



## Milestone 3.3.1 – Requirements and Specification

---

This Document (M3.3.1) describes the requirements for the spatio-temporal Interface for Europeana (WP3.3). This end user interface should provide a improved experience for retrieval and visualisation of the contents of the Europeana.



co-funded by the European Union

The project is co-funded by the European Union, through the **eContentplus** programme

<http://ec.europa.eu/econtentplus>



Österreichische  
Nationalbibliothek

EuropeanaConnect is coordinated by the Austrian National Library



### Distribution

Version	Date of sending	Name	Role in project
0.1	12.02.10	UGOE – Christian Mahnke	
0.2	19.02.10	UGOE – Christian Mahnke	
1.0	25.03.2010	Liferay, all Partners	

### Approval

Version	Date of approval	Name	Role in project
1.0	22.03.2010	Bram van der Werf	Technical Lead

### Revisions

Version	Status	Author	Date	Changes
0.1	Draft	UGOE – Christian Mahnke	12.02.10	Initial Version
0.2	Draft	UGOE – Christian Mahnke	19.02.10	<ul style="list-style-type: none"> <li>• Now using the template (As proposed by Veronika Praendl-Zika)</li> <li>• Added additional external libraries</li> <li>• Added Stakeholder Europeana.Connect (as proposed by Gitte Petersen)</li> </ul>
1.0	Final	VPZ	22.03.2010	



## Table of Contents

Table of Contents .....	3
Problem Statement.....	4
Stakeholder Descriptions.....	5
Stakeholder <i>Users</i> .....	5
Stakeholder <i>Europeana.Connect</i> .....	5
Stakeholder <i>UGOE</i> .....	5
Feature or Functionality Overview .....	6
Needs and Features .....	6
Need Basic Visualisation of a result set.....	6
Need Graphical Query builder .....	6
Need Enhanced visualisation .....	6
Need Data Editor and sharing .....	7
System Qualities.....	7
Usability .....	7
Reliability.....	8
Performance .....	8
User Interfaces .....	8
Look & Feel.....	9
Layout and Navigation Requirements .....	12
Interfaces to External Systems or Devices ¶ .....	13
Software Interfaces ¶ .....	13
Communications Interfaces ¶.....	13
System Constraints .....	13
System Compliance.....	13
Licensing Requirements .....	13
Legal, Copyright, and Other Notices .....	13
Client side.....	13
Server side .....	14
Applicable Standards.....	14
Implementation .....	14
WMS/WFS:.....	14
REST:.....	14
QA.....	14
System Documentation .....	15



## Problem Statement

The problem of:

1. There is no way to visualize the results of a search in Europeana in the context of several location and time related attributes of an specific object.
2. There is no way to create a query based on a given time and location using a graphical interface.
3. Users can't add own objects to the contents of Europeana to see how they relate (in terms of time and space) to existing objects inside Europeana.

Affects:

- The users (general public and researches) of the site.

The impact of which is:

1. Not being aware of the context of a Europeana object in time and space. Not being able to search
2. Even the current advanced search doesn't support a search for a given location, only a date. The result set can be drilled down to a specific country, which is not very accurate (at least from the point of view of researchers) since it only works for currently existing countries.
3. For research purposes there should be the possibility to add user-generated objects to a user specific repository (not the main database).

a successful solution would be:

An interface able to display objects based on their attributes on a map and a time bar, both elements should be able to communicate with each other. The user should also be able to create queries using the interface. It should be possible to add user-generated objects to be able to visualise the relations to objects within Europeana.

**For:** Users (general public)

**Who:** Are either looking into a interesting way to visualise their result set or want to create queries based on the dimensions time and space.

**For:** Researchers

**Who:** Are interested to visualize relations between Europeana and custom created objects.

**Product name:** Spatio-temporal interface (WP 3.3)

**That:** Provides a innovative way for visualising search results and provides a way to find objects for a given location and time.

