popfly but some blocks—Popfly’s ready-made components that are put together to build applications—are restricted to non-commercial use only.

In this review, we focus on Mashup Creator, as Game Creator is more complex and slightly outside of this report, as is Web Creator, a tool for creating web pages without HTML coding.

### 5.1.1 Mashup Creator

Popfly’s mashup creator is based on selecting ready-made blocks and connecting them to each other. Popfly blocks can access information from web sites, such as RSS feeds, images from Flickr, or videos from YouTube, thus in effect functioning as middlemen. In addition, there can be libraries of various functions, offering services such as a calculator or timer. They can also act as a display surfaces for the data accessed by other blocks.

A Popfly block consists of block description and block code. The description tells what the block is all about, what methods it offers, and what parameters the methods accept while the code contains the methods to be invoked when the block is used<sup>13</sup>. The block metaphor can be likened to that of objects in object oriented programming<sup>14</sup>.

Blocks one wishes to use are selected from categories or found by searching<sup>15</sup>. The categories are: New & Updated, Display, Fun & Games, Images & Video, Local Information, Maps, News & RSS, Shops, Social

---

<sup>11</sup> <http://www.popflywiki.com/FAQ.ashx>

<sup>12</sup> <http://www.popflywiki.com/FAQ.ashx>

<sup>13</sup> Walkthrough: Creating a Block with Popfly Explorer (PDF): <http://go.microsoft.com/fwlink/?LinkId=106942>

<sup>14</sup> <http://www.guardian.co.uk/technology/2007/oct/21/popfly>

<sup>15</sup> For the complete list of blocks available, see <http://www.popflywiki.com/List%20Of%20Popfly%20Blocks.ashx>.

Networks, Tools, Everything Else, and My Blocks & Data, the last category consisting of the blocks that the user in question has built. Only one category can be open at one time.

Mashups are built with available blocks by dragging blocks from the left pane to the design surface or by clicking a block on the left pane to move it to the design surface (Figure 6). The block parameters can be adjusted by double-clicking the block or by clicking the wrench icon on the toolbar next to the block. This is called “zooming in” in Popfly. The parameters are adjusted through form elements with the ones marked with an asterisk being required (Figure 7). Clicking the wrench icon gets one back to the design surface view. Another way to work on the block is through the advanced view that allows the user to “tweak” the JavaScript behind the block.

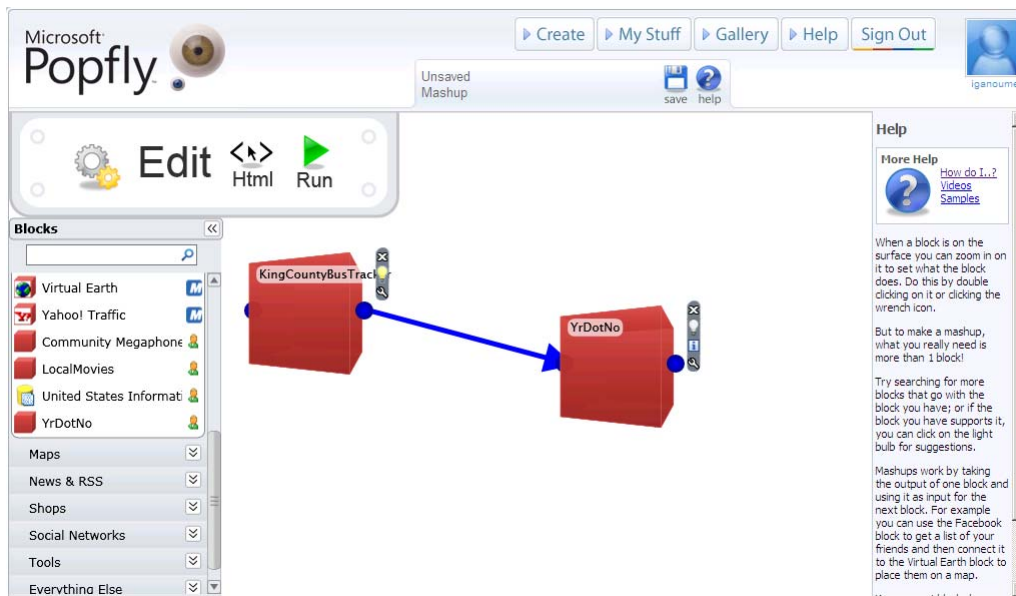


Figure 6 – Popfly design surface with the available blocks on the left

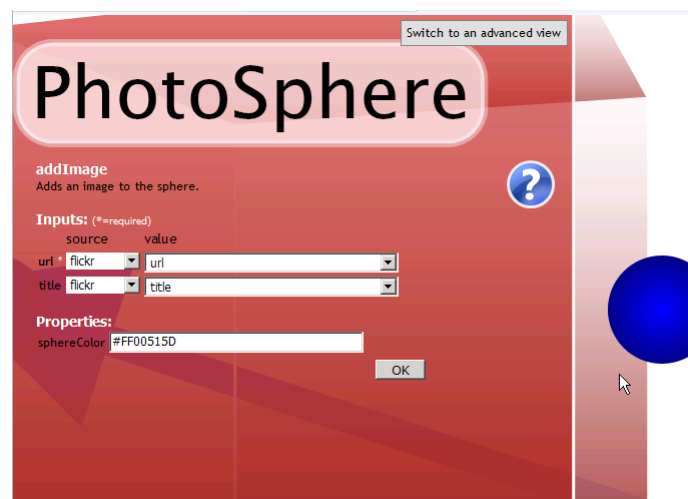


Figure 7 – Zooming into the parameters of a block in Popfly

In order to create a true mashup, more than one block needs to be added. Some blocks are not compatible, but the interface does not give any pre-warning about this. There is a pop-up warning after the fact that points out why the adding of the particular block was not possible. For instance, a “display” block already on

the design surface cannot accommodate a second “display” block as both are used to present the contents of the mashup.

A simple, logical pair to connect, for example, could be one’s holiday pictures from Flickr<sup>16</sup> (requires getting a developer key from Flickr to give to the block) and connecting this block (called unsurprisingly “flickr”) with an image display block, such as Carousel or PhotoStack, to create a photo-showing gadget that could then be used on one’s web page or embedded in, say, Facebook<sup>17</sup>.

The way to connect blocks is to click one block and then move the cursor to another block. The interface shows a blue arrow to indicate that a connection will be created. The connection is established by clicking the target block, and the blue arrow becomes permanent. The arrow can be seen in Figure 8. The connection can be undone by clicking the blue arrow, which then disappears.

The user can search for blocks that connect with the one on the design surface by clicking the light bulb icon next to the block (Figure 8). However, new blocks are suggested only in relation to the block whose light bulb is clicked, and the suggestions do not take into consideration other blocks on the surface.

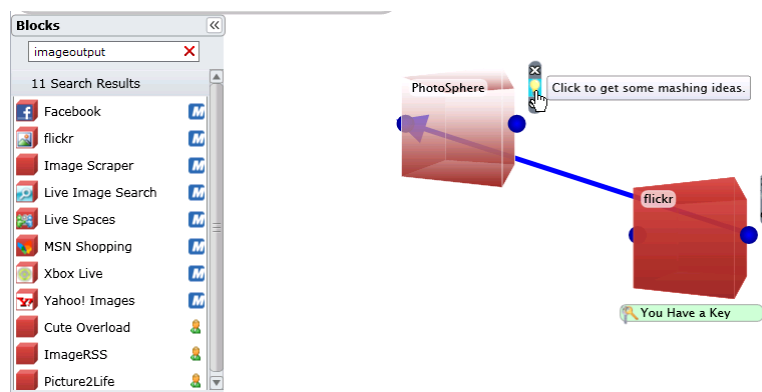


Figure 8 – Popfly shows suggestions in the left pane when the light bulb is clicked

While many basic blocks are provided “By your friends at Microsoft”, there is a growing number of blocks built by Popfly user community available. The blocks by Popfly personnel are identified with a white “M” in a blue box icon (Figure 9) while a user icon (familiar from Messenger) identifies a user-contributed block. Blocks build by Microsoft come first while the user-created blocks come below them. User-contributed blocks come with a warning “Use at your own risk” in red. The impact of the warning may be mitigated at least to some extent with a red heart on the right corner of the pop-out pane next to which the number of “fans” for the block is displayed (Figure 9).



Figure 9 – M by Microsoft versus user-built block with a warning and no. of fans

<sup>16</sup> <http://www.flickr.com>

<sup>17</sup> <http://www.facebook.com>

In addition to becoming a “fan” of a user-contributed block, other users can also comment on (and previously rate with a 5-star scale) the user-contributed blocks. In order to do that, one must first access the block details (Figure 10) by clicking the “i” icon next to the block, as only the number of fans is visible without opening the Block Details. Also, when searching for a block with tags (also accessible only when Block Details are visible), the results are listed in the order of the number of “fans”. Other users cannot add tags freely to user-contributed blocks, as only the contributor can set tags for the block.

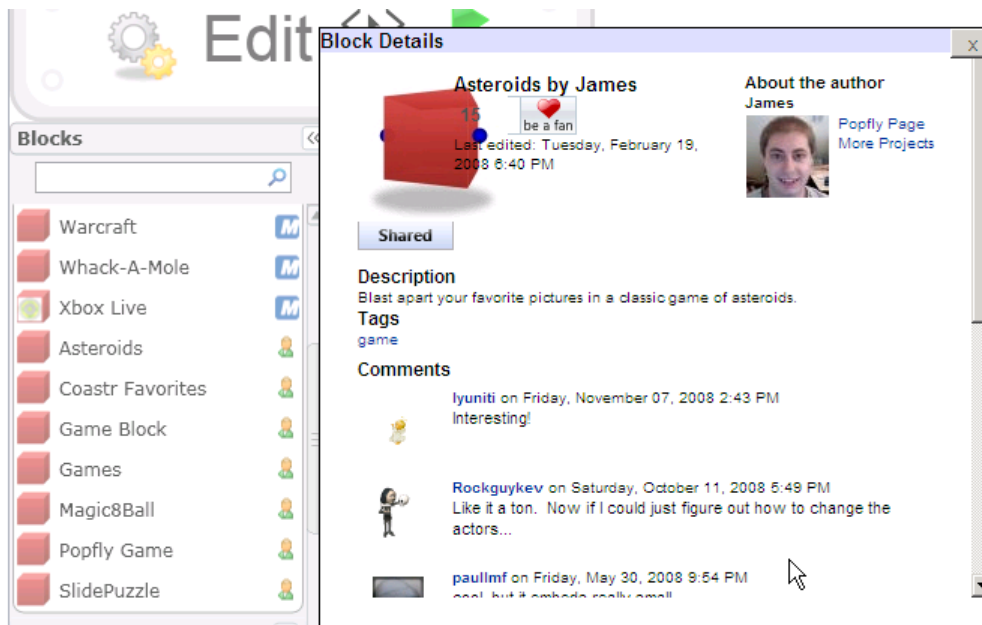


Figure 10 – Block details in Popfly

A mashup being constructed can be previewed inside Popfly simply by clicking the preview button. While the design surface view does show the blocks and their connections, this does not necessarily give a clear idea of what the final mashup would look like, and so a previewing function is a necessary part of the process.

### 5.1.2 Service composition with Popfly Mashup Creator

Building mashups with Popfly’s ready-made blocks requires no programming. The interface even helps the user to select suitable blocks to connect to each other. Of course, basic understanding of what mashups are and some kind of idea of what one is trying to accomplish helps, but with Mashup Creator, one can build mashups even through pure exploration. While the user can adjust the JavaScript code behind the block by going to the advanced view when zoomed in on the block, this is not necessary even for creating rather complicated mashups.

Users can also create blocks that others can then use. While there are basic file-editor tools for this in Popfly, users are instructed to download Microsoft Visual Web Developer Express and Popfly Explorer plugin that includes a block-building editor, as the tools offered in Popfly are rather clumsy for the purpose. Block-building requires at least rudimentary knowledge of JavaScript and XML although existing blocks can be “ripped” to see how they work and how others have done it. Presentation-layer blocks can use AJAX, DHTML, and Silverlight (XAML).

“Blocks have defined input and output parameters and operations (methods) that are used to connect them between other blocks. Each block also has an xml metadata file that describes what the block does.”<sup>18</sup> Users, however, are not allowed to create custom blocks that would require passwords or developer keys because of

<sup>18</sup> <http://www.popflywiki.com/FAQ.ashx>

the potential for abuse, and users wishing for such blocks are instructed to contact Popfly for implementing such blocks.

Popfly provides developer's wiki<sup>19</sup> and other instructions and how-tos for the users, including YouTube<sup>20</sup> videos. Furthermore, contextual help in form of built-in tutorials is displayed on the right of the design surface in the Tutorial pane. Consequently, getting started with Popfly is easy enough.

Thus, Popfly enables users to build mashups without any programming knowledge and with little understanding of the technologies under the hood. At more advanced levels, knowledge of JavaScript, XML, DHTML and AJAX begins to be required, depending on what is being built. Also, familiarity with Silverlight (XAML) is necessary for some applications.

### 5.1.3 Game Creator, Web Creator, and Popfly Space

Popfly Game Creator allows users to create games without any programming skills. Games can be created starting with a game template or from the scratch with a library of necessary ingredients. Starting from a scratch, one first chooses some actors (people, spaceships etc.) and then selects a game background. Then one places the actors, both user-controlled one and game-controlled one(s) on the canvas. Layout having thus been accomplished, behaviors, such as movements and shooting, can now be added to the actors. For the user-controlled actors, behaviors are then connected to events, such as pressing of an arrow key for movement (Figure 11) and pressing spacebar for shooting. Different projectiles can be associated with shooting, as can be sound effects, directions (of movement) and speeds (of movement). So it goes, piece by piece being chosen from the libraries without any need to understand coding<sup>21</sup>.



Figure 11 – Selecting a key for an event

Those with JavaScript coding experience can access the scripts behind behaviors and tweak them directly at code level. The final game is all the time visible and can be played at any time to see the game as it stands at the moment. Thus, any effect of code tweaking can be checked instantly.

After saving, games can be exported to Facebook (or equivalent), used as a Windows Live Gadget, Vista sidebar gadget, or shared inside Popfly. Game categories can be seen in Figure 12. One game can belong to more than one category.

<sup>19</sup> <http://www.popflywiki.com/>

<sup>20</sup> <http://www.youtube.com/>

<sup>21</sup> The whole process can be viewed at: <http://www.youtube.com/watch?v=YoCjsEF5Wbo.s>



Figure 12 – Game categories in Popfly

As the name implies, Popfly Web Creator (or Popfly Page Designer) is a tool for creating web pages and building web sites. No knowledge of HTML is necessary. However, web page making requires Visual Web Developer that is available after installing Popfly Explorer. Web Creator details are outside of the scope of this report.

Each user has his or her Popfly Space, a storage space of 100 MB per user<sup>22</sup>, where the mashups, games, and web pages are stored, and where users have their customizable profile page. There are also some social networking features, such as “Friends” and “fans”—one can be a “fan” of a person, game, or a user-contributed mashup block. However, the social networking features are not extensive.

#### 5.1.4 Current state

When Microsoft’s CEO Steven Ballmer introduced Popfly at the Web 2.0 Summit with great aplomb in Oct, 2007, just 2 months after the release of Silverlight 1.0 that was to take on Adobe’s Flash<sup>23</sup>, many pundits reacted positively<sup>24</sup> although there were some notable naysayers as well<sup>25</sup>. Although the ease of making mashups and later on games (Game Creator alpha was released in May, 2008<sup>26</sup>) with Popfly was praised, the system never really caught on with the public<sup>27</sup>, and now the whole project is under the threat of being mothballed, as Microsoft is shuffling its organization amidst layoffs<sup>28</sup>.

The question to ask is why Popfly has never caught on. Is it because users were unable to use it, they had no use for it, or never even found it? Or is it something else? Was Microsoft wrong in employing a quintessentially programming metaphor of blocks (objects of object-oriented programming) to teach neophytes programming basics? Perhaps non-programmers are simply not raring to program. Be that as it may, the Popfly saga should remind all that mashing it up with users is challenging, and simply providing easy-to-use tools alone does not guarantee success.

## 5.2 Yahoo! Pipes

*“Yahoo’s new Pipes service is a milestone in the history of the internet. It’s a service that generalizes the idea of the mashup, providing a drag and drop editor that allows you to connect internet data sources, process them, and redirect the output. Yahoo describes it as ‘an interactive feed aggregator and manipulator’ that allows you to ‘create feeds that are more powerful, useful and relevant.’ While it’s still a bit rough around the edges, it has enormous promise in turning the web into a programmable environment for everyone.”* - Tim O’Reilly (Feb. 7, 2007)<sup>29</sup>.

<sup>22</sup> [http://en.wikipedia.org/wiki/Popfly#Popfly\\_Space](http://en.wikipedia.org/wiki/Popfly#Popfly_Space)

<sup>23</sup> [http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=202404324&cid=RSSfeed\\_IWK\\_News](http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=202404324&cid=RSSfeed_IWK_News)

<sup>24</sup> For example, [http://etech.eweek.com/content/application\\_development/microsoft\\_mashes\\_it\\_up\\_with\\_popfly.html](http://etech.eweek.com/content/application_development/microsoft_mashes_it_up_with_popfly.html) and [http://www.usatoday.com/tech/products/cnet/2007-05-18-microsoft-popfly\\_N.htm](http://www.usatoday.com/tech/products/cnet/2007-05-18-microsoft-popfly_N.htm)

<sup>25</sup> <http://www.guardian.co.uk/technology/2007/oct/21/popfly>

<sup>26</sup> <http://www.informationweek.com/news/internet/webdev/showArticle.jhtml?articleID=207500450>

<sup>27</sup> [http://blogs.computerworld.com/popfly\\_never\\_caught\\_on](http://blogs.computerworld.com/popfly_never_caught_on)

<sup>28</sup> <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9126718>

<sup>29</sup> <http://radar.oreilly.com/archives/2007/02/pipes-and-filte.html>



O'Reilly certainly was not the only one to welcome Yahoo Pipes with enthusiasm<sup>30</sup>. Yahoo Pipes was born as a public beta in Feb. 2007, and has continued to attract increasing numbers of users.

Yahoo Pipes offers a visual drag-and-drop environment for building mashups that aggregate web feeds, web pages, and other services without having to write one line of code of any kind. The basic blocks—modules—are provided by Yahoo Pipes but the Pipes—the resulting mashups—can be shared, cloned, and their sources viewed if the maker has made it public. The cloned Pipes can be use as modules, which is as far as Yahoo Pipes go in allowing users to contribute modules. Thus, all the underlying modules are by Yahoo, and users can only contribute different combinations of them spiced up with different parameter settings (including which feeds to use). While this increases the security of the blocks—and in that way is comparable to Popfly's policy of not allowing user-contributed blocks that would require passwords or developer keys—it might stifle some user creativity.

While no coding whatsoever is involved, the vocabulary and – to an extent – the underlying metaphor of the Pipes come from the UNIX world. The “pipe” idea itself appears to come from the UNIX world where pipes—“|”—are used to combine data sources and to perform actions on the data<sup>31</sup>. Having basic UNIX lingo down certainly helps decipher the functionalities offered by the modules.

While the resulting mashups can be used as such, Pipes is rather a data mashup tool to provide mashed up content to be used in other applications rather than a complete Web Mashup builder<sup>32</sup> for creating standalone applications, and so the pipes built with it can also function in the background of other applications or services, for example as a source of a Twitterfeed<sup>33</sup>.

Yahoo Pipes comes with clear tutorials and documentation, and being able to start from existing pipes to figure out what makes them tick helps as well. Although the environment might take some getting used to<sup>34</sup>, there is really no reason why anybody could not build mashups with Yahoo Pipes with some perseverance. The question is, of course, how much perseverance Joe Blow has. The answer to that question tells how widely it will be used.

### 5.2.1 Service composition with Yahoo Pipes

Yahoo Pipes visual editor, a JavaScript authoring tool, consists of three parts: Library (on the left) that lists available modules (grouped under headings by function), your saved pipes, and your favorite pipes from other users, Canvas (the main area) where the pipes are assembled and tested, and Debugger (bottom) that lets one see the output of the active module at any stage of building a mashup (Figure 13). Library and Debugger panes allow resizing.

---

<sup>30</sup> For instance, <http://jeremy.zawodny.com/blog/archives/008513.html>, <http://www.mattcutts.com/blog/review-yahoo-pipes/>, and <http://radar.oreilly.com/archives/2007/02/yahoo-pipes-the.html>.

<sup>31</sup> <http://radar.oreilly.com/archives/2007/02/pipes-and-filte.html> and [http://en.wikipedia.org/wiki/Yahoo\\_Pipes](http://en.wikipedia.org/wiki/Yahoo_Pipes).

