CHAPTER 17

The Repository Mashup Map

Stuart Lewis
University of Auckland Library

Mashups can be created for many different reasons, but the most common purpose for creating a mashup, rather than using another form of data presentation, is to add value to the data by combining several sources in order to present new relationships between them. I created the Repository Mashup Map (maps.repository66.org) to do just this - combine data about repositories from different data providers and mash it up with Google Maps (maps.google.com) to visually display information about the repositories. In this chapter we'll examine the mashup to see why and how it was created and look at the data sources it uses in order to consider issues and considerations that affect us when mashing-up data feeds.

Background

Many universities around the world have repositories or are in the process of setting them up. Often these are called institutional repositories, open access repositories, or digital repositories. They are most commonly administrated by university libraries undertaking their roles as curator of, and provider of access to, university information resources. There are many reasons that a university would want or need a repository; these include

- A need to collect, archive, and manage the research outputs of an institution
- The desire to make publicly available the results and outputs of research
- A need to comply with mandates issued by funding bodies such as the National Institutes of Health, The Wellcome Trust, or the European Research Council, which mandate the depositing of publicly funded research results in open access repositories allowing free access to the materials (see www.sherpa.ac.uk/juliet for an up to date list of mandates)
- A desire to create an online showcase of the research outputs of it's faculty
- The creation of a long-term preservation archive for digital materials such as journal articles, data sets, photographs, digitized materials, or administrative documents

Since the start of the millennium, there has been a lot of effort spent on the development of repository software to facilitate the easy creation of repositories. The repository landscape now includes a wealth of software platforms, including open source, commercial, and hosted options, and the number of items held within these repositories numbers figure into many millions.

Table 17.1 shows some examples of the diverse range of repositories across the world, and some examples of their content.

Reasons for Creating the Repository Mashup Map

Because repositories are a relatively new area for libraries, tracking their growth in size and number is interesting. Two registries of open access repositories have been created to facilitate this type of tracking:

- Registry of Open Access Repositories (ROAR; roar.eprints.org) - ROAR is run at the University of Southampton, U.K. It works by automatically collecting and analyzing the content of repositories by harvesting their contents using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Once a new repository has been added and briefly checked by an administrator, the statistics collection process is fully automatic.

- Directory of Open Access Repositories (OpenDOAR; www.opendoar.org) - OpenDOAR is run by the SHERPA (www.sherpa.ac.uk) team at Nottingham University, U.K. OpenDOAR is a human-edited registry of repositories including hand-written descriptions of each repository. The SHEPRA staff members update individual repository statistics occasionally.

Both of these registries present their data in a tabular form. Although this is a suitable format for examining the data record by record, it does not lend itself to providing an overall picture of repository development across the world. Being able to visualize data in an easy format can help researchers see new information that is hidden when displayed solely in a textual format. Because repositories are located at particular places, this makes an ideal key on which to display the data; on a map.

Table 17.1 Examples of the diverse range of repositories across the world

Repository	URL	Description and Examples
MIT OpenCourseware	ocw.mit.edu	An open access repository of course materials from Massachusetts Institute of Technology
MSF Field Research	msf.openrepository.com	The repository of field research undertaken by Médecins Sans Frontières
E-LIS	eprints.rclis.org	A subject-based repository for articles about library and information studies.
BEACON eSpace	trs-new.jpl.nasa.gov/dspace	NASA's Jet Propulsion Laboratory published research repository
CADIAR	cadiar.aber.ac.uk	A typical university institutional repository in the U.K.
ResearchSpace@Auckland	researchspace.auckland.ac.nz	A typical institutional repository in New Zealand

Universities and research institutions are competitive bodies. They have to compete against each other for research funding, prestige and rankings. One side effect of this competition is that they often look sideways at what other institutions are doing in order to keep up with any developments that others are making and that they feel they need to copy. Again, a map is an ideal visualization tool as you can see what is happening at institutions near you, in your country, your continent, and across the whole world.

Repository managers also like to judge their progress in populating their repository against other repositories. They see it as a matter of professional pride to have a higher number of archived items in their repository than other repositories in similar institutions have. The vendors of repository software and services like to compare themselves to each other to see who has more installed instances of their

software in comparison to others, allowing them to make claims such as “the most widely installed open source repository solution.” This is a powerful marketing tool because we all feel safe following the leader - no one was ever sacked for buying IBM!

Plotting the Repository Mashup Map

So there was the statistical data available about repositories and a need for a visual representation allowing easy comparisons to be made, but there was one key bit of data missing - the geographic location of each repository in a form that could be plotted on a map.

OpenDOAR held the name and address of the institution that owned each repository, but this information cannot be directly used. A map does not know where your house is unless you code the location in a format that can be used with a map. To plot data on a map you require a set of co-ordinates.

I had three options available to gather this data:

1. Locate each repository by hand.
    - Description: With a tool such as Google Maps, the locations of the institutions could be found.
    - Efficiency: Low. This is a time consuming process as it requires human input to actually locate the repositories and to copy the data into the mashup database.
    - Accuracy: High. The accuracy should be high as each location is checked by hand.
    - Issues: This option does not scale well and requires human input before a repository can appear on the map.