**Unlike:** We are not aware of any with integrated time based access mechanisms for visualisation and retrieval.

**Our product:** The spatio-temporal interface is a never seen, but highly innovative approach to represent contents of the Europeana – in 4 dimensions.



## Stakeholder Descriptions

Stakeholder Summary

### Stakeholder *Users*

**Name:** Users (general public and researches).

**Description:** Users of the Interface

**Responsibilities:** The users of the spacio-temporal interface will hopefully give feedback on usability issues. Later on they are responsible for creating user generated interface specific content, like data points and saved searches.

### Stakeholder *Europeana.Connect*

**Name:** Europeana.Connect project

**Description:** The interface is developed as part of the Europeana.Connect project. The stakeholders interested to get the spacio-temporal interface incorporated in a future release of the portal software.

**Responsibilities:** The projects monitors and coordinates overall progress.

### Stakeholder *UGOE*

**Name:** UGOE (Europeana.Connect project partner).

**Description:** UGOE is responsible for WP 3.3.

**Responsibilities:** UGOE is responsible for development of the described interface. The stakeholder is interested in maximising the possibilities for reuse in it's own environment and thus is focusing on a complete client side solution on top of well-known server components. A part of this approach is the usage of REST based interfaces.

## User Environment

The user is able to use the spacio-temporal interface from any modern Browser that is able to execute JavaScript. We are currently targeting three major web layout engines: Trident (Internet Explorer) (Version 6 or later), Gecko (Firefox, Seamonkey, Camino) (Version 1.8.1 or later) and WebKit (Safari, Chrome) (Version 522.11 or later).

Mobile Browsers are currently out of scope since there are several restrictions on the user interface like size and the lack of a possibility to click and drag.

We expect future browsers to be backwards compatible with current versions of JavaScript.

On the server side we use Java to provide a maximum of portability and to ensure compatibility with future platforms.

## Feature or Functionality Overview

### Needs and Features

The following three items reflect the internal plantings of UGOE, they are not outlined in this detail in the proposal.

#### **Need Basic Visualisation of a result set**

**Need:** Basic Visualisation of a result set

**Priority:** High

**Features:** several

- Display data points (locations) on the map (for about 500 hits)
- Display data points (events, point in time) in a histogram (for about 500 hits)
- Display data points in a pageable table
- Highlight data points by a given timeframe
- Possibility to set a time frame
- Possibility to move a timeframe
- Possibility to zoom into a timeframe
- Exchangeable backend for images of maps (Google maps vs. a map in a Europeana colour scheme)
- Fuzzy visualisation for results slightly outside the selected time frame
- Connect data points on the map based on time
- Compatibility with major browsers (IE, Firefox, Safari)

Planned Release:

#### **Need Graphical Query builder**

**Need:** Graphical Query builder

**Priority:** Medium

**Features:** several

- Selection different shapes like a circle, a polygon and a country (which basically is a complex polygon)
- Selection of a time frame or a point in time

Planned Release:

#### **Need Enhanced visualisation**

**Need:** Enhanced visualisation

**Priority:** Medium



**Features:** several

- Accurate maps for a given time frame
- Visualisation for multiple result sets
- Visualisation of large result sets using heat maps
- Animations for moving time frames
- (If time permits: rendering animations to flash video for YouTube etc.)

**Planned Release:**

**Need Data Editor and sharing**

**Need:** Data editor

**Priority:** Medium

**Features:** several

- Possibility to add custom data points
- Edit user generated data points
- Place data points in time and space using the interface
- Share custom data points with other users
- Share searches and in with it's visualisation parameters with other users

**Planned Release:**

## System Qualities

### Usability

The core of the spatio-temporal interface should useable without consulting documentation. To archive this goal interface paradigms known to the user from other similar interfaces services (like Google Maps) should be reused:

The map should

- support dragging of it's contents.
- support keyboard shortcuts (zooming etc.)
- support zooming with the mouse wheel.
- display information about specific data points.

