indi-2010-05-011

Mashup Technology for Research and Education

Project	: SURFworks / Collaboration Infrastructure
Project Year	: 2010
Project Manager	: Marieke de Wit
Author(s)	: Arjan Peddemors
Completion Date	: 20 05 2010
Version	: 1.0

Summary

In the last few years, a number of tools have become available that support the creation of web applications by mixing and matching data sources and processing functionality available through the internet. Typically, these 'Mashup' tools can be used without extensive programming skills and offer a graphical user interface to create new applications, and, as such, make web application development possible for a wide group of individuals. Additionally, by building (parts of) applications, users generate functionality that can be reused by others, which can be considered a form of collaboration.
This report provides an analysis of the applicability and usefulness of Mashup tools for SURFnet users, focusing primarily on scientific researchers. It gives an overview of existing tools, including tools oriented at supporting 'scientific workflow', and provides conclusions and recommendations towards their usage in the SURFnet context.

Colophon

Programme line	: Technology Scouting
Part	: Collaboration Infrastructure
Activity	: 4.2
Deliverable	: Mashups
Access rights	: Public
External party	: Novay

This project was made possible by the support of SURF, the collaborative organisation for higher education institutes and research institutes aimed at breakthrough innovations in ICT. More information on SURF is available on the website www.surf.nl.

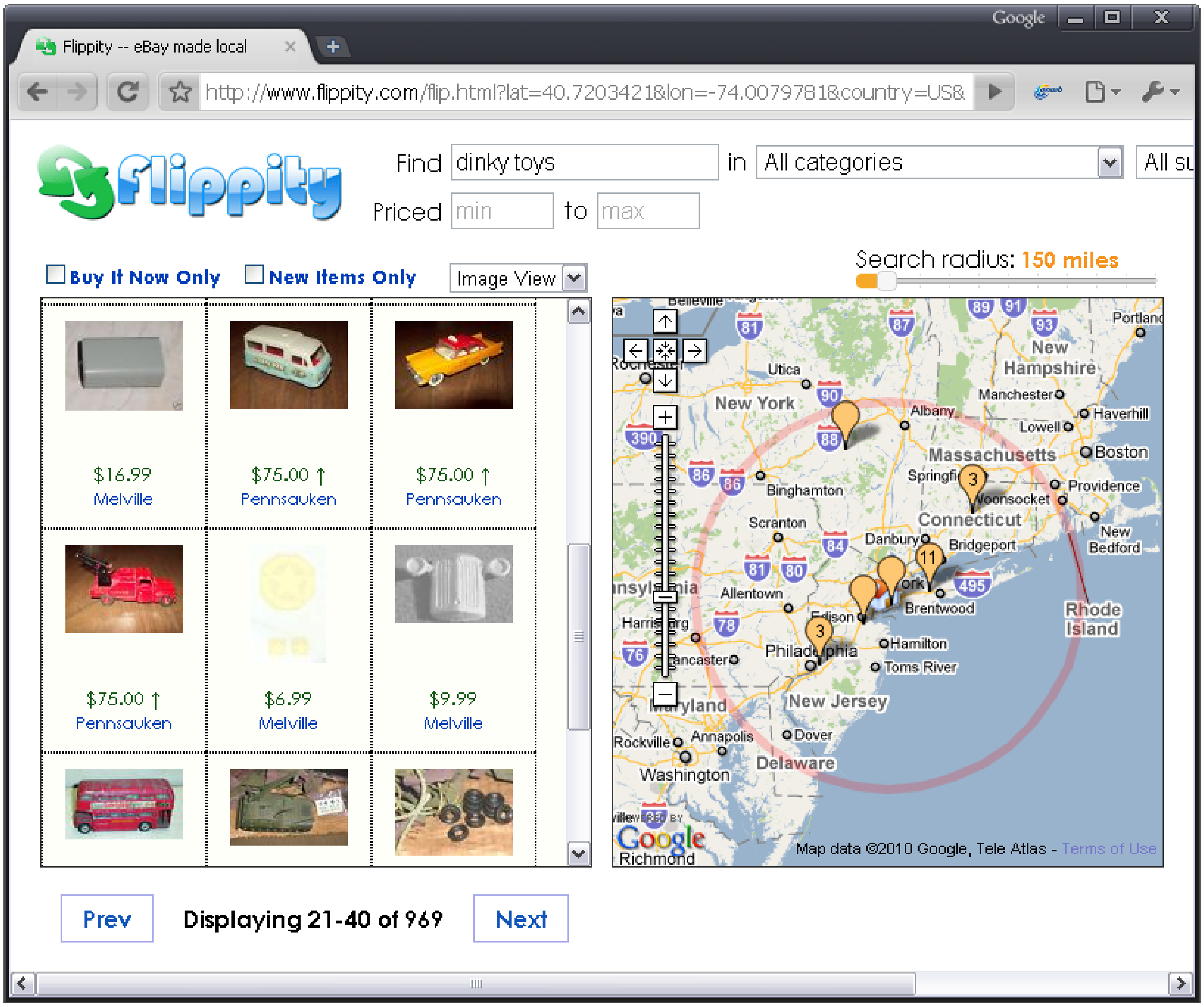
5 things you should know about Mashup technology for research and education

Scenario	A researcher is analyzing a problem by extracting information from experimental data, using Mashup technology. This data is available from multiple sources on the internet, while the data handling is done by online processing services. The researcher shares the analysis methods as well as the results with others in his network.
What is it?	Mashup technology is used to mix and match in an easy manner various resources available through the internet. It provides facilities to make the development of new web applications possible for individuals without a programming background.
For who is it?	Being a generic technology, it can be used (and is used) by individuals and groups with a broad background. In this document, we focus on the usability of currently available products from the perspective of SURFnet user (in particular researchers).
How does it work?	Mashup platforms provide functionality to quickly put together available resources to form a new application (data collection functionality, filtering functionality, etc.). Additionally, they provide tools to make this easy for users without programming skills.
What can you do with it?	In general, quickly build new web applications using existing online resources. More specifically focused on researchers, Mashup technology may be used to support daily tasks and activities of researchers and may help them to work together online.
More information	Marieke de Wit marieke.dewit@surfnet.nl

Table of Content

1. Introduction

A *Mashup* is a web application that is created by mixing and matching two or more web resources in an easy and fast way. It can be designed as a web page with embedded scripts that utilizes the resources from within a browser environment. Alternatively, it can be designed as a hosted application where resources are joined in a server environment, and to which a user connects through his browser. The resources may be widely variable, ranging from geographical map data, to language translators, to generators of instant messages. In general, any web resource that is accessible through a set of common web protocols (such as HTTP, SOAP, REST, etc.) is eligible to be used in a Mashup.



Flippity -- eBay made local

http://www.flippity.com/flip.html?lat=40.720342l&lon=-74.0079781&country=US&

Find: dinky toys in All categories

Priced: min to max

□ Buy It Now Only □ New Items Only Image View

Search radius: 150 miles

$16.99 Melville
$75.00 ↑ Pennsauken
$75.00 ↑ Pennsauken
$75.00 ↑ Pennsauken
$6.99 Melville
$9.99 Melville

Prev Displaying 21-40 of 969 Next

Figure 1: example of a Mashup application, combining Ebay data with Google map data to show a local Ebay listing

An example of a Mashup application is Flippity [2]. This Mashup combines data from two web resources: the Ebay auction and shopping website, and the Google Maps web mapping service (see Figure 1). It allows users to search for Ebay offers within a confined geographical area.

The term ‘Mashup’ is in use since the end of 2005, covering a range of technologies that have evolved over time with the rise of the web. In particular, the wide deployment of scripting languages for rich browser-side functionality, and standard protocols and formats to access resources over the internet has enabled the creation of Mashups.

Typically, developers of Mashups use tools which support the selection of existing resources and the composition of a new application by combining these resources. Such *Mashup tools* aim at making this process easy and fast, so that by using these tools, even individuals without programming skills are able to build Mashups. Often, Mashup tools are hosted services that are accessed through a web browser, although some Mashup developer environments also are (partially) based on standalone client-side programs. Most tools provide a facility to add new resources or complete Mashups to the tool environment, which allows other developers to reuse these components inside their Mashups. As such, developers using the same tool form a community that generates and reuses functionality. As we will see in the next section, it is possible to identify different kinds of users of Mashup tools.

Within the SURFnet context, a number of different *user groups* can be identified, which all may benefit from using Mashup tools. As national research and education network, SURFnet’s main users are *scientists, educators, support staff*, and *students*. For all these groups, web applications and web data sources play an important role in their daily activities. For example, scientists may benefit from combining online datasets into a new dataset that can be used as input for further analysis. Educators may be supported to generate new teaching material by joining existing (online) educational material and data sources, and use these materials in their classes. Students may work on assignments by combining and processing existing online resources in novel ways. Obviously, this only applies to activities that can be accomplished using web resources, ruling out data and processing that is available only on local computing systems. With the rise of cloud computing, however, there is a clear trend that much data storage and the processing of this data is performed online, which may make Mashup technology applicable to a wider range of daily activities of these user groups.

Mashup technology can be used to let groups of people work together at various levels. *Making Mashups* is a form of loose cooperation, where a developer uses components and data created by others. *Using Mashups* is a cooperation between the developer and the consumer(s) of the application, where a single person may have both a developer and a consumer role (multiple people inside a community using each other’s Mashups). Finally, showing the *Mashup structure and workflow* gives others insight into how results are obtained – something of high importance when analyzing, explaining or solving a problem. These cooperation aspects make Mashup technology interesting to consider in the context of the ‘Collaboration Infrastructure’ work of SURFnet, which focuses on providing a new platform that supports innovative ways to collaborate amongst SURFnet users.

The objectives of this report are twofold. We aim to provide an overview of a number of interesting Mashup tools that are expected to support collaboration between SURFnet users. For the selection of the discussed tools, we take a broad view on possible alternatives. Furthermore, our objective is to give an analysis of how these products map to the cooperation requirements of SURFnet users.

This report is organized as follows. Section 2 gives an analysis of the usage of Mashup technology in the SURFnet context and provides high-level requirements. Section 3 discusses identified products. Section 4 indicates the mapping from requirements to products and describes a proposition of how Mashup technology fits in a SURFnet collaboration infrastructure. Section 5 wraps up with conclusions and recommendations.

2. Analysis and Requirements

In this section, we provide a more detailed analysis on how Mashups are created and describe the different roles in this process. We argue that scientific researchers are an important group of users that are likely to benefit most from Mashup technology offered in a SURFnet context. Consequently, we discuss high-level requirements for using Mashup technologies in the daily activities of members of this group.

When considering functionality in Mashup applications, it is possible to discern a number of distinctive layers [4][5]. Components used within Mashups are positioned inside these layers, and are created by developers with different roles. The following layers are identified (in bottom to top order):

- **Resource Layer**. This layer provides the core building blocks of Mashup applications, being either data sources or processing functionality. Components in this layer are created by **Developers** with a technical / programming background.
- **Widget Layer**. Components in this layer use resources layer components to support task- or application oriented functionality. This could, for instance, be a filtering operation that selects elements from a particular, larger data set. These components are created by **Expert Users** or **Consultants** who have in-depth knowledge about the application domain where the widgets will be applied.
- **Mashup Layer**. In this layer, the actual Mashup is created from components at the lower layers. The Mashup is created (and consumed) by **End Users** who do not need to have programming skills, but who rely on intuitive (visual) tooling to do so.

The different layers and roles are depicted in Figure 2, in a Mashup stack.

