Spatial data visualization by usage of chosen free of charge tools for creation of composite web map applications

Jiří Ardielli[1] , Tomáš Minarčík[2], Bronislava Horáková[3]

[1] Institut geoinformatiky, Hornicko-geologická fakulta, Vysoká škola báňská – Technická univerzita Ostrava, 17. listopadu 15, 70830, Ostrava-Poruba, Česká republika
ard01@vsb.cz

[2] Institut geoinformatiky, Hornicko-geologická fakulta, Vysoká škola báňská – Technická univerzita Ostrava, 17. listopadu 15, 70830, Ostrava-Poruba, Česká republika
tomas.minarcik@vsb.cz

[2] Institut geoinformatiky, Hornicko-geologická fakulta, Vysoká škola báňská – Technická univerzita Ostrava, 17. listopadu 15, 70830, Ostrava-Poruba, Česká republika
bronislava.horakova@vsb.cz

Abstract. The web applications enabling dynamic visualization of spatial data and interactive user interface are a part of today's fast development of web applications. Many modern web map applications offer their functionality by development tools called API. Developers gaining option to use these powerful tools to easier and faster creating new web map applications. Possibility of combination more than one API of web applications and huge amount of data sources into single application is domain of new technology called Mashups. First in this article, there is described the utilization of AMapy API service, which is determines not only for rapid insertion of map into the web pages, but also for development of sophistic map applications. In the beginning of the article there are described individual properties of this API, license agreements, compatibility, functions and programming possibilities of this interface including description of basic classes, their methods and properties. The next part of the article describes web projects for building composite web applications online through internet browser. This tools are called mashup editors. With tools like this and new technologies can be developed robust web map applications, which could be concurrence to existing web-based map servers for processing and visualizing spatial data.

Keywords: Atlas, AMapy, Web 2.0, API, application, Mashups, GIS, GeoWeb.

1 Introduction

The formerly unclear content of static pages is being replaced by components dynamically capable of reacting on needs and wishes of human as well as machine users. The creator of web is not just a group of professionals experienced in web technology anymore but continuously growing mass of users who move their information, knowledge and activities into the web environment. Regarding these changes today it is often being talked about Web 2.0, a 2nd generation web. [4]

Ten years ago GIS were exclusively desktop applications. Current development is heading from map servers and thin clients to a new generation of products in a form of web services or so called mashups which in the future will be able to use, process and make distributed sources of the internet available by an effective and dynamic approach. Huge amounts of data sources and functionalities of existing as well as future map applications and services will allow its combining which will lead into formation of brand new products.

Over the past few years large software companies such as Google and Microsoft have appeared in the field of web applications development, usable in geoinformatics. Web maps' applications of these companies offer to use those function and possibilities to all internet users that until recently were only available at map servers. Those web projects also contain a number of spatial data that the users can use. With the arrival of technologies and styles of the web applications' development style, most map applications offer their own API, through which the users can use the functionalities of these applications within the scope or their own web projects.

The API is abbreviation of English words "application programming interface". This term is used by the software engineering discipline in programming techniques. It is concerned about collection of procedures, functions or classes of some library (but also possibly of some another application or some operating system's core), which can the programmer use, when using this library. The API determines the way of calling the library's functions from the source code of the program. The

functions of API are the programming complex, which the programmer is using instead of programming this on its own [11].

The so-called mashup technology came into existence due to the combination of API web applications and various data sources. Anyone can create the mashup application by using various API web project functions and not only by using the spatial data placed on the internet. To make sure that the creation of mashup will not become a privilege of users with the programming knowledge, promising web projects with graphic user interface are coming to existence on the internet. With the help of these even a user who lacks the programming languages knowledge can interactively create his own mashup application.