The timeline should

- allow a selection of a range
- adjusting the range via drag and drop
- scale up the range to the full with of the timeline



Additionally there are several controls (like the connections of data points) that are currently not found in well know services. These should be documented from within the application by using a mouse over inline popup.

The elements of the interface should communicate with each other, to propagate changes the user made. This is needed to avoid confusing side affects while navigating with the time and space controls on the result set on the same time. The results of this communication should be visible for the user instantaneously.

The interface should be implemented using Javascript instead of popular RIA Frameworks like Adobe Flash or Microsoft Silverlight to ensure a maximum of accessibility.

Usability testing should also be based on Europeana Personas (M3.2.1). Usability testing will be conducted in part in-house. We're also planning to incorporate feedback of project partners.

## Reliability

The interface is basically build on three building blocks:

- The client side visualisation
- The server side generation of images for the requested map fragments
- The server side search / processing of user queries / generation of XML based result documents

Using this architecture allows a high degree of exchangeability of components, since each block uses plain HTTP/XML for communication.

The interface runs from the viewpoint of the server (except the user generated content) stateless. State is maintained completely by the client.

## Performance

The Performance of the spacio-temporal interface depends on the overall performance of it's two tiers:

Client side visualisation:

The visualisation is done in large parts (everything except the Images for the map, see below) on the client. It heavily depends on the Javascript execution performance of the browser. Since we are not targeting mobile devices, possible performance problems on such devices are not taken into account.

Server side Map rendering and search:

The rendering of the map has to be handled on the server side using GeoServer. It's possible to use a so-called tile cache to reduce the workload of map rendering. Other possible optimisations are fixed resolutions and zoom levels or defining just a small part of the world map (eg. Europe) as view port, since the search engine is part of the Europeana core, we are not expecting any major problems.

## User Interfaces

The main part of the spatio-temporal interface (as the name suggests) is the user interface. In detail the interface consists of three elements:

- A Map
- A time line
- A table of the results

Of course there is also a forth element, the search slot (or a advanced search). Since this is already part of the existing Europeana, we like to exclude it from this document. The functionality, features and behaviour of this slot are already defined; we tend to embrace the principle of reuse of UI paradigms (see Usability). Thus this aspect is excluded from this document.

The map should provide a similar user experience like Google maps, the controls (for creating searches)

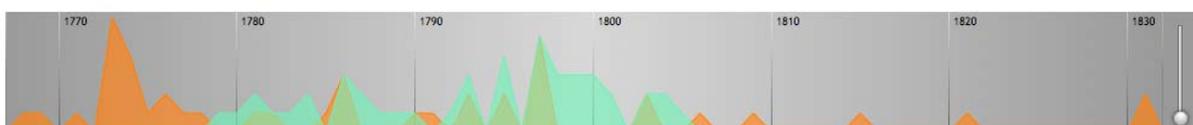
### Look & Feel

The interface is composed by two main components: The map and the timeline.