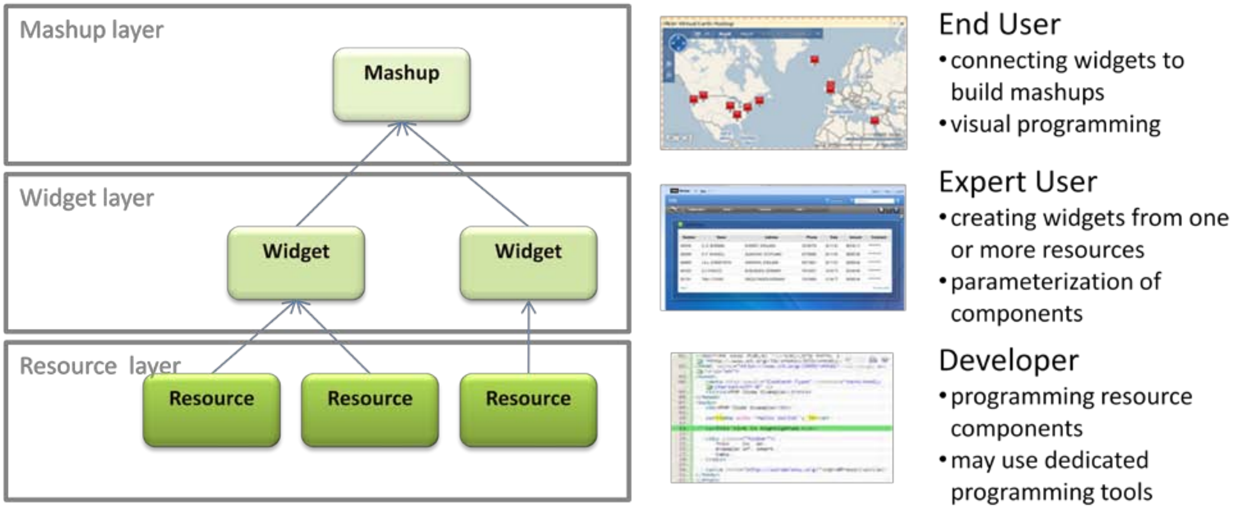
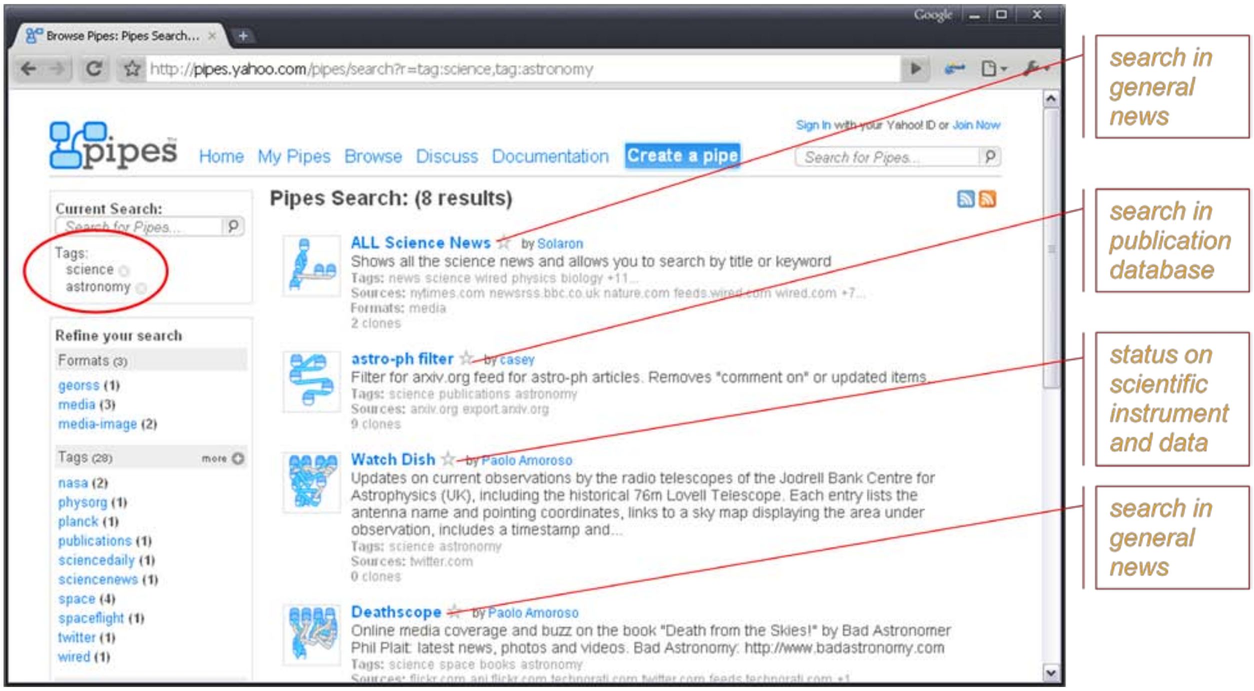

Mashup layer
Mashup
Widget layer
Widget
Widget
Resource layer
Resource
Resource
Resource
End User
•connecting widgets to build mashups
•visual programming
Expert User
•creating widgets from one or more resources
•parameterization of components
Developer
•programming resource components
•may use dedicated programming tools

Figure 2: Mashup components are ordered in layers, with different roles responsible for creating these components.

The higher layer functionality inside Mashups is usually built in a visual programming environment, where functionality is created by manipulating elements graphically (rather than in a programming language). There are currently two types of Mashup platforms: *General Purpose platforms* that offer broadly applicable functionality, and *Enterprise Mashup platforms* that offer functionality dedicated to enterprise environments. We will consider both types of platforms in this report.

When considering the user groups as identified in the previous section (scientists, educators, staff, students), we observe that their jobs are very diverse, even for members within the same group. First year medical students are most likely engaged in different activities and tasks than final year physics students. Also scientists from different domains at first glance seem to have rather different activities. Scientists as a group, however, are an interesting target group to consider for collaboration support: they basically follow the same ‘scientific method’ of collecting data in observations and experiments, and formulating and testing of hypotheses. When zooming in on aspects of this process, different scientists may use domain specific techniques, for instance when analyzing data that is very specific for their own research domain, but the process as a whole consists of a limited number of clearly distinctive steps valid for a wide range of domains. Supporting this process with collaboration tools is likely to benefit a large number of SURFnet users. Additionally, scientists may need facilities that are supported in the broader SURFnet context, such as high-performance computing resources, data storage, and different kinds of authorization and authentication (to access distributed resources). These aspects lead to the conclusion that scientists are the most interesting target group to consider in this document, and that the requirements and tool mapping are done from that group’s perspective.

A first question that arises when focusing on supporting collaboration amongst scientists using Mashup technology is: are there any examples of the usage of this technology by scientists? And if so, how is it used? To answer this question we did a quick scan on a leading, general purpose Mashup platform: Yahoo! Pipes [13]. Our impression (April 2010) is that there are very few (if any) Mashups that are supporting core steps in the scientific process. Many applications with a ‘scientific touch’ are aggregating and filtering general news sources to display news related to science in general or related to a dedicated scientific domain (see the search results depicted in Figure 3). Also, some applications support searching in publication databases for specific research domain. We did not encounter scientific data analysis on data obtained through a scientific experiment. One example that is remotely related is shown in Figure 4 and Figure 5, which provides the status of a scientific instrument (a radio telescope).



http://pipes.yahoo.com/pipes/search?r=tag:science,tag:astronomy

Tags:
science
astronomy

georss (1)
media (3)
media-image (2)

nasa (2)
physorg (1)
planck (1)
publications (1)
sciencedaily (1)
sciencenews (1)
space (4)
spaceflight (1)
twitter (1)
wired (1)

Pipes Search: (8 results)

ALL Science News ☆ by Solaron
Shows all the science news and allows you to search by title or keyword
Tags: news science wired physics biology +11...
Sources: nytimes.com newsrss.bbc.co.uk nature.com feeds.wired.com wired.com +7...
Formats: media
2 clones

astro-ph filter ☆ by casey
Filter for arxiv.org feed for astro-ph articles. Removes "comment on" or updated items.
Tags: science publications astronomy
Sources: arxiv.org export.arxiv.org
9 clones

Watch Dish ☆ by Paolo Amoroso
Updates on current observations by the radio telescopes of the Jodrell Bank Centre for Astrophysics (UK), including the historical 76m Lovell Telescope. Each entry lists the antenna name and pointing coordinates, links to a sky map displaying the area under observation, includes a timestamp and...
Tags: science astronomy
Sources: twitter.com
0 clones

Deathscope ☆ by Paolo Amoroso
Online media coverage and buzz on the book "Death from the Skies!" by Bad Astronomer Phil Plait: latest news, photos and videos. Bad Astronomy: http://www.badastronomy.com
Tags: science space books astronomy
Sources: flickr.com api.flickr.com technorati.com twitter.com feeds.technorati.com +1

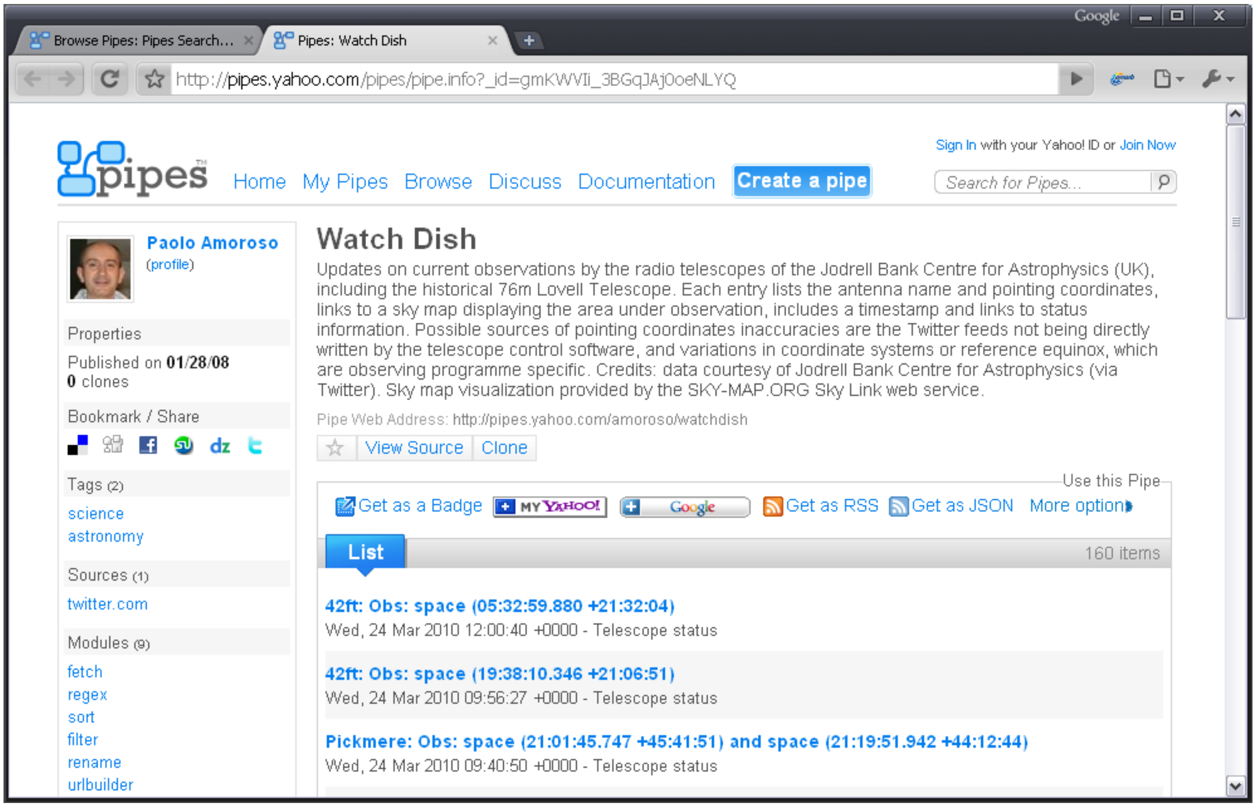
search in general news

search in publication database

status on scientific instrument and data

search in general news

Figure 3: search in the Yahoo! Pipes directory using the tags 'science' and 'astronomy'



Paolo Amoroso
(profile)

Properties

Published on **01/28/08**
0 clones

Tags (2)
science
astronomy

Sources (1)
twitter.com

Modules (9)
fetch
regex
sort
filter
rename
urlbuilder

Watch Dish

Updates on current observations by the radio telescopes of the Jodrell Bank Centre for Astrophysics (UK), including the historical 76m Lovell Telescope. Each entry lists the antenna name and pointing coordinates, links to a sky map displaying the area under observation, includes a timestamp and links to status information. Possible sources of pointing coordinates inaccuracies are the Twitter feeds not being directly written by the telescope control software, and variations in coordinate systems or reference equinox, which are observing programme specific. Credits: data courtesy of Jodrell Bank Centre for Astrophysics (via Twitter). Sky map visualization provided by the SKY-MAP.ORG Sky Link web service.