2 API of the web map applications

Many modern geoweb applications and big web portals offer their functionality by Map API. Some companies make their Map APIs freely available. Since 2005 start evolution of creating application using Map API's abroad. In Czech republic is dated the start of releasing Map API's in the end of the year 2006 and still there is a only few usable solutions. Between the mostly used API's with MapMashups are:

- Abroad:
  - Google Maps API
  - Yahoo Maps
  - Microsoft virtual earth interactive SDK
- In the Czech Republic:
  - Seznam Mapy API
  - Atlas Amapy API
- Open-source
  - OpenLayers.org
  - Mapstraction.com

3 API of the Czech web map applications

Web map applications of two biggest portals are currently mostly used in the Czech Republic. That is Atas.cz and Seznam.cz. Before, there was one more provider of Centrum.cz with their own web map application, but there was fusion of Atlas.cz and Centrum.cz into one group. These two portals allow map applications functions through their own API. Possibilities of API utilization and code samples are in next described on service example of one randomly selected provider.

3.1 Introduction to AMapy API

The AMapy API is the service offered by the ATLAS.CZ, a.s. company, which is operator of "AMAPY API" web portal. The service "AMapy API" is located at http://api.amapy.cz web address, which by help of JavaScript enables to depict not only the map's background on its pages. The user is authorized and enabled to use this service after completion of regular registration and obtaining generated unique key (GUID) sent by email, on which each user can pursue the API. Each key is bind to the certain domain name and it is not possible to use this key on another domain name. For each domain it is necessary to generate the individual key. From the local addresses (http://localhost) and from the file system it is possible to pursue the API even without generated key. The architecture of Amapy API is based on modern object language JavaScript year 2007 [1].

The map background of the Czech Republic is provided by subcontractor DPA s.r.o. company in scales 1:400 000, 1:200 000, 1:100 000, 1:50 000, 1:25 000, 1:10 000 and 1:5 000. [2]

By the help of provided programming interface the programmer can to create the map applications, to define the control elements of the map, to define marks in the map, to create the inform bubbles with variant contents and others. It is talked about so-called Mashup applications. Such application utilizes and presents data from more then one source and integrates them into one scene or one instrument. With the term Mashup application very closely relates the AJAX technology [5] (Asynchronous JavaScript and XML), which is dedicated for interactive web application development. It concerns about so-called asynchronous data transmission by the help of JavaScript language and XML markup language. In reality the AJAX is not concrete individual technology, but this term specifies the usage of several technologies together with especial target. Then, the applications developed by this technology, are enabled to change the part of its web page content without necessity of their complete reload from the server. Thereby it offers to the user pleasanter and more interactive space in comparison with classic web application. The dynamical changes present information and their depiction are ensured by the JavaScript language and by DOM (Document Object Model). DOM is an API which facilitates access or modification of content, structure or style of document or its parts [9]. By this is realized manipulation with HTML elements by help of JavaScript language. For asynchronous data exchange with web server by the help of AJAX technology is utilized XML format, HTML, JSON or plain text. Communication between server and clients is carry out by XMLHttpRequest [7] (XHR) interface through HTTP protocol.

3.2 Usage conditions and compatibility

With opportunity of this service utilization there are specified some provider's conditions. Among the most important points of general license agreements for usage of AMapy API belongs followed:

- *On non-commercial pages is possible freely to use AMapy API without restrictions.*
- *If the load exceed significantly the average values, the service will not be canceled, but Mashup's user will be contacted by the provider and there will be make an offer of additional connectivity.*
- *On commercial pages is possible freely to use AMapy API until the service is offered to the user at no charge, publicly and without any restrictions.*
- *There is no possibility to use AMapy API as an component part of paid application, or closed application like intranet.*
- *Mashup must refer to actual version of AMapy API.*

API was tested on Firefox, Opera and Internet Explorer 6 and 7 web browsers. Safari, Konquer and KHTML will be gradually followed. AMapy API supports only standard web browsers mode. [1]

3.3 AMapy API possibilities

In following list are specified possibilities, which are offered by the interface:

- *Depiction of the map*
- *Addition of control panels on map*
- *Depiction non-elastic information bubble for short text messages*
- *Definition of information elastic bubbles containing text, integrated picture gallery or potency to depict HTML pages*
- *Information elastic bubbles can contain bookmarks and also can contain footer*
- *Addition of marks to the maps*
- *Creation of own icons*
- *Possibility to draw vector line*
- *Setting up of scale and center of the map*
- *Support of more cartographic coordinate systems (S42, WGS 84, JTSK)*
- *Support for AJAX, DOM scripting, JSON and object model for map layers description*