<sup>32</sup> [http://en.wikipedia.org/wiki/Yahoo\\_Pipes](http://en.wikipedia.org/wiki/Yahoo_Pipes)

<sup>33</sup> <http://uk.techcrunch.com/2009/02/02/as-snow-hits-the-uk-the-twitter-mashups-storm-in/>

<sup>34</sup> <http://lifehacker.com/software/feeds/geek-to-live--create-your-master-feed-with-yahoo-pipes-235726.php>

Figure 13 shows the Yahoo Pipes visual editor with Library on the left listing the available modules, Canvas where the pipes are assembled, and Debugger at the bottom showing the output of the active module, in this case Pipe Output, as indicated by its orange colour. This pipe takes world news from Chinese language news

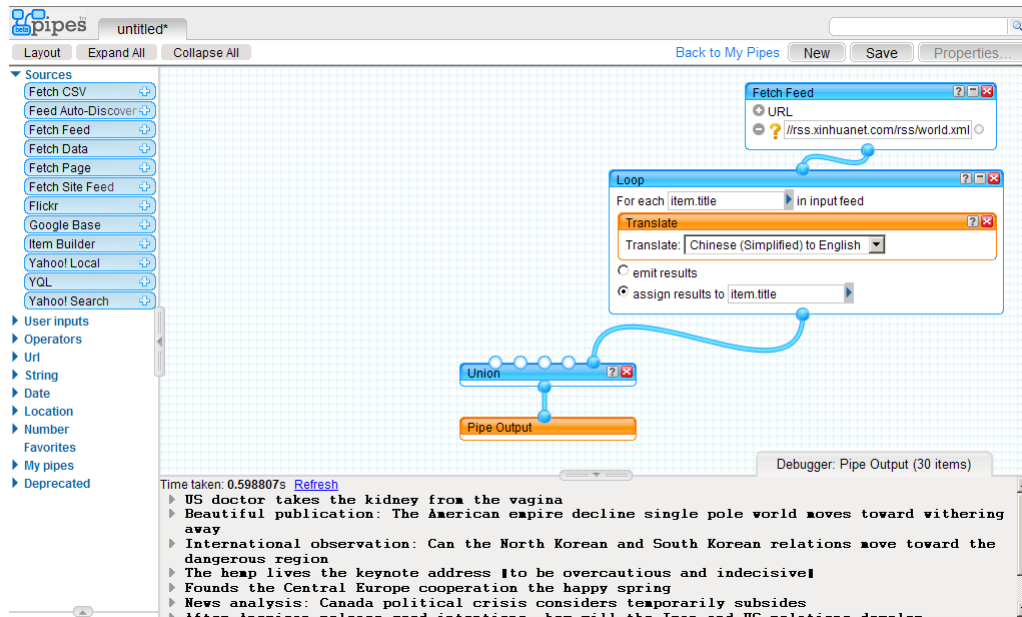


Figure 13 – Yahoo Pipes visual editor

from Xinhua (Fetch Feed module) and translates the item titles into English (Loop module with Translate module inside of it). The results then go through Union module (unnecessary here as no other sources are available to be combined with the Xinhua content) into the Pipe Output module, and the results are displayed in the Debugger.

The available modules listed in the left are grouped by their functionality. Each module performs a single, specific function, such as fetch a feed (Fetch Feed under Sources), sort (Sort under Operators), or breaking a string by a token (String Tokenizer under String).

Many modules have parameters or input fields that can be filled out like any form fields. For instance, the Fetch Feed module takes as a parameter a feed URL (Figure 14). Sometimes the number of the parameter instances can be increased. The plus sign next to the parameter name is clicked to add another instance of it.

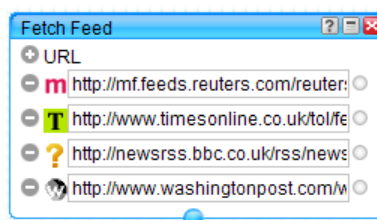


Figure 14 – The module Fetch Feed takes the URL of the feed as a parameter

Some modules accept other modules as parameters. Loop module is a good example of this. It loops the selected part of the source feed and performs the action that the added module defines on it. Figure 15 shows how the Loop module indicates with a rectangular broken line that it is ready to accept the Translate module being dragged to the canvas. The rectangle turns red when the module is ready to be dropped in and Figure 16 shows the Loop module when the Translate module has been dropped inside of it.

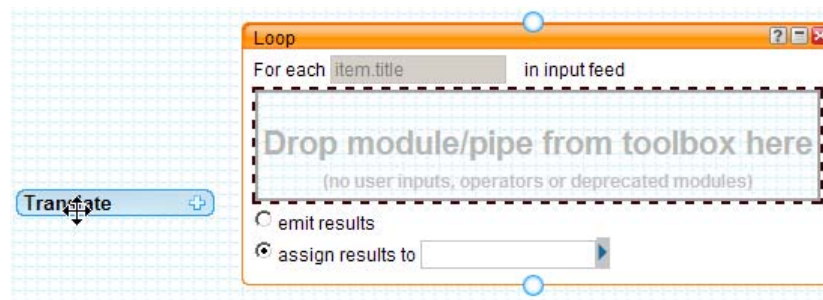


Figure 15 – Visual indication of a possible drop target

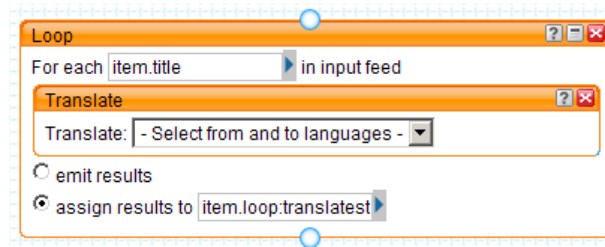


Figure 16 – The Translate module dropped into the Loop module

The module shows the parameter fields when expanded (default when dragged to the canvas). If contracted, they only show the title bar and the pipes. Collapsing an expanded module and expanding a collapsed module are both done by double-clicking the title bar. In addition, the modules have the standard maximize box in the top-right corner.

As the names of the modules show, Yahoo Pipes is “not entirely for the faint of heart”, as Tim O’Reilly puts it succinctly<sup>35</sup>. Releasing the true power of Yahoo Pipes requires some understanding of programming and UNIX terms. While figuring out what “fetching” and “sorting” might be all about does not require a Ph.D. in computer science, Joe Blow might have harder time with String Regex, String Tokenizer, or Union. The tooltips do help, such as “Merge one or more feed to create a new one” for Union, but the vocabulary used is both the strength and downfall of the Pipes. For those familiar with the lingo, everything is crystal clear and the vocabulary is empowering, but for those not familiar with it, it can further mystify already difficult concepts. (No wonder that the old UNIXheads and technically oriented pundits welcomed the Pipes with such fervor<sup>36</sup>...)

Having one’s saved pipes and one’s favourite pipes from other users available in the Library means that one can study other pipes to see what makes them tick in addition to being able to use old projects as building blocks for new, more complex projects as well as across multiple projects.

The saved pipes that you drag onto the canvas are known as subPipes. They behave like regular modules with an addition of an “open” link on their top bar that opens the module to a new tab in the editor where it can be edited like a regular Pipes project. As mentioned, SubPipes are the only way for users to make and contribute modules.

The modules are dragged and dropped—clicking on the arrow to the right of the module name also works—from the Library onto the Canvas. They are wired together by “pulling” pipes from output terminals to input

<sup>35</sup> <http://radar.oreilly.com/archives/2007/02/pipes-and-filte.html>

<sup>36</sup> For example: <http://www.mattcutts.com/blog/review-yahoo-pipes/>, <http://jeremy.zawodny.com/blog/archives/008513.html>, and <http://radar.oreilly.com/archives/2007/02/yahoo-pipes-the.html>.

terminals: Left-click an output terminal and hold the mouse button down while moving the cursor (that has changed into a four-arrow cursor) across to an input terminal. Input terminals are on top of the modules and output terminals below them. When pulling a pipe, there is a visual pipe being pulled and that pipe will remain visibly connecting the modules. Furthermore, when activating an output terminal, the terminals that are ready to accept the pipe shine orange, thus providing visual aid to the user (Figure 17).

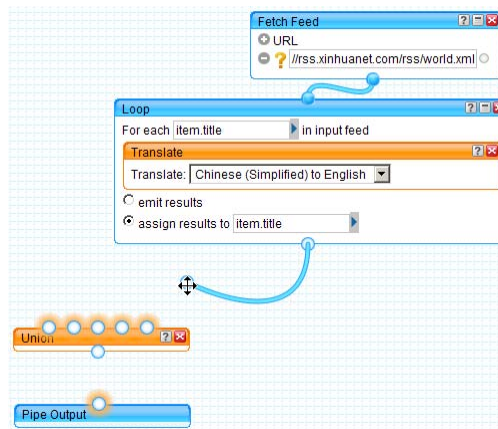


Figure 17 – The terminals that are ready to accept the pipe shine orange

Holding a cursor over a terminal will show what kind of data it receives or emits. In addition, if there is a pipe connected to the terminal, a scissor icon appears on a mouse-over. The scissors also appear if a terminal with a connection is clicked. The pipe is deleted by clicking the scissor icon (Figure 18).

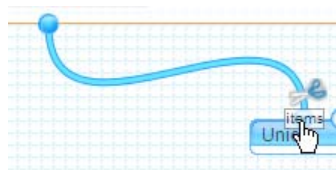


Figure 18 – Mouse-over for a scissor to sever the connection

The Debugger at the bottom pane allows previewing the output of the active module at any stage of work. By moving the cursor over the debugger tab on the right side of the Debugger, you can change which modules' output to see (Figure 19). To view the ultimate output, view the output of the Output module. Clicking the "refresh" link makes sure that one sees the current situation.

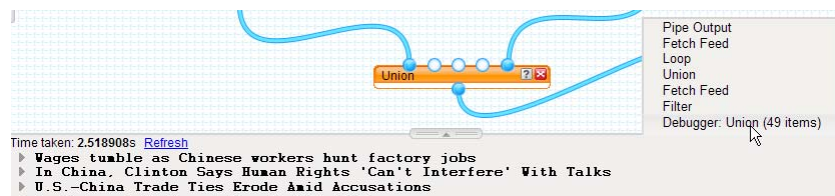


Figure 19 – The output of the Union module with the tab for changing the active module opened

Clicking the small triangles on the left side of the debugger allows one to examine in detail the output content (Figure 20).

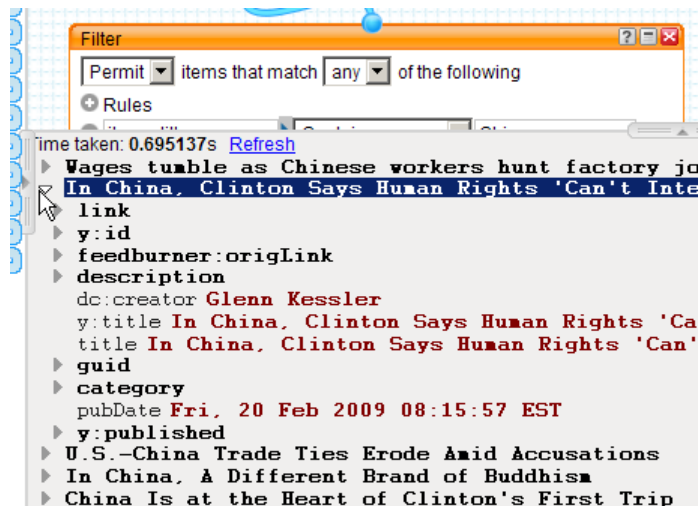


Figure 20 – One of the titles opened for further details

When a pipe is ready, it can be saved by using the “Save” button in the upper right corner of the Canvas pane. If you choose to publish your pipe, it becomes part of the pipes directory that is accessible by all users. Published pipes can be cloned and used as subPipes. (For instruction on how to share and publish a pipe but keep its inner functioning private, see: <http://jalaj.net/2009/02/10/how-to-create-private-yahoo-pipes/>.)

To run the pipe, one can go to the pipe list page (Back to My Pipes link on top of the Canvas pane) and choose a pipe to run or click directly Run Pipe... link above the Back to My Pipes link (these links are located above the canvas area). This is also where one can publish (and unpublish) pipes.

### 5.2.2 Discovering and cloning pipes

Discovering relevant or interesting pipes is challenging, as there are tens of thousands of public pipes<sup>37</sup>. The browsing pipes section offers a keyword search with a twist to help finding relevant pipes. The keyword search can be refined by adding Formats (6 possible ones), Tags (98 possible ones), Sources (99 possible ones), and Modules (44 possible ones)<sup>38</sup>. Figure 21 shows a search made with one Tag (flickr) and one Modules (sort) added to augment the search.

<sup>37</sup> For instance, Fetch module had been used in 29,330 pipes by Feb. 20, 2009.

<sup>38</sup> The situation on Feb. 19, 2009.

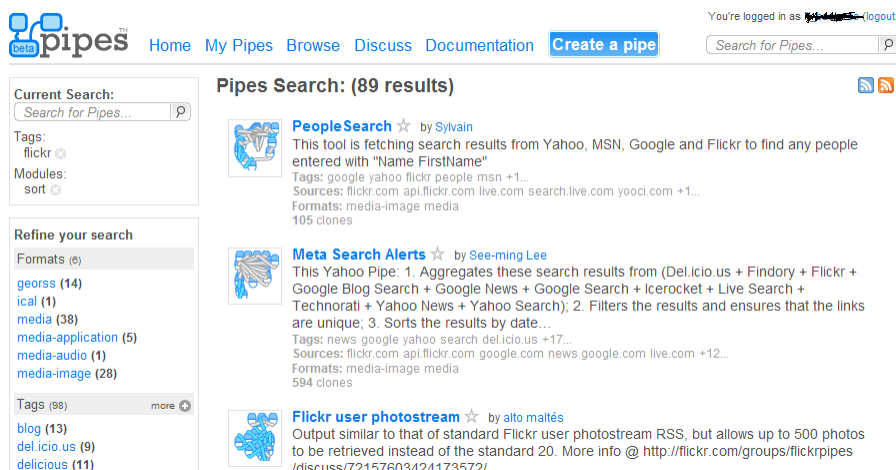


Figure 21 – A search specified by the tag "flickr" and module "sort"

On a pipe page, the pipe can be cloned or its source viewed. In addition, there is a lot of other information available about the pipe. Cloning means that the pipe is included in one’s My pipes list in the visual editor and can thus be used as a subPipe, and viewing source means that the pipe is opened in the visual editor as modules connections, not as source code.

### 5.2.3 The current state of Yahoo Pipes

While most technical pundits welcomed Yahoo Pipes with enthusiasm, there were also naysayers. While one may question if Yahoo Pipes really can turn “the web into a programmable environment for everyone”<sup>39</sup> because of the need to understand the underlying concepts and the UNIX vocabulary—even O’Reilly admits that “it’s not entirely for the faint of heart”<sup>40</sup>—Tim Anderson of IT Week<sup>41</sup> pointed to an even more fundamental issue with Yahoo Pipes. He argues that the core problem is that Yahoo Pipes does not “address the core problem of the service-oriented web: the business model. Participating in mashups works well for e-commerce sites like eBay or Amazon, because it drives sales, but that model fails for other kinds of services.”<sup>42</sup>

He sites as an example how Google “withdrew its Soap search API in favour of a JavaScript widget that users can embed on their sites. The likely reason is that Google gets no benefit from programmatic search access, whereas the widget gives Google full control of a little corner of the host web site.”<sup>43</sup>

In a way, Yahoo Pipes is in the same position as Yahoo’s Babel Fish which is a free service that offers no API because it is difficult to “monetise”<sup>44</sup>. Yahoo Pipes does not lend itself easily to monetization, either, and Anderson argues that “no business model means no long-term future.”<sup>45</sup>

Just like Popfly appears poised to go down as Microsoft reshuffles its organization<sup>46</sup> as the current recession continues, one can question the long-term viability of services that do not have business models for

<sup>39</sup> <http://radar.oreilly.com/archives/2007/02/pipes-and-filte.html>

<sup>40</sup> <http://radar.oreilly.com/archives/2007/02/pipes-and-filte.html>

<sup>41</sup> <http://www.itweek.co.uk/>

<sup>42</sup> <http://www.computing.co.uk/itweek/comment/2185589/yahoo-pipes-cracked>

<sup>43</sup> <http://www.computing.co.uk/itweek/comment/2185589/yahoo-pipes-cracked>

<sup>44</sup> Yahoo’s Kent Brewster according to <http://www.computing.co.uk/itweek/comment/2185589/yahoo-pipes-cracked>

<sup>45</sup> <http://www.computing.co.uk/itweek/comment/2185589/yahoo-pipes-cracked>

<sup>46</sup> <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9126718>

generating profits as Yahoo seems to be facing its own internal revamping<sup>47</sup>. Yahoo Inc. Chief Executive Carol Bartz, who replaced company co-founder Jerry Yang Jan. 13, 2009<sup>48</sup>, is dealing with a situation where Yahoo swung to a loss in the last quarter of the last year<sup>49</sup>. As the flagging economy is crippling the online advertisement sales, and Yahoo is struggling to keep up with Google in the online search market<sup>50</sup>, drastic measures might be expected.

It is both interesting and educational to contrast Popfly and Yahoo Pipes with iGoogle's gadgets. Popfly and Yahoo Pipes are easy-to-use tools that require next to no programming knowledge and are aimed at users, but neither has a real monetization model. iGoogle gadgets, on the other hand, require more knowledge and are aimed at both content providers and users but that have a clear business model behind them.

Last but perhaps not the least it is worth to mention that one strength and weakness of Yahoo Pipes is that it is at least currently a one-trick pony. One can make mashups with it and that is that. This is a weakness as it hinders monetization and does not allow wide areas of use, but it is also a strength in the sense that it becomes easy to learn and that it does that one thing very well indeed.

---

<sup>47</sup> <http://www.reuters.com/article/technologyNews/idUSTRE51K1FV20090222>

<sup>48</sup> [http://en.wikipedia.org/wiki/Carol\\_Bartz](http://en.wikipedia.org/wiki/Carol_Bartz)

<sup>49</sup> <http://www.marketwatch.com/news/story/Yahoo-CEO-plans-major-overhaul/story.aspx?guid={756D7C86-8B76-44EA-9D69-E65A09DF2735}>

<sup>50</sup> <http://www.marketwatch.com/news/story/Yahoo-CEO-plans-major-overhaul/story.aspx?guid={756D7C86-8B76-44EA-9D69-E65A09DF2735}>

## 6 Form-based creation

Visual programming requires a complex graphical editor to manage the structure of the program and the relationships between the components. The components are viewed from outside and connected with each other using a graphical notation. The user works from an operation-centric view and often needs to understand basic programming concepts although the notation may be easier to work with than with ordinary programming languages.

In GUI assembly tools a composite service is created from widgets—representing different web sources and services—that are laid out on a canvas. User works from a data-centric view and does not necessarily need to understand any programming. The user sees an explanation and example of what the widget will be like. He or she then simply fills in the form data and has a ready-made widget to include on the canvas.

In practice, there is little emphasis on the connections between components. Rather, the resulting service composition resembles a portal. The aggregated content is represented side-by-side, in a “salad-bar” style. This is in contrast to the “melting-pot” style of proper mashups where different components are rather put on top of each other—like locations on a map—or mixed with each other like tables whose data is gathered from several different sources.

However, the dashboard-style systems are hugely popular when compared to the proper mashup tools. Moreover, it is foreseeable that techniques to connect services on a canvas will be available in the future. For these reasons we will review two GUI assembly tools where the canvas is a web page: iGoogle and Netvibes.

### 6.1 iGoogle

*“You can easily make your own page in 30 seconds. You can add different tabs for the areas that you are interested in. For example, my iGoogle account holds tabs for my favourite blogs, politics, news headlines, cooking, photography, international publications and more! My homepage offers Map Quest, YouTube, weather updates, quotes of the day and a space for taking notes. I have RSS feeds for everything that I like to read and see on a normal basis, and it all lands in one location. The best part is decorating your iGoogle account. I feature different artists on every tab, and it really perks up my day.” - Amanda (February 3, 2009)*<sup>51</sup>

Launched in May 2005 and renamed in April 2007, iGoogle (formerly known as Google Personalized Homepage and Google IG) is an AJAX-based start-page that can be customized with themes and gadgets. In addition to using the available ones, users can make their own gadgets and themes as well. iGoogle is now available in 42 languages and over 70 country domain names<sup>52</sup>.

The iGoogle gadgets—typically simple HTML/XML and JavaScript mini-applications that can be embedded in web pages<sup>53</sup>—are similar to those on Google Desktop, and some, developed with Universal gadgets API, work in both<sup>54</sup> (at least with minor changes<sup>55</sup>). In addition to iGoogle and Google desktop, these gadgets can also run at least in Google Maps and Orkut in addition to it being possible to embed them in any website<sup>56</sup>.

While some are suggested as default when signing up (Figure 22), iGoogle start-page gadgets can be freely added and deleted.

---

<sup>51</sup> <http://agwired.com/2009/02/03/do-you-igoogle-i-do/>

<sup>52</sup> <http://en.wikipedia.org/wiki/Igoogle>

<sup>53</sup> <http://code.google.com/apis/gadgets/index.html>

<sup>54</sup> <http://www.google.com/webmasters/gadgets/guidelines.html>

<sup>55</sup> <http://code.google.com/intl/zh/apis/gadgets/>

<sup>56</sup> <http://code.google.com/intl/zh/apis/gadgets/>



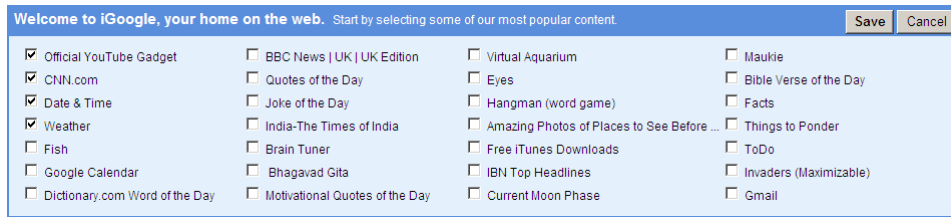


Figure 22 – Gadgets initially suggested by iGoogle

Gadgets can be re-organized on the user's iGoogle page by drag-and-drop. When the mouse cursor is on the top bar of the gadget, it changes to a four-arrow pointer to signify that the gadget can be moved. When moving the gadget, the interface gives feedback by showing where the gadget would be placed if dropped right now. In addition, the gadget becomes semitransparent and follows the rectangle (Figure 23). However, the drag-and-drop model does not work properly when trying to drag a gadget to an area that is not visible on screen.



