

AusStage Cartographic advice

Prepared by: Ian Johnson, Steven Hayes, Andrew Wilson, Steve White, Irek Golka

Archaeological Computing Laboratory, University of Sydney

Version Revision: 13th Dec 2010

Summary

The following document strives to encapsulate the results of research carried out for and general advice provided to the AusStage project by the Archaeological Computing Laboratory.

A key outcome of the project is a iconographic language that allows the complex data set managed by AusStage to be rendered on a map in a legible fashion.



The elements of this language are described in detail later in this document along with example icons and a number of “phrases” to demonstrate correct reading.

Another key outcome of the project is the recommendation to simplify the search interface by integrating all functions into a single interface.

Other parts of the document address map legends, timelines, popups and rollovers. We also cover various issues related to the use of Google Map APIs and various other 3d party APIs and

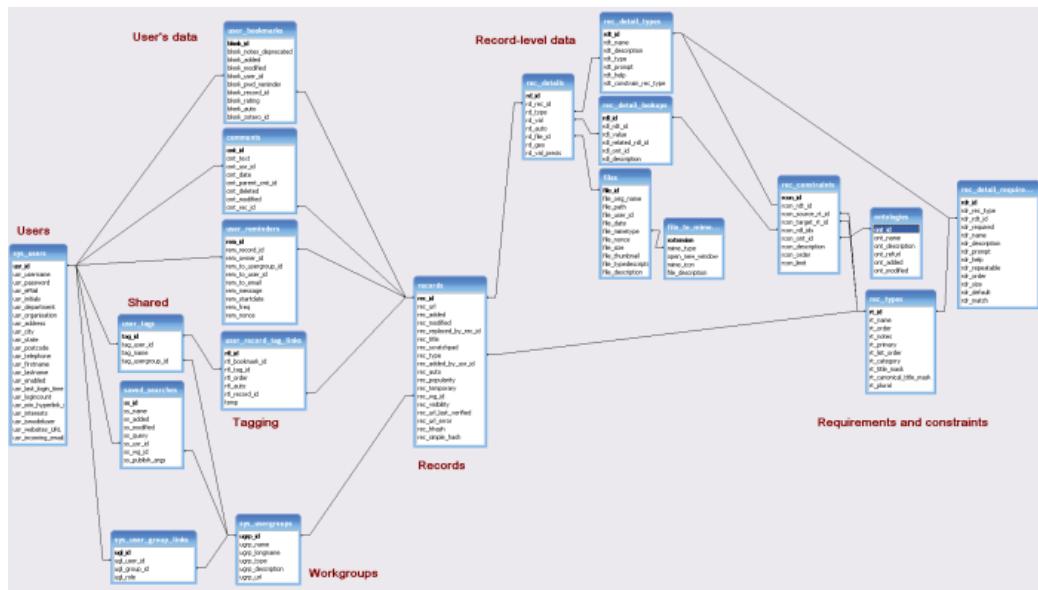
include a code example to create a low key grey background map.

Source code for the working map demo in Heurist that uses modified public mapping APIs has been provided separately.

P Documentation

In familiarising ourselves with the Ausstage database as the start of the project we made good use of the existing documentation. While most of this was very helpful we suggest the following revisions to provide more clarity for future consultants and systems integrators:

1. A few examples of working API calls (specifically the calls which generate the examples of output) to clarify the instructions (it is much easier to take a working call and modify parameters than to try and construct one from scratch using abstract specifications);
2. A full list of tables, fields, indexes and foreign keys/relationships and/or an ERD organised to promote readability (it is nearly impossible to follow the one generated by Oracle) - the example below is the structure of Heurist Vsn 2 (simplified through the omission of a few unimportant tables and relationship types);



3. A succinct 'plain english' description of the entities and their relationships to clarify the structure.

Items 2 and 3 would allow someone approaching the database for the first time to get up to speed with it and understand the structure much more rapidly than trying to unpick the structure from the end user manual, the existing interface and the existing ERD.

We recognise that the current structure is the end product of an extended accretionary development and believe that the exercise of writing a ‘plain English’ description will help focus attention on those aspects of the structure which are most in need of revision.

It would also be useful to output search metadata as internal documentation within the search results (that is, the type of search, the id searched for if applicable, and the title corresponding to the ID eg. type=Organisation, id=102, title=Melbourne Theatre Company). This is particularly required if you save a search as a file, so that you can later find out how you got to that set of results.

P Search and mapping of results

Integrated search

The final search should be an integrated experience which lead one smoothly into a map, with the ability to do another search and add the results to the map. It should be equally intuitive to search and map one type of entity as to search and map multiple types.

The Beta mapping searches are particularly confusing because of the separation of single search, multiple search and ID search, within a separate interface from the main search, with further division into entities and the embedding of Events search within Venues search.

While we recognise that the search attempts to respond to user needs directly by reproducing use-case pathways, such as enquiring about a particular contributor or event, we feel that this is at the expense of overall clarity and simplicity and risks multiplication of interfaces to respond to different pathways.

Single search and mapping interface

We recommend a single, integrated search leading to a single mapping metaphor, rather than separate searches organised around a textual view (the current search interface) and a mapping view (the beta mapping interface).

Apart from a simpler user experience, the integration of search and mapping reduces the amount of development work required by avoiding duplication.

Shifting focus

The underlying data structure is reflected in a shifting focus within the beta map search which

may cause some confusion:

1. Menu entry reads “Map by Contributor”
2. You get a list of **contributors**, with number of events and “Add” (or “Show Map” on Organisations)
3. When you add to map, you get a map of **Events**
4. What you are actually mapping is **Venues**
5. The popup box shows Venue > Contributor > Events in that hierarchy

Similarly, the current search lists Contributors when you select **Contributors** from the pulldown, but lists Events when you select **All Records**. For the first five choices, the search and the results reflect the same entity type. In the All records case, the search and the results are no longer in step.

To illustrate the potential confusion, if we search for Geoffrey Rush, we reasonably expect to see a Geoffrey Rush contributor record, but we don't. Instead we see Events. While we understand perfectly that the events listed were events in which he participated, and that's a bonus, it's not what we actually asked for and expected.

UseCase: I want to find a particular event, venue, work, person, organisation and all their associated information

Also, since it shows us Events in which he participated, why stop at that? We also might also reasonably expect to see Venues where he played, Organisations of which he is a member and Resources which mention him or are authored by him etc.

We therefore recommend that for the search screen:

either the All Records search list results for each of the entity types (preferably as five headings with a count and an expand/contract control to display the list, possibly first category opened);

or the All records search be separated into a distinct search and that the searches are made explicit along the lines of:

Search for:

[*list of entity types*] containing [*list of modes*] [Search]

Sort by [*sort order*]

Restrict by year [*years*]

Search for:

Events containing [*list of modes*] [] in any attached record

Sort by [sort order]
Restrict by year [years]

where *list of modes is* : any of the words | all of the words | exact phrase

Separate searches can give greater precision to the search, particularly if they allow search on specific fields, but require an understanding of the underlying data model (even if at a very simple level), which is fine for the AusStage researchers but not for the public. It may be a good idea to make this an Advanced level option only, and have All Records as the default search on the front page.

We recommend using the form ID:nnnn to search for IDs, and explaining this to advanced users, rather than making it a separate function, or to put it in an Advanced search area and/or place an explanation next to the search box.

It would be good to have fuzzy or soundex matching, that is if I type Jeffrey Rush I might find Geoffrey Rush.

Spatial filter on search

As Corey observes, it would be extremely useful - even in the basic search - to be able to filter geographically, so one could find, for example, all venues in Melbourne, or all events with Geoffrey Rush occurring at venues in Melbourne.

A possible approach to spatial filtering on the search interface would be as follows:

1. Record a lat/long bounding rectangle for each place in your database (if they don't have them already). This can be set easily by geocoding with a GIS such as ArcGIS;
2. Put a google map in a div on the search tab, hidden at start;
3. Add a checkbox 'geographic filter' (or 'place search' or 'filter by place' or) to open up the map div and make it an active part of the search criteria;
4. Filter the places by the current extent and display the current number within the map extent next to the map "Venues within map area = nnnn";
5. Add the spatial search to the query, along the lines of:
`MaxLong>135.6781 and MinLong<136.2545 and MaxLat>-33.5635 and MinLat<-33.1273`
(but don't make this visible in the search string);
6. The same approach could be used for temporal filtering, although this is starting to move towards an advanced search

Have a look at the Alexandria Digital Library gazetteer application for an example of filtering using the current map extent:

<http://clients.alexandria.ucsb.edu/globetrotter/> (under Advanced Time/space).

Mapping search results

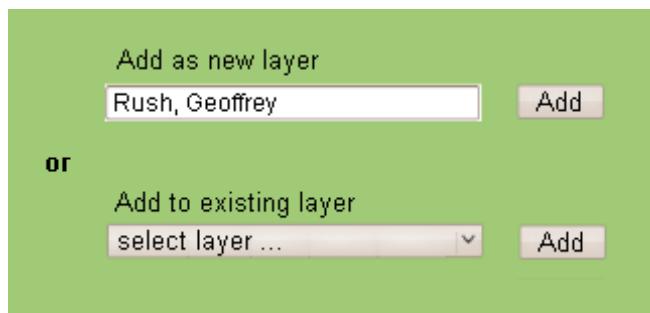
All search results should offer an Add to Map button, as in the Beta multi-contributor search. The map would show the cumulative picture of all search results added to it. This allows the user to progressively build a map visualisation through the application of searches.

Results might be added in one of two ways:

1. Individually, that is single Contributors, Events, Organisations or Venues (or several) are added as individual ‘layers’ which can be controlled and symbolised individually. The suggested mechanism is a checkbox against each entity and an **Add selected to map** button;
2. As a group, that is a set of entities from a search result can be added as one layer consisting of multiple items of the same type. In this case some capability to label the layer is needed so that the user can distinguish the results of different searches.

Option 2 allows the comparison of, for example, events of different genres through the creation of map layers for each genre. In this case the individual entities within a layer will all share a common visibility and symbolisation (eventually one might add some forms of thematic symbolisation within a layer if a use case developed).