3.4 AMapy API utilization

Services provided by AMapy portal can be used by the help of HTML markup language and by the help of JavaScript language. The API has defined classes, for which is possible by the "new" constructor to create new objects and consequently to define property's values of this objects. For some of them there are already defined events, to which there is possibility to attach the handler (method, which serves the event, is called by occurrence of given event). As well there is possibility to define the individual events to the objects and attach service events by the method *addEven* to them. Next there are declared functions, their arguments and returned data type for the classes.

3.5 Control elements and positioning of the map

The first step for utilization of AMapy service and for launching of the maps on own internet page is to place the reference to the file containing JavaScript code into head section of HTML file:

```
<script type="text/javascript" src="http://amapy.atlas.cz/api/api.ashx?guid=YOUR_GUID"></script>
```

As an argument of GUID variable there is possible to place the generated key. The Map is drawn into *DIV* element with practically haphazard dimensions:

```
<div id=" IDMap" style="width: 450px; height: 450px">
```

The map is an object of *AMap* class and is made by the constructor, where the argument is a value of *DIV* tag's identifier which defines the placement.

```
var Map1 = new AMap("IDMap");
```

The initialization of the map could traditionally starts up by the help of *onload* event of *BODY* element. However this event occurs in the moment of complete object load. In this case it happens after complete web page loading including all pictures. Here grows the risk of even one page's element non-loading (for example external pictures with advertising) and the map would not display. Thereby there will be used *domready* event, which occurs after DOM assembling. This event occurs earlier than onload event, also immediately after page's code read in.

After formation of the map there can be created and placed next objects. From among the basics belongs the control element of the map. Amapy API provides three control elements for map controlling*: AMapControl, ASmallMapControl and AMapTypeControl*.

For the definition of these control elements over the map there exist a function called *addMapPart*. When the map is side moved, the control elements do not change their position [1]. The first two mentioned elements make possible the map zooming in and zooming out and its movement into four directions. The different is only in their appearance. The last mentioned element makes possible the switching among the types of map's backgrounds, at present among air map and basic map. The user can define own elements (so-called map parts) with the help of *AMapPartDraggable* abstract class. The final map with two added control elements placed on own web page is depicted at the Fig 1.

3.6 Marks and point elements on the map

A mark graphic symbol on the map can be made by the positioning of object of *AMarker* class. First it is necessary to define a point for positioning together with coordinate system using the object of *AGeoPoint* class, and then to create a mark and consequently add it to the map by the help of map's *addOverlay* method. For addition of more number of marks, which are saved in array type variable, serve the method *addMarkers*. AMapy support three coordinate systems: the military S42, the global GPS and JTSK of Czech tourist club. The mark object can be represented by the picture or by the plain text. It can respond to a lot of events. As the mark, it is possible to use one of the predefined icons, or its appearance and properties define in *AIcon* object, by which we can define our own picture. It can be defined as a .gif file format picture or transparent .png file format, there can be shadows, description for the icon, the different appearance for three state types (normal, hover,

active), or the icon can be create by CSS cascading style sheets. Next preview of HTML language and JavaScript source code presents map creation, control elements placing on the map and addition of the mark.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Borehole Map | AMapy API</title>
   <meta http-equiv="content-type" content="text/html; charset=iso-8859-2" />
   <script type="text/javascript" src="http://amapy.atlas.cz/api/api.ashx?guid=YOUR_GUID"></script>
   <script type="text/javascript">
   var Page = {
      load: function() {
         var Map1 = new AMap("IDMap");  // creation of the map
         Map1.loadMaps();     // loading the map
         Map1.addMapPart(new ASmallMapControl());  // addition of zoom and movement control element
         Map1.addMapPart(new AMapTypeControl());  // addition of map change control element

         var point = new AGeoPoint(3460690,5551757, ACoordinateSystem.S42); // definition of the point
         var mark = new AMarker(point);  // creation of the mark
         Map1.addOverlay(mark);  // addition of the mark to the map
      }
   }
   window.addEvent('domready', Page.load.bind(Page));
   </script>
</head>