2. Look up the location of each repository by its IP (Internet Protocol) address.
    - Description: Every computer on the Internet has a unique address, known as its IP address. This address is used to ensure network traffic is sent to the correct machine. Both open and commercial databases hold the geographic locations of IP addresses.
    - Efficiency: High. The process of finding the IP address of a computer from its URL (Uniform Resource Locator) and then using a web service to find its location can be scripted. This requires no human input.
    - Accuracy: Medium. While the coverage of the free geolocation databases is fair, and the commercial databases good, not every IP address is located accurately, and some are located only to the nearest city or region.
    - Issues: One particular issue is that repositories hosted by a commercial company are often located at the company, not the institution. Sometimes the company is in a different country from the institution, and if the hosting company has lot of customers, you could find a cluster of repositories incorrectly located at the site of the server.

3. Use a geocoding service.
    - Description: Geocoding services take an address (typically a street address, city name, or ZIP code or postcode) and convert it to a map coordinate. This happens whenever you use an online map service to look up an address.
    - Efficiency: High. Addresses and ZIP codes or postcodes are held for each repository-owning institution in OpenDOAR, and can be used to automatically look up the location by means of a geocoding service.
    - Accuracy: High. Addresses are unique, and ZIP codes or postcodes refer only to a small geographic area.
    - Issues: Geocoding services are either very expensive or offer partial coverage across the world. For example Google introduced a geocoding service for U.K. locations only in July 2007.

At first the intention of the project was to make a map of U.K. repositories. Because this was a reasonably limited set of repositories (less than 100 in early 2007), I located each repository by hand. Coming from the U.K., I had a good idea about where each institution was in the country and how to locate it accurately on a map, so the job was not too difficult. However, it soon became apparent that a repository map of the whole world would be more useful, which meant that placing each repository by hand, especially in countries where I had no knowledge of the geography, was an unrealistic task.

The initial solution was to use the free IP-address-to-geographic-coordinates service "hostip" (www.hostip.info). At the same time, ROAR started using this service to collect the locations of each repository, meaning that the mashup could delegate this task to ROAR and just use that information along with the rest of the information it harvested. However, as time went on and more repositories were added, gaps in the free information started to appear. So in mid-2007, I converted the code to use a commercial web service to locate the repositories from their IP address. The accuracy was better, although not perfect.

In true Web 2.0 fashion the repository mashup map also allows users to locate repositories on the map or to update existing repository locations. The mashup achieves this by listing repositories without locations and with locations and allowing users to place repositories on a map, recording the location in the database.

At of mid 2008, a mixture of techniques was in use to locate the repositories on the map:

- A commercial geolocation database web service
- Locations held by ROAR and OpenDOAR
- Entries entered or updated by users
- Hand location of repositories

Creating the Repository Mashup Map

This next section details how I built the repository mashup map (Figure 17.1), including the decisions I had to make and the potentials benefits and trade-offs I had to consider. There are often several ways to achieve a task (e.g., aggregating a set of RSS feeds), and choosing the best solution is not always easy.

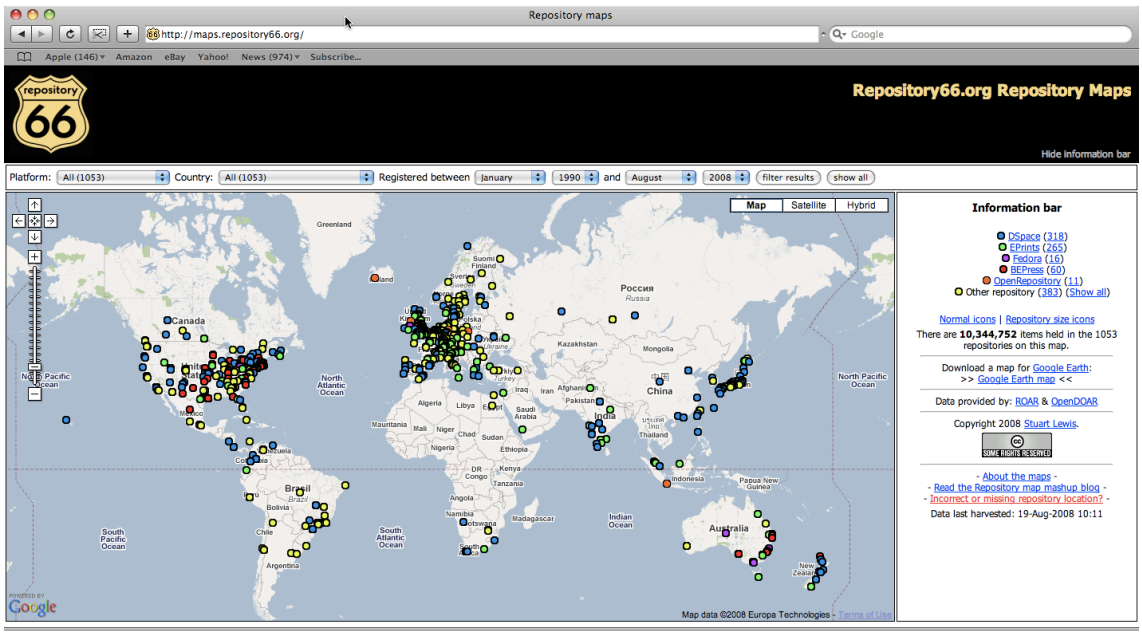

Figure 17.1 Repository Mashup Map (maps.repositroy66.org)

To be able to mashup data from external sources such as ROAR and OpenDOAR, you will need to find a way to grab the data. Sometimes this can be by screen-scraping HTML, or, if you are lucky, the data providers you use will provide you with an application programming interface (API). An API is a way of easily interacting with a web service. Using an API will typically allow you to harvest the data in a formatted fashion, often with a search facility.