Pipe Web Address: http://pipes.yahoo.com/amoroso/watchdish

160 items

42ft: Obs: space (05:32:59.880 +21:32:04)
Wed, 24 Mar 2010 12:00:40 +0000 - Telescope status

42ft: Obs: space (19:38:10.346 +21:06:51)
Wed, 24 Mar 2010 09:56:27 +0000 - Telescope status

Pickmere: Obs: space (21:01:45.747 +45:41:51) and space (21:19:51.942 +44:12:44)
Wed, 24 Mar 2010 09:40:50 +0000 - Telescope status

Figure 4: the 'Watch Dish' application in the Yahoo! Pipes platform, indicating the current observation parameters for a radio telescope

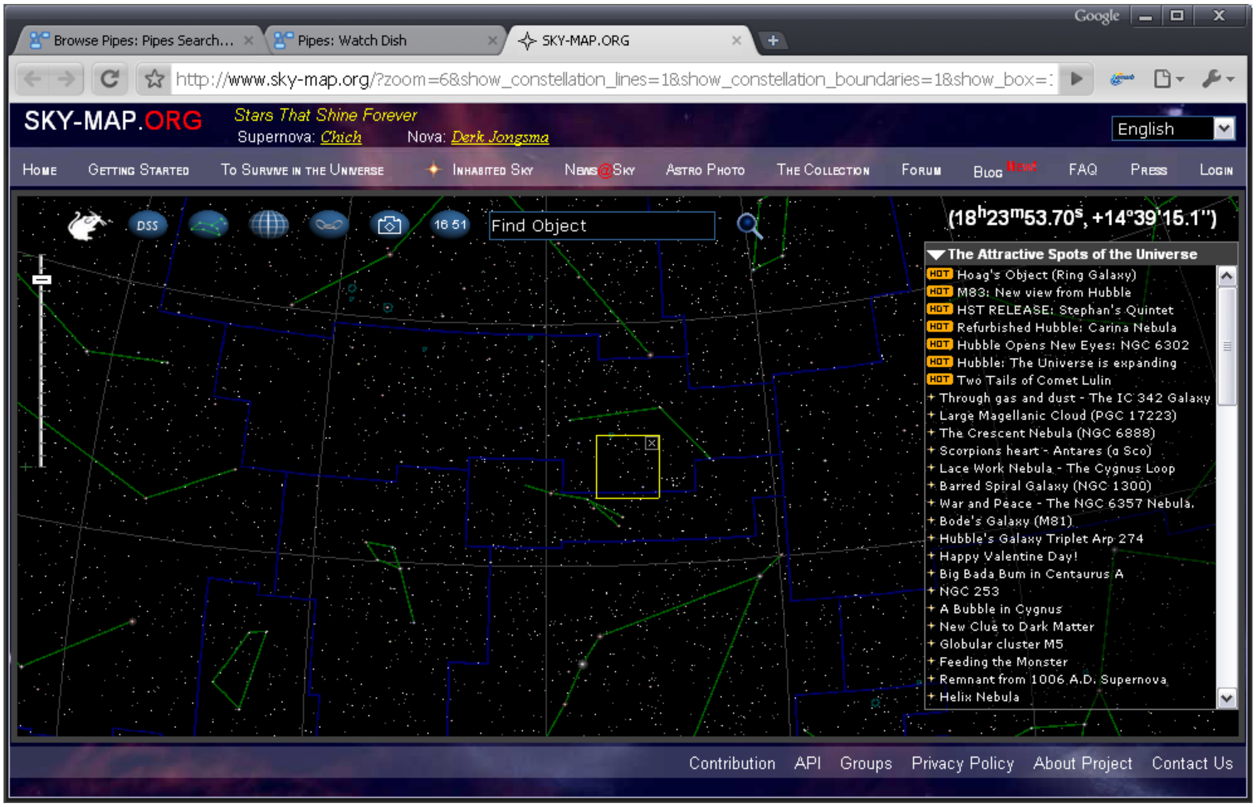

Browse Pipes: Pipes Search...
Pipes: Watch Dish
SKY-MAP.ORG
Google
http://www.sky-map.org/?zoom=6&show_constellation_lines=1&show_constellation_boundaries=1&show_box=1
SKY-MAP.ORG
Stars That Shine Forever
Supernova: Chich
Nova: Derk Jongsma
English
Home
Getting Started
To Survive in the Universe
Inhabited Sky
News@Sky
Astro Photo
The Collection
Forum
Blog
FAQ
Press
Login
DSS
16 51
Find Object
(18h23m53.70s, +14°39'15.1")
The Attractive Spots of the Universe
HOT Hoag's Object (Ring Galaxy)
HOT M83: New view from Hubble
HOT HST RELEASE: Stephan's Quintet
HOT Refurbished Hubble: Carina Nebula
HOT Hubble Opens New Eyes: NGC 6302
HOT Hubble: The Universe is expanding
HOT Two Tails of Comet Lulin
Through gas and dust - The IC 342 Galaxy
Large Magellanic Cloud (PGC 17223)
The Crescent Nebula (NGC 6888)
Scorpions heart - Antares (α Sco)
Lace Work Nebula - The Cygnus Loop
Barred Spiral Galaxy (NGC 1300)
War and Peace - The NGC 6357 Nebula.
Bode's Galaxy (M81)
Hubble's Galaxy Triplet Arp 274
Happy Valentine Day!
Big Bada Bum in Centaurus A
NGC 253
A Bubble in Cygnus
New Clue to Dark Matter
Globular cluster M5
Feeding the Monster
Remnant from 1006 A.D. Supernova
Helix Nebula
Contribution
API
Groups
Privacy Policy
About Project
Contact Us

Figure 5: sky map of the current view of the radio telescope in the Yahoo! Pipes 'Watch Dish' application

While scanning for available 'Scientific Mashups', we encountered a category of tools called 'scientific workflow' tools that specifically target at supporting researchers in their daily activities. Although most of these tools do not provide a web based GUI (running inside a web browser) and as such do not qualify as Mashup platforms, we do consider these tools in this report because of their explicit goal of supporting core steps in the scientific process (contrary to Mashup platforms).

We now define a number of high level requirements for Mashup environments to support scientists. A Mashup platform must:

- Support a common set of interaction protocols / data exchange protocols / data formats based on open standards such as REST, SOAP, XML, RSS, … This provides access to a wide range of resources on the internet, and allows interoperability between components.
- Be able to handle and process large amounts of data, because some experiments generate large amount of data and require extensive processing steps.
- Support domain specific tools and data formats, because individual research domains often use specific data analysis algorithms on domain specific experimental data.
- Deal with variety in users (Mashup creators), because the types of analysis and processing may vary considerably between scientific domains.
- Deal with variety in audience / public (Mashup users), again to support different scientific domains, but also to make results accessible to those indirectly involved in the scientific process such as publishers.
- Flexible authentication, to control access to processing and data resources. This is important in cases where experimental data contains sensitive information (such as medical data).

Additionally, when Mashup execution uses considerable computing resources, it supports the assignment of resources on individual or group basis.

- Support a ‘network effect’, where the value of the platform grows with every new user added. This helps to make the platform relevant for a large number of users.
- Be able to trace the scientific process, such that researchers can verify the results reported by their peers, using the Mashup platform.
- Support extensibility and open interfaces and preferably prevent vendor lock-in. This makes the platform less dependent on the vendors supplying the platform technology.

3. Product Overview

In this section, we discuss products that belong to three different categories identified in the previous section: 1) general purpose Mashup platforms, 2) enterprise Mashup platforms, and 3) scientific workflow environments. We use these product descriptions in the next section to determine which tools best support the identified requirements.

Currently, a considerable number of Mashup tools and platforms with varying levels of functionality are available. Here, we focus on those products that have a prominent position within their category and that represent well what can be expected from products in that category. We selected a balanced set based on an initial broad scan of Mashup products using information from papers and web search; we realize, however, that by having a limited set we inevitably exclude some tools and platforms that also have interesting features. Furthermore, we note that the selection reflects the situation of the first half of 2010, and that it is highly likely that other products will be more prominent in the future. Some tools that once were regarded as leading (such as Microsoft Popfly and the Google Mashup editor), are now discontinued, which shows that the Mashup domain is highly dynamic. Additionally, we note that, although we do not discuss them here, there are various sources on the web that provide information about technologies that support Mashup developers. An example is the ProgrammableWeb [10], which provides a directory of APIs and full Mashups, offers a channel on Mashup news, and supports a community of members.

General purpose Mashup environments, also sometimes called 'consumer Mashup environments', support a wide audience in building and using Mashups. They focus mostly on making development and deployment easy, supporting mostly the end-user role identified in the previous section. Typically, these environments are free websites that the user connects to with a browser, to build and use Mashups. Creators of Mashups are supported with visual programming tools and a repository of existing components and data sources.

Enterprise Mashup platforms have many of the features available in general purpose platforms, but provide more explicit support for users in corporate environments. Contrary to general purpose platforms, they usually run on hosts inside a company's network: the platform software is installed on local hardware. Components in an Enterprise Mashup environment offers functionality typically needed in corporate environments. Due to the high level of control over software- and hardware resources, these environments can be tailored to meet the needs of a specific corporate setting. Additionally, access to the environment can be restricted, so that sensitive data does not leave the corporate network. Enterprise Mashup platforms offer a broader range of support for the different developer roles than general purpose platforms, by providing different developer environments (requiring variable skills).

Scientific Workflow environments are quite different from mainstream Mashup platforms, because 1) they focus on supporting a dedicated group of users (scientists), 2) they provide a user interfaces which is not web-based, but goes through a dedicated client application (to be installed on the user's computer), and 3) they take into account that individual processing steps within a workflow (such as extracting information from multiple data sources) may use considerable computing resources and time. Scientific workflow environments have their roots in the research domain, as tools build by scientists for scientists. They do have, like regular Mashup environments, extensive functionality to pull in data from the web and use functionality and resources available online.

The five products considered here in more detail are the following:

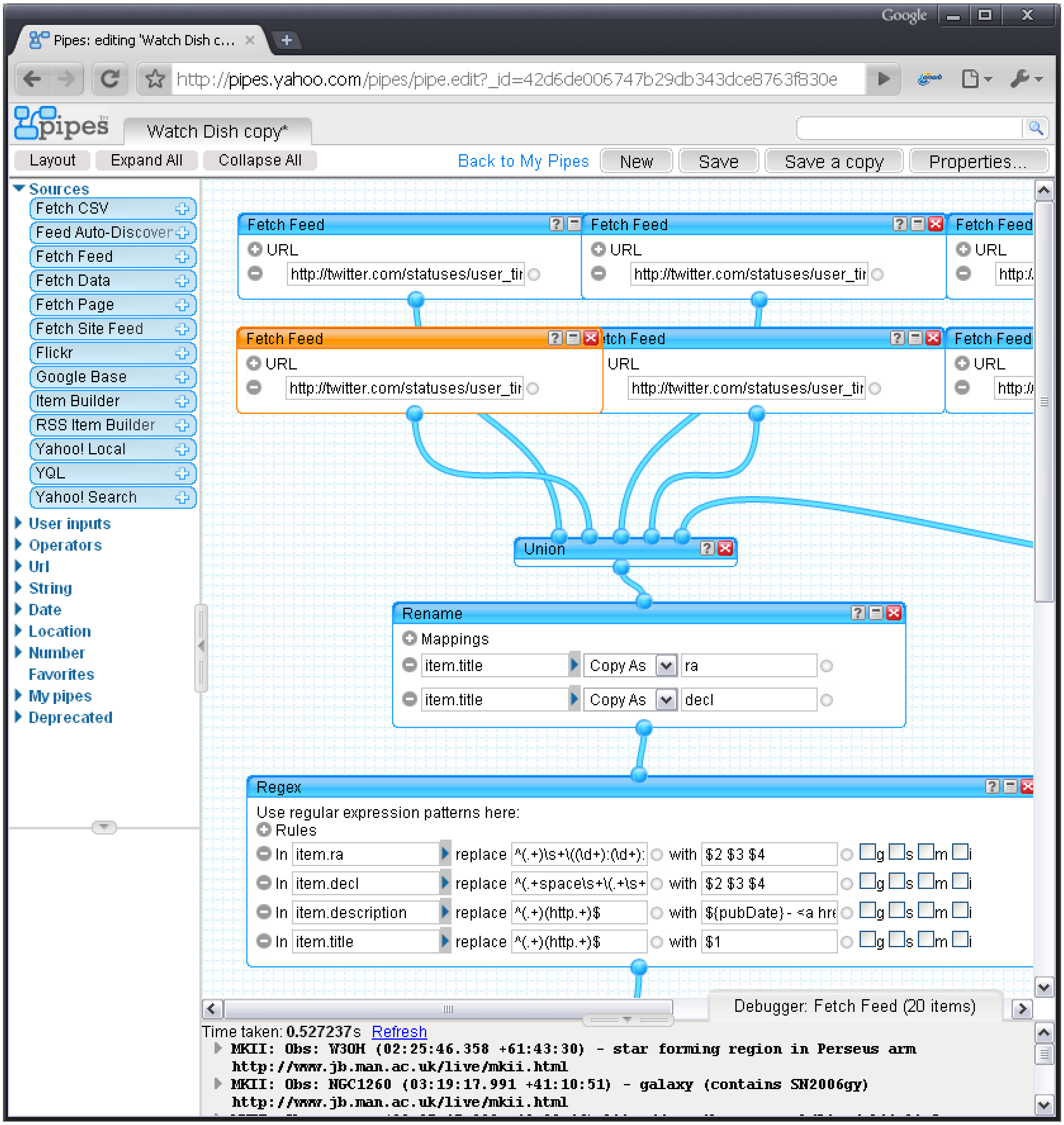
- **Yahoo! Pipes** (category: *general purpose*)
- **JackBe** (category: *enterprise*)
- **IBM Mashup Center** (category: *enterprise*)
- **Taverna** (category: *scientific workflow*)
- **VisTrail** (category: *scientific workflow*)