In each case, it will be advantageous to allow addition to an existing layer or creation of a new one (pre-populate with name of entity if a single entity):



We suggest also adding an Add to Map button to each individual record page, along the lines of:

The screenshot shows the AusStage website interface. On the left is a green sidebar with the AusStage logo and a menu of links: Search, Browse, Contribute, Research, Industry, Related, Links, About, News, Contact, Help, and Staff Only. The main content area has a dark header bar with the text 'Gateway to the Australian Performing Arts'. Below the header, there are several tabs: 'Add to map' (highlighted in grey), 'Comment on this record' (with a green plus icon), and 'Login'. The main content area displays a record for an event. The event details are as follows:

Event Name	A Midsummer Night's Dream
Venue	Albert Park Amphitheatre, Brisbane, QLD
Umbrella Event	
First date	26 Sep 1979
Opening date	
Last date	06 Oct 1979
Dates Estimated	No
Status	Professional

In the Beta version, when you select multiple items to map, you simply get a map of all the events (actually venues) which have those contributors/organisations/etc. There is no way of telling which location corresponds to which of the mapped entities, or how many there are at any location. In a later section we discuss different approaches to displaying dense data while still giving access to individual entities.

Additional sophistication could be provided by a hierarchical organisation similar to that in Google Earth, allowing control of symbology and visibility at both the individual record and layer level. This would be technically demanding and it is not obvious that there is an adequate use case.

The current Beta map of all events is far too slow to load as a startup mapping screen (although we have not seen it with collapsed groups of entities).

New beta map search interface

Feedback has been provided during development of the new search and mapping interface,, which reflects many of the ideas put forward in this report. Consequently some of the suggestions are outdated in the sense that they have already been incorporated. We have left them in their original form as a record of the consultancy.

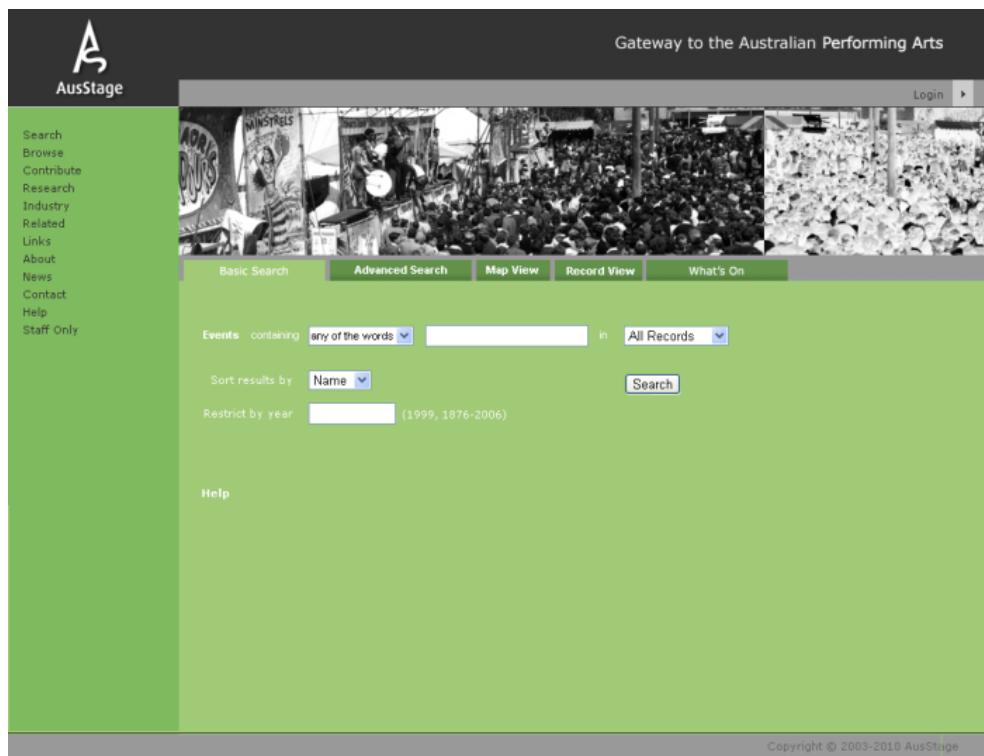
Navigation - search, map and record views

The beta search offers navigation to the textual result for the records retrieved (through a link on the name or title) with addition to the map over on the right of the listing, and once you have navigated to the textual results you have lost the search results. We would prefer to see specific entity records in a new tab within the AusStage interface (rather than a tab in the browser

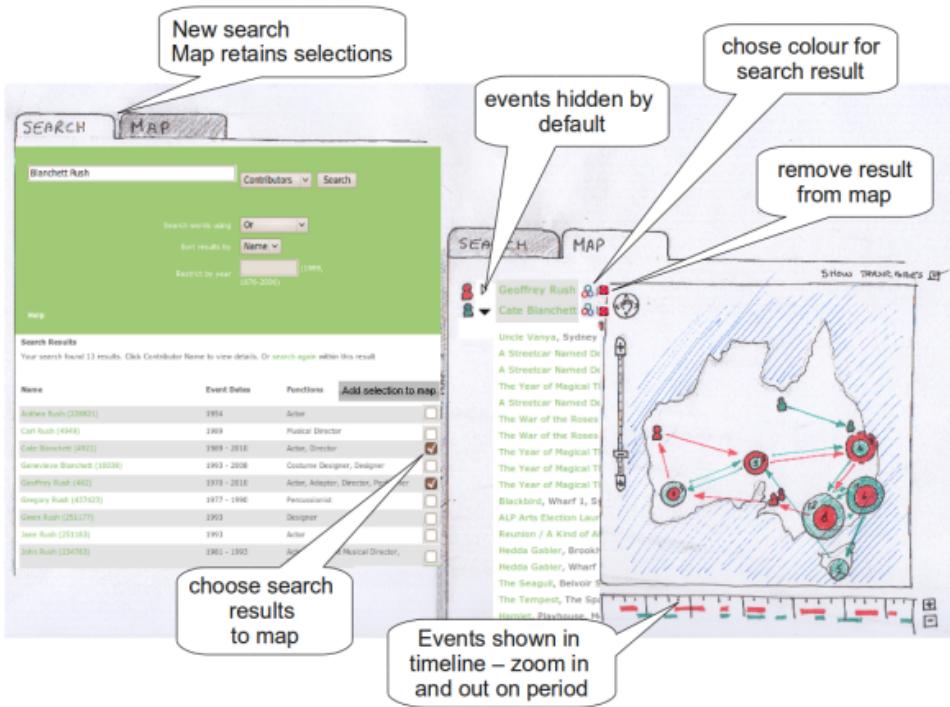
which risks spawning of multiple windows), allowing the user to quickly switch back for further searching, browsing or addition to the map.

Since you already use the tab metaphor, the use of tabs in this way should fit well into the existing interface. It overcomes the problem with the current Beta approach of adding to a map at the bottom of the list, which does not work at all well (first because you have to scroll down continually to see the result and up to add the items; secondly because scrolling the page almost invariably scrolls the map as well when using a mouse wheel - very annoying).

The illustration below shows a possible approach, including tabs for basic and advanced searches, a map view, an entity record view and a basic search function revised as suggested in the previous section.



In the illustration below [please ignore the layout of the map which is discussed later], we have simply used a combination of the existing textual search with the Beta mapping of multiple Contributors, to add search results to the map tab. The user can switch to and fro between the search view, the map view, and eventually an individual record view:



The Map tab should be available on the main interface whenever entities have been added to the map. It could be visible but rendered inactive when no results have been added (in which case it needs a popup message along the lines of ‘Please add search results to map to activate’).

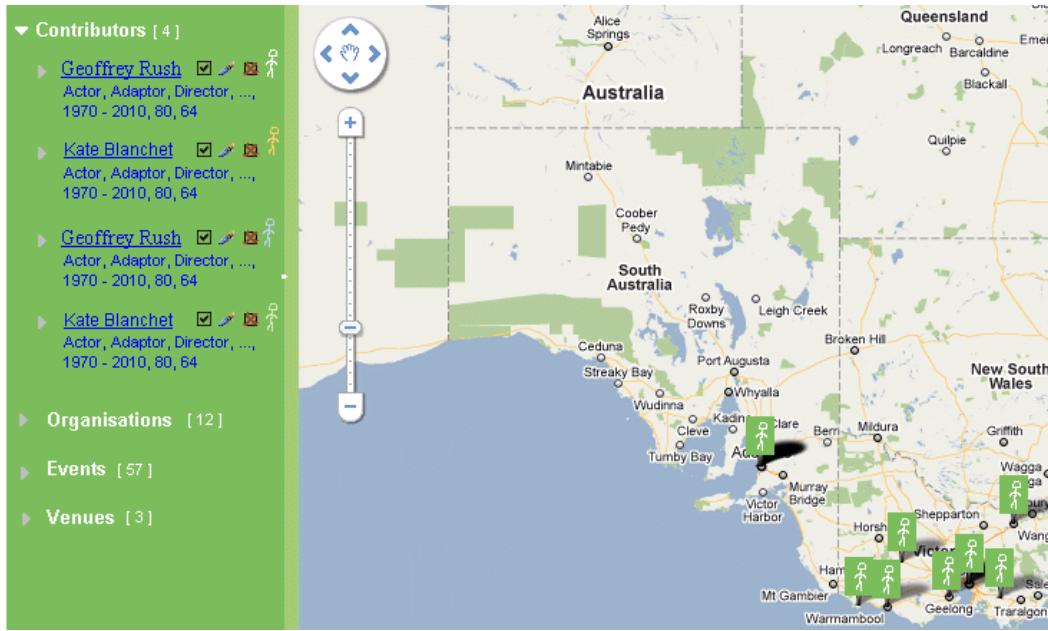
General clarification of location

We recommend a clear distinction between **locations** and **venues**, where venues are treated as a database entity equivalent to events, contributors and organisations, and that all four of these be mappable with their own symbols (currently events do not have a mapping symbol).

Cartographic presentation

We now turn to the structure of the Map tab and its legend. The following section will look at methods of representation of dense clusters, associated information and trajectories.