Figure 23 – Drag and drop in process

The gadgets do not report on their state, which can be bewildering for users. For instance, when the Iltalehti feed gadget no longer was able to update the feed, it had no mechanism for telling the situation to its users. Furthermore, incompatible changes in the feeds can lead to undeletable gadgets, since the gadget's menu is not accessible in error state. The only way to get rid of it was from the canvas view of iGoogle.

As of July 2008, Google rolled in a left navigation bar that listed all the gadgets on the page. The new interface also enabled using tabs: As the number of gadgets one has grows, it makes sense to group them by themes or topics instead of having one mile-long page (Figure 24).

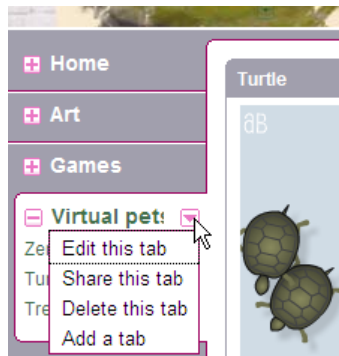


Figure 24 – iGoogle tabs

When adding a tab, iGoogle wants to suggest gadgets for it based on the name given to it with “I’m feeling lucky” feature familiar from the Google search engine (Figure 25). If the user does not unselect the “I’m feeling lucky” checkbox (which is selected by default), iGoogle simply opens the new tab with gadgets already in it instead of giving a list from which to select as when signing up. The gadgets on the tabs can also be dragged and dropped into other tabs.

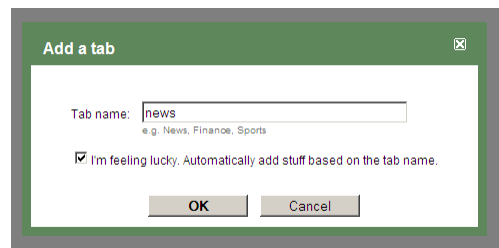


Figure 25 – iGoogle suggestions for gadgets based on the name of the tab

The list of gadgets on the tabs on the left navigation bar also makes it easy to move into full-screen view of a gadget, known as “canvas view”, a feature added in October, 2008<sup>57</sup>. Until then, iGoogle had been all about “quick, at-a-glance access to the information”<sup>58</sup>. Now much richer content is possible. Instead of only checking your Gmail inbox on iGoogle, now it became possible to use different features of Gmail directly on iGoogle. Likewise, watching videos and playing games is easier in canvas view. Mini-applications can be watched full-screen inside iGoogle, as can an entire website “with ads and all”<sup>59</sup>.

<sup>57</sup> <http://igoogledeveloper.blogspot.com/2008/10/big-canvas-big-opportunity.html>

<sup>58</sup> <http://igoogledeveloper.blogspot.com/2008/10/big-canvas-big-opportunity.html>

<sup>59</sup> <http://www.techcrunch.com/2008/10/16/igoogle-goes-wide-introduces-canvas-pages/>

In effect, iGoogle becomes more than a start page from which to spring somewhere else: Now users can stick around and “explore their personalized content without leaving iGoogle”<sup>60</sup>. The benefits to Google are clear. It is no surprise, consequently, that Google sees the canvas view as increasing the potential to use iGoogle as a “distribution mechanism”<sup>61</sup> that gadget providers can “monetize”<sup>62</sup>. Some content providers fight this iGoogle-centricity by having the clicked headlines open in their own websites while others have embraced the concept and use iGoogle as a new way both to distribute their content and advertisements (Figure 26).

As it is, advertisement is beginning to pop up inside the gadgets as the developers are looking to profit from their work. In addition to individual gadget developers placing Google’s AdSense ads in their gadgets, Marissa Mayer, Google’s VP of search products and user experience, says that “*some companies who already have established business models, [are] really using Google Gadgets almost like a form of an advertisement. It’s a distribution mechanism. If you’re the NYTimes or Netflix, you might not be a user’s homepage. But now you have a place on their homepage. Your brand is in front of that user every day... [Additionally] we’re starting to see real business models in other gadgets. The most successful gadgets actually have tens of millions of pageviews a month.*”<sup>63</sup>

On the negative side, the interface upheaval of adding the left navigation bar, canvas view and tabs, was not universally welcomed, as such headlines as *iGoogle personalises personal pages on other people's behalf*<sup>64</sup> and *iGoogle Users iRate About Portal's Changes*<sup>65</sup> indicate. Part of the PR debacle was caused by Google’s insensitive handling of the testing, with users assigned to test groups without an opt-out and without a warning or information how long the test would last<sup>66</sup>. The result of forcing changes on people is the appearance of such devices as a Firefox extension called iGoogle Tab Remover that, well, removes the iGoogle tabs from the left<sup>67</sup>.

While Google might be big enough to take this kind of criticism on the stride, the rest of us lesser mortal

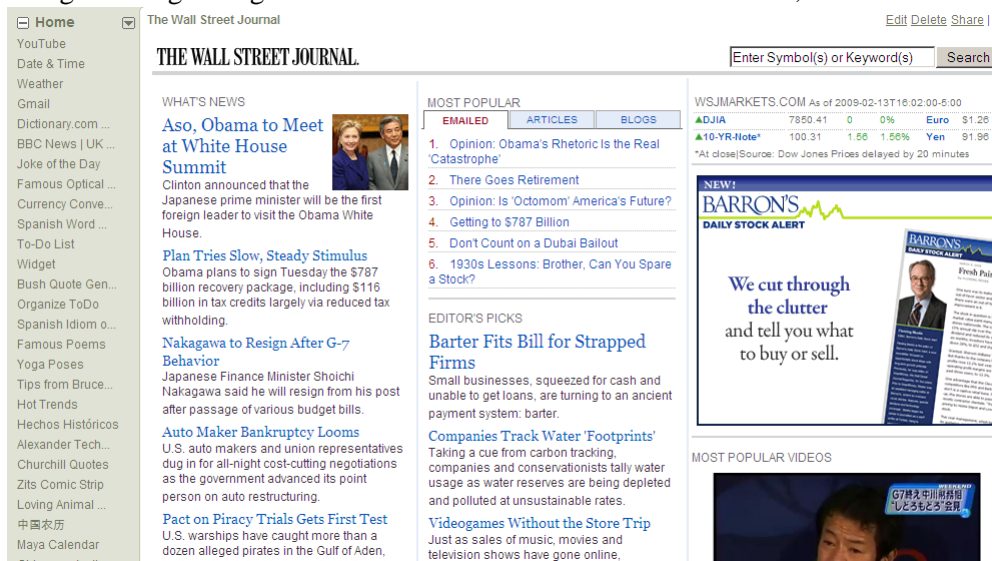


Figure 26 – iGoogle canvas view

<sup>60</sup> <http://www.techcrunch.com/2008/10/16/igoogle-goes-wide-introduces-canvas-pages/>

<sup>61</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>62</sup> <http://igoogledeveloper.blogspot.com/2008/10/big-canvas-big-opportunity.html>

<sup>63</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>64</sup> [http://www.theregister.co.uk/2008/09/01/google\\_personal\\_homepage\\_brouhaha/](http://www.theregister.co.uk/2008/09/01/google_personal_homepage_brouhaha/)

<sup>65</sup> <http://www.informationweek.com/news/internet/google/showArticle.jhtml?articleID=211201740>

<sup>66</sup> <http://en.wikipedia.org/wiki/Igoogle>

<sup>67</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/02/13/AR2009021302823.html>

may not be, as Kevin Rose, founder of Digg.com<sup>68</sup>, found out in the DVD code brouhaha when a user revolt led him to post the code in his blog after digg.com had tried to remove it from the site<sup>69</sup>. Even Facebook has been forced to bow to user demands<sup>70</sup>. The likely lesson here is that in making drastic changes, one needs to engage the users instead of dictating to them.

In the beginning of 2009, Google introduced a theme creator with which users can build their own themes<sup>71</sup>. While building themes was possible earlier on, too, using Google's API and then submitting the theme to the design gallery or with 3<sup>rd</sup> party tools, such as igThemer that let less technically savvy users create themes<sup>72</sup>, the new tool brings theme-building to everybody.

Themes are templates that determine how iGoogle looks like. The image on the top bar is an important part of a theme. Each tab can have its own theme. Themes can change according to the current time of the time-zone or weather. (One theme, Theme of the day, circulates different themes, which can lead to interesting situations, as one anonymous user comments: "...I was at a business conference the other day and to my surprise there were half naked women on my start page."<sup>73</sup>)

### 6.1.1 Discovering gadgets in iGoogle

Finding gadgets in iGoogle is strongly based on recommendations. When signing up, some suggestions are made (Figure 22). Also, when adding a tab, gadgets are added to the tab automatically unless the user unchecks the "I'm feeling lucky" feature (Figure 25). In addition, when a gadget is added, a link to recommendations based on the addition is displayed (Figure 27). However, the recommendations occasionally seem to have nothing to do with the added gadget or iGoogle is unable to make recommendations: "Sorry, there are no recommendations available for..." If all this is not enough, iGoogle also has the "You might also like..." link available in the gadget menu, in the extra space on the right if viewing a non-expanding gadget in canvas view mode, and the gadget page<sup>74</sup> also has "you might also like..." suggestions.

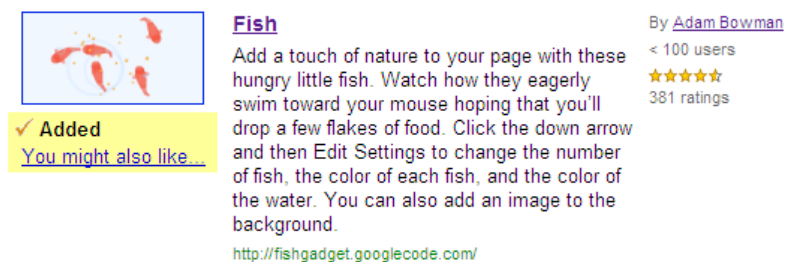


Figure 27 – Suggestions after adding a gadget

The options to recommendations for finding gadgets are searching with keywords and browsing gadgets by categories (All categories, News, Tools, Communication, Fun & Games, Finance, Sports, Lifestyle, Technology, and Politics) that can be sorted by Hottest, Most users, and Newest. With thousands of gadgets, finding things from the categories is not very efficient, thus the need for extensive use of recommendations. Furthermore, there are many very similar or practically identical gadgets going around, and the current interface does not really help in choosing between them.

<sup>68</sup> <http://www.digg.com>

<sup>69</sup> <http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2007/05/03/MNG4RPK18J1.DTL&hw=User+revolt&sn=001&sc=1000>

<sup>70</sup> <http://news.bbc.co.uk/1/hi/technology/7896309.stm?lss>

<sup>71</sup> [http://news.cnet.com/8301-17939\\_109-10143715-2.html](http://news.cnet.com/8301-17939_109-10143715-2.html)

<sup>72</sup> [http://news.cnet.com/8301-17939\\_109-10143715-2.html](http://news.cnet.com/8301-17939_109-10143715-2.html)

<sup>73</sup> <http://www.google.com/ig/directory?type=themes&url=skins/sampler.xml>

<sup>74</sup> The page that introduces the gadget in detail, such as <http://www.google.com/ig/directory?type=gadgets&url=www.artbible.info/art/artoftheday.xml>.

As will be discussed, iGoogle is currently placing greater emphasis on social side of things, and so there are ways to learn about potentially interesting gadgets through one's friends. The gadget menu has an item for sharing the gadget (Figure 28).

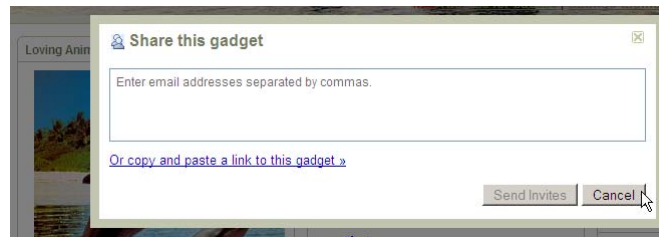


Figure 28 – Sharing a gadget

In addition to sharing individual gadgets, users can share whole tabs with any number of gadgets in them with the “Share this tab” option in the tab menu (Figure 24). The sender can choose which gadgets to include in the shared tab and whether or not to include his or her current setting. Of course, there are also numerous sources on gadgets on the Internet that give recommendations and tips about them<sup>75</sup>.

### 6.1.2 Building gadgets

There are two ways to create gadgets in iGoogle: The easy way with little choices and the hard way with plethora of options and imagination as the only limit. The easy way is to click the “Add stuff” link—there is no direct link to making gadgets on iGoogle start-page—and then click “Try now” in Create your own gadget area on the right. The next page offers ready-made wizards for making specific types of gadgets, and making the gadget is “as easy as filling out a simple form”<sup>76</sup> (Figure 29). No programming knowledge whatsoever is required. On the other hand, the gadgets made this way offer no feed connectivity or mashup possibilities. The seven types available are introduced below (the texts are directly from <http://www.google.com/ig/gmchoices>):

- *Framed Photo*: You love your pics. So will your loved ones. Share a series of your favorite photos with friends and family.
- *GoogleGram*: Grace that someone special's home page with a gift of flowers or candy whose messages change every day.
- *Daily Me*: What are you doing? What's on your mind? Share your day's quotes, links, and ideas with everyone in your life (Figure 29).
- *Free Form*: Channel and display your creativity with this all-purpose gadget that lets you meld text and image in any way.
- *YouTube Channel*: Are you the one who always finds the good stuff on YouTube? Create a video channel for your friends.
- *Personal List*: With this gadget you can publish your own personal "Top Ten" or simply send a set of chores to your sweetie.
- *Countdown*: Share a countdown to an upcoming holiday, a friend's party or your wedding anniversary.

---

<sup>75</sup> For instance, <http://googlegadgetblog.com/>

<sup>76</sup> <http://www.google.com/ig/gmchoices>

## Daily Me

Tell your friends and family what's on your mind. When you're happy with how it looks, press "Create Gadget."

The screenshot shows the 'Daily Me' gadget creation wizard. On the left, a preview of the gadget is shown with the title 'The Daily Jess' and a photo of a woman. Below the photo are several status update fields: 'Current Status...' (About to go for a run), 'Lately I'm thinking...' (What shall I do for my birthday?), and 'Currently reading...'. On the right, the configuration options are: Title (The Daily Jess), Photo (http://www.google.com/ig/modules/gadgetmal), 'Things about me' section with dropdown menus for 'Current Status...', 'Lately I'm thinking...', 'Currently reading...', and 'This is really interesting...', and a 'The Soapbox' text area. At the bottom, there is a 'Background' dropdown set to 'Blue' and two buttons: 'Preview Changes' and 'Create Gadget'.

Figure 29 – Wizard for Daily Me

The hard way is to make gadgets using the Google Gadget API with HTML, XML, and JavaScript. Also, Flash and Silverlight content can be embedded in gadgets. iGoogle gadget tutorial claims that all that is needed to build gadgets is “a basic understanding of HTML” as the tutorials will teach one all the XML one needs—although for “more sophisticated gadgets,” learning some JavaScript becomes necessary<sup>77</sup>. The extensive documentation available online through iGoogle naturally includes the Gadget API reference for more advanced practitioners: <http://code.google.com/intl/en/apis/gadgets/docs/reference/>.

Besides the core gadget API, Google has the following gadgets API extensions<sup>78</sup>:

- *OpenSocial*
  - Utilize functions for creating social gadgets.
  - Available for gadgets on containers that support the OpenSocial API, such as Orkut and the iGoogle sandbox.
- *Maps*
  - Add geo-targeted functionality by overlaying your data on maps.
  - Available for gadgets on Google Maps (Mapplets).
- *Finance API*
  - Retrieve and display stock information from Google Finance.
  - Available for gadgets on iGoogle.
- *Ads*
  - Create gadgets that serve as dynamic, rich-media advertising.
  - Available for gadgets running via Google AdSense.
- *Calendar*
  - Build gadgets than run within a calendar and display time-based information.
  - Available for gadgets on Google Calendar.
- *Spreadsheets*
  - Build gadgets that interact with the content of a spreadsheet.
  - Available within a spreadsheet, or on another web page that supports gadgets, such as iGoogle.

<sup>77</sup> <http://code.google.com/intl/en/apis/gadgets/docs/legacy/gs.html>

<sup>78</sup> <http://code.google.com/intl/en/apis/gadgets/docs/overview.html>

Thus, the level of desired sophistication determines the level of programming requirements. Building Flash or Silverlight applications is, of course, another world.

While gadget files can be written with any text editor, iGoogle also has Google Gadgets Editor (GGE) (Figure 30) that allows the developer to preview the gadget and that provides a selection of example gadgets with which to start creating and that can be added to one's iGoogle page—although the examples are not included in the gadget version of GGE. GGE is used to write legacy gadgets, that is, environments that do not yet support the OpenSocial API<sup>79</sup>. For developing gadgets with the OpenSocial API, Google is preparing iGoogle developer sandbox that is currently only open to developers.



Figure 30 – Google Gadget Editor

Gadgets are public by default<sup>80</sup>. While making them private is possible, it requires extra effort and how to do it is not highlighted in the tutorials.

### 6.1.3 Popular gadget types in iGoogle

In order to understand what gadgets have become popular among iGoogle users, we went through all the available 4948 gadgets, distributed on 707 pages (between Feb. 16 and Feb. 13, 2009). Seventy gadgets had over 100,000 users.

When these gadgets are grouped by their categories (Table 1), we notice that category News totally dominate the scene. In fact, 49 of the 130 gadgets—more than one third—fell into the news category. Another major category was finance-related gadgets, 10 out of 130. These two categories overlapped by three financial news gadgets. With news and finance-related gadgets, as with many others, users were reduced to passive consumers of information. In fact, only 20 out of 130 gadgets were about enabling users to do something (Table 2).

Very few of the gadgets were mashups of any kind; most of them simply showed feed information from one source. In fact, all but about 26 gadgets were made by a content provider itself<sup>81</sup>. The 26 gadgets were

<sup>79</sup> [http://code.google.com/intl/en/apis/gadgets/docs/dev\\_guide.html](http://code.google.com/intl/en/apis/gadgets/docs/dev_guide.html)

<sup>80</sup> <http://code.google.com/intl/en/apis/gadgets/docs/basic.html>

developed by 13 developers, out of which 5 are likely to be normal users. Others are by companies or people clearly affiliated with a company that either builds gadgets or provides content.

Table 1 – Left: Popular gadgets grouped by categories. Right: News gadgets by categories.

News	35
Weather	7
Dictionary/translating	6
TV program	5
Sports news	5
Jokes	5
Virtual pet	4
Stock market	3
Financial/business news	3
Blogs	3
Quotes	3
Games	3

News	35
Sports news	5
Financial/business news	3
Celebrity news	2
Movie news	1
Music news	1
Weird news	1
PC and console news	1

Table 2 – Gadgets where user is not a passive consumer of information.

Dictionary/translating	6
Games	3
Telephone directory services	2
Currency converters	2
Yellow pages	1
Calendars	1
Wikipedia search	1
Craigslist search	1
To-do list (Arabic)	1
On-screen keyboard (Arabic)	1
Calculator	1

In summary, the popular gadgets mainly get their content from a single source and are typically developed by the company that provides the content. Majority of the other developers are companies or people from companies that are directed related to content providing or ICT. Google itself was behind 16 of the 130 gadgets (over 10%).

If the iGoogle gadgets give any indication, getting the content providers involved or at least making it easy to create gadgets that can use feeds provided by the major content providers is essential for success. On the other hand, is iGoogle really about enabling users to create gadgets or about building a new way to distribute content? Marissa Mayer, Google’s VP of search products and user experience, speaks of iGoogle as a “distribution mechanism”<sup>82</sup>, and when evaluating the lessons learned from iGoogle, this has to be born in mind.

<sup>81</sup> Game and virtual pet makers were considered to represent their own content. Furthermore, this number is more an educated guess than exact fact, as inducing the situation in some cases was not simple. However, it is accurate enough to allow us to see the trend.

<sup>82</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>



## 6.1.4 Creating themes

iGoogle home page advertises Themes for causes and Artist themes (Feb. 18, 2009) but other themes must be found from behind the “Add stuff” link. In addition, like gadgets, themes can be recommended from the theme page to other users. The following page has a tab for themes and on the right a section called Create your own theme. The tool for creating themes is similar to the tool for creating gadgets with a form (Figure 31). It lets you download photos, crop them to the size and determine colors with ready-made palettes. The ready themes are then published and shared. A similar tool is expected to hit Gmail this year<sup>83</sup>.