In the general purpose category we consider one product, while for the other categories we look at two products. The reason for selecting only one general purpose product is that there are very few of them that offer broad functionality, and Yahoo! Pipes is a very strong leader in this category.

3.1. Yahoo! Pipes

Yahoo! Pipes [13] is a general purpose Mashup environment, offered as a free online service by Yahoo!. It supports developing Mashup applications with a visual programming editor, using a collection of building blocks. Furthermore, it hosts the Mashups created with this editor, i.e., it provides the connectivity and computing resources to run Mashups. The site has a directory of existing Mashups (in the order of tens of thousands entries), which can be searched based on keywords, tags, or used components. Mashups can be used without restriction and do not require an account. The creation of Mashups requires a Yahoo! account. An example of a Mashup created with Yahoo! Pipes is described in Section 2 (see also Figure 4).

The Yahoo! Pipes editor offers different types of components to build Mashups. The available categories are: Sources, User Inputs, Operators, URL, String, Date, Location, and Number. This set of components is supplied by the platform and cannot be extended by users, which means that special (non-generic) operations cannot be executed by Yahoo! Pipes Mashups. Figure 6 shows an example of an editing session, using a copy of the 'Watch Dish' application (Section 2). A Mashup is created by pulling components from the panel on the left onto the main canvas in the window, set parameters on these components, and then connect the output from one component to the input of another component (blue 'pipes'). Input and output must be compatible, otherwise a connection cannot be made. The debugger (bottom part of the window) allows inspecting the output of components as they operate within the Mashup. When editing is finished the Mashup can be published on the Yahoo! Pipes site. Additionally, the Mashup output may be added to other websites.



Fetch Feed URL: http://twitter.com/statuses/user_tir

Rename – Mappings:

item.title | Copy As | ra

item.title | Copy As | decl

Regex – Use regular expression patterns here:

In item.ra replace ^(.+)\s+\((\d+):(\d+): with $2 $3 $4 □g □s □m □i

In item.decl replace ^(.+space\s+\(.+\s+ with $2 $3 $4 □g □s □m □i

In item.description replace ^(.+)(http.+)$ with ${pubDate} - <a hre □g □s □m □i

In item.title replace ^(.+)(http.+)$ with $1 □g □s □m □i

Time taken: 0.527237s Refresh

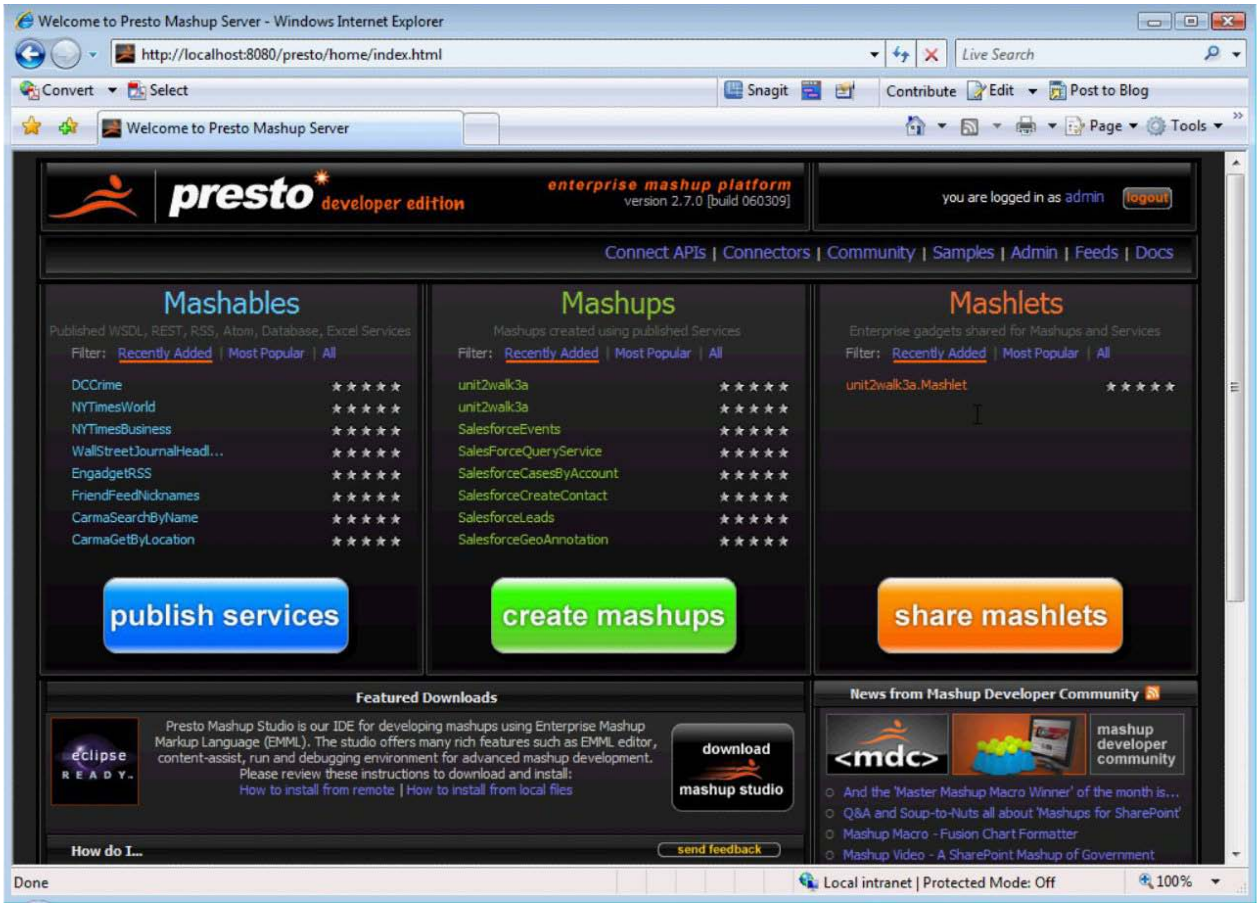
```
MKII: Obs: W3OH (02:25:46.358 +61:43:30) - star forming region in Perseus arm
http://www.jb.man.ac.uk/live/mkii.html
MKII: Obs: NGC1260 (03:19:17.991 +41:10:51) - galaxy (contains SN2006gy)
http://www.jb.man.ac.uk/live/mkii.html
```

Figure 6: editing a copy of the 'Watch Dish' Mashup inside the Yahoo! Pipes editor. Building blocks (components) are added to the main canvas from the panel on the left. The output from one component is fed into the input of another using connectors. The debug panel at the bottom shows the output of the selected (orange) 'fetch feed' source component.

Mashups in the Yahoo! Pipes environment have limited resources to run. They can be executed a maximum of 200 times per 10 minutes; when exceeding this limit, an error is generated. Also, no control is possible over the amount of processing power assigned to a Mashup. It is not clear if there is a limit in the amount of input data a Yahoo! Pipes Mashup can handle.

3.2. JackBe

JackBe [7] offers an enterprise Mashup platform, called Presto, in the form of a software package to be installed on local (corporate) hardware, and in the form of a cloud service running within the Amazon EC2 cloud environment [1]. The standalone software was released at the end of 2007 and can be considered relatively mature, while the cloud service was released very recently in April 2010. The cloud service is essentially the same product as the standalone version, with restrictions in terms of access to intranet resources, and with extensions to let developers more easily shares their components and Mashups. The JackBe site reports a customer base of more than 50 large enterprises and government organizations.


Welcome to Presto Mashup Server - Windows Internet Explorer
http://localhost:8080/presto/home/index.html
Live Search
Convert
Select
Snagit
Contribute
Edit
Post to Blog
Welcome to Presto Mashup Server
Page
Tools
presto developer edition
enterprise mashup platform
version 2.7.0 [build 060309]
you are logged in as admin
logout
Connect APIs | Connectors | Community | Samples | Admin | Feeds | Docs
Mashables
Published WSDL, REST, RSS, Atom, Database, Excel Services
Filter: Recently Added | Most Popular | All
DCCrime
NYTimesWorld
NYTimesBusiness
WallStreetJournalHeadl...
EngadgetRSS
FriendFeedNicknames
CarmaSearchByName
CarmaGetByLocation
Mashups
Mashups created using published Services
Filter: Recently Added | Most Popular | All
unit2walk3a
unit2walk3a
SalesforceEvents
SalesForceQueryService
SalesforceCasesByAccount
SalesforceCreateContact
SalesforceLeads
SalesforceGeoAnnotation
Mashlets
Enterprise gadgets shared for Mashups and Services
Filter: Recently Added | Most Popular | All
unit2walk3a.Mashlet
publish services
create mashups
share mashlets
Featured Downloads
eclipse READY.
download mashup studio
How do I...
send feedback
News from Mashup Developer Community
<mdc>
mashup developer community
Done
Local intranet | Protected Mode: Off
100%

Presto Mashup Studio is our IDE for developing mashups using Enterprise Mashup Markup Language (EMML). The studio offers many rich features such as EMML editor, content-assist, run and debugging environment for advanced mashup development. Please review these instructions to download and install:
How to install from remote | How to install from local files