We suggest a fairly conventional legend/navigation on the left of the map tab. From a screen real-estate point of view, the map could be expanded to take up the full width of the screen, replacing the AusStage site navigation panel with the map legend, but the map and legend could also be displayed completely within the main panel as suggested by the previous mockup of tabs. The former is better for mapping, the latter is a bit clearer from the navigation viewpoint, and easier to implement, therefore we recommend this approach in the first instance.



The map legend should have the following functions:

- Show name and enough information to identify (see example above). It would also be adequate to simply display the names, and this is both simpler and avoids the need to interpret the summary information (which could be done with a key at the bottom of the legend panel, a popup legend key or rollovers on the individual entries)
- Show type of entity for each entity, using an icon <Irek to suggest icons>
- Toggle visibility of the entity on the map (conventionally an eye icon - Irek)
- Show entities in order added, most recent at top (default order)
- (optionally) group entities by type (all contributors together, all events together etc.), in collapsible sections as illustrated above
- Delete from legend (= remove from map)
- Change symbology (since entities are likely to have fixed icons, this will be primarily selection of colour, but could also include size and labelling)
- Show trajectory (see trajectories section).

P Map layers

The above discussion of search results leads directly into the use of map layers, both to represent search results and as background on which to visualise them.

It should be noted that **layers** on the map are completely independent from the representation of clusters of objects on the map, which we discuss in the next section. The layers come into effect:

1. to provide a key to the objects displayed (whether individually or as clusters) on the map;
2. to set visibility of entities that have been added to the map, ie. they are a filter which is applied before symbolisation/clustering is carried out;
3. to set and show symbology which will be used when entities are shown as individual icons.

Entities

In the discussion above, we have focused on the addition of individual search results - that is single records of Venues, Events, Contributors and Organisations - to the map, treating them effectively as individual map layers which can be individually controlled. Every entity added is thus their own layer with their own symbology (chosen by default to be different from any already on the map, so for instance using the palette sequentially) and behaviours.

What you could do is to provide some grouping mechanism so, for example, all entities of the same type could appear together and have a super-layer control to hide/show them, or you could allow drag-and-drop reordering (we think that would be the first thing worth adding if you have time) and possibly have super-layers into which entities could be dropped.

We think that the extra programming effort, and complexification of the interface, in attempting to group together similar entities into layers is probably not warranted at this time. Should a convincing use case develop at a later date, this could be revisited, but the sort of more sophisticated analysis that might call for such groupings is probably better handled by exporting data to a fullblown GIS.

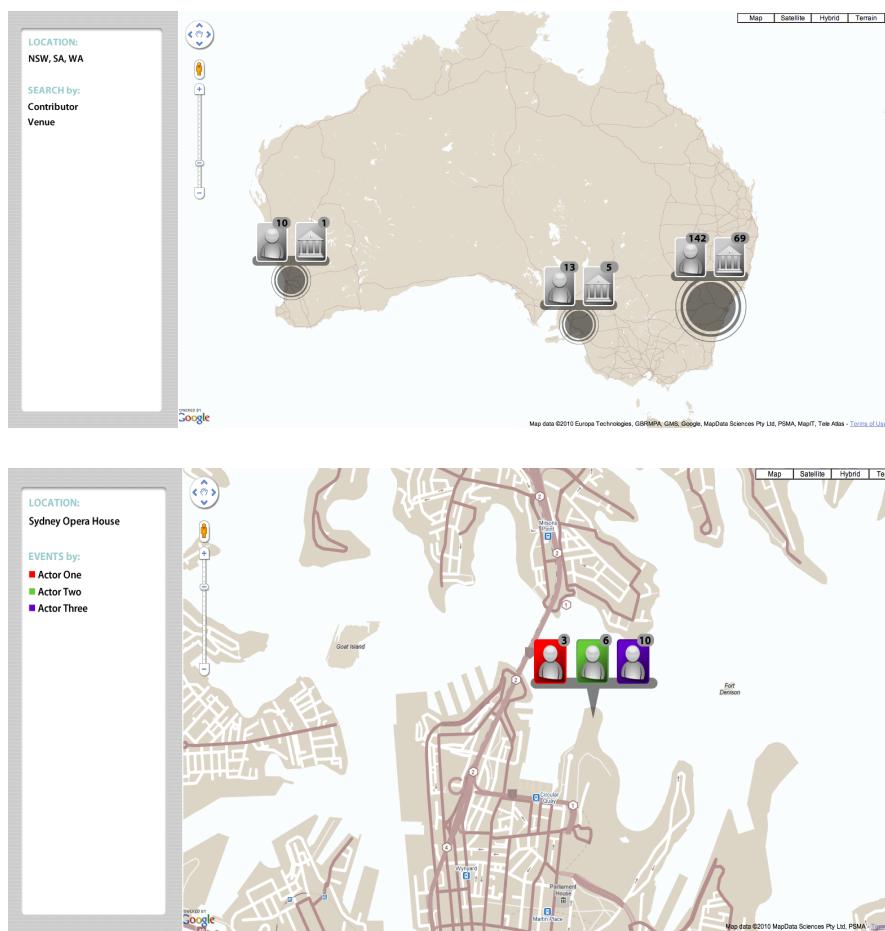
Background map

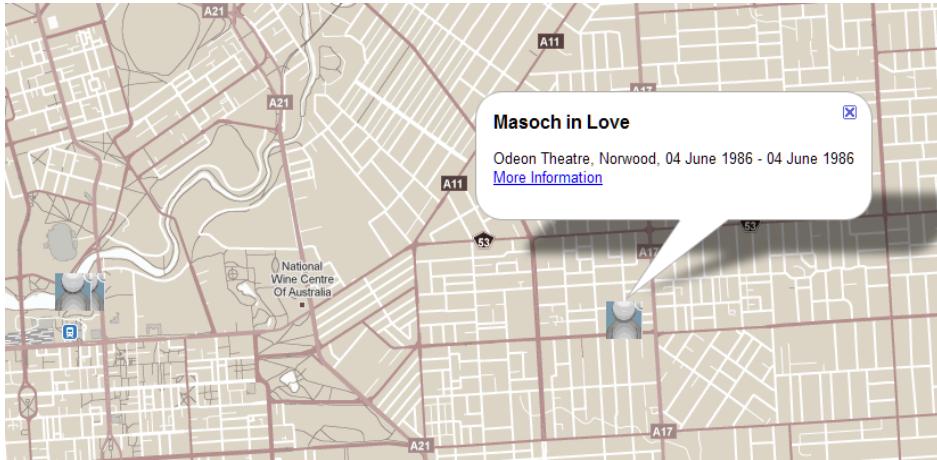
A key requirement is to mute the tone of the underlying map to improve readability. The new Google Maps API Vs3 allows selective presentation of the Google *Map* layer through the choice of a colour palette for many of the object types that appear on the standard map layer.

Although the terrain map is a good option for larger scales, use of it automatically blocks zoom in to local views, which is incompatible with the needs of AusStage mapping of venues. We

therefore suggest adopting the use of a simplified and toned-down Google Map layer as the default at all scales (one could also use Javascript to switch between terrain view and the standard Google map at high zoom, or to change the rendering of the Map layer, but this is significantly more work for fairly small benefit).

We note that Jonathan particularly likes the greyscale maps from the Literary Atlas of Europe. The illustrations below follow something of that metaphor using one of the proposed colour palette tones.





The following code (which we will provide, with eventual modifications, as a file) is used to generate the maps shown above:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0px; padding: 0px }
      #map_canvas { height: 100% }
    </style>
  </head>
  <script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
  </script>
  <script type="text/javascript">
    var map;
    var sydney = new google.maps.LatLng(-33.8,151.2);
    function initialize() {
      var stylez = [ {featureType: "poi.park", elementType: "all", stylers: [ {invert_lightness: true},
        {visibility: "simplified"}, {hue: "#ccff00"} ]},
        {featureType: "water", elementType: "all", stylers: [ {hue: "#00ffdd" },
        {visibility: "simplified"}, {lightness: 95} ] },
        {featureType: "landscape", elementType: "all", stylers: [ {lightness: -10}, {hue: "#ffb300"} ]},
        {featureType: "road.highway", elementType: "all", stylers: [ {visibility: "simplified"}, {lightness: 59}, {hue: "#ffffff"}, {saturation: -79}, {gamma: 0.44} ]},
        {featureType: "road.arterial", elementType: "all", stylers: [ {visibility: "simplified"}, {lightness: 20}, {hue: "#ffffff"}, {saturation: -79}, {gamma: 0.44} ]},
        {featureType: "road.local", elementType: "all", stylers: [ {visibility: "simplified"}, {lightness: 20}, {hue: "#ffffff"}, {saturation: -79}, {gamma: 0.44} ]},
        {featureType: "administrative", elementType: "all", stylers: [ {hue: "#0099ff"}, {lightness: 82}, {visibility: "off"} ]},
        {featureType: "poi", elementType: "all", stylers: [ {visibility: "off"} ]} ]
      var mapOptions = { zoom: 14, center: sydney };
      map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
      var styledMapOptions = { map: map, name: "austage low key colour concept" }
      var testmap = new google.maps.StyledMapType(stylez,styledMapOptions);
      map.mapTypes.set('austage1', testmap);
      map.setMapTypeId('austage1');
    }
  </script>
```

```

</head>
<body onload="initialize()">
<div id="map_canvas" style="width:100%; height:100%"></div>
</body>
</html>