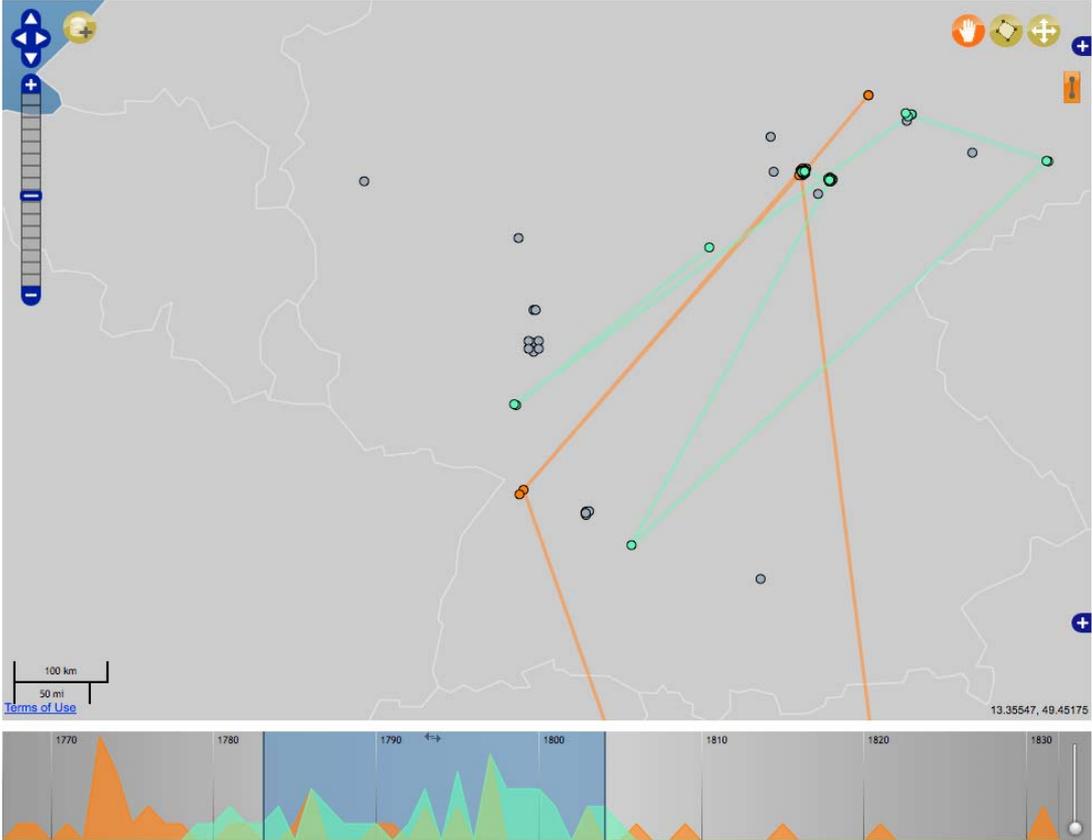
To be able to give a quick overview there is also a list of the search results. This list will be very similar to the existing Europeana result set list view.



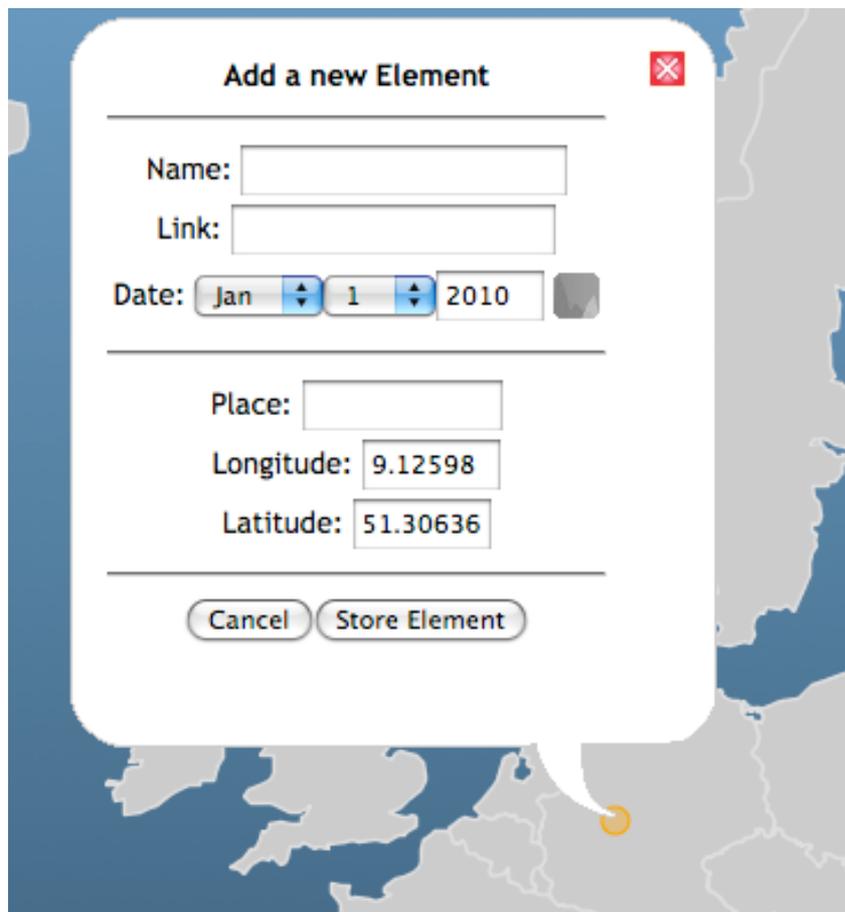
The map view.