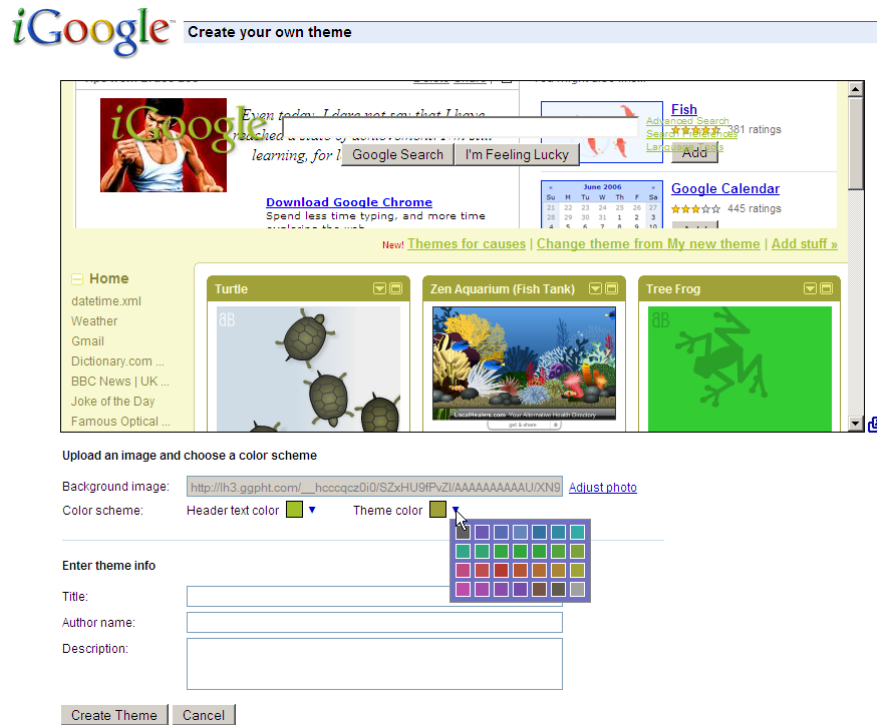


Figure 31 – Creating a theme

However, it is possible to create themes with more freedom by defining the theme in an XML file that contains key-value pairs for the attributes, such as background and text color<sup>84</sup>. Using it requires signing up for the sandbox that is currently only open to developers. While a theme can be simply a color setting for the header background and iGoogle logo, it can also include multiple images, dynamic behavior, and be localized for different countries and languages.

A theme has one or more skins, quintessentially a collection of header image, logo, and color settings. To be dynamic, a theme needs to have more than one skin that then can change according to the time of the day. A theme can be localized by including localized metadata fields for particular languages and countries.

## 6.1.5 Social future

Even if social networking has not been the strong area for Google thus far, Orkut notwithstanding, this is poised to change<sup>85</sup> as iGoogle is getting social through its own data portability system, Friend Connect<sup>86</sup> (Figure 32). Social features are beginning to pop up in iGoogle gadgets now that developers have OpenSocial API available. For example, “the My Google Book Search Library gadget, which lets you search

<sup>83</sup> [http://news.cnet.com/8301-17939\\_109-10143715-2.html](http://news.cnet.com/8301-17939_109-10143715-2.html)

<sup>84</sup> [http://code.google.com/intl/en/apis/themes/docs/dev\\_guide.html](http://code.google.com/intl/en/apis/themes/docs/dev_guide.html)

<sup>85</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>86</sup> <http://www.techcrunch.com/2008/05/12/google-confirms-friend-connect/>

books and create a personal digital library, now asks people who have OpenSocial profiles if they want to allow the gadget to ‘know who I am and access my profile’ and ‘post updates to my Friends group.’”<sup>87</sup>

With canvas views, iGoogle appears poised to be “taking on more of the trappings of a social network”<sup>88</sup>. All that is missing is a buddy list and activity feed<sup>89</sup>, but these could be available through Gmail contacts—“we know your contacts” says Mayer, Google's VP of search products and user experience—and possible voyeuristic feeds<sup>90</sup>, such as Google Latitude, which is already now available as a gadget iGoogle. Another sign that shows the coming of the social networking is that iGoogle is about get a chat feature like the one in Gmail<sup>91</sup>.

As to the social future of iGoogle, Mayer stated Jan. 9, 2009, that “*It certainly is conceivable as we introduce chat and we introduce open social and make iGoogle an open-social container, that means developers would have access to some of your friend information.*”<sup>92</sup>



Figure 32 – Friend Connect

Social networks appear poised to be at least one of the—if not the—foundation of the future advertisement online. They give both better financial rewards and elicit bigger conversion rates than traditional banner advertisement<sup>93</sup>. For example, SocialMedia, a social advertisement company, claims that “people are 200 times more likely to respond to the social ad.”<sup>94</sup> Facebook will face growing competition from Google and others in the fight over the advertising dollars in the tightening economy<sup>95</sup>.

In addition, tracking users through such platform as iGoogle is more efficient than through IP numbers or cookies because a computer may be used by many people but an iGoogle account is typically private. This leads to being able to profile users better. The better profiles translate into better search results and better targeting of advertisements.

As an example of profiles improving search results, Mayer sees knowing enough about users translating into not needing to type long queries: “when you type things into the search box it basically augments your

<sup>87</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/01/19/AR2009011901072.html>

<sup>88</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/01/19/AR2009011901072.html>

<sup>89</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/01/19/AR2009011901072.html>

<sup>90</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>91</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>92</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>93</sup> [http://news.cnet.com/8301-10784\\_3-9974220-7.html](http://news.cnet.com/8301-10784_3-9974220-7.html) and [http://www.forbes.com/2007/11/06/facebook-ad-platform-tech-internet-cx\\_wt\\_bu\\_1106techfacebook.html](http://www.forbes.com/2007/11/06/facebook-ad-platform-tech-internet-cx_wt_bu_1106techfacebook.html)

<sup>94</sup> [http://news.cnet.com/8301-10784\\_3-9974220-7.html](http://news.cnet.com/8301-10784_3-9974220-7.html)

<sup>95</sup> [http://money.cnn.com/2009/02/16/technology/hempel\\_facebook.fortune/](http://money.cnn.com/2009/02/16/technology/hempel_facebook.fortune/)

query,' she said. 'If I type "broadway shows" into Google it would actually come back and understand that I like musicals and like peppy to sad...on the whole that's what we're really shooting for.'"<sup>96</sup>

As mentioned earlier, although iGoogle ostensibly offers ways for users to build and share gadgets, Google sees it more as a "distribution mechanism"<sup>97</sup> that allows monetization<sup>98</sup>. Besides bigger content providers, such as The Wall Street Journal and NYTimes, using iGoogle to generate advertising revenue, smaller and individual gadget developers are increasingly adding AdSense ads to the gadgets. As "[t]he most successful gadgets actually have tens of millions of pageviews a month"<sup>99</sup>, it is no wonder that companies are actively involved in developing gadgets. It is a win-win situation, as both Google and gadget makers and content providers can profit and the users get better-targeted content.

Or perhaps there are also some losers. For example, Cusimano.Com Corporation makes gadgets displaying works of Ansel Adams, Salvador Dali, and Andy Warhol, apparently without paying for any IP rights. Gadgets can use feeds from different sources, and the source provider might not profit from this kind of proliferation financially. Should the gadget maker get the money or what is the share of the content provider? Interesting IPR questions are waiting to be answered. Another loser might be our right to privacy—if we still have such thing in the first place<sup>100</sup>.

Be that as it may, iGoogle is coming social, perhaps even a social networking platform of kinds, and it is becoming more and more a part of the online advertisement push that emphasizes profiling and social advertising. User developed gadgets and themes are just part of the trappings.

## 6.2 Netvibes

*"Netvibes was founded in September 2005 ... as a way for people to take control of their daily digital lives. The problem he [Tariq Krim, Netvibes founder] was trying to solve was 'How can I have everything that matters to me on the Internet -- such as my RSS feeds, the weather, my social networking, competitive sports, my start page -- all in one place in a digital dashboard?' This idea evolved into personalized pages for individuals and personalized pages for brands and publishers."* - Freddy Mini, Netvibes CEO<sup>101</sup>

Netvibes<sup>102</sup> is a personalized start page made with AJAX, akin to iGoogle, My Yahoo, and Pageflakes. Due to the similarity with iGoogle, we discuss it by contrasting it with iGoogle.

The Netvibes Ecosystem consists of the AJAX-based Netvibes framework and user-submitted widgets providing feeds, podcasts, and *universes* that are publicly viewable customized pages of another entity<sup>103</sup>. As in iGoogle, users can create tabs to organize their widgets, and individual widgets can open to take up all the canvas area (Figure 33). The tabs show with a number in parentheses how many unviewed content items—for instance, news items—there are in each tab.

---

<sup>96</sup> <http://www.zdnet.com.au/news/software/soa/Welcome-to-iGoogle/0,130061733,339275160,00.htm>

<sup>97</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>98</sup> <http://igoogledeveloper.blogspot.com/2008/10/big-canvas-big-opportunity.html>

<sup>99</sup> <http://vator.tv/news/show/2009-01-08-igoogle-to-add-chat-and-become-more-social>

<sup>100</sup> "You have zero privacy anyway," Scott McNealy, the chief executive officer of Sun Microsystems, told a group of reporters and analysts: "Get over it." (<http://www.wired.com/politics/law/news/1999/01/17538>)

<sup>101</sup> According to: <http://www.dailyfinance.com/2009/02/25/growth-matters-netvibes-makes-europe-more-social/>

<sup>102</sup> <http://www.netvibes.com/>

<sup>103</sup> <http://en.wikipedia.org/wiki/Netvibes>

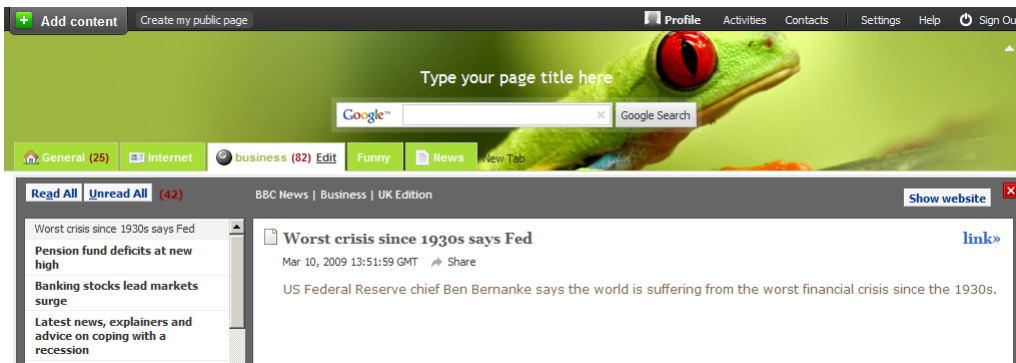


Figure 33 – Netvibes tabs and canvas

When *Add content* is expanded on top of the page, widgets can be dragged and dropped on the tabs (Figure 34). The widgets can be searched for by a keyword or browsed by category (with Essential widgets category separated from others). In addition, the user can create a feed with the feed address. If a user is in habit of adding feeds, there is also a Firefox extension<sup>104</sup> for adding feeds to Netvibes, an example of how different ecosystems use each other.



Figure 34 – Add content pane in Netvibes

Organizing widgets in tabs also works by dragging and dropping. The visual cues to the movement are very similar to those used in iGoogle (Figure 35).

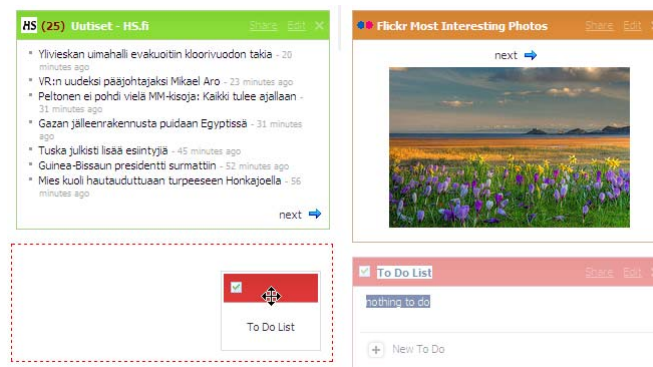


Figure 35 – Drag and drop in progress in Netvibes

Netvibes allows adding content by location. Thus, the widgets available in *Add content* change according to the location that the user sets. The number of items changes greatly when the location changes (Table 3). As

<sup>104</sup> <http://eco.netvibes.com/tools/firefox>

the categories in Table 3 show, Netvibes is a much more news-based system than iGoogle, and works easily as a brand-monitoring dashboard for companies<sup>105</sup>. Even the widgets behind Fun & games contain a lot of topic-related article feeds and only few actual games. This contrasts strongly with iGoogle’s virtual pets and large number of games.

Table 3 - Number of widgets based on location and category.

Browse categories	Location Finland	Location UK
Featured widgets	6	34
News	12	239
Business	1	78
Sport	6	112
TV, movies & music	24	160
Tools & technology	10	172
Fun & games	3	30
Lifestyle	4	85
Shopping	0	20
Travel	1	39

Essential widgets (42 for all locations), on the other hand, focus on doing things with widgets for calendar, email viewers, Facebook, weather, and searches (maps, videos, shopping etc.).

Finally, *Add content* area offers a wizard to personalizing the view. First, the system wants to know one’s location and then interest (Figure 36), the choices of which are the same as the categories in Browse categories (except for Featured widgets which is missing). The wizard then recommends the most popular widget for each category that one selects. Otherwise, recommenders do not play a major role in Netvibes.

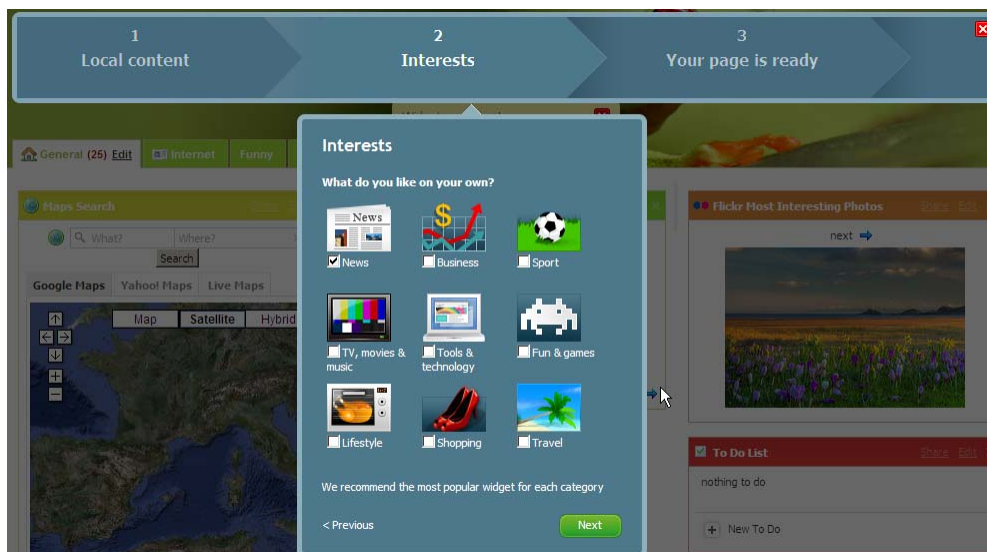


Figure 36 – Netvibes add content wizard

Users can share their tabs in Netvibes just like in iGoogle. In addition, Netvibes offers other ways both to track one’s own and one’s friends’ activities (Figure 37) and to subscribe to other users’ universes. The system is reciprocal, as one can see who is reading one’s “stuff”.

<sup>105</sup> <http://vator.tv/news/show/2009-01-29-netvibes-the-new-social-media-dashboard>

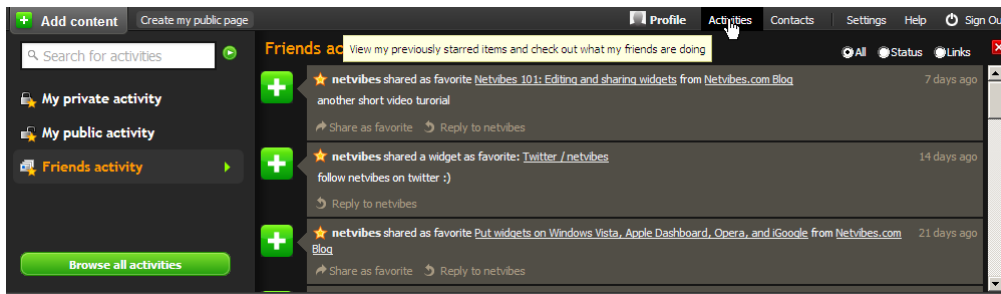


Figure 37 – Tracking activities

In addition to widgets, one can select themes for the start page. Unlike in iGoogle, however, the selected theme is common to all tabs. In addition to selecting widgets directly from one’s Netvibes start page, Netvibes has an ecosystem<sup>106</sup> where one can browse and search for all the widgets available. At the time of writing (March 3, 2009), there were 178,738 widgets and 944 universes available. Table 4 lists the 10 most popular widgets in Netvibes.

Table 4 - Popular Netvibes widget.

Name	Installs	Stars	Description
Weather	26,972,434	4	Weather forecasts for a city of choice
Flickr	25,513,492	-	Flickr widget – tagged pictures
Websearch	23,109,768	-	Integrated searches in major search engines
ToDoList	22,300,532	3	To-do list with add/remove and dragging
Mail	8,843,632	5	Assistant to create an email widget, with notification
PostIt	8,088,119	5	Sticky notes
Flash	5,931,503	-	Flash widget – animations, etc.
Facebook	5,739,027	4	Facebook widget - check the profile and status updates
Google News	5,038,945	3	Google News widget
Ebay	3,109,480	2	Ebay widget

Users, developers and publishers—as Netvibes puts it<sup>107</sup>—can create widgets to Netvibes using Netvibes Universal Widget API (UWA) which is a widget framework that uses XHTML for structure, CSS for styling and JavaScript/Ajax for behavioral/DOM control. In addition, the framework supports iframes and plugins, such as Flash. UWA furthermore subscribes to “write once, run everywhere” ideology, and the widgets developed with it can run in various widget platforms, including among others iGoogle, Windows Vista Sidebar, Apple Dashboard, Live.com, iPhone, Opera, blogs, and MySpace.

In summary, Netvibes is like iGoogle’s cousin and not far removed. In many ways, the two services mirror each other, one getting it slightly better in one aspect, another getting it slightly better in another. Although Google is internationally clearly stronger, Netvibes also has a significant market share in Europe<sup>108</sup>. Perhaps more significantly, as Vincent Chang, Netvibes’ spokesperson, says: “*Netvibes also has the Web’s largest collection of universally compatible widgets, including official news and video content from more than 1,000 of the world’s leading brands and media companies. We measure our success by our share of leading brands -- we have 1,000 advertisers including CBS to the New York Times, Nissan, Nokia, and Ogilvy -- and by that measure we’re doing well.*”<sup>109</sup>

Freddy Mini, Netvibes CEO, sees widgets as a “significant advertising market opportunity” because they are not based on the user visiting a site but “maintain engagement with the audience after the audience leaves the

<sup>106</sup> <http://eco.netvibes.com/>

<sup>107</sup> <http://dev.netvibes.com/>

<sup>108</sup> <http://www.dailyfinance.com/2009/02/25/growth-matters-netvibes-makes-europe-more-social/>

<sup>109</sup> <http://www.dailyfinance.com/2009/02/25/growth-matters-netvibes-makes-europe-more-social/>

site”<sup>110</sup>. Mini estimates that the market for widgets is going to hit \$5 billion if widgets prove to provide brand name advertisers greater revenue for spent dollars.<sup>111</sup>

### 6.3 Yahoo! Widgets

*“Desktop Widgets blur the line between the web and the desktop by pulling the content out of the browser and integrating it into your desktop.”* – Yahoo Widgets blog<sup>112</sup>

Arguably, Konfabulator was the platform—the widget engine—that brought widgets to foreground and on the desktop in the first place<sup>113</sup>. Originally released as Mac-only platform to support widgets on the desktop in Feb. 2003 that cost US\$24.95, Konfabulator was sold to Yahoo in 2005 and became freeware<sup>114</sup>. Just prior to this, Apple released its widget engine, Dashboard that bears many similarities to Konfabulator. Consequently, Apple has faced accusations of essentially having stolen the ideas behind Konfabulator. Apple argue that the whole idea actually originated with Apple—where Arlo Rose whose brainchild<sup>115</sup> Konfabulator had worked—and that dashboard is, in fact, different and better.<sup>116</sup>

Dashboard is Yahoo Widgets’s biggest competitor on the Mac desktop while, prior to Microsoft Vista and its support of gadgets with Windows Sidebar<sup>117</sup>, it faced competition on the Windows desktop from DesktopX (shareware), Kapsules (freeware), and AveDesk (freeware)<sup>118</sup>. In Linux, there is gDesklets that supports widgets (called applets in this context) on the Gnome desktop<sup>119</sup>. However, perhaps the most serious and the most similar<sup>120</sup> competitor to Yahoo Widgets is Google Desktop that—like Yahoo Widgets—is multi-OS platform for bringing widgets on the desktop. Google desktop supports Mac, Windows, and—unlike Yahoo Widgets—Linux, and many gadgets available in iGoogle also work on it.

Having Yahoo Widgets on the desktop requires installing Yahoo Widget Engine, quintessentially “a JavaScript runtime environment combined with an XML interpreter”<sup>121</sup>. Yahoo Widget Engine 4.0 included a feature called the dock (Figure 38) that shows all currently open widgets, and allows seeing all widgets inside the user’s “My Widgets” folder. Mouse-overing a widget icon on the dock allows closing it and seeing its preferences. Also, the Dock allows revealing all the widgets in the Heads-up Display<sup>122</sup>: All the other applications are dimmed the widgets are brought to the foreground. The dock also has an auto-hide option.<sup>123</sup>

---

<sup>110</sup> According to: <http://www.dailyfinance.com/2009/02/25/growth-matters-netvibes-makes-europe-more-social/>

<sup>111</sup> <http://www.dailyfinance.com/2009/02/25/growth-matters-netvibes-makes-europe-more-social/>

<sup>112</sup> <http://widgets.yahoo.net/blog/?p=16>

<sup>113</sup> While many claim the fame for coming up with the concept, here is one early story from 1984 about “desk Accessories” that might have pretty good case: [http://folklore.org/StoryView.py?project=Macintosh&story=Desk\\_Ornaments.txt](http://folklore.org/StoryView.py?project=Macintosh&story=Desk_Ornaments.txt). The term widget, on the other hand, was first applied to user interface elements during Project Athena in 1988 ([http://en.wikipedia.org/wiki/GUI\\_widget](http://en.wikipedia.org/wiki/GUI_widget)).