Harvesting Data From ROAR

ROAR offers two data-harvesting interfaces:

1. An OAI-PMH interface (roar.eprints.org/oai.php?verb=ListRecords&metadataPrefix=oai_dc)
2. A complete download of all repository records (roar.eprints.org/index.php?action=rawlist)

The choice of which to use was easy. Rather than making successive web requests, which are required by OAI-PMH, it was easier to download all of the data in one go and parse it into individual records myself.

Harvesting Data From OpenDOAR

OpenDOAR provides a useful API (www.opendoar.org/tools/api.html). There are two ways of using it:

1. Perform a search, for example all repositories in New Zealand (www.opendoar.org/api.php?co=nz)
2. Download all the data at once (www.opendoar.org/api.php?all=y)

As with ROAR, I found it easier to download all the data at once and then process it myself.

As previously mentioned, the Repository Mashup Map is created by mashing up the data held in ROAR and OpenDOAR and combining it with geographic location coordinates supplied by various means. To mashup the data, several design decisions needed to be made.

The Common Identifier

To mashup data, you often need a common identifier that can be used to match items in each of the data sources that you are using. Often these common identifiers are unique to a given object, allowing very fine-grained matching of objects in different systems. For example in the case of a book mashup that pulls information from different sources, you could choose the ISBN, as it will uniquely identify the same item each system. In the case of repositories, the unique key used by both ROAR and OpenDOAR is the OAI-PMH location (a URL) of each repository. The OAI-PMH location is the web address that can be used by software to harvest metadata from a repository in order to reuse the metadata or to collate statistics.

You can see example OAI-PMH responses by trying these URLs:

- cadair.aber.ac.uk/dspace-oai/request?verb=Identify
- cadair.aber.ac.uk/dspace-oai/request?verb=ListSets
- cadair.aber.ac.uk/dspace-oai/request?verb=ListMetadataFormats
- cadair.aber.ac.uk/dspace-oai/request?verb=ListRecords&metadataPrefix=oai_dc

Data Source Authority

When combining two data sets which provide different types of data about a group of similar objects, you have to decide whether to use all of the data provided by both systems or to consider one data source as authoritative and the other as secondary. This can be explained most easily though a traditional Venn diagram (Figure 17.2).

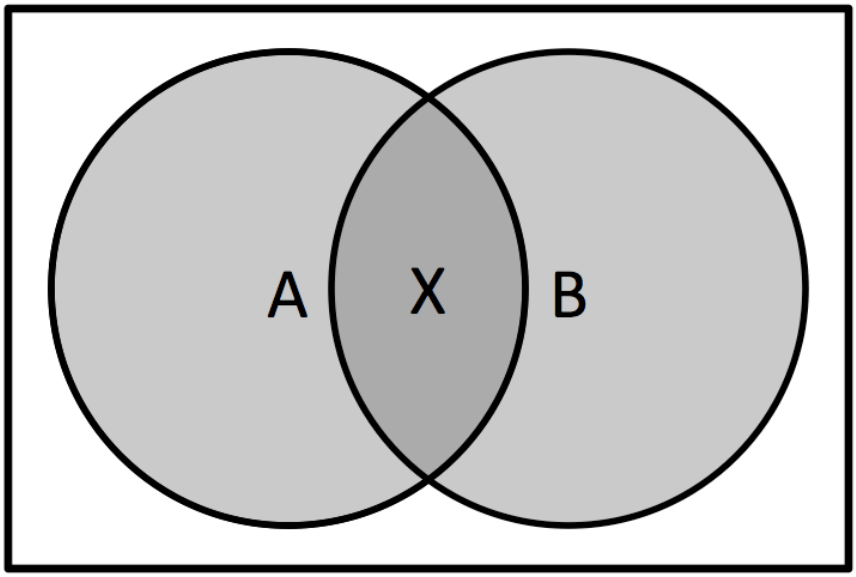

Figure 17.2 Traditional Venn diagram (en.wikipedia.org/wiki/Venn_diagram)

- Use all the data - Use data from both systems, A and B. Where they crossover at X, you have richer data; where they differ, you have less data, but at least you have the object.
- Use one authoritative system - When one system is considered authoritative, you lose any data that is stored only in the other system. In the diagram, this means you would lose data from section B.

The decision will depend on your mashup. If you are creating a mashup of products sold by an online store with reviews about those items from a product review site, there is no point in showing reviews for products that you cannot buy. Therefore, it is best to consider the store's product list as the authoritative data source. If on the other hand your mashup is to compare prices of products in different online stores, then you want to show every product, even if it only for sale in one store.