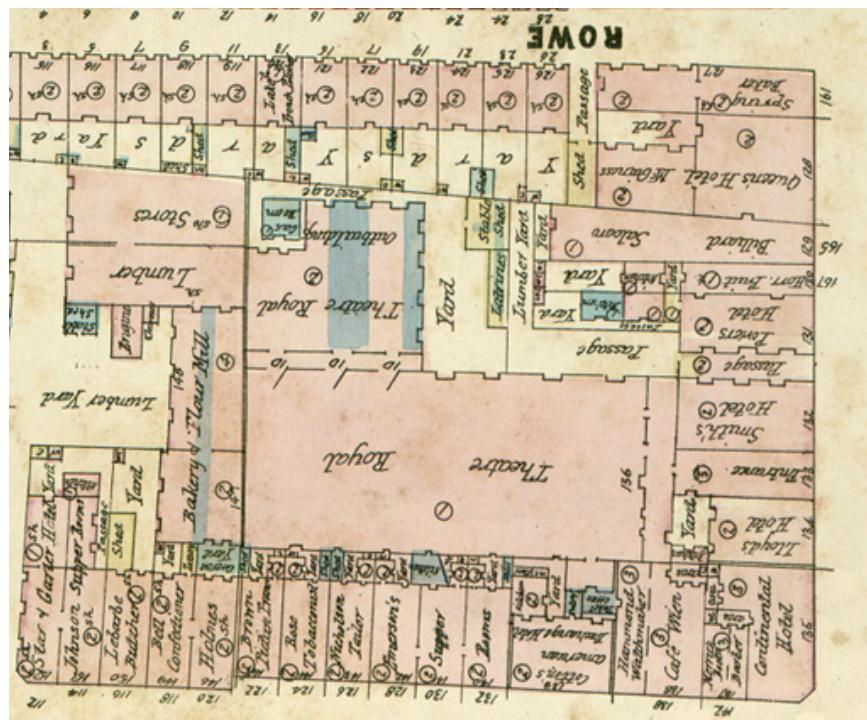
```

Census data

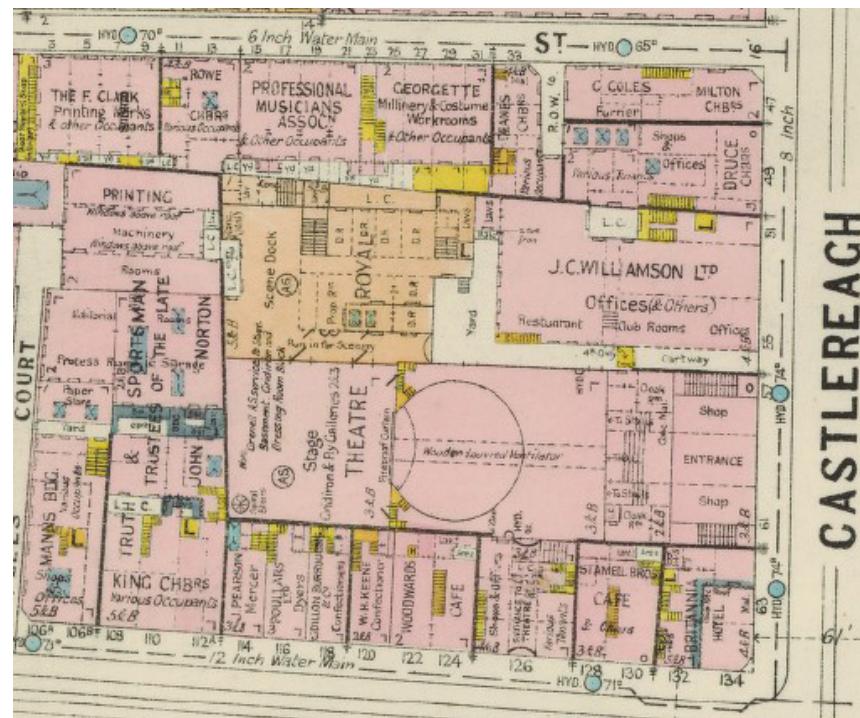
We doubt whether census data form a particularly useful background relative to the effort required to create map layers from them. There is very little direct relationship between the Basic Community Profile census data and the AusStage dataset, so while it is possible to generate background layers that represent population statistics it is not easy to see how they could be used.

Historic map data

As a future option we suggest that it might be interesting to provide a range of georeferenced historic maps of the major concentrations of theatres, and to tie these to the map zoom and timeline. In this way they could be displayed on top of the default modern background, when appropriate, to place historical venues/events in contemporary context. This would only be relevant when the map is zoomed in to specific city areas for which appropriate historical map coverage would be available.



Corner of King & Castlereagh Streets Sydney in 1880
showing the Theatre Royal and its stage door access.



The same corner in 1920 showing the theatre
and the adjacent J C Williamson offices.

Building and using prototype maps

Heurist

A Heurist workgroup has been created called Austage which can be accessed by logging in at:

<http://heuristscholar.org/h3-iq>

username: austage

password: blanchet

Note: this is an alpha development version of Heurist developed with Firefox as the target platform, so we recommend the use of Firefox, although Chrome also seems to be fine. Irek is testing the mapping functions on Firefox and Safari as he goes along.

A number of demonstration dataset searches can be accessed by looking down the left hand column of the application under **Workgroups > Austage**. To view a map click on one of the saved searches (for example “Blanchett test set”) and select one or more records using the standard click, shift and ctrl keys. Clicking on the map tab in the right hand pane should draw a map and render your selected items as a marker overlay. You can select or deselect items from the list and the map should dynamically redraw.

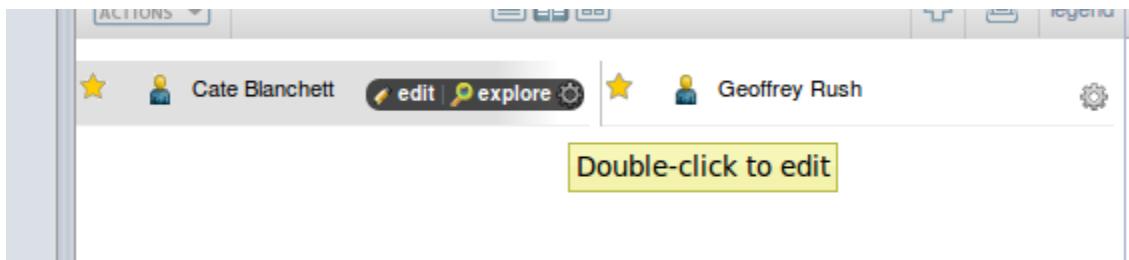
The screenshot shows the Heurist 3.0.0 application interface. On the left, there's a sidebar with navigation links like '+ Add Record', 'My bookmarks', 'All records', and 'Workgroups' (ACL, Austage, Workgroup page, Austage Blog, Saved searches (shared), Blanchett test set, Rush or Blanchett, Rush test set, Bali Paintings, Carly Todhunter, Dictionary of Sydney, eScholar Internship, Hayes M.Phil, Rethinking Timelines, Steven Hayes, USyd Archaeology). Below that are 'Example collections' and 'Collected: 0'. A central grid displays 12 record cards, each with a title, date range, and icons. To the right is a map of Australia and the surrounding regions, with markers for various locations. A timeline at the bottom shows the years 1990 and 1991.

Changing background colours and icons

We have endeavoured to provide the facility to easily change map icons and background colours in order that different designs can be assessed. To make such changes it will help to understand the data structure used in the prototypes.

Using the KML import facility in Heurist (accessed from the "My Heurist" menu item) data from Austage can be used to create records. In our case we have created "Event" type records. The Heurist KML parser has done a good job of matching field names to detail types in Heurist record definition. Each event record contains a performer detail which points to one or more person records - in this case Blanchett or Rush. Those interested in importing KML should bear in mind that the performer information was added later using a system called the detail updater. We are happy to explain this process as required.

You can access the person record for Blanchett or Rush by selecting the "Austage people" saved search. You can edit the record by hovering over the little gear icon at the bottom right of the record listing and selecting **edit** from the resulting menu.



Both records have a value set in the additional data section called Hexadecimal colour which will take any valid HTML colour code. The mapping application will use this colour value (if present) to render the background colour of an icon.

Cate Blanchett's person record also includes a thumbnail image which will be used if present in place of the standard person icon in the map.

A screenshot of a web-based form. It includes fields for 'Curriculum Vitae file' with a browse button and a placeholder 'Upload your CV here, and any other documents of relevance'. There's also a field for 'Reference to a heurist image record' with a remove button and a placeholder 'Reference to a heurist image record - use in preference to a thumbnail'. Finally, there's a section for 'Photograph (thumbnail)' showing a thumbnail image of a person, the filename 'cate.jpeg', its size '[1.6 kb]', and a delete button.

APIs for creating marker overlays

Each event record in the demonstration set has a spatial coordinate pair. An inverted white triangle is rendered using these coordinates with the lowest apex of the triangle sitting on the point. An open source javascript API called ELabel (<http://economy.org.uk/gmap/elabel.htm>) is used to render a second marker overlay, offset so as to appear to sit on the top of the triangle has the property of being able to take and render any reasonable HTML content. It is this overlay component that renders the image of Cate Blanchett in the example.

Full source code showing how we have implemented Elabel into our Heurist mapping process will be made available soon.

GMaps API version

At time of writing (2010-11-08) Google Maps Vsn 3 API was not compatible with timemap.js. The latter API is very useful for synchronising the behaviour of data on a Google map and a Simile timeline. For this reason the examples use GMaps Vsn 2 and do not make use of timemap.js.

Because the clustering and layer management functions of GMaps V3 are superior to Vsn 2 we

will eventually incorporate them into Heurist once timemap.js has been updated to work with Vsn 3, but that will be after completion of the consultancy. We recommend the the use of Vsn 3 and timemap.js when the compatibility issue has been resolved (see next section).

eLabel is not compatible with GMaps Vsn 3. As mentioned later in this document, InfoBubble is part of Vsn 3 and provides essentially the same functionality as eLabel in Vsn 2.

Assessing GMaps Vsn 3

While the absence of an updated timemap.js suggests that AusStage should delay use of GMaps API Vsn 3, we have formed a good overall impression of the API. It is more streamlined and object oriented than Vsn 2 and provides a few new interesting functions (e.g. the ability to natively read KML structures and combine these with other structures). There are however some limitations imposed by Vsn 3 - mainly due to the newness of the code.

Many existing APIs have yet to be migrated to the new version, the overall feature set is still small compared with the more mature API and documentation is still quite thin on the ground. There are many as-yet unanswered discussion group posts and this simply reinforces the feeling that it would be wise to wait a little while longer before fully implementing solutions based on this new API.