<sup>114</sup> [http://en.wikipedia.org/wiki/Yahoo\\_Widgets](http://en.wikipedia.org/wiki/Yahoo_Widgets)

<sup>115</sup> According to Rose, the idea came to him when he saw skinnable mp3 player, and thought that it would be a cool thing for the desktop, too (<http://www.konfabulator.com/cartoon/partOne.html> and <http://gigaom.com/2006/11/06/widgets-keynote-by-konfabulators-arlo-rose/>).

<sup>116</sup> For example, <http://www.macworld.com/article/35200/2004/06/konfabulator.html>, [http://www.macsimumnews.com/index.php/archive/widget\\_wars\\_thoughts\\_on\\_the\\_konfabulator\\_dashboard\\_brouhaha](http://www.macsimumnews.com/index.php/archive/widget_wars_thoughts_on_the_konfabulator_dashboard_brouhaha), and [http://daringfireball.net/2004/06/dashboard\\_vs\\_konfabulator](http://daringfireball.net/2004/06/dashboard_vs_konfabulator)

<sup>117</sup> <http://www.microsoft.com/windows/windows-vista/features/sidebar-gadgets.aspx>

<sup>118</sup> [http://en.wikipedia.org/wiki/Yahoo\\_Widgets](http://en.wikipedia.org/wiki/Yahoo_Widgets)

<sup>119</sup> <http://en.wikipedia.org/wiki/GDeskletss>

<sup>120</sup> <http://widgets.yahoo.net/blog/?p=16>

<sup>121</sup> [http://en.wikipedia.org/wiki/Yahoo\\_Widgets](http://en.wikipedia.org/wiki/Yahoo_Widgets)

<sup>122</sup> Originally called Konsposé and renamed by Yahoo ([http://en.wikipedia.org/wiki/Yahoo\\_Widgets](http://en.wikipedia.org/wiki/Yahoo_Widgets))

<sup>123</sup> [http://en.wikipedia.org/wiki/Yahoo\\_Widgets](http://en.wikipedia.org/wiki/Yahoo_Widgets)



Figure 38 – Yahoo Widgets Dock

Having their own rendering engine allows desktop widgets to be graphically fancier than their web cousins. Living outside of the browser also allows access to local resources, potential offline use and background downloading in addition to greater interaction with the OS through desktop interaction<sup>124</sup>.

Widgets for Yahoo Widgets are available on the Yahoo Widgets website<sup>125</sup>. The same site has support for developers who wish to build widgets. Installing the Yahoo Widgets Engine brings in 14 bundled widgets with it. Currently there are 5,391 widgets available to Yahoo Widgets on the site divided into 20 categories: utilities (702), news (574), games (501), radio (477), search (426), clocks (326), webcams (289), countdown (223), weather (161), communication (135), and so on. Table 5 lists the top ten most popular widgets by the number of downloads.

Table 5 – Most popular Yahoo Widgets (yahoo.com, 2/2009).

Name	Downloads	By	Description
Yahoo Weather	4,834,555	Yahoo	Weather report
Picture Frame	3,558,868	Yahoo	Flickr/Yahoo Photos upload and mgmt
Analog Clock	3,006,650	Yahoo	Analog Clock with styles and colours
Yahoo Finance	2,124,207	Yahoo	Information on indices, stocks and funds
Digital Clock	1,146,361	Yahoo	A Digital Clock with alarm and styles
Yahoo Widget Gallery	1,088,571	Yahoo	Search widgets from Yahoo Widget Gallery
Day Planner	1,069,699	Yahoo	Get calendar events/tasks, many sources
Yahoo Mail Checker	1,038,406	Yahoo	Unseen mail count for Yahoo Mail address
Sys Monitor	789,080	A.Kreisl	System monitor for local system
Battery	786,024	Yahoo	Battery status

Although category *News* plays an important role with 574 widgets, it is interesting to notice that *Utilities* have gained the top position. Eight out of ten most popular utilities widgets reveal information about the system, state of CPU, battery, or file system. Only two of them are not related to the system, namely My Coke Rewards Widget<sup>126</sup> and JC Sticky Deluxe<sup>127</sup>.

The prerequisites for writing Yahoo Widgets shows an already familiar list of languages: JavaScript, HTML, Flash, and XML (JavaScript and XML being the core with HTML and Flash added in Konfabulator 4.5) in addition to being able to use some image editor, such as Photoshop, Paint Shop Pro, or Gimp<sup>128</sup>. The development environment now also offers a rudimentary debugger<sup>129</sup>. AppleScript can also be used, but then the widget is Mac-only, as can shell scripts and COM applications, resulting in a Windows specific widget. However, most widgets, made with JavaScript and XML, work for both Mac and Windows. Yahoo Widget Engine 4.0 included SQLite, allowing developers to create and modify databases. The same version included

<sup>124</sup> <http://widgets.yahoo.net/blog/?p=16>

<sup>125</sup> <http://widgets.yahoo.com/>

<sup>126</sup> “The My Coke Rewards Widget lets you add cap codes to your account directly from your desktop and receive the latest news on great rewards at MyCokeRewards.com” (<http://widgets.yahoo.com/tags/utilities/?s=downloads&p=>)

<sup>127</sup> “Sticky Deluxe is a cross-platform sticky note / memo taker. It is an electronic version of the venerable 3M Post-It ® note.” (<http://widgets.yahoo.com/tags/utilities/?s=downloads&p=>)

<sup>128</sup> [http://widgets.yahoo.com/static/downloads/Widget\\_Creation\\_Tutorial\\_1.1.zip](http://widgets.yahoo.com/static/downloads/Widget_Creation_Tutorial_1.1.zip)

<sup>129</sup> <http://widgets.yahoo.net/blog/>: Konfabulator 4.5: Cool Developer Features (Dec. 14, 2007) and Yahoo Widgets 4.5 is Here! (Nov. 29, 2007).



support for Canvas that “allows developers to create Widgets without a single image or create simple drawing programs”<sup>130</sup>.

With Yahoo facing financial problems and likely re-organizations and Yahoo Widgets being battered left and right by Dashboard in Mac, Vista Sidebar on Windows, and Google desktop on both, the future of Yahoo Widgets is shrouded in mystery. Then again, Yahoo does also have a start page with widgets, My Yahoo which is closely related to their search engine, and as a content provider, Yahoo does need outlets for the content, so perhaps Yahoo Widgets is better insured against the future than, say, Yahoo Pipes at the end of the day.

---

<sup>130</sup> [http://en.wikipedia.org/wiki/Yahoo\\_Widgets](http://en.wikipedia.org/wiki/Yahoo_Widgets)

## 7 Script based creation

All the previous techniques discussed are good tools for end-user development if the users lack programming skills. But if the user wants to customize the developed application more and is capable of doing basic programming, script based solutions come into the picture. Bolin et al. (2005) introduced a programming system embedded in the Firefox web browser that enables end-users to automate, customize, and integrate web applications without examining their source code. Screenshot of the system, Chickenfoot, can be seen in Figure 39.



Figure 39 – Chickenfoot user interface.

Chickenfoot uses a high-level pattern language based on keyword matching that enables the user to identify pages without knowledge of the page's HTML structure or XPath. In a user study Bolin et al. (2005) found that the keyword matching used in Chickenfoot correspond closely to the names users actually generate for page and it was useful for generating complex page structures.

Next, we will turn to Firefox add-ons that can be created by the end-user.

### 7.1 Firefox extensions

*“One advantage the open source Firefox browser has over Internet Explorer is the wealth of add-ons that are available to users. These little extras can customize the browsing experience.”* - Steven M. Cohen (Information Today)<sup>131</sup>

Firefox was born out of Mozilla getting bloated with all kinds of features. Dave Hyatt and Blake Ross began to design a stand-alone browser without the extra baggage forced on everybody<sup>132</sup>. The idea was to make a

<sup>131</sup> <http://www.linuxinsider.com/story/Firefox-Add-Ons-Addictive-Browsing-Enhancers-66015.html?wlc=1235390349&wlc=1235473333&wlc=1235571498>

browser that is lean and mean but that can easily be extended with add-ons. In effect, each user could have the features that they wanted and needed instead of a monster that would have had forced everything on every single user. Firefox, originally called Phoenix, replaced together with Thunderbird the bloated Mozilla Suite as the focus of the Mozilla Organization in 2003, and after some name-related hassles, the new browser was named Mozilla Firefox on Feb. 9, 2004<sup>133</sup>. Since then, Firefox has re-challenged Internet Explorer and has gained a market share of over 20%<sup>134</sup> (give or take depending on who is counting).

As mentioned, allowing add-ons to complete the browsing experience was very much part of the concept behind Firefox. Its success has had Internet Explorer (IE) to follow the suit, as IE8—currently an RC1 beta—accepts 3<sup>rd</sup> party add-ons, too<sup>135</sup>, although a strong developer community, such as exists behind Firefox, is missing<sup>136</sup>. There are currently 6843 add-ons available on <https://addons.mozilla.org/> (AMO) for Firefox, the official Mozilla Foundation website that acts as a repository for add-ons for Mozilla software, including Mozilla Firefox, Mozilla Thunderbird, SeaMonkey, and Mozilla Sunbird<sup>137</sup>.

Firefox add-ons can be divided to three groups: Extensions, themes, and plugins. Extensions either modify the existing functionality or add new functionality to Firefox, and they allow customization so that each user can have a Firefox that fits his or her needs<sup>138</sup>. Plugins, on the other hand, are binary components that help the browser display specific content that it could not display natively, such as multimedia or PDF files, directly inside the browser<sup>139</sup>. Themes are skins that change the look and feel of Firefox interface, allowing users to personalize the browser to their tastes<sup>140</sup>. While themes may simply change some colors on the interface, they can be used to change every piece of its appearance. In the following we will limit the discussion to the extensions and ways to build them.

Firefox extensions modify the existing or add new functionalities to the stand-alone browser. The 6471 extensions available constitute the majority of add-ons available on AMO. There is a great range of extension types available, as can be seen from the 16 categories they are divided to in AMO: “Alerts & Updates”, “Appearance”, “Bookmarks”, “Dictionaries & Language Packs”, “Download Management”, “Feeds, News & Blogging”, “Language Support”, “Photos, Music & Videos”, “Privacy & Security”, “Search Tools”, “Social & Communication”, “Tabs”, “Toolbars”, “Web Development” and “Other”. Naturally, one extension can be in several categories.

In addition to categories, AMO offers keyword search that can be narrowed down with the categories (drop-down), default being “all add-ons.” The search also offers an advanced version that opens under the keyword search (Figure 40). It is also possible to organize the extensions by popularity, ranking, and recent update.

---

<sup>132</sup> [http://en.wikipedia.org/wiki/Firefox\\_\(browser\)](http://en.wikipedia.org/wiki/Firefox_(browser))

<sup>133</sup> [http://en.wikipedia.org/wiki/Firefox\\_\(browser\)](http://en.wikipedia.org/wiki/Firefox_(browser))

<sup>134</sup> <http://marketshare.hitslink.com/report.aspx?qprid=2>

<sup>135</sup> <http://www.pcmag.com/article2/0,2817,2339704,00.asp>

<sup>136</sup> [http://www.informationweek.com/blog/main/archives/2009/02/ie8\\_better\\_but.html](http://www.informationweek.com/blog/main/archives/2009/02/ie8_better_but.html)

<sup>137</sup> [http://en.wikipedia.org/wiki/Mozilla\\_Add-ons](http://en.wikipedia.org/wiki/Mozilla_Add-ons)

<sup>138</sup> <https://developer.mozilla.org/en/Extensions>

<sup>139</sup> <https://developer.mozilla.org/en/Plugins>

<sup>140</sup> <https://developer.mozilla.org/en/Themes>

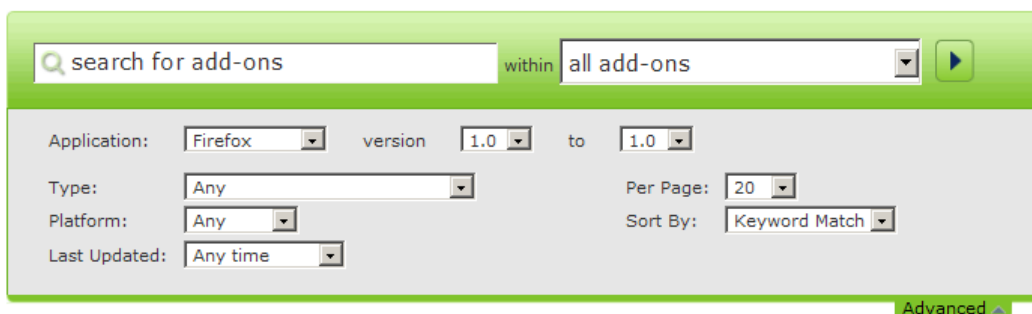


Figure 40 – Advanced search opened

The results page shows each add-on in its own box with an image, name, author, star rating, the number of reviews the rating is based on, and the number of weekly reviews in addition to the categories it belongs to and a short description. The boxes are color-coded. The recommended add-ons are on a gradient pastel green background, normal ones on gradient pastel blue, and the experimental ones on a gradient pastel red. Experimental and recommended add-ons also have a plaque in their respective colors on the left upper corner.

If the add-on is for an older version of the Firefox, then the Add to Firefox button on the right is white as it is with the experimental ones. This indicates that add-ons that are for older versions and do not yet have an updated version for the current browser cannot be added. Experimental ones require signing in to install, and they are meant only for advanced users for testing before being made available for general users<sup>141</sup>. Many are in prototype, alpha, or beta state, and have not been tested by an AMO editor. By forcing users to sign in the Mozilla Foundation is making sure that they understand that they are taking a risk. This also underlines that the other add-ons have been tested by an AMO editor and have been found safe. Experimental ones are not shown by default, and making them visible requires checking the checkbox.

However, the editorial checking has not always prevented security problems, as the case of GreaseMonkey (currently the 5<sup>th</sup> most downloaded extension by weekly downloads) showed<sup>142</sup>. The developers had a lax attitude towards security, which backfired badly and embarrassingly for the whole open source community: GreaseMonkey had a huge security hole that would allow data to be leaked to remote sites<sup>143</sup>. The problem was solved quickly and with transparency that did credit to the accomplices, but the case underlines the need to take measures to protect the users of an ecosystem from intended or accidental problems related to user-created applications and services.

Although the Firefox source code is “maintained by a large number of developers and reviewed by many security experts”, a particular extension is “less likely to be reviewed by as many people”<sup>144</sup>. An extension’s vulnerability is also browser vulnerability and can expose users to possible harm. A system is only as secure as its least secure link.

Firefox also has an in-built Add-ons manager that gives recommendations, offers access to AMO (Browse All Add-ons and See all Recommended Add-ons links), and keyword searching (Figure 41). Add-ons can be enabled, disabled, and uninstalled in the manager (through the Extensions, Themes, and Plugins tabs).

<sup>141</sup> <https://addons.mozilla.org/en-US/firefox/pages/experimentalAddons>

<sup>142</sup> <http://www.rietta.com/firefox/Tutorial/security.html>

<sup>143</sup> <http://www.rietta.com/firefox/Tutorial/security.html>

<sup>144</sup> <http://www.rietta.com/firefox/Tutorial/security.html>

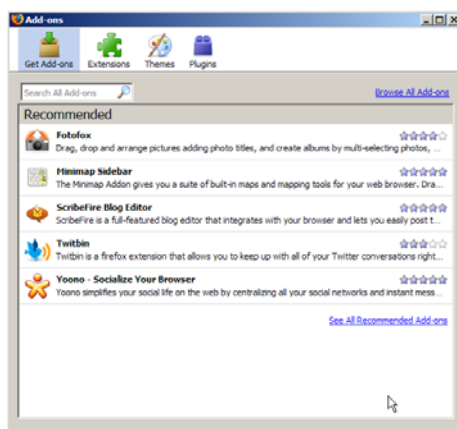


Figure 41 – Firefox add-on manager

Looking at the 10 most popular (by the number of weekly downloads) Firefox extensions, it appears that they are typically made by software developers (Table 6). Both the total download and weekly download numbers are impressive. Interestingly, the most popular categories are *Download management* (4) and *Web development* (4), followed by *Privacy and security* (2) and *Appearance* (2).

Table 6 – Most popular Firefox extensions (23.2.2009).

Extension	Weekly downloads	Total downloads	Category	Review
Video DownloadHelper	563,024	31,036,200	Download management	1232
FlashGot	551,440	58,302,576	Download management	135
Adblock Plus	495,882	41,941,220	Privacy, security	886
NoScript	494,980	42,383,194	Web dev, privacy, security	418
Greasemonkey	252,961	15,422,730	Web development	199
DownThemAll!	207,505	26,157,115	Download management	556
Firebug	184,367	13,091,276	Web development	454
IE Tab	182,488	20,778,657	Web development, appearance	496
PDF Download	174,392	12,715,150	Download, alerts, updates	206
Cooliris	172,276	12,098,436	Feeds, news, blogs, search Photos, music, videos	773

If we compare the most popular extensions to the ones one and a half years ago, we notice that little has changed. According to Wikipedia.org<sup>145</sup>, the most popular extensions in Oct. 30, 2007, were NoScript (now no. 4), Tab Mix Plus (now no. 21 – keeping in mind that many features offered by Tab Mix Plus have become part of the Firefox distribution), FoxyTunes (now no. 27), Adblock Plus (now no. 3), StumbleUpon (now no. 16), Foxmarks Bookmark Synchronizer (now no. 11), DownThemAll (now no. 6), and Web Developer toolbar (now no. 23). While there are new kids on the block, nothing has substantially changed in the applications or the types of applications that are popular among the Firefox users.

MozillaZine<sup>146</sup> maintains a list of extensions that are known to be problematic. Of the ten most popular, six were included in the current list: Adblock Plus (can interfere with Flash content, most often on Mac OS), NoScript (do not use both FlashBlock and NoScript together, as NoScript blocks JavaScript required by FlashBlock and includes Flash-blocking functionality), Greasemonkey (Greasemonkey scripts do not work on sites blocked by NoScript), DownThemAll (may cause web browsing to slow to a halt to the point that nothing loads and changes certain "network.http" preferences that will need to be reset, even if you disable or uninstall the extension), Firebug (may cause excessive CPU usage; Firefox may hang when a pop-up window is accessed), and IE Tab (memory leak). Thus, while sometimes the cause is simply overlapping and contradictory extensions, some extensions cause problems, such as memory leaks, even alone.

<sup>145</sup> [http://en.wikipedia.org/wiki/Firefox\\_\(browser\)](http://en.wikipedia.org/wiki/Firefox_(browser))

<sup>146</sup> [http://kb.mozillazine.org/Problematic\\_extensions](http://kb.mozillazine.org/Problematic_extensions)

Although developers are not paid for developing extensions, many of them do solicit donations, making many extensions effectively donationware.

### 7.1.1 Building Firefox extensions

A Firefox Extension is simply “a collection of files and folders that have been compressed into a file with a .xpi extension. The .xpi file (pronounced zippy) is nothing more than a .zip file that has been renamed.”<sup>147</sup>

Writing extensions for Firefox does not require extensive development background. However, a working knowledge of JavaScript is necessary, as most of the work is done with it. Setting up an extension development environment is also rather straightforward. It involves simply creating the following folder structure<sup>148</sup>:

```
MyExt /
      chrome/
      chrome/chromeFiles/
      chrome/chromeFiles/content/
      defaults/
      defaults/preferences/
```

Extensions—or more typically parts of them—can also be developed in C++ especially if there is a “need for high-performance beyond what can be delivered by JavaScript code” or one needs to use 3<sup>rd</sup> party C/C++ libraries<sup>149</sup>. However, this is again where only eagles dare, as it is the domain of C++ developers with years of experience. Others would do fine to stick to JavaScript<sup>150</sup>.

Most extensions also need to add or change some graphical elements in Firefox. Firefox’s interface is written in XUL and JavaScript<sup>151</sup>. XUL—XML User Interface Language, pronounced “zool”—is “an XML grammar that provides user interface widgets”, such as buttons, menus, toolbars, and trees.<sup>152</sup> User actions are bound to functionality with JavaScript. Although XUL follows the XML grammar, using it still requires understanding XUL DOM. XUL overlays allow attaching UI widgets to a XUL document at run time. They are in effect .xul files that specify XUL fragments, such as widgets, to be inserted, removed, or modified.<sup>153</sup>

Thus, while the tools for extending Firefox are not very complex, using them still requires specialized knowledge and willingness to go through some learning. Mozilla Developer Center<sup>154</sup> offers materials to start with to lower the curve, but it is no surprise that all the developers behind the most popular extensions shown in Table 6 have a developer background.