To avoid potential duplication of data in the Repository Mashup Map, I decided to make one source the authoritative source. If all sources were used (A + X + B) then duplicates could (and do!) arise where a common identifier in A and B refer to the same object but are subtly different. As of August 2008, the Repository Mashup Map used ROAR as its authoritative source because the collection policy of OpenDOAR is stricter than that of ROAR. However, I continue to monitor the situation to provide the best aggregated resource to users of the maps.

Live or Offline Processing

Mashing up data requires processing power. It takes effort to combine data sources, and this can either take place “live,” as the user requires it, or “offline,” in advance of the user’s requiring it. Performing the processing offline is advantageous as the mashup user does not have to wait while the data is compiled. When data from different sources is being mashed up, the latency of different data sources could mean a slow and unsatisfactory mashup experience for the user. However, if the data is time critical and constantly changing (e.g. a mashup of items for sale on eBay and their geographic locations), then offline processing may not be an option as the user requires up-to-date information. One solution offering the advantages of both options is to cache a copy of data for a time (perhaps an hour) and then to refresh the data once it has expired. This means the data is reasonably up-to-date but not being refreshed slowly for each user.

Offline processing will not be an option if the whole of a data set is not available or you do not have the processing power or storage space to deal with it. If you were mashing up data from a large online store such as Amazon (www.amazon.com), you can only query it rather than download it all. (Even if you

could download it all, the average mashup-maker does not have the resources to process it.)

The Repository Mashup Map uses data sources that are manageable (in the order of thousands rather than tens of thousands) and that are not updated very often (OpenDOAR rarely, ROAR daily). Therefore, it made sense to compile the mashup data offline every day or two rather than mashup the data for each user. An additional benefit of processing the data offline is the it is less prone to errors. If one of the data sources is temporarily not working, then the site will continue to provide a full service rather then being degraded during the service outage.

Client or Server Filtering

Very often with a mashup, the user is only interested in a portion of the complete data set. When viewing a mashup of crime statistics and houses for sale, the user will most likely be interested only in certain house types in a few areas (e.g., large family homes in the suburbs). This filtering of the complete data set can either take place on the server providing the data or in the browser (client) viewing the data.

Some mashups, such as a mashup of products and reviews, require the user to enter a query before any data is shown. Only when users requests reviews for a particular product or product type will they see any mashedup data. In this case, because the data set is likely to be large, it is probably most efficient to let the web server perform the work and then send the resulting smaller data set to the client.

However, if the mashup is restricted to a smaller and more specialized data set in which the user by default will see all of the mashed up data (e.g., a map of zoos and wildlife parks in a country) then it would probably be most sensible for filtering to take place on the client's machine. Because the client machine already has all of the data, it would be silly and slow for the mashup to request just a subset of the data to be resent (just zoos and wildlife parks with monkeys) when the filtering can be performed locally.

The Repository Mashup Map uses the second of these options. The entire data set of repositories and their locations is downloaded at once. If the user wishes to filter on one aspect, such as software type or date of creation, then this filtering is performed on the browser end.

Send Enough or All of the Data

When one is creating a mashup and deciding when and what data to send to a client, an additional decision that needs to be made relates to when to send the data. On first loading, mashups often just initially show a low level of detail (product names, price and star rating) but then display a higher level of detail (product reviews and photos) once the user has selected them. There are three options of deciding what data to send to the client at which point:

1. Send all the data at once.
    - Advantages: Once the mashup has loaded, the client has all of the data the client requires, so the client will experience a responsive mashup that provides additional detail quickly.
    - Disadvantages: The mashup will be slower to load as all the data needs to be transmitted at once.
2. Send the low-level detail first, and then the high-level detail once the user requires it.
    - Advantages: The mashup loads quicker as it is not loading the entire data set.
    - Disadvantages: Users may not perceive the mashup as responsive because it will take a short amount of time for the higher detail level to be fetched from the server.

3. Send the low-level detail first, and send the high-level detail in the background.

    - Advantages: The mashup loads quickly, and then in the background, while the user is interacting with it, the high level data is downloaded, meaning that the mashup will appear responsive when the user requests the high level of detail of an item whose data has been downloaded in the background.

    - Disadvantages: It is technically difficult to do this and beyond the scope of an amateur mashup creator.

Initially I decided to make the Repository Mashup Map send all the data at once. However, as the data contained in ROAR and OpenDOAR grew, the data file that was downloaded became half a megabyte in size. For users with a slow Internet connection, the downloading of the file was prohibitively long. Therefore in mid-2008 I switched to the second option and only downloaded the simple statistics of each repository. Full details, such as the repository description, were loaded only when the user requested to see it. Because each user of the mashup will probably look only at the descriptions of a few repositories, this small trade-off of loading time versus a slightly less responsive mashup made sense.

Data Source Licensing

Data sources often come with licenses describing what you are allowed to do with the data, what you are not allowed to do with the data, and whether there are any attributions to the data owners that must be shown. The data from ROAR and OpenDOAR are both shared under a Creative Commons (www.creativecommons.org) license ("by attribution" and "by attribution, noncommercial, sharealike" respectively).