The timeline showing a combined search .



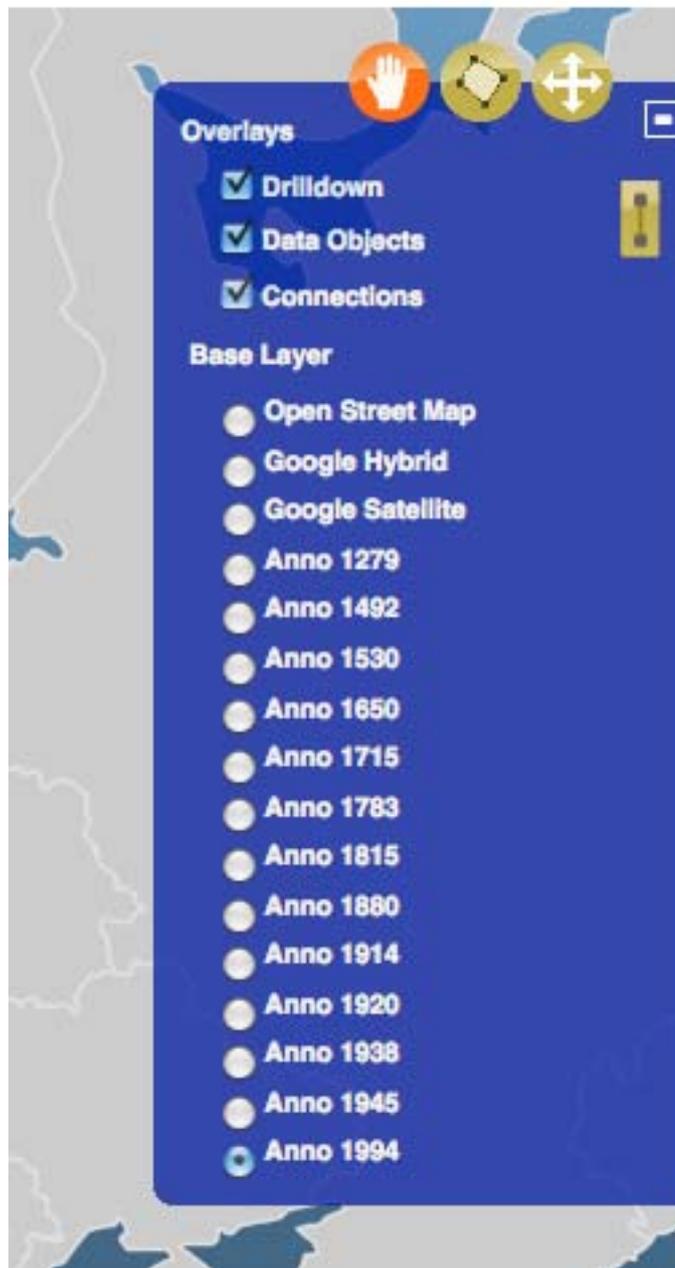
Map and timeline showing a combined search with connections.



The image shows a 'Draft interface for adding own data points' as a white modal dialog box titled 'Add a new Element' with a red close button in the top right corner. The dialog is overlaid on a map of Europe. The form contains the following fields and controls:

- Name:** A text input field.
- Link:** A text input field.
- Date:** A date picker showing 'Jan', '1', and '2010' with dropdown arrows for the month and day, and a calendar icon.
- Place:** A text input field.
- Longitude:** A text input field containing the value '9.12598'.
- Latitude:** A text input field containing the value '51.30636'.
- Buttons:** 'Cancel' and 'Store Element' buttons at the bottom.