<body>
   <div id=" IDMap" style="width: 450px; height: 450px;">
   </div>
</body>
</html>
```

3.7 Cut out and scale of the map

For the depiction of the map, where occurs the set of marks, it is useful to set up suitable map scale and to set up area, where all of the marks are situated. For this purpose there exists setBestZoomAndCenter method, which accepts an array of the marks as an argument (object of AMarker class), or it accepts array of geo-points (object of AGeoPoint class). Herein it is necessity to point out the non-documented property overlays of objects of AMap class, which is array type and contains set of objects (AGeoPoint, AMarker) placed in the map. For the current map the method returns the optimal center and scale for the given group of geo-points (or marks) so that all marks will be depicted together.

3.8 Bubble elements

Bubble is a graphic element, which represents some extensive legend. Generally there are distinguished two types of bubbles:

- *Elastic bubble – possible to depict at a point or mark and fill up some content.*
- *Non-elastic bubble – by this bubble it is possible to give short information, which is not bind to any position. It depicts as a temporary hint at a left bottom part of the map.*

The elastic bubble is depicted by showBubble method or by onClickShowBubble property of AMarker object. There can be placed maximally one bubble at the same time on the map.

ShowBubble method can be called on object of AMap class (it is necessary to pass on AGeoPoint object– also placement of bubble to the point), or can be called on object of AMarker class (it is not necessary to pass on the placement - it is a property of object of AMarker class – also the placement of bubble to the mark). By call of the method, there is created the ABubble object. Method is defined by position and content, which can be the HTML string, HTML element, array of objects of ABubbleTab (bookmark) or array of URL paths to the pictures (photos even with previews). The content is distinguished according to attribute "content" and thereby there exist several kinds of elastic bubbles:

- *bubble with text – value of "content" attribute: HTML string, thus text string, or even the URL with picture*
- *bubble with picture – value of "content" attribute: HTML element (picture) made by help of JavaScript document.createElement method*
- *bubble with photos – value of "content" attribute: array with the URLs of pictures*
- *bubble with bookmark - value of "content" attribute: array of objects of ABubbleTab class*