If is a good idea to license your own data in order to protect your own rights. If the data you are using requires it, you may be required to pass on a similar set of rights (a share-alike license).

The Repository Mashup Map licenses its data with a Creative Commons "by attribution, noncommercial, sharealike" license.

The Mashup Website

This chapter has so far concentrated on why the mashup was created and how the data was collated. This section describes how the mashup website is constructed, and what technologies power it.

The Mapping

The mapping element naturally takes up the largest area of the mashup webpage and is powered by the Google Maps API (code.google.com/apis/maps). The API allows a map to be added to any web page with very few lines of code. Interactive features can easily be included on the maps such as a zoom bar, mouse event handling for panning around the map, and buttons to switch between a traditional map view and a satellite view. The initial view of the map can be controlled by code and is set by default to show the whole world.

I could have used an alternative mapping API such as Yahoo! Maps Web Services (developer.yahoo.com/maps) or Bing Maps (www.microsoft.com/maps). However for purely pragmatic reasons, I chose to use Google Maps. (Sometimes decisions I make are just based on "what I know" rather than anything more scientific!

The Dots

The markers are placed on the map via JavaScript. The XML (eXtensible Markup Language) file containing the repository data is downloaded using Javascript that then processes the data and places the markers. The markers are either displayed in variable sizes (Figure 17.3) or all the same size (Figure 17.4), depending on the number of items in each repository. The markers are sized according to a

logarithmic scale (en.wikipedia.org/wiki/Logarithmic_scale) to ensure that small repository markers are not too small and large repository markers are not too big.

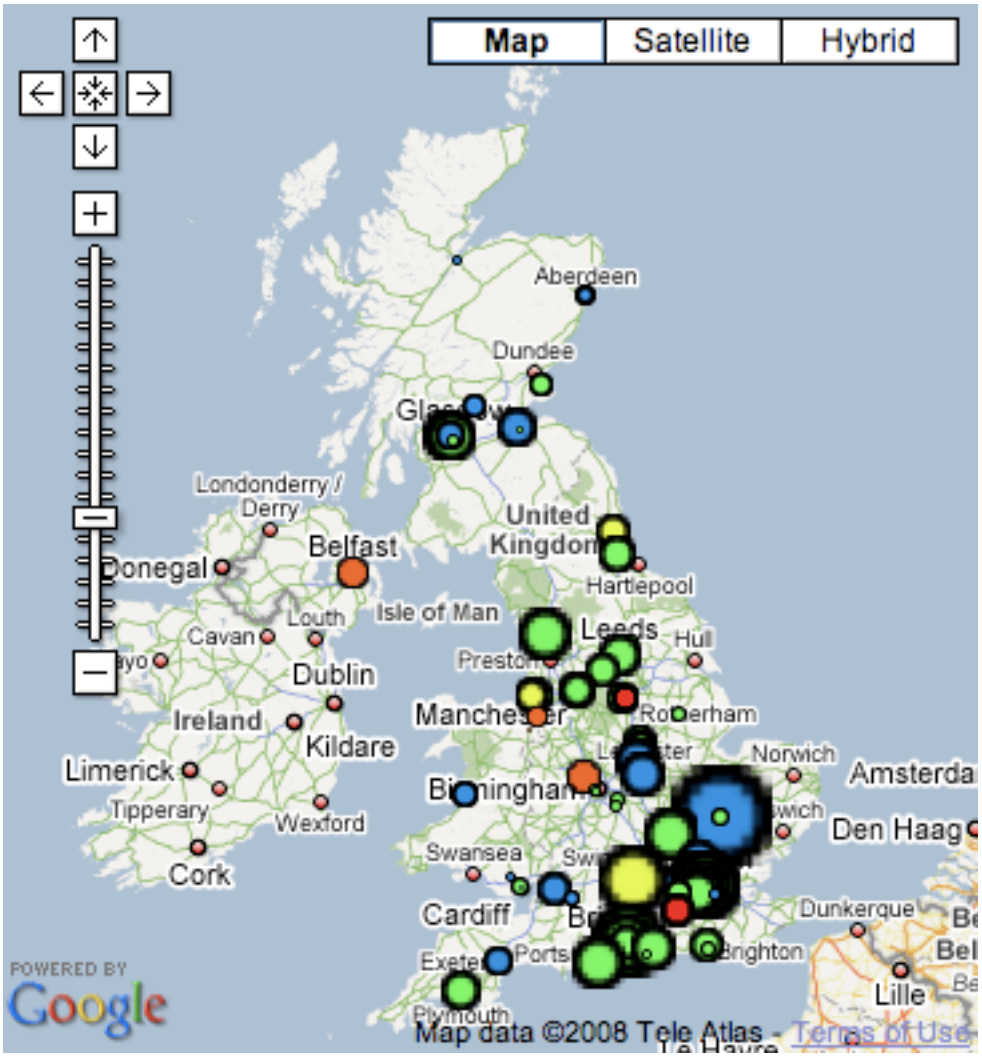

Figure 17.3 Markers are displayed in varied sizes.