One consolation about the tools is that because of this approach, the extensions are by and large platform independent<sup>155</sup>. If one wishes to distribute the extension one has made, there are two options. The simpler one is to simply host the extension on one’s own website. However, AMO in most cases brings the extension to people’s fingertips more efficiently. First one needs to create an AMO account, and to log in. After submitting the extension through the system, it takes several days before it shows up in AMO because of the reviewing process. “Once the add-on has been reviewed, it will be made available for downloading. Reviews

---

<sup>147</sup> <http://www.rietta.com/firefox/Tutorial/conf.html>

<sup>148</sup> <http://www.rietta.com/firefox/Tutorial/conf.html>

<sup>149</sup> [https://developer.mozilla.org/en/Creating\\_Custom\\_Firefox\\_Extensions\\_with\\_the\\_Mozilla\\_Build\\_System](https://developer.mozilla.org/en/Creating_Custom_Firefox_Extensions_with_the_Mozilla_Build_System)

<sup>150</sup> [https://developer.mozilla.org/en/Creating\\_Custom\\_Firefox\\_Extensions\\_with\\_the\\_Mozilla\\_Build\\_System](https://developer.mozilla.org/en/Creating_Custom_Firefox_Extensions_with_the_Mozilla_Build_System)

<sup>151</sup> [https://developer.mozilla.org/en/Building\\_an\\_Extension](https://developer.mozilla.org/en/Building_an_Extension)

<sup>152</sup> [https://developer.mozilla.org/en/Building\\_an\\_Extension](https://developer.mozilla.org/en/Building_an_Extension)

<sup>153</sup> [https://developer.mozilla.org/en/Building\\_an\\_Extension](https://developer.mozilla.org/en/Building_an_Extension)

<sup>154</sup> <https://developer.mozilla.org/en/Extensions>

<sup>155</sup> <http://www.rietta.com/firefox/Tutorial/env.html>

can take a varying amount of time depending on how many pending submissions there are and the availability of people to perform the reviews.”<sup>156</sup>

### 7.1.2 Future directions

*“Firefox’s rich add-on ecosystem allows us to bend the browser to our will and deck it out with features to do almost anything.”* - David Chartier (Ars Technica)<sup>157</sup>

Google’s direction is to make the browser the new OS<sup>158</sup>. With Firefox’s extendibility, it is also becoming an ecosystem of its own, providing all kinds of functionality that a netizen may need, ranging from synchronizing bookmarks, calendars, and todo-lists to being “a great tool for drafting posts, saving Web snippets, finding topical content, and more”, even up to the point of becoming a “blogging machine”<sup>159</sup>. In addition, Firefox is also becoming social, as such extensions as Glue<sup>160</sup> and StumbleUpon, and 214 other add-ons in the Social & Communication category testify.

Moreover, Firefox offers even feed readers, such as Wizz RSS News Reader or Homeland Security Threat Level, and extensions for making mashups directly in the browser, such as Intel MashupMaker or Open Mashups Studio & Runtime. In addition, there are extensions for adding social elements to the pages one views, thus combining social and mashup approaches, such as Socialbrowse. Thus, Firefox is becoming a complete platform for many Internet-related activities, not simply a tool for viewing page content as-is.

Firefox is also going head to head with Adobe’s Flash and Microsoft’s Silverlight, as the graphics rendering system Cairo, part of Firefox 3, attests and Mozilla CEO Mitchell Baker confirms<sup>161</sup>. In addition, the Mozilla Foundation has announced that “it is giving a \$100,000 grant, by way of the Wikimedia Foundation, to help develop an open-source standard for Internet video”<sup>162</sup>.

Another direction is mobile browsers. Mozilla’s head of mobile business, Jay Sullivan, sees add-ons as integral part of Firefox’s mobile browser strategy: “We will also be the first mobile browser that supports add-ons...add-ons will be a huge part of the experience”<sup>163</sup>. Another key factor is providing support for web developers: In addition to HTML, JavaScript, and CSS, developers “will also be able to access the camera, accelerometer, location and so on.”<sup>164</sup>

Interestingly, Sullivan states that “Mozilla would target Windows Mobile, Maemo, Symbian and LiMo as platforms for its browser”, as iPhone forbids it and both Android and BlackBerry are written in Java while Firefox is not, which makes the situation tricky.<sup>165</sup>

Now that Google has released Chrome, Firefox is slowly coming to a crossroad, since Google has been a financially important power behind it. For instance, in 2006, The Mozilla Foundation released “an audited financial statement for the year 2006 showing that it took in \$66 million for the year, \$61 million of which were search royalties, largely from Google.”<sup>166</sup> While Google’s support for Firefox has not dried up yet, Firefox’s financial future is going to go through changes eventually.

---

<sup>156</sup> [https://developer.mozilla.org/en/Submitting\\_an\\_add-on\\_to\\_AMO](https://developer.mozilla.org/en/Submitting_an_add-on_to_AMO)

<sup>157</sup> <http://arstechnica.com/web/guides/2009/02/pimp-my-browser-how-to-turn-firefox-into-a-blogging-machine.ars>

<sup>158</sup> <http://googlesystem.blogspot.com/2008/09/google-os-is-actually-browser-google.html>

<sup>159</sup> <http://arstechnica.com/web/guides/2009/02/pimp-my-browser-how-to-turn-firefox-into-a-blogging-machine.ars>

<sup>160</sup> <https://addons.mozilla.org/en-US/firefox/addon/3481>

<sup>161</sup> [http://apcmag.com/firefox\\_to\\_go\\_headtohead\\_with\\_flash\\_and\\_silverlight.htm](http://apcmag.com/firefox_to_go_headtohead_with_flash_and_silverlight.htm)

<sup>162</sup> <http://bits.blogs.nytimes.com/2009/01/27/mozilla-spends-money-to-make-web-video-free/>

<sup>163</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/02/19/AR2009021901914.html>

<sup>164</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/02/19/AR2009021901914.html>

<sup>165</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2009/02/19/AR2009021901914.html>

<sup>166</sup> <http://www.paidcontent.org/entry/419-mozilla-releases-2006-financials-61-million-in-search-royalties>

## 7.2 Facebook

*“Facebook.com is the current ‘uber’ mashup today. As a massive social networking site, Facebook has become a cultural phenomenon. It mashes up many different creative services into a unified social experience online. There are hundreds of applications being mashed together at Facebook.”* - Paul Gil (January, 2009)<sup>167</sup>

In this review, we look at Facebook as a mashup environment rather than try to look at every aspect of it.

Facebook, formerly Thefacebook, is a free-access social network service that enables people to communicate and share content with other people. The name refers to the paper facebooks that some US institutions give to incoming students and staff as a way to get to know people on campus<sup>168</sup>. If Facebook was a country, it would be the 6<sup>th</sup> largest in the world with over 175 million active<sup>169</sup> users<sup>170</sup>. Facebook was founded by Mark Zuckerberg (current CEO) and launched in February 2004. First the use was limited only to Harvard students, then it expanded to other colleges, university and high school students, and finally to anyone aged 13 and over.<sup>171</sup> The only thing needed to join up is a valid e-mail address for account opening confirmation.

Facebook has grown fast. At the end of year 2004, it had reached nearly 1 million active users and a year after more than 5.5 million active users. After September 2006 when Facebook opened to everybody, the growth has continued unabated with over 175 million active users in February 2009.

The fact that more than three billion minutes are spent on Facebook each day and more than 15 million users update their statuses at least once each day shows how engaging user experience Facebook offers. Facebook is now available in more than 35 languages and more than 60 new translations are in development. More than 70% of Facebook users are outside the United States.<sup>172</sup> The fastest growing user group is 35-54 year olds with a growth rate of 276% over six months. However, the growth rate of over 55 year olds is not far behind at 194% and the number of 25-34 year olds is also doubling every six months.<sup>173</sup>

---

<sup>167</sup> <http://netforbeginners.about.com/od/m/f/whatismashup.htm>

<sup>168</sup> <http://en.wikipedia.org/wiki/Facebook>

<sup>169</sup> Facebook defines users active if they have returned to the site in the last 30 days.

<sup>170</sup> <http://news.therecord.com/News/article/494005>

<sup>171</sup> <http://en.wikipedia.org/wiki/Facebook>

<sup>172</sup> <http://www.facebook.com/press/info.php?statistics>

<sup>173</sup> <http://www.istrategylabs.com/2009-facebook-demographics-and-statistics-report-276-growth-in-35-54-year-old-users/>





Figure 42 – The Facebook profile of Mark Zuckerberg

An example of Facebook’s user profile is shown in Figure 42. Facebook’s factsheet says the following concerning Facebook’s structure (which differs slightly from the contents of example image): “Facebook’s simplified navigation gives users easy access to core site functions and applications. Profile, Friends, Networks and Inbox – pages core to the user experience on Facebook – have a prominent place at the top of the user’s profile page. Facebook applications – Photos, Notes, Groups, Events and Posted items – are displayed on the left side bar, along with any third-party applications a user has added to their account.”<sup>174</sup>

Nevertheless, the fact that there are a number of courses, offered for instance even in Finnish libraries<sup>175</sup>, indicates that the complexity of the service can be daunting for some users. These courses are targeted at all adults interested in the service and having the basic prerequisite of computer literacy. Interestingly, there are also courses on Facebook application developers<sup>176</sup> and advertisers<sup>177</sup>. More ominously, there are also courses in some countries to help parents understand and monitor their children’s use of Facebook<sup>178</sup>.

<sup>174</sup> <http://www.facebook.com/press/info.php?factsheet>

<sup>175</sup> [http://www.iltalehti.fi/espoo/200811178603110\\_eo.shtml](http://www.iltalehti.fi/espoo/200811178603110_eo.shtml)

<sup>176</sup> For example <http://www.fcsovelto.fi/kurssit/Kurssi.aspx?ID=2701&n=Facebook-sovelluskehitys>

<sup>177</sup> For example <http://www.fcsovelto.fi/kurssit/Kurssi.aspx?ID=2705&n=Facebook%20markkinoinnin%20tukena>

<sup>178</sup> For example <http://facebookforparents.org/>

### 7.2.1 Using Facebook applications

In Facebook, the mini-applications that users can contribute to the Facebook ecosystem are simply known as applications. Facebook's decision to open the platform to applications developers has been called a "masterstroke" and it—together with the resulting applications—has contributed to Facebook's popularity<sup>179</sup>.

The numbers involved in their use statistics are staggering. For example, each month there are more than 850 million photos and five million videos uploaded and more than 24 million pieces of content (links, news stories, notes, etc.) shared<sup>180</sup>. The default applications<sup>181</sup> made available for any new user on top right of the home page (Figure 43) consist of the following:

- *Photos* application allows unlimited number of uploads and a possibility to arrange photos in different albums. Privacy settings, such as visibility, can be adjusted separately for each album. Photos can also be tagged with the names of people in the photos (if they have not denied it via their privacy settings) and others can comment them.
- *Groups* application allows users to search and browse all Facebook groups, see their own groups, their friends' groups, and create new groups.
- *Notes* application lets users to blog or to import an external blog to Facebook. Users can view their own and their friends' notes collectively on one page. Notes can also be tagged with friends so that those friends can browse notes written about themselves and their friends.
- *Gifts* application contains icons that cost one dollar and can be sent to Facebook friends.
- *Video* application allows users to upload video files to their profile, send video messages, and check their friends' videos.
- *Events* application collects all information about users' forthcoming and past events together. The application also lets users know about interesting things coming up among their friends and in the groups they belong to. Users can also organize their own events.
- *Posted Items* application, including its sub-functionality is an easy way to share anything on the Internet (websites, blogs, videos etc.) and on one's Facebook profile (photos, notes etc.) with help of the Publisher.
- *The Facebook Wall*, a core component of Facebook, is a history log and timeline for Facebook experience (in the middle of Figure 42) that can also be considered an application. It is a tool for leaving comments on one's friends' pages. It also logs the user's Facebook activity stream based on the user's settings.

---

<sup>179</sup> [http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/03/nielsen\\_globalfaces\\_mar09.pdf](http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/03/nielsen_globalfaces_mar09.pdf)

<sup>180</sup> <http://www.facebook.com/press/info.php?statistics>

<sup>181</sup> <http://www.facebook.com/press/product.php> and <http://en.wikipedia.org/wiki/Facebook>

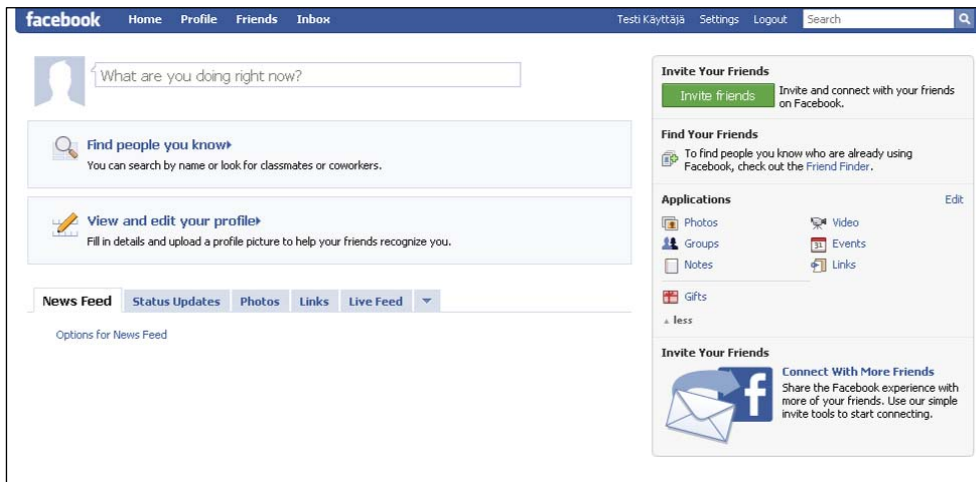


Figure 43 – Facebook home page for a new user

## 7.2.2 News feeds and requests

One feasible way to find new applications in Facebook is through one's friends. An average Facebook user has 120 friends<sup>182</sup>. Cameron Marlow, a Facebook sociologist<sup>183</sup>, says that the number of friends a user has is very different than the number of friends the user frequently interacts with. Men on average leave comments and respond to postings of 7 and women to 10 friends. Two-way communication (e-mail and chat) is even lower: men interacting with 4 and women with 6 of their friends on average. While the number of friends of some Facebook users can be over 500, such users are not that different on these interaction numbers: men leave comments for 17 and communicate with 10 friends while for women the numbers are 26 and 16.<sup>184</sup>

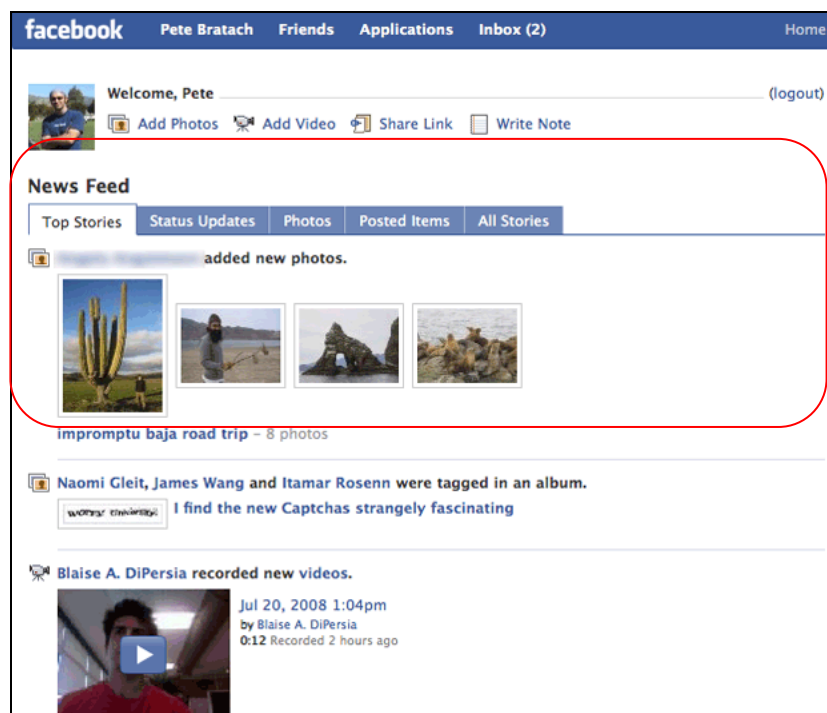


Figure 44 – Example of news feeds

<sup>182</sup> <http://www.facebook.com/press/info.php?statistics>

<sup>183</sup> <http://latimesblogs.latimes.com/technology/2009/03/monkeysphere.html>

<sup>184</sup> [http://www.economist.com/science/displayStory.cfm?story\\_id=13176775&source=hptextfeature](http://www.economist.com/science/displayStory.cfm?story_id=13176775&source=hptextfeature)

Figure 44 shows Photos application's News Feed story about photos that the user's friend has uploaded. This feed is by default sent to **all** friends the user has. This way even new users get information about applications used by their friends. Applications can also send notifications to those users who have allowed these kinds of notifications.<sup>185</sup> Users can specify to some extent which of their activities will initiate a feed. They can also do the same adjustments on what they see of their friends' activities. Users can also check their friends' profiles to see which applications they are using.

In March 2009 Facebook informed its users on their profile homepages that the Facebook home page is going to be changed in the near future. More information was presented to interested users as a homepage tour<sup>186</sup>. Posts of users' friends will be streamed in real time, and users' control over different feeds is going to be increased by creating more detailed filters based on applications and friend lists (groups of friends that users can make). The stream evolves fast and users might have difficulties in keeping track of everything interesting that is happening. The new section of Highlights will help with this by presenting only the pieces of information with which the user's friends have been interacting the most.

Facebook applications can create requests that appear on users' homepages on the top right corner (Figure 45). Actions of the user's friends typically initiate requests that require the user to take some action (for example, Photos application requesting a photo tag confirmation).

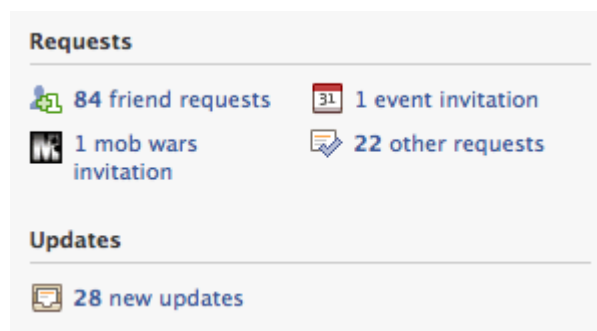


Figure 45 – Examples of different kinds of requests on profile page

### 7.2.3 Application directory

All applications can be found in Facebook Application directory (Figure 46). There are four categories for different types of applications: for Facebook Profiles, for Desktop, for The Web and by Facebook. Only the first one is divided to sub-categories: Alerts, Business, Chat, Classified, Dating, Education, Events, Fashion, File Sharing, Food and Drink, Gaming, Just for Fun, Messaging, Mobile, Money, Music, Photo, Politics, Sports, Travel, Utility, and Video.

The application list contains limited amount of information about the application, namely the icon, name, developer's name, and textual description. Facebook suggests the application developers to use "beautiful imagery and concise, descriptive text"<sup>187</sup>. Some applications also provide the number of monthly active users and reviews. More information about an application can be accessed from its About page.

In the Application directory, the default tab in For Facebook Profiles is Applications You May Like that contains the applications the user's friends are using. Applications can also be sorted based on their age (Newest) or amount of active users (Most Active Users). In contrast, before August 2007, Facebook measured the popularity of applications based on the number of users. Criticism from users, however, caused a shift in the focus from the highly viral, yet useless applications to more engaging applications.<sup>188</sup> When

<sup>185</sup> [http://developers.facebook.com/get\\_started.php?tab=anatomy](http://developers.facebook.com/get_started.php?tab=anatomy)

<sup>186</sup> [http://www.facebook.com/sitetour/homepage\\_tour.php](http://www.facebook.com/sitetour/homepage_tour.php)

<sup>187</sup> [http://developers.facebook.com/get\\_started.php?tab=anatomy](http://developers.facebook.com/get_started.php?tab=anatomy)

<sup>188</sup> "A shift to engagement". <http://www.facebook.com/developers/>

applications are filtered by language, users also get the Recently Popular sorting option, but on the whole the relationship between different selections and produced results is a bit confusing.

Table 7 presents a list of the most popular English applications of each category in March 2009 based on the number of monthly active users, that is, users that have used the application within the last 30 days. For Facebook Profiles category is on its own league based on the number of active users, matched only by Facebook® for BlackBerry® Smartphones (For Your Desktop category) and Facebook Mobile (By Facebook category). Because the For Facebook Profiles category has the most popular applications, we included ten of its most used applications. Furthermore, the application types in this category vary greatly, which can already clearly be seen in these ten applications. It is important to realize that the situation is constantly evolving, and even during this review some of the top five applications have changed their order and some categories even got new items.

Table 7 - Popular Facebook applications.