Object of *ABubbleTab* class is defined by the name and by the content (content – almost similar as by bubble – can contain HTML string, HTML element or array with pictures).

There is possible to set up ability of maximizing to the bubble. This is done by the help of "true" value assignment to the *maximizable* attribute of object of *ABubble* class. By default this is set up by calling *showBubble* method through its *options* argument. After the maximization of the bubble, the content stays without changes. The maximized bubble can even contain complete HTML page. This can be ensured by attachment of value (URL) to *maxUrl* attribute. This attribute is also own attribute of objects of *ABubble* class and at once there is with it automatically set up *maximizable* attribute to the value "true". It does not depend on content (attribute content) of given bubble and not even on if it contains bookmarks. The *maxUrl* attribute exist only one for all the bubble.

For depiction of elastic bubble by the object of AMarker class, after occasion of OnClick event, there is not necessary to define the handler of this event. AMarker class has namely defined the onClickShowBubble attribute, which can have defined the content of the bubble.

The bubble can contain footer, which is dark blue stripe under bubble content. In this footer there can be contain HTML string, DOM element (e.g. picture) or array of *ABubbleFooterTab* objects, which represents bookmarks placed in footer. Assignment the footer content is done through *footerContent* attribute by calling of *showBubble* method.

Non-elastic informs bubble is depicted by *showInfoBubble* method of object of *AMap* class. Inform bubble is placed in the left bottom corner of the map and serves to show short informative texts. After a short time period the bubble is automatically hidden. It can contain HTML element (e.g. picture) or HTML string and it is possible to set up the duration of bubble depiction.

3.9 Line elements

Line element is defined by its starting and ending points, and by points, which are border for individual segments of broken lines, as object of *APolyline* class. There can be assigned values as color, thickness and transparency to the rendered track.

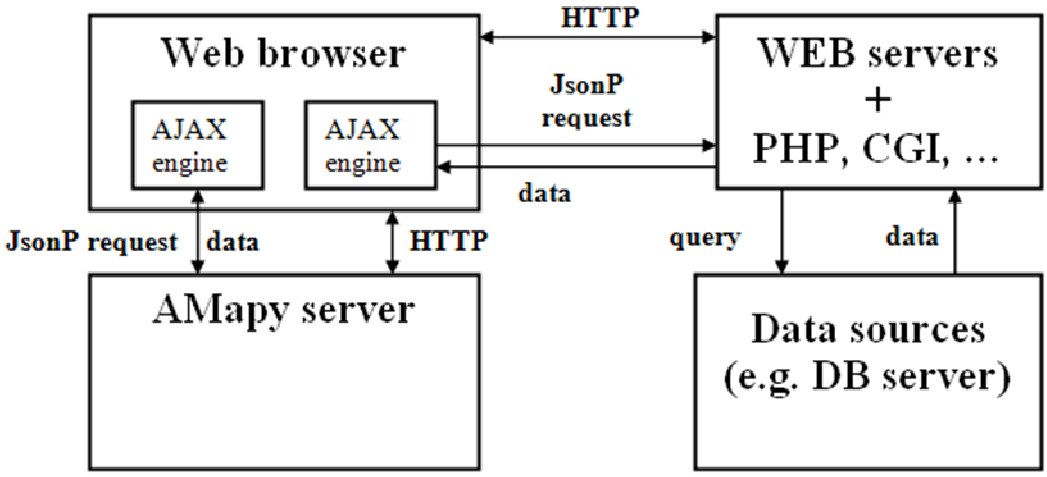

Fig 1 Architecture of the system

3.10 Architecture of the system