Steve White - our chief programmer - has however taken a close look at both APIs and has determined that modifying existing Open Source JavaScript code to work with the newer API is fairly straightforward, if a little time consuming. In any case, we expect that most of the libraries written for Vsn 2 will be migrated over the next six months.

Steve has focused in particular on libraries that provide html markers and clustering functionality to determine the impact of migrating these libraries to Vsn 3. He found that these libraries are based mostly on primary calls in the two API's, for which the functionality was available in both versions but in some cases accessed via different functions. This requires slight changes to the existing libraries. In one case a different object needed to be accessed to get the functionality, but the impact to the library code is still minor. It should be noted that these libraries are rather small and focus on a limited set of Google Maps functionality.

The upshot of Steve's investigations is that he feels that it is quite safe to continue developing applications in GMaps API Vsn 2 for now, as the transition to the new API presents few challenges (less than a day's programming for the two libraries) for a capable programmer.

Dynamically-generated markers

As proof of concept for developing a flexible iconic system, Steve was able to modify the two GMaps libraries (ClusterMarker.js and ELabel.js) in a matter of hours. These modifications are aimed at demonstrating that one can use information-specific markers (html-generated) in conjunction with clustering, rather than relying on pre-defined markers. Steve suggests that the modification for a callback hook to generate content-specific cluster markers would represent a couple of hours work for someone familiar with the libraries. Source code and commentary will be provided.

Dealing with density

Density clusters

The classic use of clusters is to cope with scaling by representing large numbers of icons appearing at one point on the map (eg Venues in Sydney on an Australia-wide map).

Density clusters can represent the number of entities at each point by one (or a combination of):

- Symbol size - typically non-linear scaling and/or minimum size limited
- Symbol colour and/or density - no more than 3 - 4 values can be effectively distinguished
- Numerical count

Density clusters can be made explodable so that clicking on them generates a daisy of individual items, as illustrated below. This is useful in providing individual clickable links to further data for each entity.

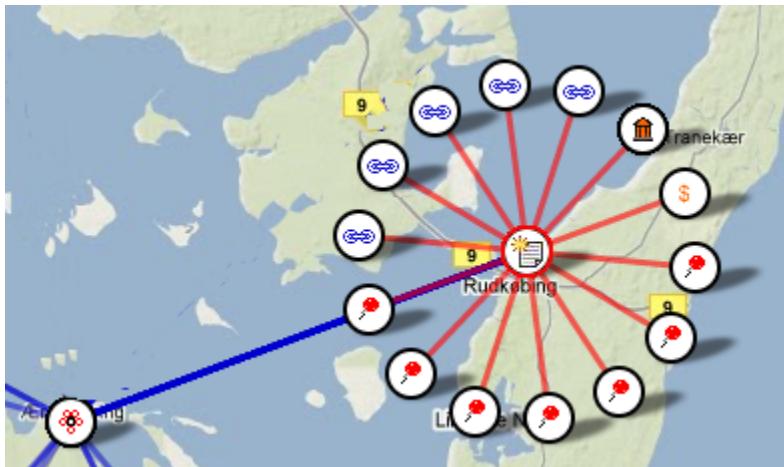
Google Maps API 3 density clusters simply show the number of all types of entity together; they do not allow users to see the count of different entity types at a given location until the marker is clicked. Other mechanisms need to be employed if users are to be able to see at a glance what they are likely to get when they click.

We have successfully used - and subsequently quite easily modified - a clustering library written for GMaps Vsn 2. Having studied the relevant parts of the Vsn 2 and Vsn 3 object models implicated in clustering logic we now feel confident that implementing what we have prototyped in Vsn 2, using the more recent Vsn 3 API, will be a straightforward and predictable task.

Relationship clusters

Exploded clusters offer more than just a method of dealing with density clusters. They also offer the opportunity to represent non-spatial entities (Contributors, Organisations, Works and Events) attached to a spatial location (Venue) as individual clickable icons. Since this is very much aligned with the conceptual structure of AusStage, we believe this may be a suitable device for representing AusStage data.

Exploded relationship clusters are scalable to display variable numbers of entities at a point and allow rollover or clicking on the icon to display detailed information and/or navigation.



In this notional example, the cluster is centred on a real location (a Venue) but shows Events, Organisations, Contributors and Resources attached to the venue. The cluster is indicated by a special icon which can be an animated GIF (eg a pulsating point).

Note that these clusters can be distinguished from density clusters through the use of different icons and/or the use of an animated GIF to contrast with a static density cluster indicated by symbol size and/or colour and/or count.

The code that we developed - which we will make available Open Source on the Rethinking Timelines website if you wish to make use of it - not only allows filtering of the clusters with time, but also the 'petals' of the daisy with time, including both the marker and the ray. It manages messaging in such a way that the cluster will collapse to a point if time filtering reduces it to less than 2 icons, but will re-expand if the time filter is changed back to include multiple icons.

Clustering multiple entities

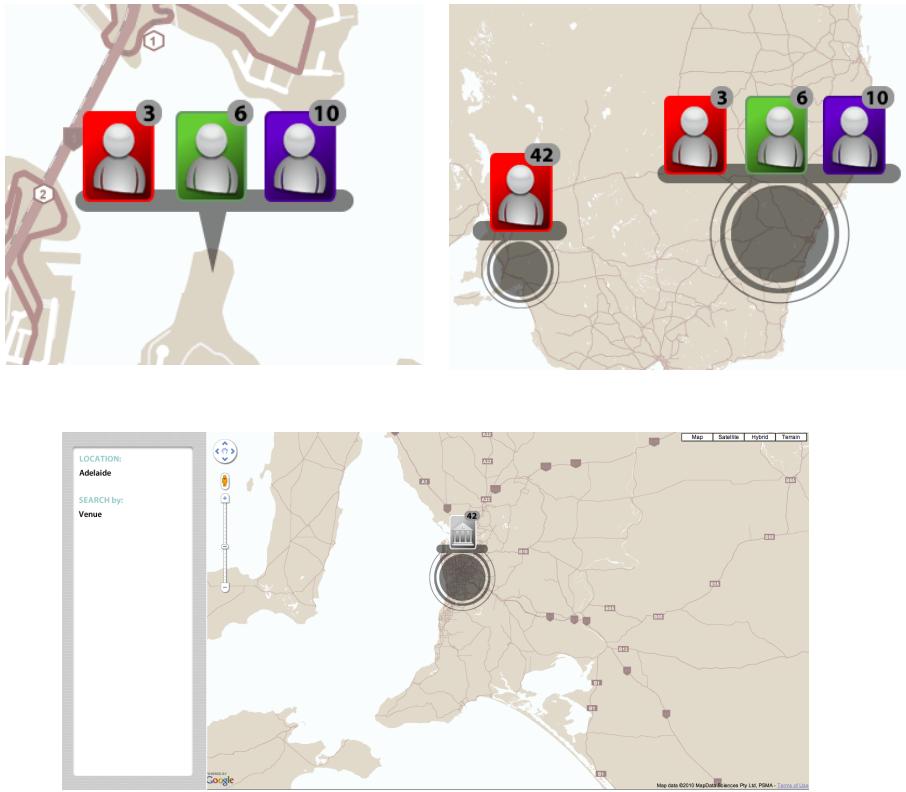
AusStage deals with four types of entity: Contributors, Organisations, Venues and Events. Venues are the only directly mappable entity, while Events linked to Venues provide temporal context to enable timelines to be generated.

In line with our general general recommendation that the search function should appear as an integrated system, we are proposing a similarly integrated iconography where combinations of all four entity types can appear on a map simultaneously and still convey meaning. The design challenge is therefore to devise an icon schema that allows a high level of information density while remaining simple clear and readable.

Original concept (October)

The following examples use a common contributor icon in comparison mode. The numbers

represent events (so there is an implied grouping) and the colour allows the user to quickly distinguish between contributors. All icons sit on a common “platform” which considerably simplifies implementation and at the same time clearly points to a venue.



When zoomed out to larger geographic area a clustering element replaces the pointer below the platform. This is more appropriate as comparisons are being made across a broad area - in this case Sydney and Adelaide. The size of the cluster element gives a very quick idea of the number of events (apologies for the “42” which somewhat contradicts this - 4 would be more appropriate) in a particular region. Events are probably a reasonable proxy for other quantities.

While there is no intrinsic limit to the number of icons and the range of colours that can sit on a given platform we suggest that users will learn to limit their selections to that which will create reasonably practical maps.

Revised concept (November)

Our iconographic language can be summarised thus:

- Search results are rendered on the map in the form of aggregations. Aggregations can consist of up to four icons, aligned horizontally;;

- The four icons represent one of the four possible entity types;
- A lozenge-shaped area beneath each icon may contain a number. This number represents the number of mapped entities (Contributors, Events, Venues or Organisations) associated with a particular geographic location or cluster;