Draft interface for adding own data points.



Controls for different (mostly historic) maps.

## Layout and Navigation Requirements

The interface needs to be able to scale itself to the size of the screen resolution of used display. The controls for map and time navigation will be centred on the screen. These parts of the interface should be visible without scrolling. If a search returns any results, these are displayed on the map. Additionally a list will be shown below the main controls. If there is more the on search, each is shown in a distinct colour. For usability reasons only two parallel searches are displayed. The underlying implementation should support multiple searches.



## Interfaces to External Systems or Devices ¶

The interface relies heavily on the search capabilities of Europeana and the GeoServer for the rendering of maps.

## Software Interfaces ¶

The Software will interact with the Europeana Solr Local search engine via HTTP. The interfaced to be used will follow the REST paradigm. The maps will be provided a WMS/WFS compliant map server. The server components need a J2EE Container as runtime environment. A specification for interacting with a data store for user generated content is out of scope for this document (-> T 3.3.2)

## Communications Interfaces ¶

The interface only uses stateless HTTP for communication.

## System Constraints

Client side components are implemented in HTML and JavaScript. It is planned to use Maven for dependency and packaging description of the client software.

If needed the existing JavaScript code will be embedded in a GWT environment.

Server side components like GeoServer will be deployed as provided by the providing project. Currently there is no need for a custom packaging or deployment process.

## System Compliance

### Licensing Requirements

There are no requirements to acquire a licence for commercial third party software. We are planning to release the source code under an open source licence.

### Legal, Copyright, and Other Notices

We are planning to implement the spatio-temporal interface using the following third party libraries:

#### Client side

**Name of library or external implementation:** Simile Timeplot

**Licence:** Simile Licence<sup>1</sup> (similar to Apache or BSD Licence)

**URL:** <http://www.simile-widgets.org/timeplot/>

---

<sup>1</sup> <http://simile.mit.edu/license.html>



**Name of library or external implementation:** OpenLayers

**Licence:** Custom licence<sup>2</sup>

**URL:** <http://openlayers.org/>

**Name of library or external implementation:** Explorer Canvas

**Licence:** Apache Software Licence 2.0

**URL:** <http://code.google.com/p/explorercanvas/>

**Name of library or external implementation:** WebFX Slider

**Licence:** Apache Software Licence 2.0

**URL:** <http://webfx.eae.net/dhtml/slider/slider.html>

### Server side

**Name of library or external implementation:** GeoServer

**Licence:** GNU GPL V2

**URL:** <http://geoserver.org/display/GEOS/Welcome>

Additional third party components are a SQL compliant database, a Java application server and a Java virtual machine and several Java libraries (like Log4J). Since these are described in the coding guidelines, they are not part of this list.

In general it is possible that the need for additional libraries is identified during the development process. In these cases open source (as defined by OSI) libraries and external implemented features will be used.

## Applicable Standards

### Implementation

The implementation uses besides well-known standards like HTTP and XML several lesser known ones as well:

#### WMS/WFS:

Web Map Service/Web feature Service is a group of standards specified by the Open Geospatial Consortium. They are used to provide map and geo feature data over the web.

#### REST:

REpresentational State Transfer is more a concept than a standard for interaction with resources on the web.

#### QA

Code quality during the development process will be monitored by the continuous integration facilities of Europeana Labs.

---

<sup>2</sup> <http://svn.openlayers.org/trunk/openlayers/license.txt>



## System Documentation

The interface for the end user should be as easy as possible (see Usability). Controls that are not widely used in existing similar applications should be documented using inline popups.

Additionally there should be tutorial screen casts that give the users a hands on experience on the capabilities of the interface. If there are accessibility concerns with screen casts, FAQs are also possible.

Documentation for developers will be delivered by code documentation and a detailed document outlining the architecture and the interaction between the used 3<sup>rd</sup> party components.