From above described functionality and property of AMapy API there is evident the possibility of its extensive exploitation in many ways. The architecture of the system is drawing at the Fig 1. The core of the system is web browser. In the first page load, the data is taken from the web servers and from AMapy server by HTTP protocol. Next, by utilization of web map application, the data is taken from the servers by usage of AJAX. This data transfer is running on the background, the results are immediately depicted on the page and user is not disturbed of necessity of sending requests to the servers.

4 Online tools for creation of composite web map applications

The largest software companies working in the area of web application development see the potential in the mashup applications. Web projects allowing the users to create their own mashup applications in a well-arranged interactive graphical environment serve as evidence. The development of these mashup editors is at its very beginning, but it already contain many functions for easy and effective mashup application creation.

Google, Microsoft and Yahoo! offer the most sophisticated mashup editors. Microsoft and Yahoo! have chosen the way of interactive graphic development environment. Google bet on the development environment that agrees with a narrower user group, in which basic knowledge of programming is necessary.

4.1 Microsoft Popfly

Graphic user interface of Microsoft popfly is based on Microsoft silverlight technology, which was developed for creation of multimedia internet applications. Silverlight allows develop very impressive web applications with graphical effects. Main parts of graphic user interface is container, application workspace and field for searching components. Inside container are a lot of components sorted into categories.

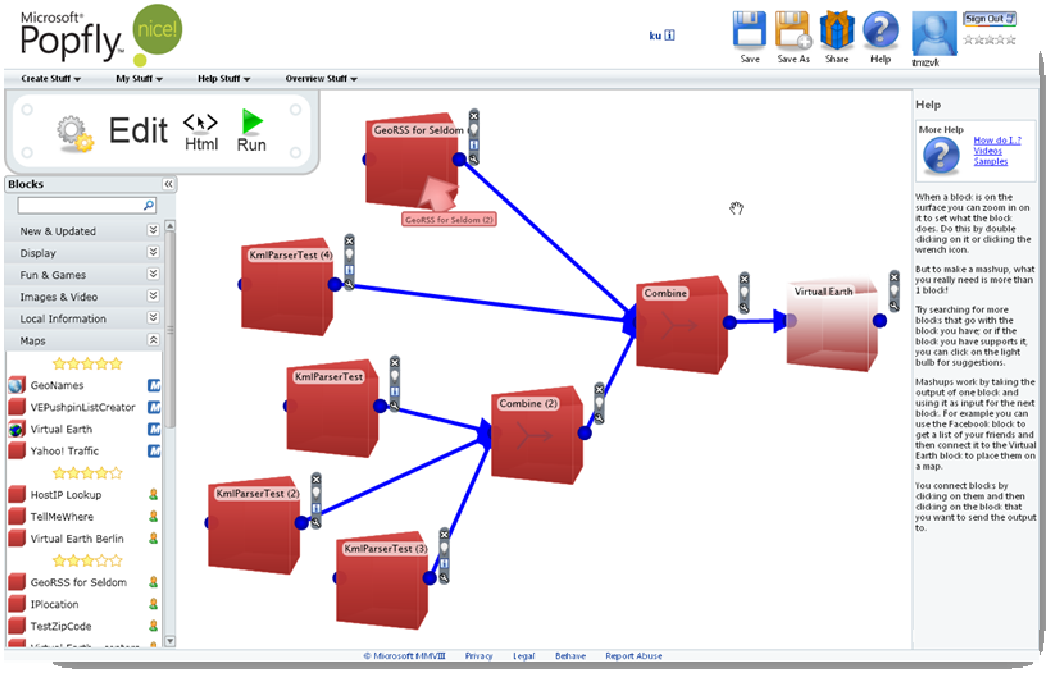

Fig 2 Microsoft Popfly workspace

The principle of the mashup application creation is based on piling the components on the worktop and their interconnection. The users have the opportunity to evaluate the individual components through assigning them golden stars. The highest ratings have the components created by Microsoft developers [8]. Inserting components from the holder into the application workspace is done through drag-and-drop control. The components can be moved elsewhere in the application workspace and can be interconnected through links represented by blue arrow. Each component contains several control elements such as the removal of component from application workspace, gaining information about the component or the possibility of entering the input parameters through a form.

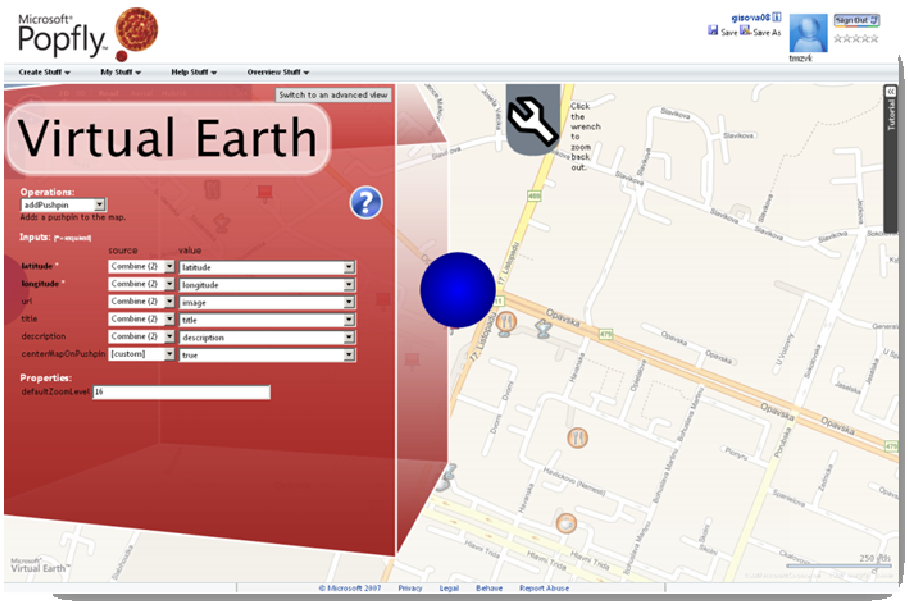

Fig 3 Microsoft Popfly - component setup