<b>TOP 10 applications For Facebook Profiles</b>			
<b>Name and developers</b>	<b>Monthly users</b>	<b>Reviews and rating (/5)</b>	<b>Description</b>
Causes By R. Kabir	25,534,451	216 (3.7)	Start and join the causes, collect donations
Slide FunSpace By Slide	17,842,284	1 250 (2.2)	Find & share videos, posters, graffiti, and so on.
Top Friends By Slide	17,410,152	1 253 (2.2)	Share the profile with friends, includes customized skins, musics, and so on.
Super Wall By RockYou!	16,177,148	550 (2.6)	Share videos, pictures, graffiti, and so on
We're Related By FamilyLink	16,077,640	758 (2.7)	Build your family tree and see who you are related to on Facebook
Movies By Flixster	10,918,928	206 (2.8)	Compare the movie taste with friends, share reviews, discover new movies, test knowlege
Texas HoldEm By Zynga	10,325,550	3,506 (4.4)	Play Texas Hold'Em Poker with Facebook friends.
Pass a Drink By socialreach.com	9,815,458	10 (-)	Pass a drink or a round of drinks to your friends to show them some love.
Pet Society By Playfish	8,227,229	47,928 (4.7)	Play games, decorate house and bring gifts for your friends when you and your pet visit them
Bumper Sticker By Harris Tsim	7,962,317	6,480 (2.3)	Create own stickers, stick them to your friends.
<b>TOP 5 applications For Your Desktop</b>			
<b>Name and developers</b>	<b>Monthly users</b>	<b>Reviews and rating (/5)</b>	<b>Description</b>
Fb for BlackBerry By RIM	4,075,476	378 (3.4)	Facebook access from Blackberries – status updates, photo uploads, messaging, pokes, etc.
Fb for iPhoto By Facebook	-	147 (3.4)	Tag, caption, and export photos from your iPhoto library directly to Facebook.
Flock, By Flock	241,111	57 (4.7)	Flock's People sidebar for Flock web browser
Toolbar for Firefox By Facebook	-	130 (3.9)	Integrate your Facebook life into your browser.
Digsby, By dotSyntax	89,217	164 (4.7)	Combination of IM, email, and social networks
<b>TOP 5 applications For The Web</b>			
<b>Name and developers</b>	<b>Monthly users</b>	<b>Reviews and rating (/5)</b>	<b>Description</b>
Fb for Palm, By Palm	131,620	161 (3.3)	Integration of Facebook with Palm smart phones
Pool By M. Smith	64,415	302 (3.6)	A pool game.
Fast and Furious By A. Fedorov	58,016	-	A car chasing game.
Vimeo By Vimeo	22,446	5 (2.2)	Connection to Vimeo video sharing website.
INQ1 By INQ mobile	19,200	5 (5.0)	INQ Social Mobiles.
<b>TOP 5 applications By Facebook</b>			
<b>Name and developers</b>	<b>Monthly users</b>	<b>Reviews and rating (/5)</b>	<b>Description</b>
Facebook Mobile	4,953,213	572 (2.6)	Mobile device interface to Facebook.
Facebook Groups	-	607 (3.5)	Create and join groups.
Facebook Gifts	-	791 (1.9)	Virtual gifts in Facebook.
Facebook Video	-	803 (2.8)	Video upload, video messages, tagging, etc.
Discussion Boards	-	3,566 (4.4)	Discussion forums.

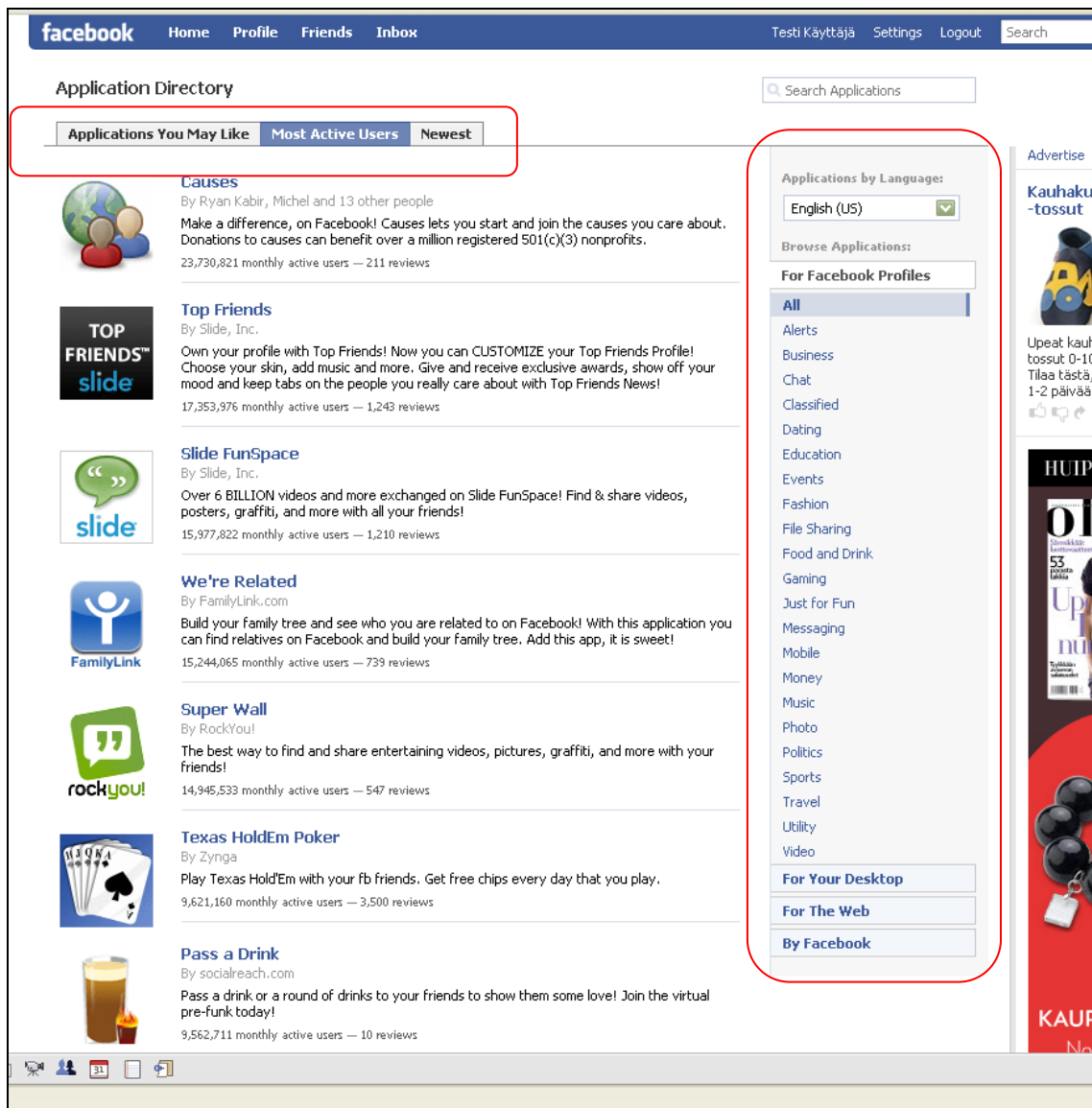


Figure 46 – Application directory

In addition to Facebook's built-in Application directory, there are even applications to help the user find interesting applications, such as Application finder<sup>189</sup>. With this application, the user gets various options for finding applications, such as a feed on recently reviewed, top ranked, lowest ranked, and a possibility to browse applications by a tag cloud. The application is developed by AppSORAMA, a team that develops Facebook applications based on user requests. However, Application finder has only 15,899 monthly active users even though it has been available at least since June 2007, which attests to its lack of popularity. Applications also need to be submitted into the finder in order to be included.<sup>190</sup>

Applications such as Application finder underline the great variety among the Facebook applications that include applications for time/content management, games, personality quizzes, among many, many other types.

<sup>189</sup> <http://apps.facebook.com/appsdirectory/>

<sup>190</sup> <http://www.facebook.com/apps/application.php?id=2448217719>

## 7.2.4 Application initialization process

To start using a Facebook application, the user just needs to click its link on the list page to get to the application's About page (Figure 47) that displays more information about the application, its developers (Facebook or third party, see the red square in Figure 47), reviews, number of fans, and discussions (not visible in Figure 47). Every user can become a fan of an application in addition to becoming a fan of different pages. The About page lets the user see a few of the application's fans by name and profile picture, sometimes friends of the user looking at the page, which may influence the decision to start using or ignore the application. Reviews allow users to leave opinions about how others felt about the application.

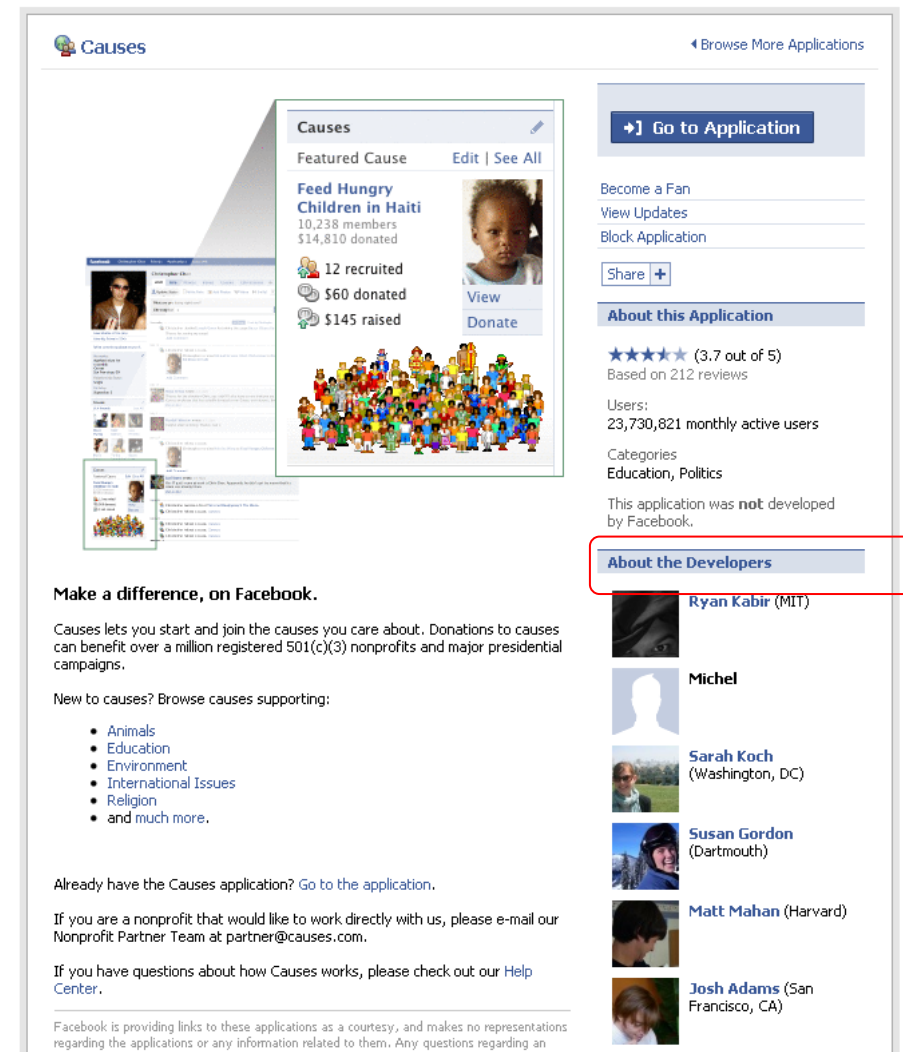


Figure 47 – The About page of Causes application

The About page's function is to motivate users to try the application<sup>191</sup>. If the user decides to continue by clicking "Go to application", he or she needs to allow the application to access undefined parts of the user profile information, photos, friends' information and other content that it needs to work. This is especially problematic with third party applications not developed by Facebook because the user cannot know exactly what information will be used and how. It is also worth to notice that this is only an example of one application's initialization process. The actual requirements for using an application and steps involved in initializing it may vary greatly. In contrast, starting to use a default application mentioned before does not require these steps at all. Starting to use an application might also require the user to invite some of his or her friends to use it too before even being able to use it him or herself.

<sup>191</sup> [http://developers.facebook.com/get\\_started.php?tab=anatomy](http://developers.facebook.com/get_started.php?tab=anatomy)



The Canvas Page is the main page of an application on Facebook<sup>192</sup>. A Facebook application can take on many forms and be integrated in different areas of the Facebook profile. Figure 48 shows some of these. The profile box is generally a place to show the most recent updates and actions of the user. Besides the Boxes tab, a default place for all users applications, users can create individual tabs for applications.<sup>193</sup> The Info tab allows users to structure information related to applications as a list of text and images<sup>194</sup> (Figure 49).



Figure 48 – The integration of Smiley application with profile box and tabs

<sup>192</sup> [http://developers.facebook.com/get\\_started.php?tab=anatomy](http://developers.facebook.com/get_started.php?tab=anatomy)

<sup>193</sup> [http://developers.facebook.com/get\\_started.php?tab=anatomy](http://developers.facebook.com/get_started.php?tab=anatomy)

<sup>194</sup> [http://developers.facebook.com/get\\_started.php?tab=anatomy](http://developers.facebook.com/get_started.php?tab=anatomy)

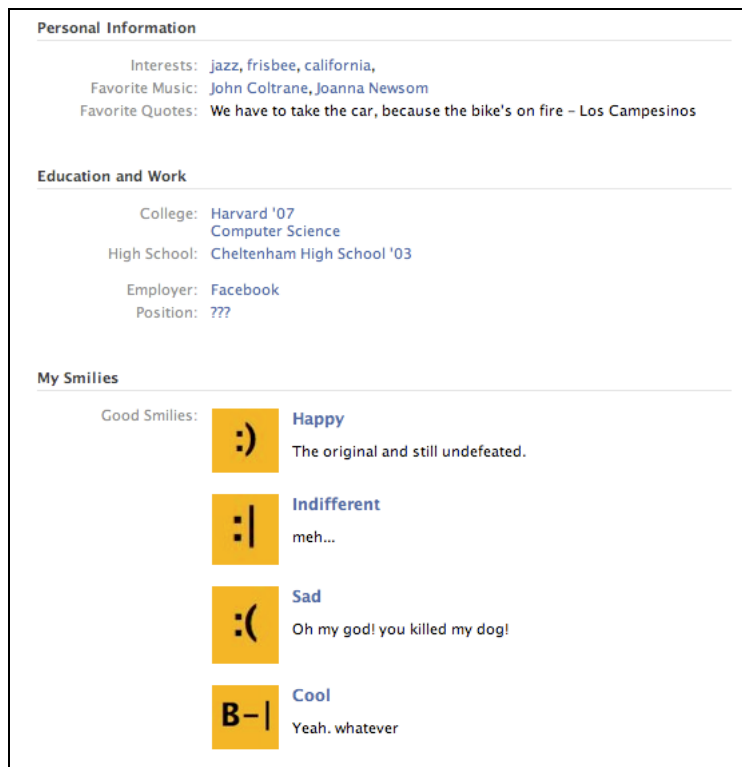


Figure 49 – The integration of Smiley application with Info tab

### 7.2.5 Security of Facebook applications

Users can change privacy options of every application separately from the application's Privacy settings page. The settings include story posting options, places where the application is presented (profile box, tab, and info section) and to whom.

There have been numerous findings of third party Facebook applications installing spyware. The example from January 2008 concerns a malicious Facebook application *Secret Crush* that led many users to install the infamous Zango spyware. The application appeared to users as an innocent Facebook request (Figure 50).<sup>195</sup> After responding to the request, the user finds a message informing that one of his or her friends has a secret crush and invites him or her to find out more by using the Secret Crush application (a common approach for Facebook applications). The dialogues are well designed to ensure that the user believes the cover story.<sup>196</sup>

After the user has chosen to add this application to his or her profile, the terms of use are displayed the same way as in all third-party application installations. Unfortunately, users tend to be rather careless about giving access to their personal information. In this example, many users were too eager to find out who had a crush on them to mind about privacy issues.<sup>197</sup> After accepting the terms, the user needs to invite at least five friends to use the application before being able to use it to find out who might have a crush on him or her.<sup>198</sup>

<sup>195</sup> <http://www.fortiguardcenter.com/advisory/FGA-2007-16.html>

<sup>196</sup> <http://www.fortiguardcenter.com/advisory/FGA-2007-16.html>

<sup>197</sup> <http://www.fortiguardcenter.com/advisory/FGA-2007-16.html>

<sup>198</sup> <http://www.fortiguardcenter.com/advisory/FGA-2007-16.html>



Figure 50 – Secret crush application making a request

Secret Crush application is a social worm. It does not rely on phishing or any sort of user-space customization feature abuse “but on pure social engineering which is based on simple manipulation strategies such as “escalation of commitment””. It is psychologically hard for users to discontinue the process even if they do not want to invite friends just after disclosing their personal information to the application. So, most users invited at least five friends to complete the process. The person having the crush is still not revealed but the user is asked to click “download now” which leads to a copy of the infamous Zango spyware.<sup>199</sup>

A more recent example is from February 2009. The Error Check System application gives users a notification that one of their friends has had troubles viewing their profile. The notification has a link to a page where it is recommended that users install an application to fix the error, but what it really does is collect information about users and their friends.<sup>200</sup>

Facebook is taking actions against harmful third party applications. They have established guiding principles for the developers of social applications. Social application should be meaningful, i.e. provide value to users, trustworthy meaning that they respect users, and well-designed so that they are usable and scaleable.<sup>201</sup> According to the principles, a trustworthy application should be:

- *Secure*: Protects user data and honors privacy choices for everyone across the social graph
- *Respectful*: Values user attention and honors their intentions in communications and actions
- *Transparent*: Explains how features will work and how they won't work, especially in triggering user-to-user communications<sup>202</sup>

Facebook has also other more concrete means to address the issue, for example an optional Application Verification Program (since 11/2008<sup>203</sup>). This program tries to ensure that an application is respectful, transparent, and meets the principles of trustworthiness. The visual symbol for the program-certified applications is the verification badge (Figure 51).<sup>204</sup>

Costs associated with the application review process are 375 dollars per application. For current students and registered non-profits with official documentation the fee is 175 dollars. The first waves of verified applications, starting in early 2009<sup>205</sup>, are expected to number in the hundreds<sup>206</sup>. The program is an example

<sup>199</sup> <http://www.fortiguardcenter.com/advisory/FGA-2007-16.html>

<sup>200</sup> <http://www.kauppaletti.fi/5/i/talous/uutiset/etusivu/uutinen.jsp?oid=2009/02/19344&ext=rss>

<sup>201</sup> [http://developers.facebook.com/get\\_started.php?tab=principles](http://developers.facebook.com/get_started.php?tab=principles)

<sup>202</sup> [http://developers.facebook.com/get\\_started.php?tab=principles](http://developers.facebook.com/get_started.php?tab=principles)

<sup>203</sup> <http://developers.facebook.com/news.php?blog=1&story=168>

<sup>204</sup> <http://developers.facebook.com/verification.php>

<sup>205</sup> <http://developers.facebook.com/verification.php>

<sup>206</sup> <http://developers.facebook.com/news.php?blog=1&story=168>

of how an ecosystem can respond to privacy and security problems caused by third-party applications. The program has caused criticism because of the review fee and the fact that approval needs to be regained every year<sup>207</sup>.

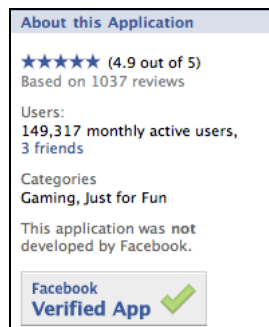


Figure 51 – Example of a verification badge on an application

## 7.2.6 Facebook platform

Facebook development platform was launched on August 2006<sup>208</sup>. Now there are more than 660 000 developers and entrepreneurs from more than 180 countries making applications, more than 52 000 applications available, and 140 new applications added every day. More than 95% of Facebook members have used at least one application built on it.<sup>209</sup>

Facebook development platform has even inspired Stanford University to offer a computer science course called “Create Engaging Web Applications Using Metrics and Learning on Facebook” (Fall 2007) in which the participants created many successful Facebook applications<sup>210</sup>. Different APIs (including authorizing APIs) are available in Facebook platform for third party applications for fetching user data but also to access Facebook core features like notifications, invites and feeds<sup>211</sup>.

Facebook Markup Language (FBML) is a variant-evolved subset of HTML with some elements removed. It allows developer users to some extent customise their applications.<sup>212</sup> “It is the specification of how to encode content so that Facebook’s servers can read and publish it, which is needed in the Facebook-specific feed so that Facebook’s system can properly parse content and publish it as specified<sup>213</sup>.” Facebook also offers a specialised Facebook JavaScript (FBJS) library. The official FBML documentation can be found on the Facebook Developers Wiki.<sup>214</sup>

Basic applications can be up and running in minutes but for that the developer user needs to A) be well versed in some coding language that has a client library for Facebook’s API (such as PHP, Ruby on Rails,

---

<sup>207</sup> [http://www.pcworld.com/businesscenter/article/154099/facebook\\_app\\_verification\\_fee\\_draws\\_criticism.html](http://www.pcworld.com/businesscenter/article/154099/facebook_app_verification_fee_draws_criticism.html)

<sup>208</sup> <http://www.facebook.com/press/info.php?timeline=>

<sup>209</sup> <http://www.facebook.com/press/info.php?statistics>

<sup>210</sup> <http://www.facebook.com/group.php?gid=5378622985>

<sup>211</sup> [http://en.wikipedia.org/wiki/Facebook\\_markup\\_language](http://en.wikipedia.org/wiki/Facebook_markup_language)

<sup>212</sup> [http://en.wikipedia.org/wiki/Facebook\\_markup\\_language](http://en.wikipedia.org/wiki/Facebook_markup_language)

<sup>213</sup> [http://en.wikipedia.org/wiki/Facebook\\_markup\\_language#Facebook\\_Markup\\_Language](http://en.wikipedia.org/wiki/Facebook_markup_language#Facebook_Markup_Language) and [http://oren.blogs.com/praxis/2007/05/facebook\\_markup.html](http://oren.blogs.com/praxis/2007/05/facebook_markup.html)

<sup>214</sup> [http://en.wikipedia.org/wiki/Facebook\\_markup\\_language](http://en.wikipedia.org/wiki/Facebook_markup_language)

JavaScript, or Python), B) have a basic understanding of the Internet, SSH, MySQL, and Unix, C) be familiar with Web hosting fundamentals, and D) have a place to host the application<sup>215</sup>.