- The aggregate icon sits on either an arrow pointing to a venue or on a circular cluster icon. The latter is located at the notional centroid of a clustered group. If present, the circular cluster icon will contain a number denoting the total number of venues <check with Steven, or is it entitites> grouped in the cluster.

In line with common practice in the use of clusters, clicking on the cluster will zoom the map in to a point where the cluster “explodes” into smaller components. This may result in a map showing individual locations, smaller clusters or a combination of both.

Base objects (primitives)

The proposed representational language requires the following seven distinct graphical objects:

 *	Contributor
 *	Venue
 *	Event
 *	Organisation
 *	Lozenge showing enumeration - rendered over the lower portion of the four entity icons as required
 *	Cluster including the number of locations - cluster markers will sit at the notional centroid of the clustered venues
 *	Pointer for non-clustered icon groups - the apex of the pointer to sit at the geographical location of the relevant venue

Example expressions (sentences)

	One contributor in six events at one venue. N.B. it is acceptable to exclude the venue icon as all non clustered icons will appear at a venue location
	Four contributors in 12 events at one venue
	Three events and one organisation at one venue
	A cluster of three venues with a total of five contributors and 16 events

N.B. A maximum of four icons can appear in the aggregate icon. One of the principal reasons for this is to ensure the clustering algorithm can work reliably. This is because the total “footprint” of icons to be clustered is an important parameter in the calculation of clusters. Keeping this footprint limited in size ensures that there are no unexpected results as icons and clusters are recursively recalculated.

Programmatically building icons

While most GMaps associated libraries provide methods to render custom markers, the iconographic language we are proposing requires a quite sophisticated method of building composite icons. For this reason we are proposing an “IconMaker” library which will encapsulate the rules of our iconographic language and supply the appropriate aggregate icon as required.

We have already confirmed the basic validity of this approach with some prototype code which is almost ready for demonstration (23 Nov). The following paragraphs describe the high level logic involved.

- Assume an array of ten items in a set of selected search results. Six of these represent events by a single contributor at a common venue, the other four represent events at

different venues by a single contributor;

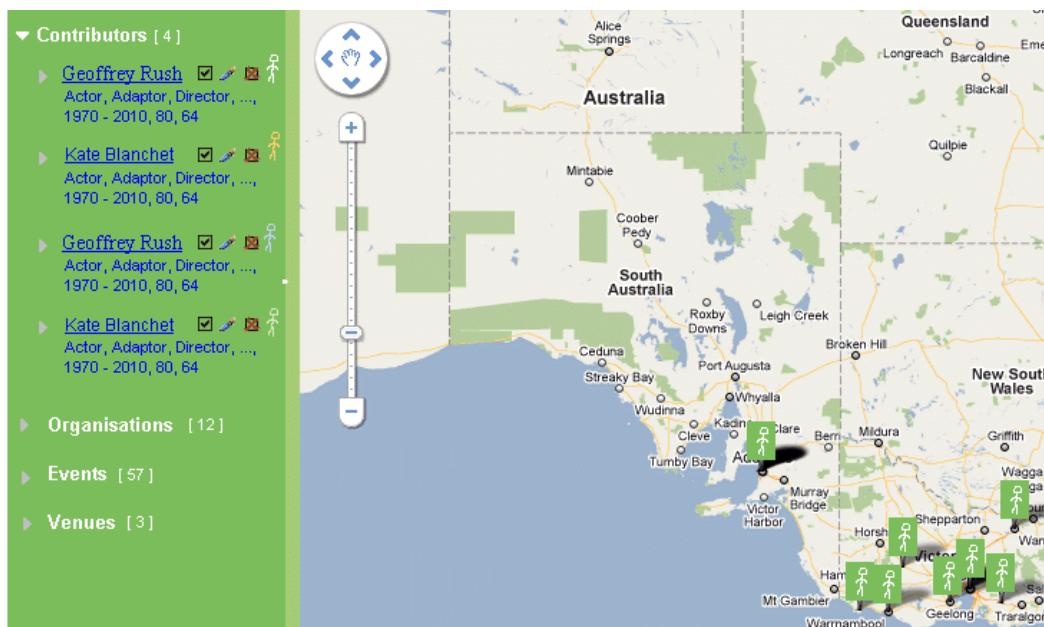
- Pass the array to the IconMaker library and it constructs five markers and associated icons. Markers can be loaded with any metadata we require and so the indices of the six items that make up the marker are also recorded. The result is one marker object with an icon denoting six events and four marker objects with icons denoting single events;
- Pass the new markers to a clustering library. This algorithm determines if any of the five icons associated with the markers will overlap at the current zoom level - this is why it is vital to pre-generate icons in the previous step.

Suppose that at a given zoom level the four single performance icons overlap and the one multi-performance icon stands alone. The clustering algorithm passes this information back to the IconMaker library which ignores the multi-performance icon and creates a cluster icon which combines the four single performance icons into one.

The result is two markers containing all the metadata for the ten search results encapsulated in two icons.

Map legend

Users are able to choose to add items discovered through searching onto the map via the *Add to Map* buttons. These chosen items will be displayed as a legend. We recommend a legend on the left of the map, since it provides navigation for the map, can be made scrollable to handle large numbers of layers and people are familiar with this metaphor.



Since there is no inherent connection between entities in the database, each entry in the legend

will represent one entity - a venue, an event, a contributor, a work. The advantage of such an approach is intuitive simplicity and the ability to see the relationships between entities, eg. where did one actor play and how does it overlap with the appearances of another.

The legend can also display entries for the background map(s), allowing these to be turned on and off or adjusted for transparency. In the first instance the background is likely to be simply the Google Maps map view adjusted for colour rendering. It may be useful to provide a transparency setting to allow the user to drop this layer back further if desired. It is probably worth including the layer at the bottom of the legend even if one can do little with it to give a sense of context.

Key map

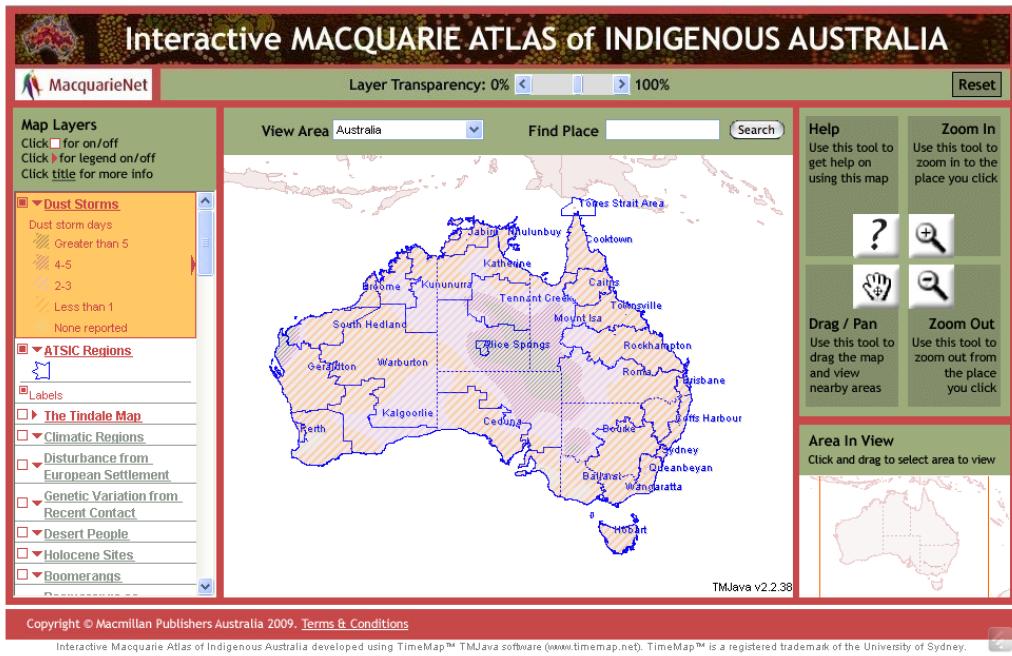
It may be useful to provide a small key map with navigation, as illustrated for the Macquarie Atlas of Indigenous Australia (below). [link to website](#)

Ordering and Grouping

If a user does several searches and adds all the entities found to the map, there is a risk that the legend will become extremely long. While long legends can still be scrolled, there are a couple of additional options to make more sense of this.

- New entries should be added at the top of the legend
 - It may be useful on the AusStage map to scroll the legend back to the top when new entries are added so that they are within the user's field of view. Ideally this would be smooth scroll which would only occur when the map tab was visible;
- Reordering
 - It should be possible to reorder the entries by dragging them up and down, either through hold and drag on the text or on a specific drag point/icon.
 - It would be convenient - but is probably unnecessary - to be able to drag multiple layers.
 - If this is implemented it would be useful to automatically select any newly added layers;
- Deletion
 - It should be possible to quickly and easily delete entries in the legend, preferably as multiples.
 - Deletion of single entries should probably be immediate (preferably with Undo), but multiple entries should have a verification warning;
- Groups by type
 - It would be possible to form expand/contract groups of entities by type, as illustrated in the legend above. This allows immediate simplification of long menus. However it makes it hard to manage manual reordering;
- Group by addition
 - Entries could also be grouped into a higher level layer whenever two or more entries are added to the map simultaneously. These groups would need to be fissionable. We think this adds needless complexity and is unlikely to reflect real

user needs.



[link to website](#)

Individual legend entries

A considerable amount of functionality can be packed into a legend without making it incomprehensible.

- Entity type icon
 - The standard icon usefully distinguishes different types of entity in the legend, and displays colour coding for individual entities.
- Summary information