Developers could build their own components. Scripting language, which can be used for developing new components is JavaScript. Also can be used AJAX technology, DHTML or XAML.

Microsoft Popfly is one of the powerful web tools, which allows building even complicated and most sophisticated applications based on mashups technology.

4.2 Yahoo! Pipes

Yahoo! Pipes is interactive web editor for developing mashup applications with the aid of graphic tools and new technologies. The graphical user interface of Pipes is based on drag-and-drop environment created by AJAX technology.

The Pipes environment consists of three main parts. In the left part there is the container of the individual modules, ordered into categories. The application workspace occupies the biggest part of the window. Here it is possible to insert modules and to chain them. Debugger occupies the bottom part of the window. This show the output for each used module and if there is a mistake, it shows information about this mistake.

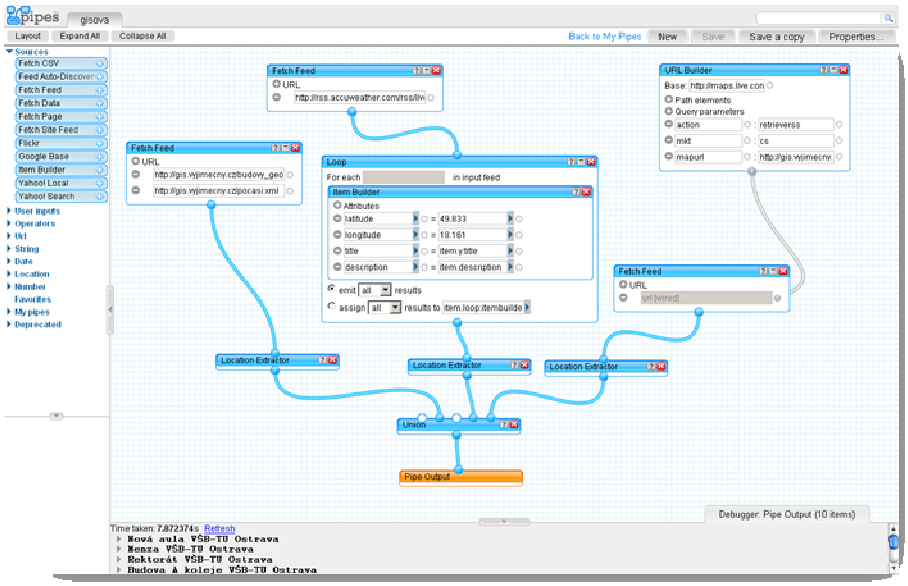

Fig 4 Yahoo! Pipes workspace

Own created working mashup is possible to share with other Pipes users. Pipes offers possibilities to use own mashup otside Yahoo! network. There are a lot of options and data formats for export results such as RSS, JSON, PHP, KML etc. [8][12]

4.3 Google Mashup Editor

One of the biggest internet company Google Inc. developed their own tool for creating mashup applications called Google Mashup Editor (GME). GME is still in beta version and is available just for users who gained permission to testing. GME is oriented on software developers rather then nonspecialist users. To create mashup application in the GME environment it is necessary to know a few existing markup and scripting languages. Google has also created a new special markup language for GME, where marks are introduced by GM name space. All GM marks rewrite into the programme code of Javascript language after the mashup application starts running. Besides the GM marks it is possible to use the actual code in the Javasript language or the capabilities of XHTML, CSS, XML when programming mashup. GME also offers the support for data processing from RSS formats. The values can be approached in an editor using Xpath. [3], [8].

Graphic user interface of GME consists of three main parts. First part and also first screen what user see after login to GME is called Editor. In editor users could write programming code. Syntax of programming code is highlighted and possible to save or check syntax. Second part of GME is called Feed Browser. There is possible to load any RSS or GeoRSS channel and GME highlights individual parts of code of channel. Get to values of that parts of code is possible through XPath in Editor. Last