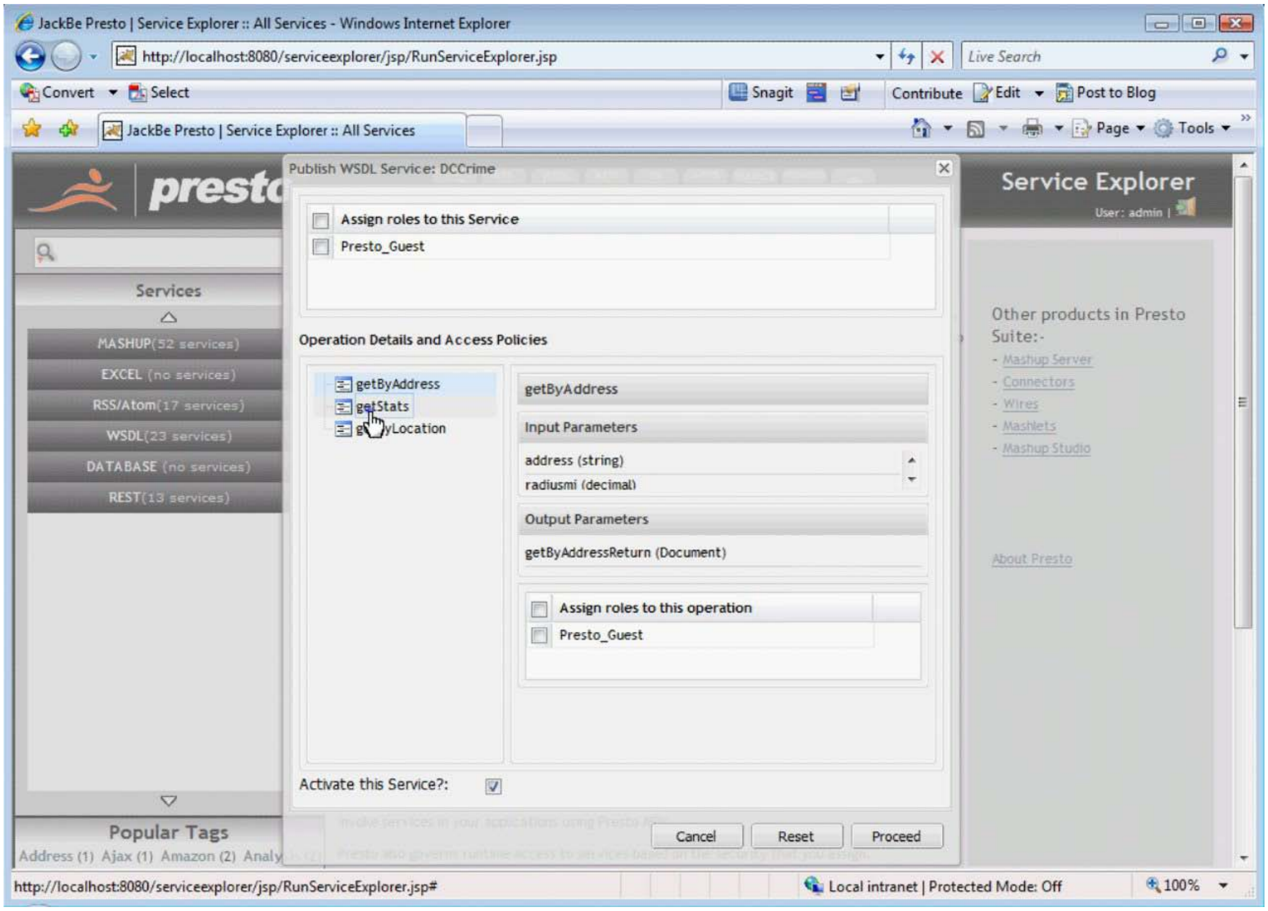
- And the 'Master Mashup Macro Winner' of the month is...
- Q&A and Soup-to-Nuts all about 'Mashups for SharePoint'
- Mashup Macro - Fusion Chart Formatter
- Mashup Video - A SharePoint Mashup of Government

Figure 7: JackBe Presto home page, showing an overview of available components.[1]

Compared to Yahoo! Pipes, the JackBe platform has a richer set of tools to develop Mashups and Mashup components, targeting all developer roles identified in Section 2. The platform identifies 'Mashables', 'Mashlets', and Mashups. Mashables are basically data sources and elementary services in the resource layer, while Mashlets are components positioned in the widget layer. The platform provides a repository of the available components at the different layers, as depicted in Figure 7.

[1] JackBe screenshots are taken from the JackBe tutorial by Fig Leaf Software available at http://training.figleaf.com/tutorials/Presto.cfm

Presto supports two different composers for different kinds of tasks. The Presto Wires composer is a web browser based visual programming environment focusing predominantly on creating Mashlets and Mashups. This tool is comparable to the visual editor supplied by Yahoo! Pipes: it allows for selecting and configuring components and linking these components to form a complete Mashup. The screenshots in Figure 8 and Figure 9 show examples of the usage of this composer. The default set of components provides functionality that is useful in corporate settings, such as connectors to databases, customer relationship management (CRM) systems, content management systems (CMS), and MS Excel sources.



Publish WSDL Service: DCCrime

☐ Assign roles to this Service
☐ Presto_Guest

Operation Details and Access Policies

getByAddress
getStats
g…yLocation

getByAddress

Input Parameters

address (string)
radiusmi (decimal)

Output Parameters

getByAddressReturn (Document)

☐ Assign roles to this operation
☐ Presto_Guest

Activate this Service?: ☑

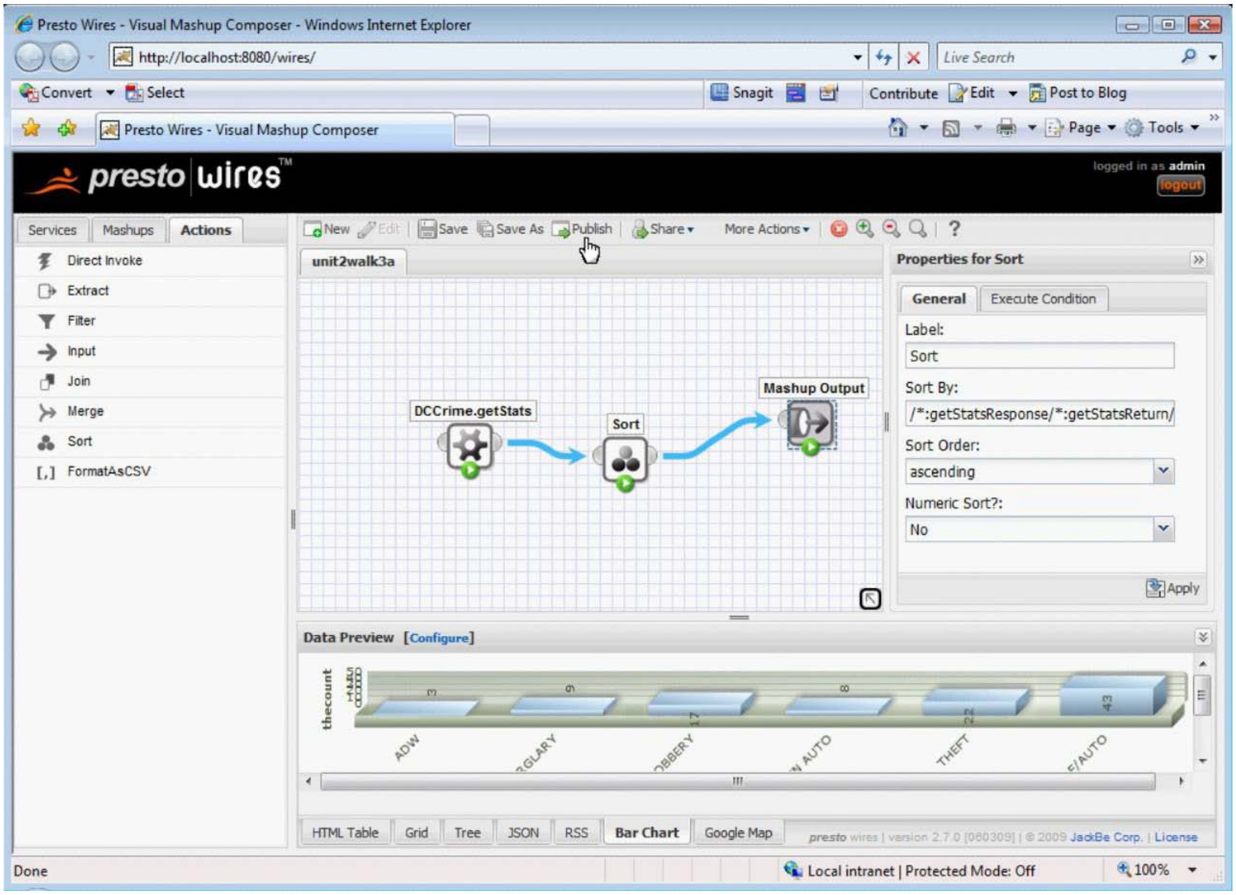
JackBe Presto | Service Explorer :: All Services - Windows Internet Explorer
http://localhost:8080/serviceexplorer/jsp/RunServiceExplorer.jsp
Live Search
Convert
Select
Snagit
Contribute
Edit
Post to Blog
JackBe Presto | Service Explorer :: All Services
Page
Tools
presto
Services
MASHUP(52 services)
EXCEL (no services)
RSS/Atom(17 services)
WSDL(23 services)
DATABASE (no services)
REST(13 services)
Popular Tags
Address (1) Ajax (1) Amazon (2) Analy
Service Explorer
User: admin |
Other products in Presto Suite:-
- Mashup Server
- Connectors
- Wires
- Mashlets
- Mashup Studio
About Presto
Cancel
Reset
Proceed
http://localhost:8080/serviceexplorer/jsp/RunServiceExplorer.jsp#
Local intranet | Protected Mode: Off
100%

Figure 8: configuration of a web services data source within the JackBe Presto environment

The Presto Mashup Studio is an Eclipse plugin which can be used to create program components and Mashups in the Java programming language. Eclipse is a well-known Integrated Development Environment (IDE) with strong support for Java. This composer is targeting developers that create functionality at the resource layer, and provides full access to all features of the JackBe platform. Obviously, using this composer requires skills to go beyond the skills of many end users. It allows for extending the platform with arbitrary functionality, however, which gives the Presto environment considerable flexibility.

The JackBe Presto platform is deployed as a standalone server in the corporate network. This means that there is control over who has access to the environment and also over the amount of resources that are used to run Mashlets. It also means that whenever processing data (inside a Mashable or a

Mashlet) that takes considerable resources, the platform configuration and design of the components can be such that performance is sufficient. There is currently little information about scalability of high demanding components in the cloud environment, although in theory this should be supported by flexibly extending the usage of cloud resources.



Label: Sort

Sort By: /*:getStatsResponse/*:getStatsReturn/

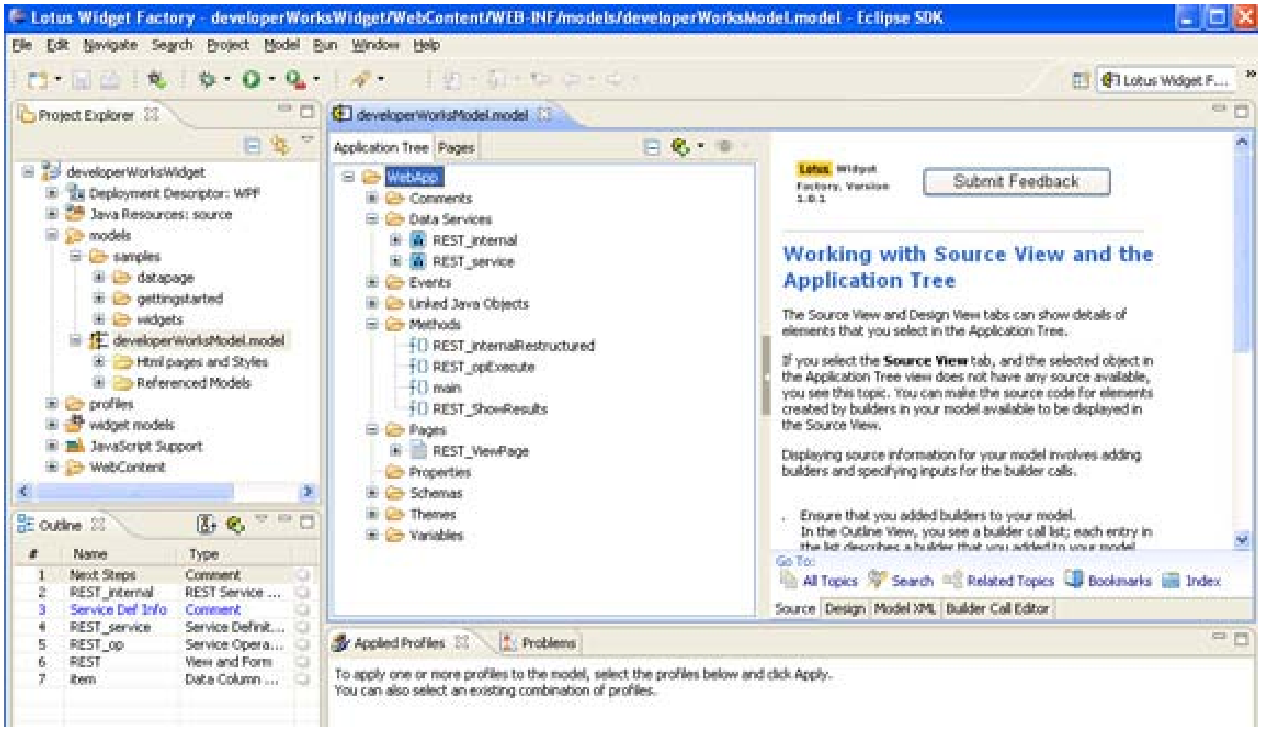
Sort Order: ascending

Numeric Sort?: No

Figure 9: wiring together individual components in the JackBe Presto environment

3.3. IBM Mashup Center

The IBM Mashup Center software [6] is an enterprise Mashup platform with similar features as the JackBe platform. It is available as a standalone software package that typically is installed in a corporate network, and accessed by users from within this network. Mashup Center may be run as a service in the cloud, using the Amazon EC2 cloud environment; IBM supports the easy uploading of the platform software to an Amazon node. Contrary to JackBe, IBM Mashup Center is not directly available as a cloud service (i.e., it requires uploading and administration from the platform operator itself).