The Data

In order to display the data held about each repository, a window appears when a user clicks on a repository marker on the map. The window that pops up uses tabs to make the best use of the space. The initial tab shows basic data about the repository: its name, date of creation, number of items it holds, and content types it holds.

Additional tabs show the description of the repository from OpenDOAR, a growth graph showing how the number of items in the repository has grown over time, and a search tab. The search tab allows the user to search the contents of the repository using Google, Google Scholar (scholar.google.com), or Bing (www.bing.com) (Figure 17.5).

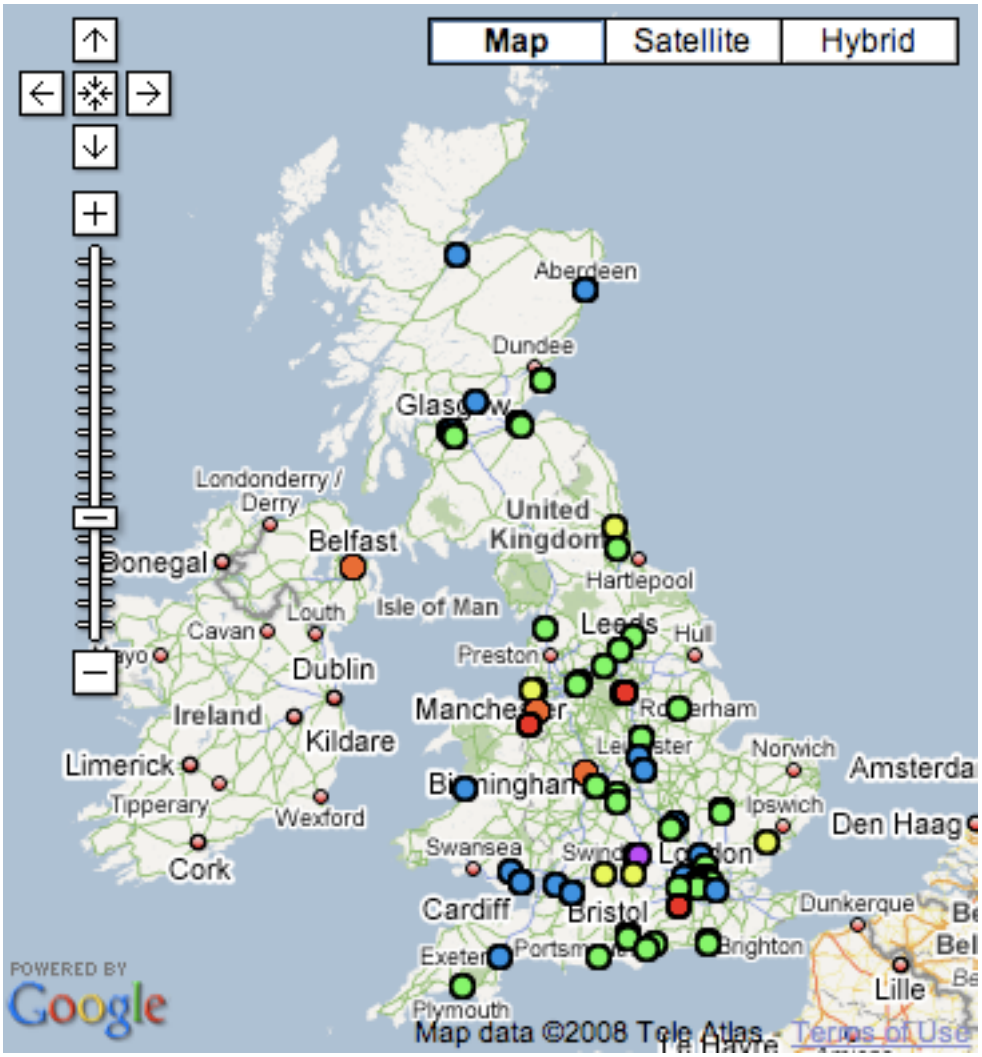

Figure 17.4 Markers are displayed all the same size.

The growth graphs

Growth graphs showing the how the number of items in each repository have changed over time are generated for the site (Figure 17.6). There were several possible solutions for generating the graphs:

- Generate graphs when the data is collated - The software written to collect the data could have generated the graphs via a software library designed to generate graphs.

- Generate graphs on the client using JavaScript - The data could be sent in its raw form to the web browser that can make use of a JavaScript charting library. Potential downsides of this option are an extended initial loading time as the charting library is also downloaded and an increased chance that the site will be incompatible with some browsers.

- Use an online charting service - Google provide an online charting library (code.google.com/apis/chart) that generate charts on the fly for inclusion in webpages. The URL of the image has to be specially constructed to contain all the chart data and metadata (type, colors, scales, etc.).