part is called Sandbox. In Sandbox are displayed results of coded mashup in the Editor. There user can see how his mashup application looks like.

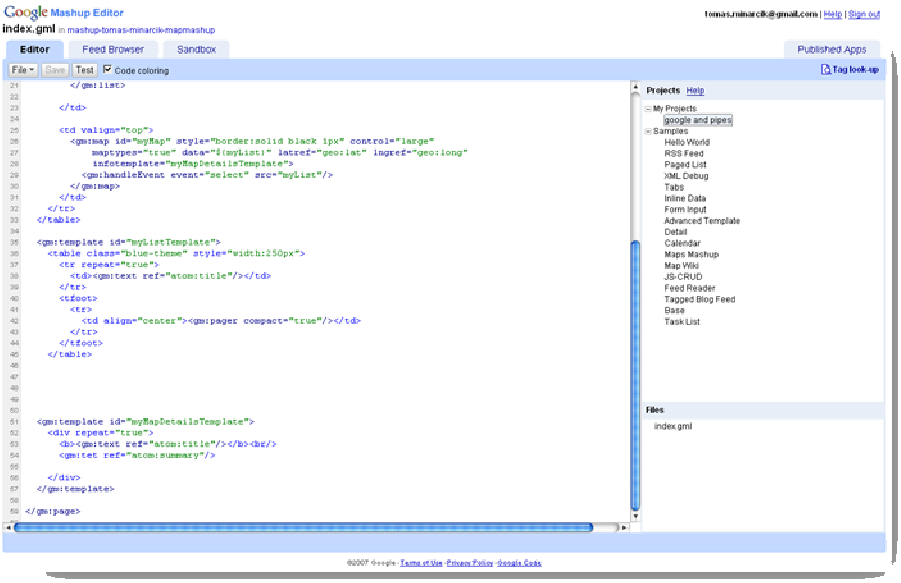

Fig 5 Google Mashup Editor

5 Conclusion

Web applications are capable of cooperation with desktop applications, however more important is the fact that they are capable of cooperation among each other. Result of this is not just a passive publication but a real sharing of data and functionality.

Web applications allow to create shared information content effectively and dynamically. Result of a wide decentralization are growing problems how to effectively search, process and use this content. New web is becoming a store of data and functions of giant dimensions.

Question is how to share these sources and further effectively reuse by exerting minimal cost and achieving high benefits. First typical examples are mashups and the highest demand is after mapping mashups which use the freely available mapping services over API.

Results from mashup editors or mapping API's could be combined in one robust web map application. This step requires some deeper programming skills.

The big problem of the mashup applications is data, which in most cases, especially in the case of the spatial data, is very expensive, and only big companies can afford such a big investment.

References

1. Atlas.cz. <http://amapy.atlas.cz/api/>, AMapy API.
2. DPA – Data processing agency. <http://www.dpa.cz/?typ=text&pid=reference>.
3. Google Inc. <http://editor.googlemashups.com/>, Google Mashup Editor.
4. Hinchcliffe, D. All We Got Was Web 1.0, When Tim Berners-Lee Actually Gave Us Web 2.0.<http://web2.wsj2.com/all_we_got_was_web_10_when_tim_bernerslee_actually_gave_us_w.htm>
5. Holzner, S. *Mistrovství v AJAXu*. Computer Press, 2007, Brno. ISBN 978-80-251-1850-4.
6. Kosek, J. *PHP : tvorba interaktivních internetových aplikací : podrobný průvodce*. Grada, 199, Praha. ISBN 80-7169-373-1.
7. Lacko, Ľ. *Ajax : hotová řešení*. Computer Press, 2008, Brno. ISBN 978-80-251-2108-5.

8. Minarčík, T., Ožana, R. FMG 2008, Webové projekty pro tvorbu MapMashups aplikací.
9. Resig, J. *JavaScript a Ajax : moderní programování webových aplikací*. Computer Press, 2007, Brno. ISBN 978-80-251-1824-5.
10. The PHP Group. <http://www.php.net/>, PHP: Hypertext Preprocessor.
11. Wikimedia Foundation Inc. <http://cs.wikipedia.org/wiki/API>.
12. Yahoo! Inc. http://pipes.yahoo.com/pipes/, Yahoo! Pipes.