#	Name	Type
1	Next Steps	Comment
2	REST_internal	REST Service ...
3	Service Def Info	Comment
4	REST_service	Service Definit...
5	REST_op	Service Opera...
6	REST	View and Form
7	item	Data Column ...

Working with Source View and the Application Tree

The Source View and Design View tabs can show details of elements that you select in the Application Tree.

If you select the **Source View** tab, and the selected object in the Application Tree view does not have any source available, you see this topic. You can make the source code for elements created by builders in your model available to be displayed in the Source View.

Displaying source information for your model involves adding builders and specifying inputs for the builder calls.

- Ensure that you added builders to your model. In the Outline View, you see a builder call list; each entry in the list describes a builder that you added to your model.

To apply one or more profiles to the model, select the profiles below and click Apply.
You can also select an existing combination of profiles.

Figure 10: creating IBM Mashup Center widgets inside the Eclipse IDE[2]

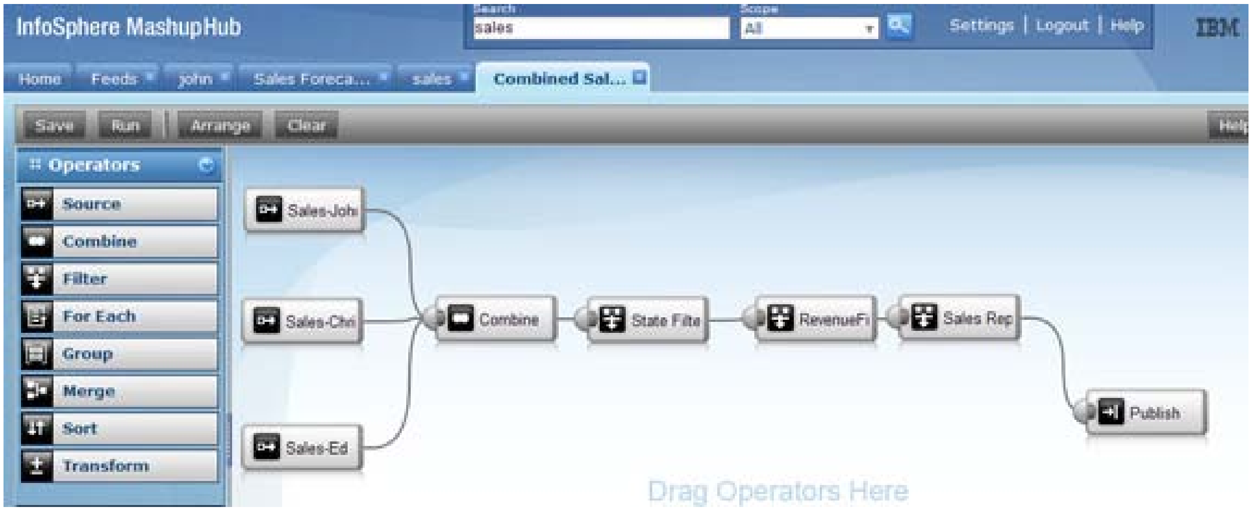


Figure 11: combining different resources (feeds) into one, using the IBM Mashup Center visual programming environment

Developers are supported with different tools to generate functionality at different layers in the Mashup stack. At the highest level, widgets can be linked together to form full Mashups. Widgets themselves can be created with a browser-based widget builder and with a widget factory plugin for the Eclipse IDE (Figure 10). Data resources can be processed using predefined operations in a visual

[2] IBM Mashup Center screenshots are taken from the IBM Mashup Center web site and other IBM web sites.

programming manner (Figure 11). A catalog within the main web interface provides access to existing components, and allows sharing components between users.

Like JackBe, IBM Mashup Center runs on hardware under control (either locally, or in the cloud) of the organization who purchased and uses the platform software. This implies that the amount of computing resources can be adapted as needed.

3.4. Taverna

Taverna [11] is an open source software tool for designing and running scientific workflows. It is a standalone client-side program, typically running on the machine of the user, which uses local as well as remote (internet) data sources and data processing functionality. Like general purpose and enterprise Mashup platforms, Taverna has a visual editor to generate workflows. Workflows can use multiple data sources as input and perform various operations on this data to generate an output, something which is very similar to what Mashups do. The Taverna user base is reported to comprise more than 350 organizations. An example of an editing session inside the Taverna client is shown in Figure 12.

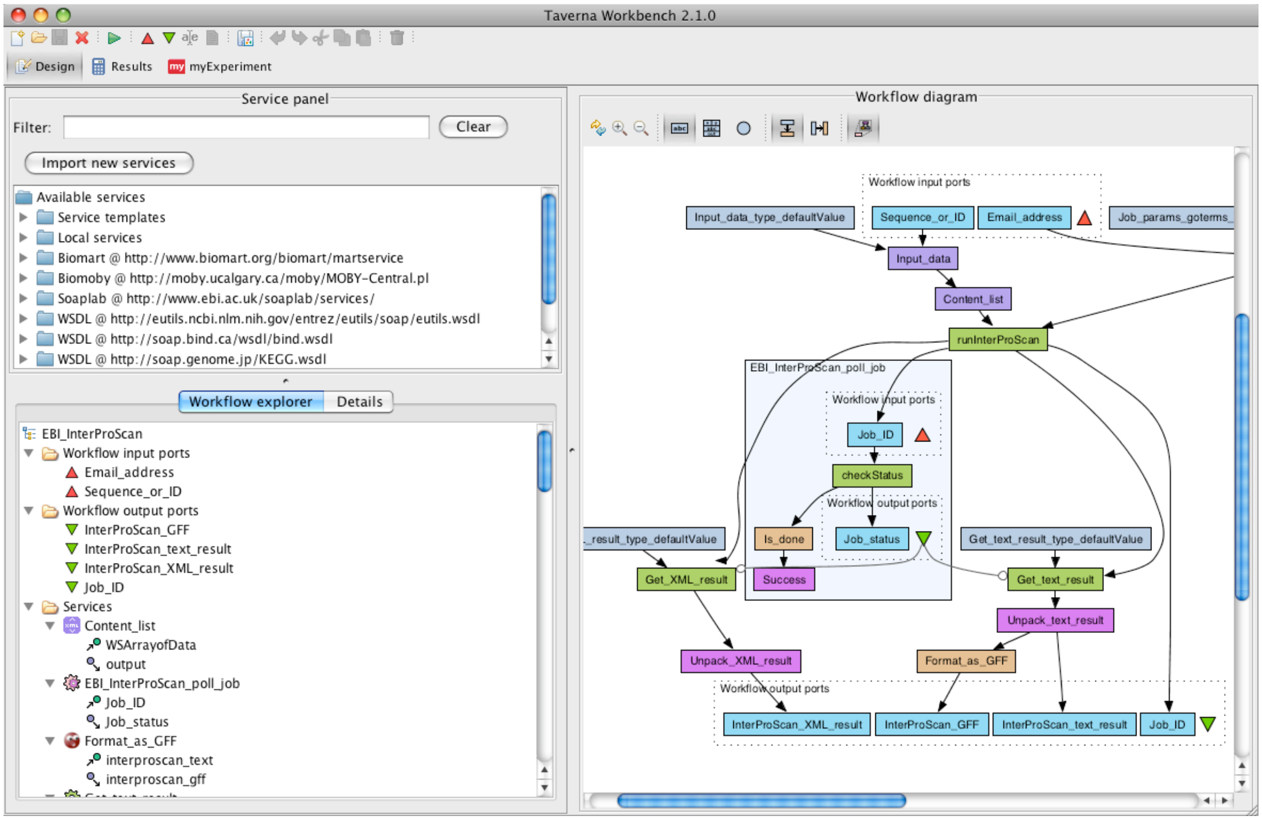
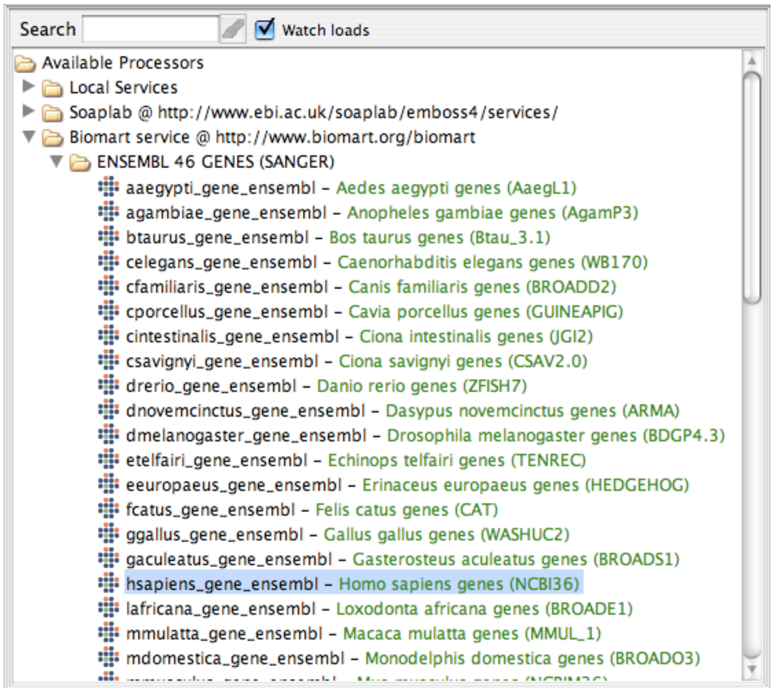


Figure 12: screenshot of a workflow editing session within the Taverna Workbench[3]

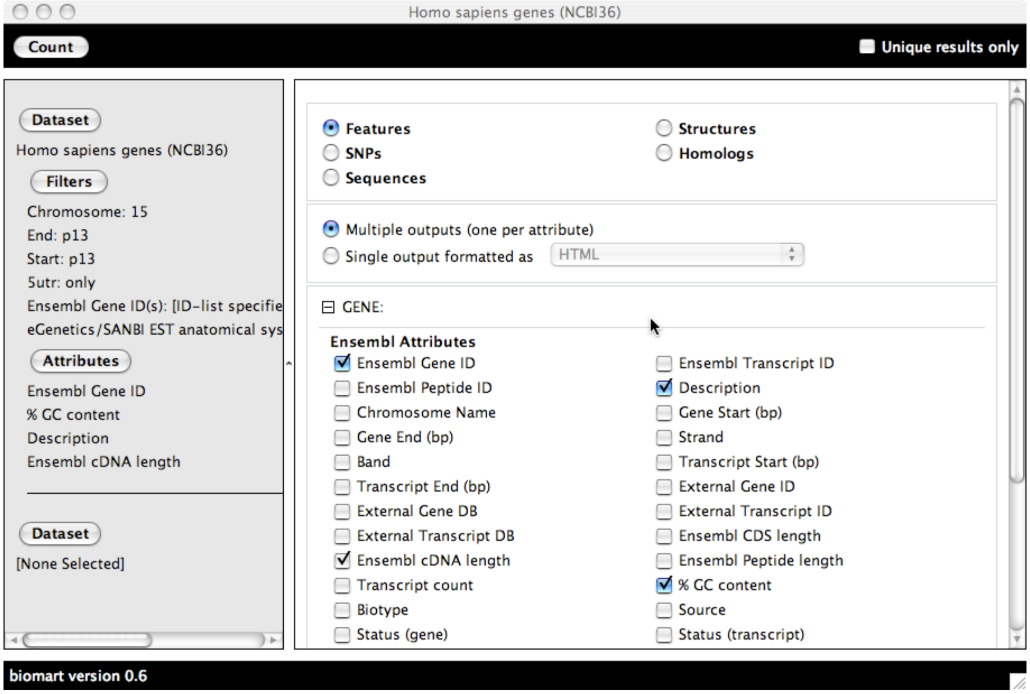
[3] Taverna screenshots are taken from the Taverna web site and user manual.