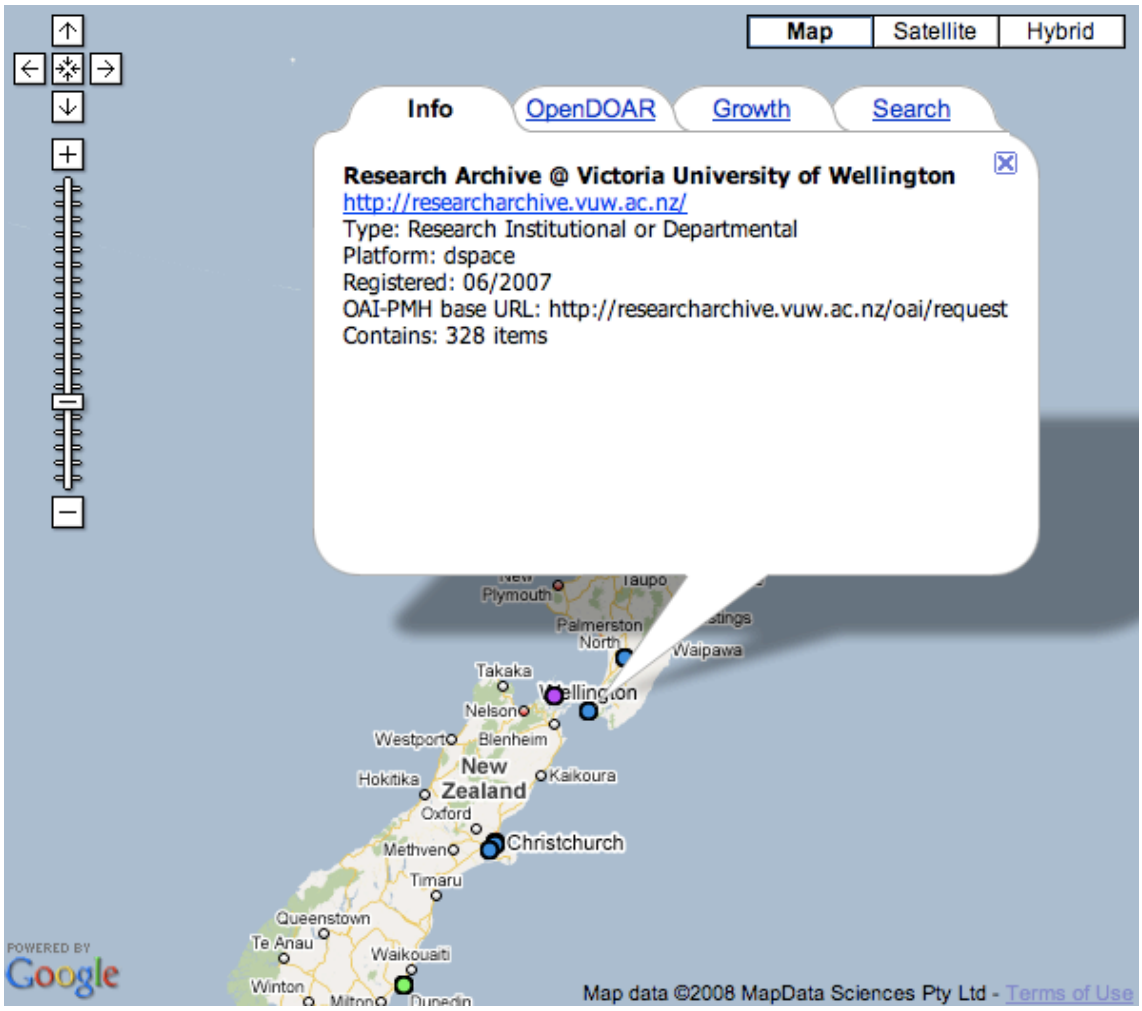

Figure 17.5 The pop-up window tabs make the best use of the space to display the data held about each repository.

The Google charts option was chosen as it seemed most flexible. It also reduced the amount of bandwidth used by the mashup web server as Google servers would serve the images. From a lazy programmer's perspective (there is nothing wrong with being lazy - I'm just making my job easier and quicker!) this was the quickest option to develop.

The filters

Several filtering options are available when one is selecting which repositories to display. The first filter added was for the different software platforms that are used by each repository. This showed interesting patterns of use throughout the world. Some countries have a complete mix of platforms whereas others showed more prominent usage patterns.

The second filter added shows the state of repository population at any time in the past. The user can select a month and year and see which repositories existed at that time. This allows a user to see how and where repositories developed over time (adoption trends).

A third filter was added during the editing of this book chapter (a good mashup keeps adding new features!). It allows the user to filter on repositories in one country only.

Also added during the final editing of this chapter was the automatic zoom function. Imagine using the filters to select “All repositories in the United Kingdom”; it is pretty predictable where the results will be. Therefore the mashup makes use of the Google Maps API functions that allow you to center the

maps and to zoom so that the map markers are shown at an appropriate level. In this case it zooms in automatically to show the U.K.