 - The example above illustrates a way of packing a brief description into the legend entry. This can be supplemented by a popup box on rollover.
- Expand/contract (show/hide summary/symbology)
 - A rotating triangle next to the entry can be used to hide or display the summary information.
 - When collapsed the legend entry should be restricted to one line, using an ellipsis to indicate truncation.
 - The layer icon can also be placed within the expandable section, but we think it is better displayed at all times, along with controls below.
- Rollover
 - On rollover, popup more detailed information about the entity eg. full name or title, address of venue, number of events etc. Could also include a thumbnail image where this is available.

- Alternatively, rollover could serve to highlight the entity and clicking could open a popup window (which might also include navigation). However, this does not permit multiple selects.
- Selection
 - Clicking on the text of the entry (label, summary information) should select and highlight the entry in the legend. It should simultaneously highlight all occurrences of the entity on the map and timeline.
 - It would be useful to allow multiple selection with the normal shift and control modifiers.
- Navigation
 - Each legend entry should provide a navigation, or view page icon, which opens the full textual record for the entity. Ideally the record would be opened in a tab in the interface, but failing that a re-used separate window would be OK.
- Show/hide map layer
 - It should be possible to temporarily hide a layer without deleting it from the map through a visibility checkbox (most likely an 'eye' icon as used in most drawing software).
 - Hiding a map layer could also advantageously apply an automatic contract operation; the user can re-expand the entry if they wish.
- Delete map layer
 - It should be very easy to remove a map layer so that the user can avoid unwanted accumulation of layers in one session. This could be achieved either with the delete button (operating on all highlighted layers) or with a delete icon next to each entry.
 - There's some risk users might not realise that they can delete highlighted layers, so a delete icon may be needed in parallel with or instead of the use of the delete key.
- Change symbology
 - Since there are standard icons for each entity type, symbology will probably be limited to colour. Colour can be assigned automatically by selecting colour sequentially from the standard palette. Clicking on the icon could give a pulldown of colours from this palette with a 'custom' option, as is used by many programs.
 - Transparency could be used, but is probably not relevant to entity layers. However it might be relevant to the background map in the case that you want to include, for example, historical map layers on top of the Google Maps underlay. Transparency could replace colour as a pulldown slider from the layer icon.

Interaction

It is important that the map window respond dynamically to changes in the legend, eg. switching a layer on and off, rather than requiring a full screen redraw. In this way users will be able to easily compare subsets of their chosen search results and discern patterns.

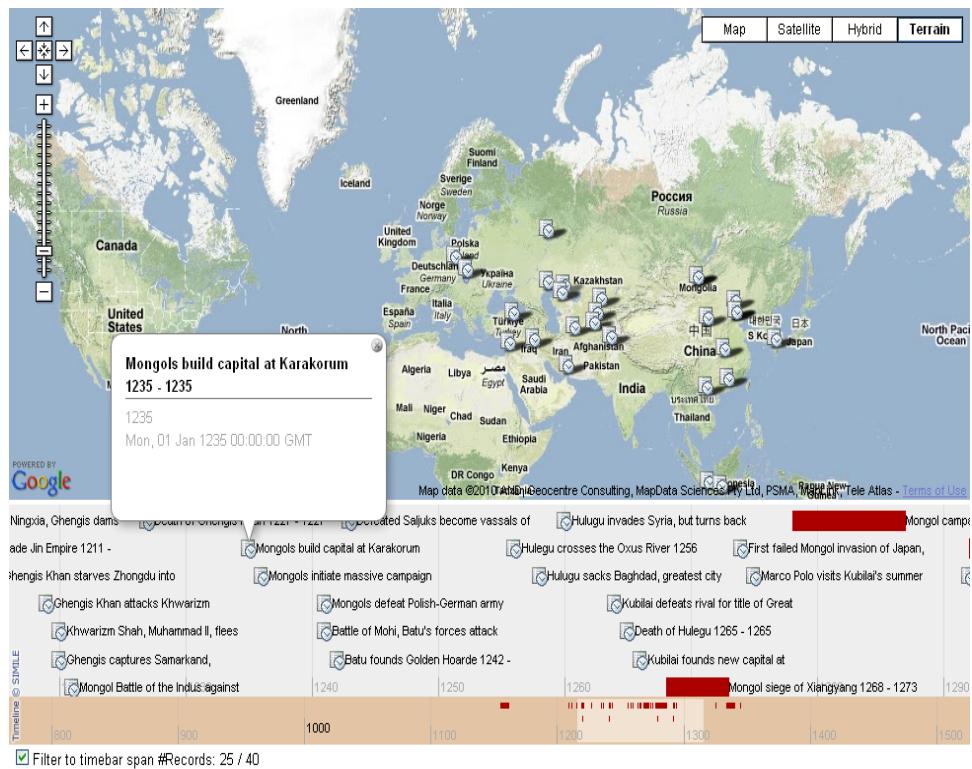
Timeline

We suggest the use of the Simile timeline to provide a temporal view coordinated with the map view, as illustrated below (from the Heurist map interface).

Note the checkbox **Filter to timebar span** which filters the entities on the map to those which fall within the current view of the timebar. As the timebar is dragged to and fro, the map is filtered dynamically, leading to the possibility of animation which can be a powerful mean of discerning temporal pattern on an otherwise busy map.

The degree of filtering can be changed by zooming the timebar in and out - timebar zoom in/zoom out buttons need to be placed in proximity to the checkbox, along with an explanatory text along the lines of

[] Filter to timebar span [+] [-] Narrow/enlarge and then drag timebar to change time filter



Navigation to further information (via popup) should be available on both map and timeline.

In all our prototype work we have relied on a relatively simple implementation of the Simile

timeline API that we have been using for general map viewing in Heurist.

In some of our more customised interface work (eg. the Dictionary of Sydney, specifically <http://www.dictionaryofsydney.org/map/24129>) we have integrated Google maps and Simile timeline functionality using a third party Javascript API called **timemap.js*** (available from <http://code.google.com/p/timemap/>). This library enables user interaction with a timeline to be reflected on an accompanying map and vice versa. We can supply an example of its use with PHP from the Heurist mapping interface.

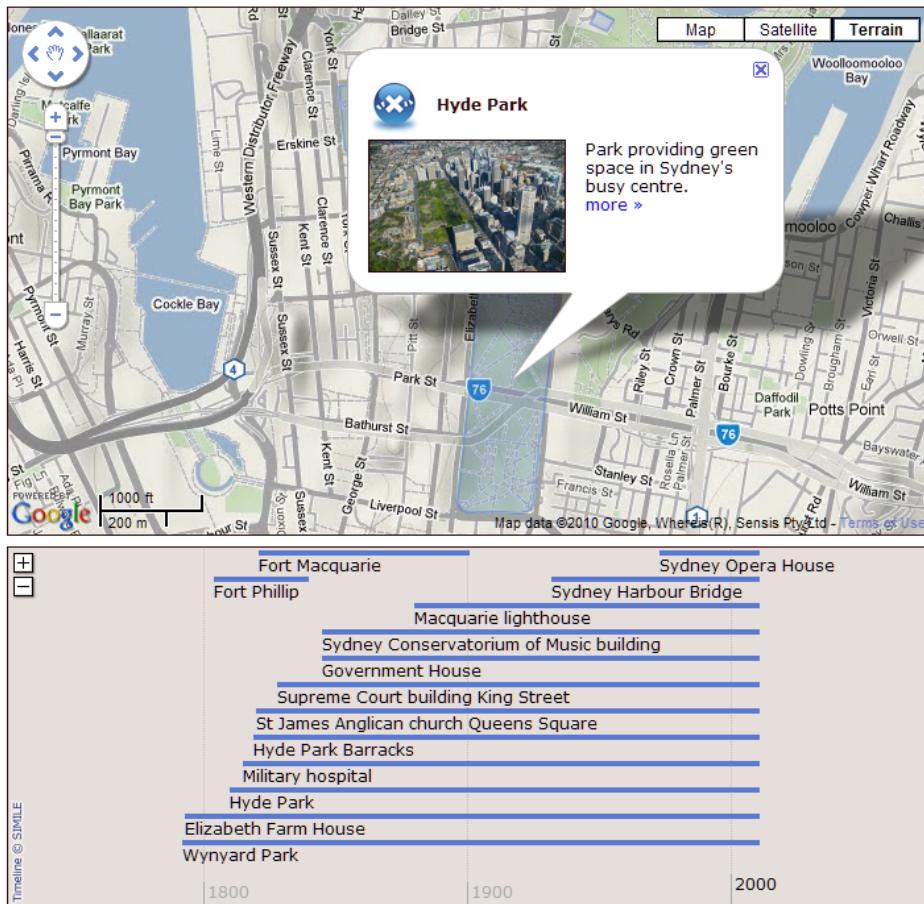
(footnote*: not to be confused with TimeMap™ (<http://www.timemap.net>), the ACL's Java mapping framework developed nearly 10 years ago, and still actively supported)

Unfortunately timemap.js is one key library that is currently not available in a form compatible with GMaps API Vsn 3. Nick Rabinowitz, the author of timemap.js, will update it to work with GMaps Vsn 3 with a proposed delivery date of 7th January 2011, for a maximum cost of US\$1500. AusStage has agreed to fund up to A\$1500 and ACL will cover any shortfall.

Note: Clicking on the map and clicking on the timeline should both pop up a box on the map rather than in different places; an example of this behaviour in action can be found at

<http://timemap.googlecode.com/svn/trunk/examples/artists.html>

Take for instance the case of a time series of events occurring at one place. A well-designed timeline should provide clickable elements that highlight and produce an informative popup window over the location of the relevant venue, tying the map and timeline together. Various other techniques are also made possible through timemap.js linking.



Interaction with timeline

All event durations should be shown on a timeline below the map. They should appear as a horizontal bar with a solid colour chosen to fit in with the overall colour scheme. Clicking on a bar should generate a popup window on the map situated over the relevant venue or cluster. The popup window should provide full details of the event and venue.

The timeline itself should include a zoom in and zoom out feature which will change the time scale displayed and the arrangement and form of icons on the map. For example, if fourteen events shown at one venue are spread over a period of five years, zooming the timeline into a maximum range of two years will dynamically reduce the number of events displayed on the map. As the timeline is zoomed all of the numbers displayed in the lozenges and cluster icons should change dynamically.

These display changes are performed by filtering a client-side array of temporo-spatial objects reflecting the selected query results (we can supply a code example). It should be noted that we are not proposing that the set of query results be regenerated server-side through user interaction with the map or timeline; interaction simply displays different aspects of selected

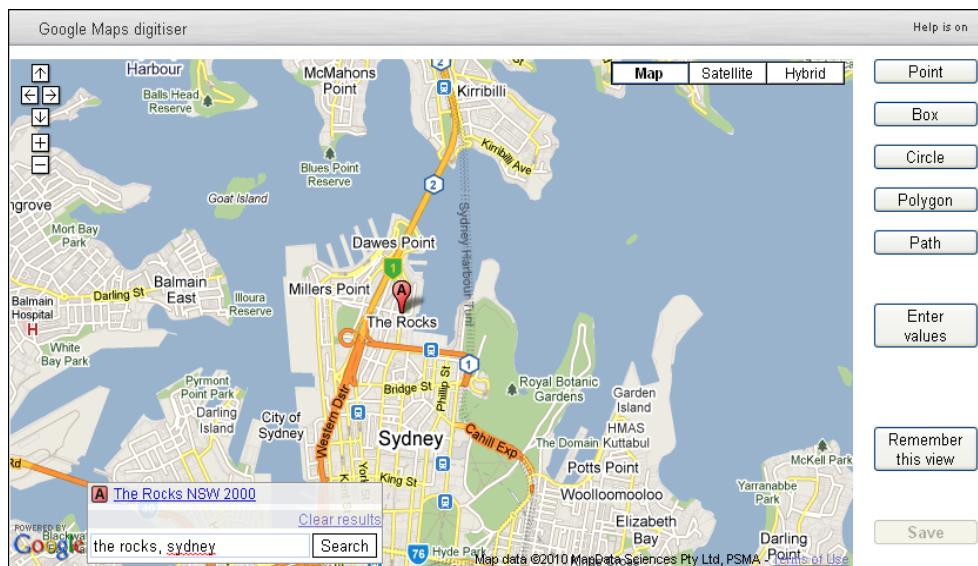