Search ☑ Watch loads

- Available Processors
  - Local Services
  - Soaplab @ http://www.ebi.ac.uk/soaplab/emboss4/services/
  - Biomart service @ http://www.biomart.org/biomart
    - ENSEMBL 46 GENES (SANGER)
      - aaegypti_gene_ensembl – Aedes aegypti genes (AaegL1)
      - agambiae_gene_ensembl – Anopheles gambiae genes (AgamP3)
      - btaurus_gene_ensembl – Bos taurus genes (Btau_3.1)
      - celegans_gene_ensembl – Caenorhabditis elegans genes (WB170)
      - cfamiliaris_gene_ensembl – Canis familiaris genes (BROADD2)
      - cporcellus_gene_ensembl – Cavia porcellus genes (GUINEAPIG)
      - cintestinalis_gene_ensembl – Ciona intestinalis genes (JGI2)
      - csavignyi_gene_ensembl – Ciona savignyi genes (CSAV2.0)
      - drerio_gene_ensembl – Danio rerio genes (ZFISH7)
      - dnovemcinctus_gene_ensembl – Dasypus novemcinctus genes (ARMA)
      - dmelanogaster_gene_ensembl – Drosophila melanogaster genes (BDGP4.3)
      - etelfairi_gene_ensembl – Echinops telfairi genes (TENREC)
      - eeuropaeus_gene_ensembl – Erinaceus europaeus genes (HEDGEHOG)
      - fcatus_gene_ensembl – Felis catus genes (CAT)
      - ggallus_gene_ensembl – Gallus gallus genes (WASHUC2)
      - gaculeatus_gene_ensembl – Gasterosteus aculeatus genes (BROADS1)
      - hsapiens_gene_ensembl – Homo sapiens genes (NCBI36)
      - lafricana_gene_ensembl – Loxodonta africana genes (BROADE1)
      - mmulatta_gene_ensembl – Macaca mulatta genes (MMUL_1)
      - mdomestica_gene_ensembl – Monodelphis domestica genes (BROADO3)
      - mmusculus_gene_ensembl – Mus musculus genes (NCBIM36)

Figure 13: selecting data from the Biomart data repository.



Homo sapiens genes (NCBI36)

☐ Unique results only

Dataset

Homo sapiens genes (NCBI36)

Filters

Chromosome: 15
End: p13
Start: p13
5utr: only
Ensembl Gene ID(s): [ID-list specifie
eGenetics/SANBI EST anatomical sys

Attributes

Ensembl Gene ID
% GC content
Description
Ensembl cDNA length

Dataset

[None Selected]

● Features ○ Structures
○ SNPs ○ Homologs
○ Sequences

● Multiple outputs (one per attribute)
○ Single output formatted as HTML

⊟ GENE:

Ensembl Attributes

☑ Ensembl Gene ID	☐ Ensembl Transcript ID
☐ Ensembl Peptide ID	☑ Description
☐ Chromosome Name	☐ Gene Start (bp)
☐ Gene End (bp)	☐ Strand
☐ Band	☐ Transcript Start (bp)
☐ Transcript End (bp)	☐ External Gene ID
☐ External Gene DB	☐ External Transcript ID
☐ External Transcript DB	☐ Ensembl CDS length
☑ Ensembl cDNA length	☐ Ensembl Peptide length
☐ Transcript count	☑ % GC content
☐ Biotype	☐ Source
☐ Status (gene)	☐ Status (transcript)

biomart version 0.6

Figure 14: detailed configuration of data items from the Biomart data repository

Taverna can use different kinds of data sources within a workflow, some of which have domain specific formats and semantics. Users of Taverna come from research domains where large amounts of experimental data and structural data analysis play an important role, such as Bioinformatics, Chemistry, Astronomy, Data Mining, etc. Therefore, the data sources that are preconfigured in this environment come from these domains. An example of adding a specific data source to a scientific workflow within Taverna, and its detailed configuration, is provided in Figure 13 and Figure 14.

Taverna supports a wide range of processing components that are common in scientific settings (some of which are domain specific). An example is the integration of the 'R' statistical computing environment with Taverna, a tool which is used in a wide range of scientific domains (see Figure 15).

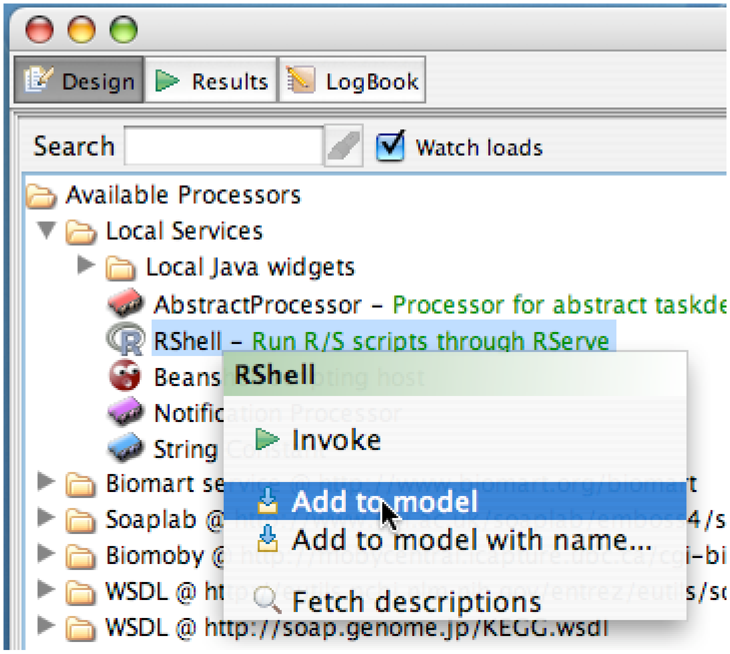
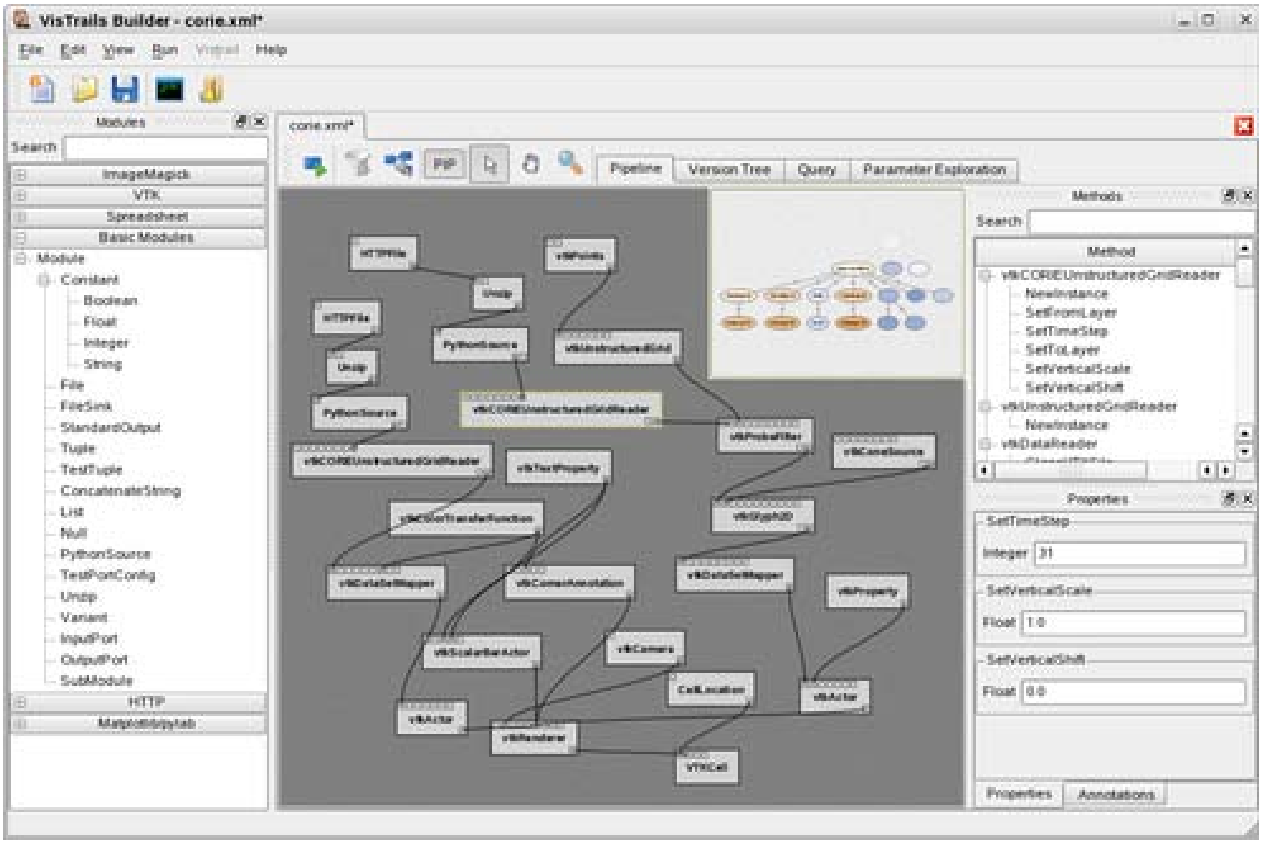


Figure 15: adding an 'R' (statistical language) processing item to a Taverna workflow

Taverna integrates with the myExperiment virtual research environment [9] for finding, using and sharing scientific workflows. The myExperiment website is comparable to the component and Mashup repositories and catalogues available for Mashup platforms. Furthermore, Taverna has functionality to offload processing of steps in the workflow to other sites and machines. Taverna supports the installation of plugins, and its source code is available under a free software license, which means that it is fully extensible.

3.5. VisTrails

The VisTrails open source workflow system [12] is mostly similar to Taverna, but focusing more strongly on visualization, while having less advanced data import and workflow sharing features. Like Taverna, it is a client-side program, installing on the machine of the user. It has a workflow canvas, which can be used to add components to a workflow in a visual manner (see Figure 16).



SetTimeStep
Integer 31

SetVerticalScale
Float 1.0

SetVerticalShift
Float 0.0

Figure 16: the VisTrails workflow editor[4]

[4] This VisTrails screenshot is taken from the VisTrails web site.

4. Discussion and Proposition Analysis

In this section, we discuss the overview of the different products described in the previous section, using the requirements identified in Section 2. Furthermore, we give an analysis on the form in which Mashup technologies could be provided in a SURFnet context, keeping researchers in mind as the target group.

We compare the different products and platform by going through the individual high-level requirements.

Support a common set of interaction protocols / data exchange protocols / data formats such as REST, SOAP, XML, RSS, …
The products discussed support multiple protocols and formats, although each have their specific focus. The enterprise Mashup products focus on handling data that is common in corporate setting, while the scientific workflow products focus on data formats that are common in science (and even more specific: for particular scientific domains). For the general purpose and enterprise Mashup platforms, additional data connector components would need to be developed, to make them attractive for researchers. At first glance, the enterprise Mashup platform support a wider range of web protocols and data formats.

Be able to handle and process large amounts of data, because some experiments generate large amount of data and require extensive processing steps.
This is where the scientific workflow products shine, because they have explicit support for long and expensive (in terms of usage of computing resources) processing steps. They also have better facilities to offload jobs to other machines, and to run parts of the processing steps in parallel. To some extent, in an implicit manner, enterprise Mashup environments can also deal with expensive steps, because the software is running under own control, and can therefore be placed on more capable hardware. Also, these environments support adding dedicated low-level components, which may be used to introduce distributed processing features. For general purpose platforms, the assumption seems to be that Mashup execution does not require extensive resources and that processing steps require a short execution time, which makes them less suitable to fulfill this requirement.

Support domain specific tools and data formats, because individual research domains often use specific data analysis algorithms on domain specific experimental data.
The scientific workflow products are most likely to best support this requirement, because they already do have domain specific features, and are extensible to support others when need arises. Enterprise Mashup platforms may be able to support this as well, although they have the disadvantage of not incorporating generic scientific tools such as the ‘R’ statistical computing system. New extensions may depend on this generic functionality to run. General purpose Mashup platforms do (by nature) not support domain specific functionality and offer little possibility for extensions.