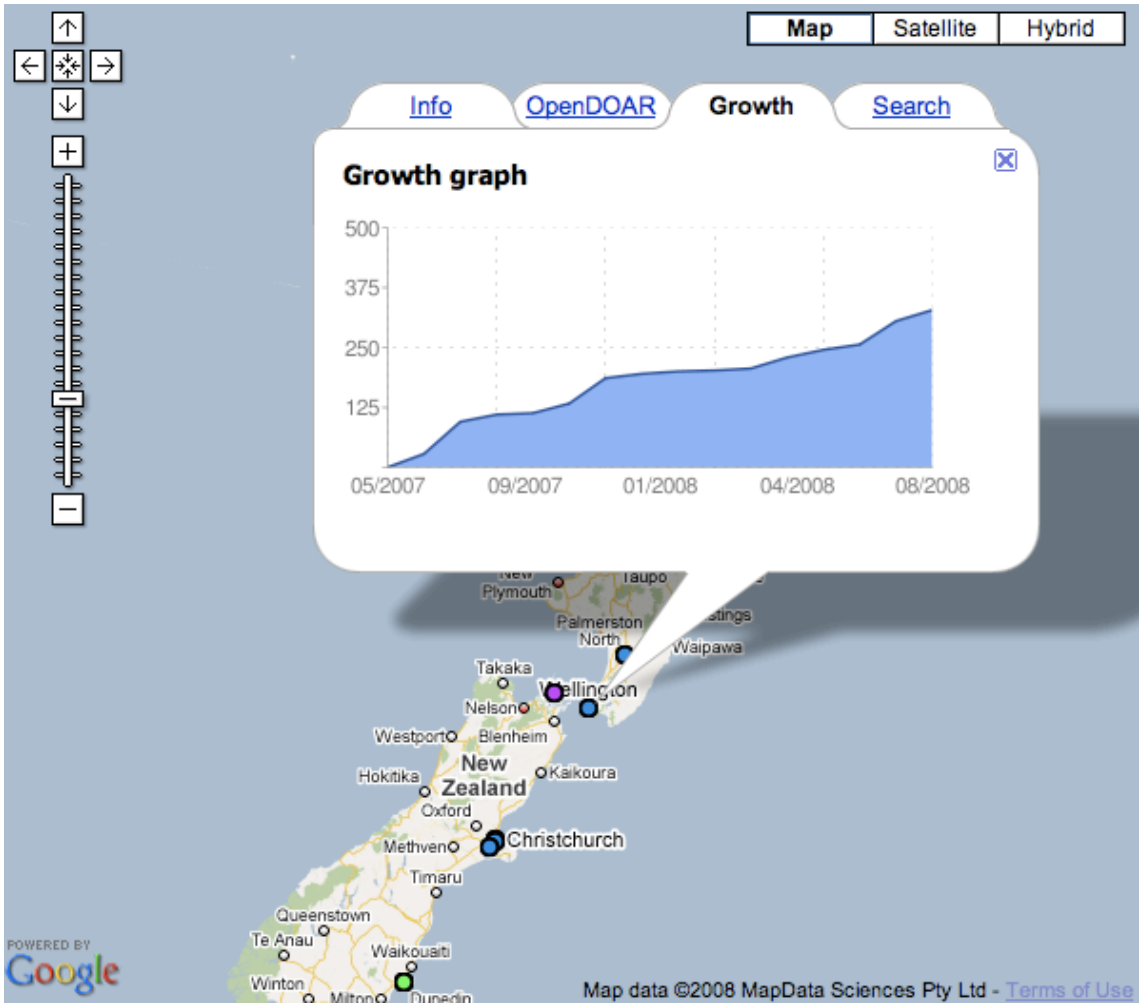

Figure 17.6 Growth graph showing the number of items in the repository

User Contribution

Users can update the position of a current repository marker if it has been incorrectly placed or can be placed more accurately, or they can add the location of a repository that has not yet been placed on the map. A smaller Google Map is shown to the user, who can move around it and zoom in or out to find the correct location, which is then emailed to the author as a suggestion to update the location database. I try to thank all contributors with a short personal email just so they know their contribution has been accepted and considered worthwhile and to say "Thank You!"

The Blog

The site uses a blog powered by WordPress (wordpress.org) to provide a mechanism to update users on developments and to allow users to comment on the mashup. Once again, resource issues made the choice easy. WordPress uses technologies available from my web host (PHP and MySQL) and was quick to setup and configure.

Future developments

There are several possible areas of future developments of the Repository Mashup Map, and they include

- A timeline animation - Automate the date filter so that a Play button can be pressed to automatically view the change in the repository landscape over time with new repository markers appearing and growing.

- Extra filters - New filters such as repository content type could be added.

- Extra search options - Extra search targets can be added for each repository (perhaps repository-specific search services such as OAIster [www.oaister.org] or Intute Repository Search [www.intute.ac.uk/irs]). Another search option might be to search groups of repositories, such as all repositories of a given type.

- Dynamic filter selection boxes - When you select a filter (e.g., "All DSpace repositories"), the other filter selections (e.g., countries) should get rewritten so that instead of showing how many repositories there are in each country, they show how many DSpace repositories there are in each country.

I would be pleased to hear of any other suggestions! Please either email me or leave a comment on the blog.

Author Bio

Stuart Lewis is the IT Innovations Analyst and Developer at the University of Auckland Library (he formerly worked at Aberystwyth University), a committer with the open source DSpace repository platform, and the creator of the maps.repository66.org mashup site. He is married to Lauren and has two fun young kids, Jacob and Lily.

Email Stuart at stuart@stuartlewis.com

October 2009