Facebook Platform is constantly evolving, and so the developer users need to keep up with the changes. Every week new code is revealed to developers a day before it is released, so they can test their applications against the changes on [www.beta.facebook.com](http://www.beta.facebook.com).<sup>216</sup>

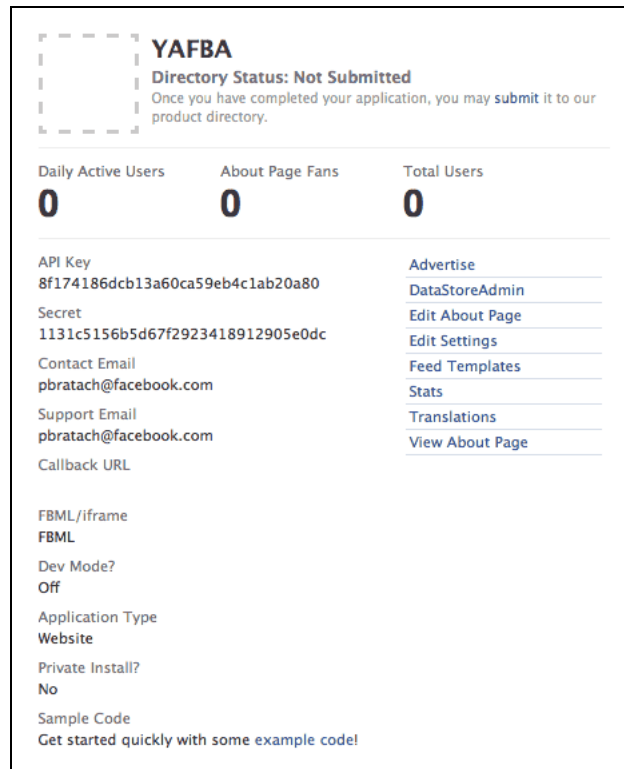


Figure 52 – Basic information about application being created

While setting up an application (Figure 52), the developer user gets an API key that identifies the current application to Facebook. It is passed along with all the API calls that the application makes. Every application also gets an application Secret that Facebook uses to authenticate the requests the application makes. This key should never be shared with anyone.<sup>217</sup> On the server side, there needs to be Facebook PHP client library installed, along with MySQL with memcached. Some hosting services provide a preconfigured environment for Facebook applications (for example, Joynet)<sup>218</sup>

### 7.2.7 The current state of Facebook

Facebook is part of the race for becoming a major part of users' everyday life and browsing. One indicator of this is the Facebook Toolbar for Firefox<sup>219</sup> that makes the following claim: "The Facebook Toolbar for Firefox is a free extension for Firefox that lets you extend your Facebook experience into your everyday

<sup>215</sup> [http://developers.facebook.com/get\\_started.php?tab=tutorial](http://developers.facebook.com/get_started.php?tab=tutorial)

<sup>216</sup> [http://developers.facebook.com/get\\_started.php?tab=tutorial](http://developers.facebook.com/get_started.php?tab=tutorial)

<sup>217</sup> [http://developers.facebook.com/get\\_started.php?tab=tutorial](http://developers.facebook.com/get_started.php?tab=tutorial)

<sup>218</sup> [http://developers.facebook.com/get\\_started.php?tab=tutorial](http://developers.facebook.com/get_started.php?tab=tutorial)

<sup>219</sup> <http://developers.facebook.com/toolbar/>

browsing.” With the Facebook toolbar (Figure 53), users can check what their friends are up to by opening the Facebook friends sidebar that includes their profile pictures, statuses, and a possibility to interact with them on Facebook. The toolbar also allows searching Facebook content anywhere. Firefox 2 (or higher) and IE7 allow users also to add Facebook to their search boxes. Icons on the toolbar show how many new pokes, friend requests, messages, event invitations, and group invitations the user has. Automatic pop-ups will notify the user when his or her friends update their accounts or interact with the user on Facebook. The Share button lets the user import any web content to Facebook, share the current web page by sending it to Facebook friends, or post it to one’s profile.<sup>220</sup>



Figure 53 – Facebook Toolbar in the Firefox browser

The Facebook Connect<sup>221</sup> is the next evolution of the Facebook Platform, a single sign-on service that enables Facebook users to login to different partner web sites using their Facebook account. This is done with trusted authentication: Users thus bring their real identity information with them wherever they go but they can also control who can see which pieces of the information based on the same rules that they have set on Facebook. Facebook Connect enables sharing information and actions from current sites with their friends in Facebook via Facebook Feed, requests, and notifications (Figure 54).<sup>222</sup>



Figure 54 – Example of social distribution via Facebook Connect

Share buttons can be located on Facebook pages or on partner websites as instructed (Figure 55). If users click on a Share link next to any content, they can send that content in a message or post that content to their profile. A message will appear to recipient’s Inbox and behaves like any message sent from inside Facebook and it also includes the media attachments. If the shared contents are posted to user’s Profile, they appear in Posted Items box.<sup>223</sup>

As far as business models and monetizing are concerned, Facebook CEO Zuckerberg said in October 2008: *“What every great internet company has done is to figure out a way to make money that has to match to what they are doing on the site. I don’t think social networks can be monetized in the same way that search did. But on both sites people find information valuable. I’m pretty sure that we will find an analogous business model. But we are experimenting already. One group is very focused on targeting; another part is*

<sup>220</sup> <http://developers.facebook.com/toolbar/>

<sup>221</sup> <http://developers.facebook.com/connect.php>

<sup>222</sup> <http://developers.facebook.com/news.php?blog=1&story=198> and <http://developers.facebook.com/connect.php>

<sup>223</sup> <http://www.facebook.com/press/product.php>

*focused on social recommendation from your friends. In three years from now we have to figure out what the optimum model is. But that is not our primary focus today.”*<sup>224</sup>

The screenshot shows the 'Share Partners' page on Facebook. At the top, it says 'Share Partners' with a link 'Back to About: Facebook'. Below this is a paragraph: 'Add a link or button to your site to make it easier for your content to be shared on Facebook. Copy and paste the following code blocks into your site and replace <url> with the link you want to Share.'

There are four sections, each with a visual example and a code block:

- Icon Only:** Shows a small Facebook icon. The code block contains: `<script>function fbs_click() {u=location.href;t=document.title;window.open`
- Link Only:** Shows the text 'Share on Facebook'. The code block contains: `<script>function fbs_click() {u=location.href;t=document.title;window.open`
- Link and Icon:** Shows the text 'Share on Facebook' with a small Facebook icon. The code block contains: `<script>function fbs_click() {u=location.href;t=document.title;window.open`
- Share Button:** Shows a 'Share' button with a Facebook icon. The code block contains: `<script>function fbs_click() {u=location.href;t=document.title;window.open`

At the bottom, there are two questions: 'How do I make sure the Share Preview works?' and 'What is the blue bar that appears over my webpage? Is there a way to prevent it from appearing?'

Figure 55 – Instructions for adding Share feature to any web page content

<sup>224</sup> <http://faz-community.faz.net/blogs/netzkonom/archive/2008/10/08/mark-zuckerberg.aspx>

## 8 Evaluation of mashup tools for end-user development

Gammel and Storey (2008) have explored and compared the current mashup tools (including Popfly and Yahoo Pipes! that were introduced earlier) across six different themes (Levels of Abstraction, Learning Support, Community Support, Searchability, UI Design and Software Engineering Techniques).

They found that the mashup makers provide many features to support end-users, but there is still much room for further improvement. High learning barriers still remain, especially between different levels of abstraction and notations. Features of the compared products can be seen in Figure 56 (Gammel and Storey, 2008).

Feature	MP	YP	IMC	GME	SMC	IMM
Reuse of Complete Mashups	✓	✓	✓	✓	✓	✓
Mashup and Widget Parametrization	✓		✓			
Automatic Reuse of Data Extractors						✓
Programming-By-Example			✓			✓
Visual Dataflow Languages	✓	✓	✓			
Vis. Workflow/Process Orchestration Languages					✓	
Dialog-based Wiring of Widgets			✓			
Textual DSL Editors	✓			✓	✓	
Textual DSLs in Dialog Fields	✓	✓	✓		✓	✓
Extension APIs	✓		✓		✓	✓
Integration of Different Abstraction Levels	✓		✓		✓	✓
Tutorials, Help, API Documentation	✓	✓	✓	✓	✓	✓
Discussion Forums	✓	✓	✓	✓	✓	
Using Shared Artifacts as Examples	✓	✓	✓	✓	✓	✓
Context-Specific Suggestions	✓					
Sharing Mashups	✓	✓	✓	✓	✓	✓
Sharing Mashup Elements	✓	✓	✓		✓	✓
Tagging	✓	✓	✓			
Rating	✓	✓	✓			✓
<i>Discussion Forums</i>	✓	✓	✓	✓	✓	
Artifact-Centered Discussion	✓		✓			
Social Networks	✓					
Text-Based Search	✓	✓	✓			
Browsing Mashups by Structural Properties		✓				
Simple Categorization of Mashup Elements	✓	✓	✓		✓	
<i>Context-Specific Suggestions</i>	✓					
Automatical UI Generation		✓			✓	
Selecting & customizing UI	✓					
Visual UI composition			✓		✓	✓
Textual UI composition				✓		
Debugging Output	✓	✓	✓			
Version Control			✓		✓	

Figure 56 – Features of reviewed mashup tools (MP = Microsoft Popfly, YP = Yahoo Pipes!, IMC = IBM Mashup Center, GME = Google Mashup Editor, SMC = Serena Mashup Composer, IMM = INtel MashMaker)

Gammel and Storey (2008) suggest doing more to compare different notations for the same tasks, e.g. visual dataflow languages vs. dialog-based wiring, and using evaluation frameworks such as the Cognitive



Dimensions of Notations (Green et al. 1996) to analyze the strengths and weaknesses of such notations. Regarding UI development Grammel and Storey (2008) suggest an integration of different design mechanisms, such as using automatic generation to provide a starting point and then being able to modify it using a visual form editor, to combine the strengths of the different approaches.

Another important UI improvement could be providing a means for advanced information visualization, which could help to leverage the users perceptual system and further increase the usability of mashups. They also see programming-by-example good for website data extraction, data transformations and filtering, well as for drag & drop gestures in the UI. Using it for other tasks in mashup makers could be challenging. Grammel and Storey (2008) also note that an interesting feature that could support learning finding could be context based suggestions.

## 9 Summary and discussion

This review has concentrated on tools and technologies for end-user service composition. The main focus is on research and development of mashups tools, and on the use of end-user development approaches in those tools. The review starts with tools for use-time service composition: Mozilla Ubiquity and Intel MashMaker. These tools make it possible to combine other services with the page the user is currently examining. In the rest of the review the tools are categorized into main end-user programming approaches.

First, under the programming-by-example umbrella we discuss two tools that support programming by demonstration, Karma and Vegemite, and one tool that is based on example modification, d.mix. While Karma and Vegemite are use-time composition tools, they aim to generalize users' actions to create reusable services. The support for example modification in d.mix is twofold: via site-to-service maps d.mix helps users to understand the relationship of the presentation of a Web page to the API calls needed to produce the page, and through an active wiki it maintains a collaboration forum where users—from end-users to developers—can share their code.

Second, in the category of visual programming we discuss Microsoft Popfly and Yahoo! Pipes, both of which are sophisticated graphical editors for connecting service components visually together. Although the metaphor behind Yahoo Pipes is from the pipes of UNIX and the metaphor behind Popfly is from object-oriented programming, the two interfaces share a lot in common. Both have a library of ready-made components to which users are able to contribute new components. The interfaces to components basically consist of traditional form elements. In both, connections are drawn between the objects to depict the data flow. Both services aim to bring making mashups (and in case of Popfly, much more) to non-programmers. But despite the attractive visual programming interfaces these systems provide, they have failed to attract large numbers of users. One possibility is that they are still too difficult for end-users but on the other hand are too restricted to developers.

Third, under the title form-based creation we present two dashboard-like GUI assembly tools, iGoogle and Netvibes, that allow a creation of a service portal with a combination of form-based parameterization of widgets and drag-and-drop of widgets to a dashboard. The user sees an explanation and example of what the widget will be like, and then simply fills in the form data and has a ready-made widget. However, the scope of the types of applications possible to make with this approach is by necessity somewhat limited. Making a theme is even simpler, since instead of creating a new interface, a theme just changes the looks of an existing interface. iGoogle offers both a simple, form-based approach that brings skinning to everybody's fingertips in addition to a more complex but much more powerful approach that involves some coding, mirroring its approaches to user-created gadgets. Firefox's approach to skinning is similar to iGoogle's programming option. The popularity of iGoogle themes argues that a possibility of skinning is an important aspect of any such ecosystem.

Fourth, major environments for script-based service creation are studied, focusing on Firefox and Facebook. The idea is to give users the ability to program services and applications without having to set up C++ or Java environments. What users need to know is typically some combination of JavaScript, CSS, XML, and HTML. Firefox extensions represent this approach. All offer tutorials and examples from which to start developing one's own solutions. While the requirements here are not that high, some web programming knowledge is necessary. However, this has not prevented users from creating applications and services that have enriched the Internet experience of literally millions of other users

Lastly, there is a summary of the properties of mashup tools from the perspective of the support for end-user programming.

The *operation-centric* end-user programming paradigms, namely visual programming and script-based creation, require certain understanding of what mashups and other such services are all about. They require the user to have a clear concept of the service to be created, and this can mean that a complete newbie to programming and web technologies might have difficulties coming up with a working service. The *data-centric* approaches – such as use-time composition approaches, programming by demonstration, and form-

based creation – are easier for non-programmers but do not easily extend to the creation of arbitrary, robust, and reusable services. They tend to lead either to throwaway on-the-fly compositions or to restricted portal-type, parameterized widget collections.

All services reviewed offered different approaches to finding applications and services salient to a particular user. All offered ways to rate and comment the applications, although some brought the ratings to the foreground while some, for instance Popfly, did not emphasize them. The popularity of an application was measured in different ways. For instance, while Firefox add-ons could be arranged by weekly downloads, Facebook widgets in turn could be arranged by active users (who has used the widget within one month). No matter how popularity was measured, it seems an important sorting criterion for the applications.

Furthermore, all divided the applications to categories to help finding the relevant ones. Categories could be augmented with other information, as in Yahoo Pipes when using them for searching. Tags are also beginning to enter the picture although they were not utilized in any remarkable way or to any great effect in these services.

Recommenders were used at least to some extent in most repositories to help user find widgets or add-ons. Emphasis was on recommending other applications based on the current application, that is, on item-based and not user-based recommendations. In any case, making user-based recommendations is difficult if the platform does not require sign-in is. However, most do require it, and so there could be room for utilizing the wisdom of crowds to a greater degree. Now finding anything beyond the most popular can take some effort, and some worthy applications can simply be buried in the mass of applications.

Almost all tools have some form of social scaffolding or ecosystem support for service composition. The possibility to utilize the complementary capabilities of different users can provide interesting new approaches to end-user development. Even the most naïve end-users could rely on components, wrappers, connectors, etc. developed by users with more expertise or development capabilities. For expert users there would be a lot of opportunities to contribute to the ecosystem, show their skills, find new users for their tools, and so on. However, would that be enough to motivate expert users in the long run?

From the perspective of a platform provider, the services composed by users do not exist in a vacuum but are—or at least should be—part of its overall business plan. iGoogle is bringing social API to gadget makers to have social networks in iGoogle to compete with Facebook about the probable next big thing in online advertisement, namely social advertisement. Yahoo Pipes and Popfly offer less ways to monetize, and not surprisingly Popfly's future is clouded and Yahoo itself is going through major changes, leaving Yahoo Pipes's future in the air as the company appears poised to focus more on key strategic products, such as search ads<sup>225</sup>. Even the best platform cannot fly if the platform ends up dismantled due to financial reasons.

Dörner et al. (2008) have noted that Service Oriented Architectures (SOA) promise the development of a new generation of adaptive and adaptable software applications that will partially eliminate the need of transformation from users' requirements to UML or other modelling languages. This has moved the development tasks closer to the context of users, making SOA an interesting topic for end-user development (EUD) research. If SOA can be extended with structures for in-use-modifications (even beyond software technologies), it will be possible to design a new generation of user-adaptable systems regarding to Dörner et al. (2008).

The examples we have discussed have shown the different benefits and drawbacks of different kinds of End-User development systems. What could be the next generation system be a like? It could try to combine the different approaches described, allowing different EUD techniques used depending on the task at hand, time constraints and the level of expertise of the user. Context-awareness will be also a new hot topic in EUD and offers interesting possibilities as noted by Grammel and Storey (2008). Koripää et al. (2005) have presented a context framework and a tool called Context Studio for facilitating easy customization of context-aware features into existing mobile terminal applications. Such tools require a vocabulary model that provides the

---

<sup>225</sup> [http://www.businessweek.com/technology/content/feb2009/tc20090226\\_871329.htm?chan=technology\\_technology+index+page\\_top+stories](http://www.businessweek.com/technology/content/feb2009/tc20090226_871329.htm?chan=technology_technology+index+page_top+stories)

representation of contexts and actions, and interaction for creating the rules for user activities. Integrating context-awareness to a system that combines the different end-user development paradigms in a usable and meaningful way is challenge for the future.

## 10 References

- Anderson, C. and Andersson, M.P. (2006). *The Long Tail*. Hyperion.
- Barrett, R., Maglio, P. P., and Kellem, D. C. (1997). How to personalize the Web. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI'97)*, ACM Press, 75–82.
- Blackwell, A. F. (2002). What is programming. In *Proc. 14th Workshop of the Psychology of Programming Interest Group (PPIG'02)*, 204–218.
- Bolin, M., Webber, M., Rha, P., Wilson, T., and Miller, R. C. (2005). Automation and customization of rendered web pages. In *Proc. 18th Annual ACM Symposium on User Interface Software and Technology (UIST'05)*, ACM Press, 163–172.
- Burnett, M., Cook, C., Pendse, O., Rothermel, G., Summet, J., and Wallace, C. (2003). End-user software engineering with assertions in the spreadsheet paradigm. In *Proc. 25th International Conference on Software Engineering (ICSE'03)*, ACM Press, 93–103.
- Dörner, C., Pipek, V., Weber, M., and Wulf, V. (2008). End-user development: new challenges for service oriented architectures. In *Proc. 4th International Workshop on End-User Software Engineering (WEUSE'08)*, ACM Press, 71–75.
- Ennals, R., and Garofalakis, M. (2007). MashMaker: Mashups for the Masses. In *Proc. 2007 ACM SIGMOD International Conference on Management of Data (SIGMOD'07)*, ACM Press, 1116–1118.
- Fujima, J., Yoshihara, S., and Tanaka, Y. (2007). Web application orchestration using Excel. In *Proc. IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, IEEE Computer Society, 743–749.
- Grammel, L. and Storey, M.-A. (2008). An end user perspective on mashup makers. [http://lars.grammel.googlepages.com/paper\\_mashup\\_makers.pdf](http://lars.grammel.googlepages.com/paper_mashup_makers.pdf)
- Green, T. and Petre, M. (1996). Usability analysis of visual programming environments: a cognitive dimensions framework. *Journal of Visual Languages and Computing* 7, 2, 131–174.
- Halbert, D. C. (1984). *Programming by Example*. Doctoral Thesis. University of California, Berkeley.
- Hartmann, B., Wu, L., Collins, K., and Klemmer, S. R. (2007) Programming by a sample: rapidly creating web applications with d.mix. In *Proc. 20th Annual ACM Symposium on User Interface Software and Technology (UIST'07)*, ACM Press, 241–250.
- Hoyer, V., and Fischer, M. (2008). Market overview of enterprise mashup tools. In *Proc. 6th International Conference on Service Oriented Computing (ICOSC 2008)*, LNCS 5364, Springer, 708–721.
- Korpipää, P., Malm, E., Salminen, I., Rantakokko, T., Kyllönen, V., and Känsälä, I. (2005). Context management for end user development of context-aware applications. In *Proc. 6th International Conference on Mobile Data Management (MDM'05)*, ACM Press, 304–308.
- Lieberman, H., Paternó, F., Klann, M. and Wulf, V. (2005). End-user development: an emerging paradigm. In H. Lieberman, F. Paternó, and V. Wulf (Eds.), *End User Development*, Kluwer, 9–16.
- Lin, J., Wong, J., Nichols, J., Cypher, A., and Lau, T. A. (2008). End-user programming of mashups with vegemite. In *Proc. 13th International Conference on Intelligent User Interfaces (IUI'09)*, ACM Press, 97–106.

- Little, G., Lau, T. A., Cypher, A., Lin, J., Haber, E. M., and Kandogan, E. (2007). Koala: capture, share, automate, personalize business processes on the web. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, ACM Press, 943–946.
- MacLean, A., Carter, K., Lövsstrand, L., and Moran, T. (1990). User-tailorable systems: pressing the issues with buttons. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI'90)*, ACM Press, 175–182.
- Myers, B. A. (1986). Visual programming, programming by example, and program visualization: a taxonomy. *SIGCHI Bulletin* 17, 4, 59–66.
- Nardi, B. A. (1993). *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press.
- Repenning, A. (1991). Creating user interfaces with agentsheets. In *Proc. 1991 IEEE Symposium on Applied Computing*, IEEE Computer Society, 191–196.
- Repenning, A. (1993). *Agentsheets: a Tool for Building Domain-Oriented Dynamic, Visual Environments*. Doctoral Thesis. University of Colorado at Boulder.
- Smith D. (1975). *Pygmalion: A Creative Programming Environment*. Report No. STAN-CS-75-499, Department of Computer Science, Stanford University.
- Wong, J., and Hong, J. I. (2007). Making mashups with Marmite: towards end-user programming for the web. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, ACM Press, 1435–1444.
- Wong, J. and Hong, J. (2008). What do we “mashup” when we make mashups? In *Proc. 4th International Workshop on End-User Software Engineering (WEUSE'08)*, ACM Press, 35–39.