Deal with variety in users (Mashup creators), because the types of analysis and processing may vary considerably between scientific domains.
Enterprise Mashup platforms support this requirement well, because they have tools for the different creator roles. The general purpose tool only support the end-user role (high in the Mashup stack), while the scientific workflow systems are generally more on the middle and lower level.

Deal with variety in audience / public (Mashup users), again to support different scientific domains, but also to make results accessible to those indirectly involved in the scientific process such as publishers.
General purpose platforms run as a public web service and are therefore easy accessible to a large audience. Enterprise Mashup platforms may also be run as a public web site, providing easy accessibility in that way. Workflow products, however, require the installation of client software and are less oriented at the occasional user.

Flexible authentication, to control access to processing and data resources.
It is unclear whether the discussed products do support this requirement. The enterprise environment may be best suited to support this, because authentication and authorization are most relevant for enterprises. General purpose platforms look least equipped to support this, because all functionality is accessible to all users. Scientific workflow products have explicit support for processing that requires many resources (such as facilities to offload jobs to other machines). This may help to control which users have access to processing facilities, and also how many resources are allocated to (parts of) a Mashup.

Support a 'network effect', where the value of the platform grows with every new user added.
This is best supported by general purpose Mashup platforms like Yahoo! Pipes, because new members create new Mashups that are, in general, accessible by others, who can learn and benefit from the efforts of their fellow members. There is a single repository where all components are gathered. To some extent, it may also be supported by JackBe through their cloud offering, where a similar community repository exists. When enterprise Mashup environments run within corporate intranets, however, sharing between users only takes place between people having access to the system. To make a platform attractive to users, it is important that the community has a certain size; by isolating groups of users within corporate networks, this size is not quickly reached. The same problem occurs with scientific workflow systems, because the software runs locally on the user's machine. Taverna has community support through the myExperiment site, which is crucial to build up a repository of work others have made. Note that currently the number of users of scientific workflow platforms is not very high (relative to the total number of scientists), and the usage of these kinds of tools is not yet part of normal scientific practice for a wide range of disciplines. Also note that to gain a large number of members, it is necessary to be internationally oriented; research is conducted in an international setting and scientist must be able to cooperate freely with their peers abroad.

Be able to trace the scientific process, such that researchers can verify the results reported by their peers, using the Mashup platform.
All tools support looking at the details of how Mashups of others were created.

Support extensibility and open interfaces and preferably prevent vendor lock-in.
The open source scientific workflow products are clearly doing best in this respect, because vendor lock-in is less of an issue when it is possible to take the code and change it according to you own wishes. Enterprise Mashup tools also have considerable possibilities to extend their functionality, although vendor lock-in is more likely. General purpose platforms have little opportunity for extensions, which makes them less suitable to fulfill this requirement.

Table 1 gives an overview of the Mashup products in relation to the high-level requirements as described above.

	Support Protocols	Large Amounts of data	Domain Specific Tools	Variety in users	Variety in audience	Flexible Authentication	Network effect	Trace Scientific Process	Support extensibility
General purpose Mashup Service	+/-	-	-	End users	++	Unclear	++	+	-
Enterprise Mashup Service	+	+	+/-	All kinds of roles	+/-	Unclear, support expected	+/-	+	+
Scientific Workflow Service	+	++	++	Expert roles	-	Unclear	+/-	++	+

Table 1: overview of Mashup products in relation to the high-level requirements

From the discussion above, it is clear that no single product or product type is completely supporting scientific Mashups. The Enterprise Mashup products and scientific workflow products look more appropriate than the general purpose products, although they also do not meet all requirements. This situation is depicted in Figure 17.

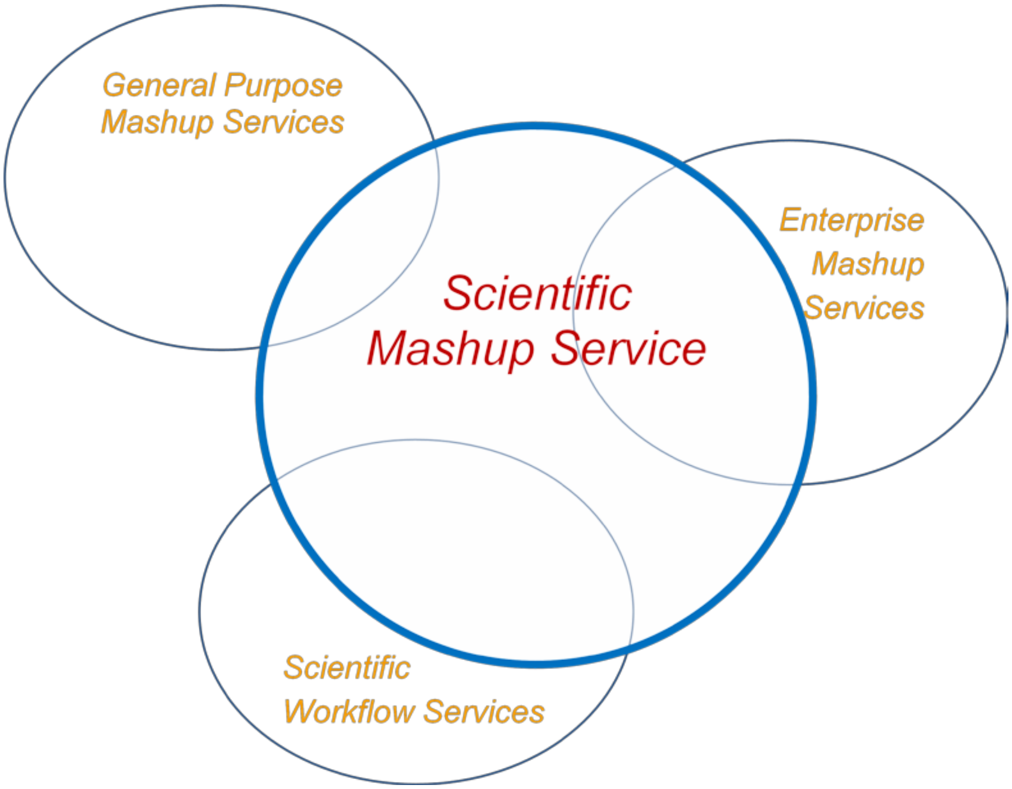

General Purpose Mashup Services
Scientific Mashup Service
Enterprise Mashup Services
Scientific Workflow Services

Figure 17: overlap between existing platforms and the requirements for scientific mashups

A suitable way to position Mashup technology for collaboration within the SURFnet context, is to provide a Mashup service to scientists and couple this with other computing and storage resources available through an international network to which SURFnet contributes. In that way, researchers can use the Mashup service to collaborate and exchange workflows, while at the same time they use the platform to execute these workflows on distributed computing facilities, partly available within the SURFnet environment and partly within other (international) networks. To access these resources, users must be authenticated and authorized, which is also a role that can be adapted by SURFnet (in cooperation with others).

An outline of such a setup is depicted in Figure 18, where a Scientific Mashup service is used by a community of researchers. The Mashups running in this environment may require substantial resources, which are provided through high-performance computing (HPC) services. An authentication and authorization service controls which Mashups and individuals running these Mashups have access to the HPC resources.
In this setup, SURFnet will act as a broker in the collaboration environment. Mashup services can be provided by SURFnet or external service providers.

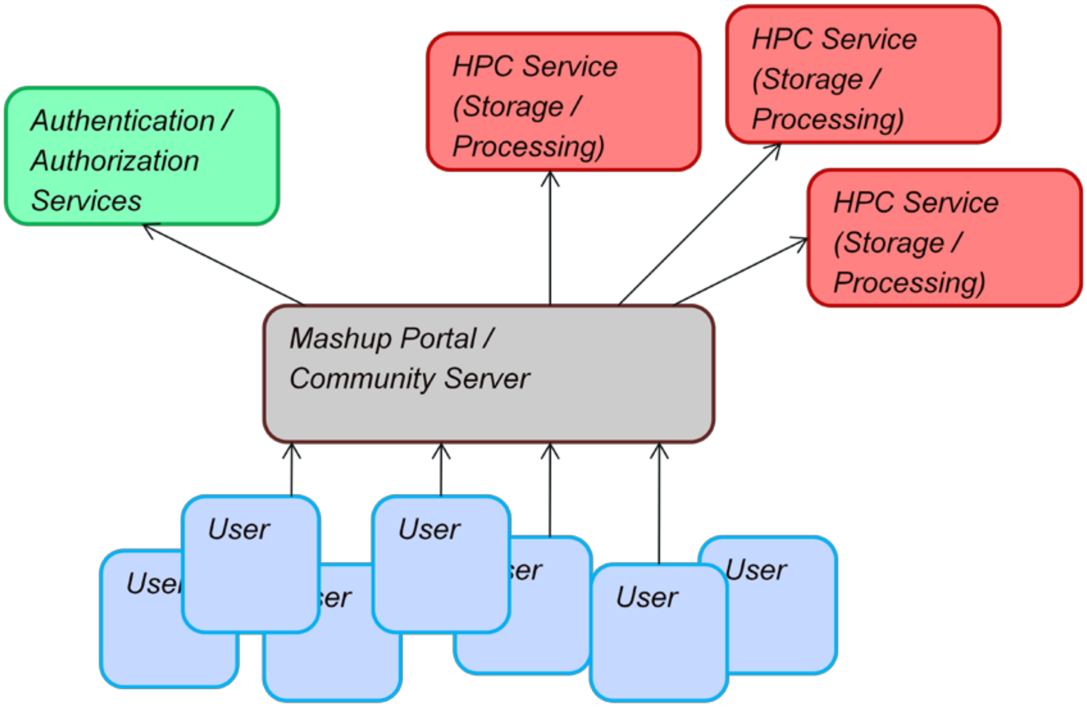

HPC Service (Storage / Processing)
HPC Service (Storage / Processing)
Authentication / Authorization Services
HPC Service (Storage / Processing)
Mashup Portal / Community Server
User
User
User
User
User
User
User
User

Figure 18: a Mashup service in the broader SURFnet context

5. Conclusions and Recommendations

In this report, we have given an overview of existing Mashup technologies and scientific workflow environments, to support an important SURFnet user group – researchers – in their daily research activities. We have formulated a set of requirements to indicate which features and characteristics are most important for researchers using Mashup technology. We have shown that the different platforms are quite diverse, and we have indicated that no single type of platform is providing a good mapping to the requirements. Overall, the Enterprise Mashup products and scientific workflow products are more suitable to fulfill the requirements than the general purpose Machup services, although they do not match completely.

We recommend that these conclusions are checked with the actual users, because their remarks may steer the requirements in a different direction, in which case a certain product category may be suitable to consider.

Furthermore, we recommend that any initiative to support the scientific process and scientific collaboration is done within an international context, because scientists operate at an international level.

Finally, we recommend to keep following the developments in the domain of Mashups and related technologies, because this domain is currently highly dynamic, and new features and product changes appear often.

6. References

[1] Amazon Elastic Compute Cloud (EC2), http://aws.amazon.com/ec2/

[2] Flippity, http://www.flippity.com/

[3] Howe, B., Green-Fishback, H., and Maier, D., Scientific Mashups: Runtime-Configurable Data Product Ensembles, In *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, 2009

[4] Hoyer, V., Stanoevska-Slabeva, K., Janner, T., and Schroth, C., Enterprise Mashups: Design Principles towards the Long Tail of User Needs, In *Proceedings of the IEEE International Conference on Service Computing (SCC'08)*, 2008

[5] Hoyer, V., and Stanoevska-Slabeva, K., Towards a Reference Model for Grassroots Enterprise Mashup Environments, In *Proceedings of the 17th European Conference on Information Systems (ECIS'09),* 2009

[6] IBM Mashup Center, http://www-01.ibm.com/software/info/mashup-center/

[7] JackBe Enterprise Mashup Software, http://www.jackbe.com/

[8] Keppler project, https://kepler-project.org/

[9] myExperiment Virtual Research Environment, http://www.myexperiment.org/

[10] ProgrammableWeb- Mashups, APIs, and the Web as a platform, http://www.programmableweb.com/

[11] Taverna workflow system, http://www.taverna.org.uk/

[12] VisTrails, http://www.vistrails.org/

[13] Yahoo! Pipes, http://pipes.yahoo.com/