search results. It is possible to perform database queries through map and timeline interaction - www.realestate.com.au provides a working example of this - however we believe that the search and select workflow we are proposing will not work well if it is regenerated whenever the user changes the spatial or temporal context.

Positioning/spatial search

UseCase: I arrive at the map interface and want to navigate to a place/town.

Once you have the map you need to be able to navigate rapidly to a location - whether a town/city or a specific location. Here are a number of alternatives which could be used in the AusStage interface:

- Put in a Google search and zoom to location, as in our digitising tool (which also has a further behaviour, viz adding a popup with an action - creating a point):



- Zoom-to-this-location links within the marker popup, along the lines of:



- Drag rubber band box to zoom

KeyDragZoom (Google Maps Utility Library): This class allows a user to zoom in to an area on the map by holding down a hot key (e.g. Shift) while dragging a box around the area. Version 2.0 adds an optional visual control for turning zoom mode on and off. See

<http://google-maps-utility-library-v3.googlecode.com/svn/tags/keydragzoom/2.0.1/examples/visual.html>

- this would probably require selecting a special tool on the map, since people are more used to dragging the map around and double-clicking to zoom.

UseCase: I want to find an upcoming event in a particular area

This combines the timeline with jump to place functionality. Seeing events on a timeline associated with the map would give a sense of what's available within close spatial and temporal reach.

P Popups and Rollovers

Rollovers

We suggest using rollovers on markers to provide a first level of information, followed by a click to obtain further information in a popup, as at present. Use of rollovers will make the map more responsive to enquiry.

Rollover information could be as simple as: “Theatre Royal, 27 events, 35 contributors” but could also include a listing of events and an instruction ‘click marker for full listing’.

Fortunately, placemarks in Google Maps support rollover actions, although some Javascript coding is required (see <http://code.google.com/apis/earth/documentation/placemarks.html>).

Rollovers can also be implemented in the Simile timeline widget - see for example http://groups.google.com/group/simile-widgets/browse_thread/thread/4e3313b2bc300992

(although this will probably require some modification to the Simile code).

Layout of popup

Currently, the presentation of the results in the popup changes according to the search path. For example, if you search on an Event you get Venue followed by an Events, whereas if you search on a Contributor you get Venue followed by Contributors followed by Events. We find this somewhat confusing, particularly as there is little differentiation in style between the different entity types in the list. We therefore recommend:

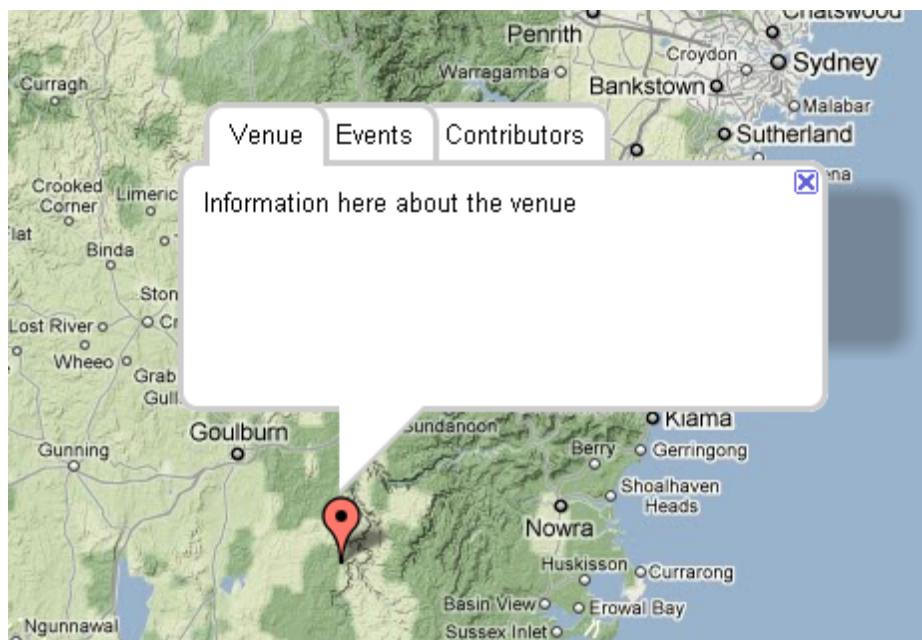
1. Maintaining a fixed layout of the popup dialogue, starting with Venues at the top (since this is what is represented by the marker), followed by Events (since this is central to all searches), followed by the Contributors to each event.
2. Making a clear style distinction between these categories. If they occur as a top-to-bottom list (as at present), the Venue would sit at the top with Events as headings and Contributors as list elements. Each entry should also have a small version of its corresponding type icon next to it to reinforce the sense of structure.;
3. An alternative approach would be the use of a tabbed information box (see below) to separate out the information for each entity type, but this would not allow the nesting of Contributors within each Event.

We also recommend including shortcut zoom links in the popup as illustrated under Positioning/Spatial Search.

<<To do: ask Irek to have a look at the popup and make further recommendations>>

InfoBubble

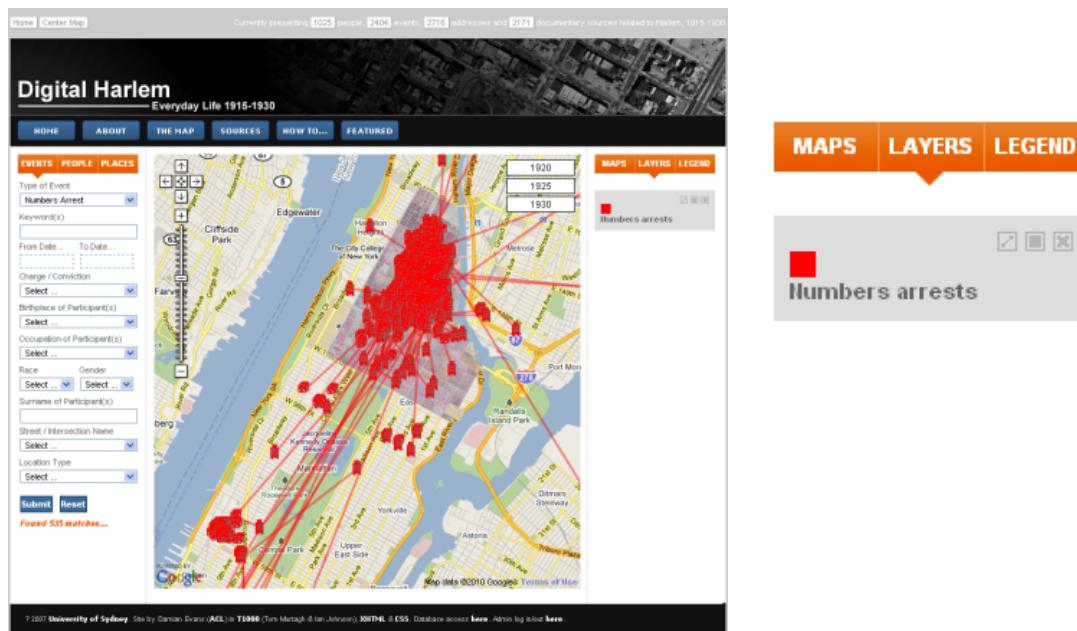
InfoBubble (<http://google-maps-utiliity-library-v3.googlecode.com/svn/trunk/infobubble/>) gives the opportunity to put multiple tabs on a popup. This could be used to separate out the information for the Venue, the Events associated with the Venue, and the Contributors at that Venue.



Trajectories

P UseCase: I want to show trajectories of a production eg. from city to regional centres and/or changes in the composition of a production (eg. change of actors in different venues).

One way of handling trajectories is that shown on the Digital Harlem application (<http://acl.arts.usyd.edu.au/harlem/>) illustrated below. In this case, there is a control on the legend (shown enlarged on the right) which switches on connections between (in this case) the location of events and the residential location of the participants. We suggest that a similar control be included on the AusStage map legend, defaulting to Off.



The current Digital Harlem application simply draws lines without time filtering. We are currently working on a modification to draw directional arrows between events (crimes in this case) to indicate their sequence, and should be able to illustrate that next week.

It should not be hard to add arrows to the AusStage trajectories, since the direction can be determined from the dates/times, but the arrows are only likely to be informative when dealing with fairly sparse sets of connections obtained by selecting entities with a small number of events and/or through temporal filtering.

The Harlem example illustrates the problem with many highly clustered points (more than 500 in this case) and a mixture of short and long range trajectories; individual points cannot be distinguished when zoomed out, but complete trajectories cannot be seen once zoomed in. A better approach to trajectories is therefore to combine trajectories with time filtering using an

interactive time bar such as the Simile timebar shown in the Timelines section above.

Rollovers

Rollovers on markers could be used effectively to highlight trajectories which start/end with an event and to provide further information on the event. Further discussion in Popups and Rollovers section.

Tours and productions

While tours and productions would be particularly useful in linking together events to plot trajectories, the current data structure does not permit this. Once the data structures have been developed (following the strategy recommended in the report by ???? it would be possible to add Tours and Productions to the map legend and use these to switch trajectory lines on and off. It would then be possible to further time filter the individual elements of a tour or production.

Miscellanea

We really like the ability, at the bottom of the search results, to do the same search on Google, Picture Australia and Australia Dancing (presumably that could be extended to Trove, and to other types of search - there's a nice example in the ECAI Iraq Atlas of using Cheshire to do library searches and return resources, see <http://ecai.org/iraq/EraEmpireProfile.asp?EraID=1>).

It would be good if this - as well as the Add to Map button, was also available on all the Entity pages, and was a bit more prominent (eg. at the top of search results rather than the bottom ?).