

DESKTOP GEOMORPHOLOGIC INFORMATION SYSTEM TOWARDS SERVER SOLUTION

Jakub, ŠILHAVÝ¹, Karel, JEDLIČKA²

^{1,2}Geomatics section of Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 22, 306 14, Pilsen, Czech Republic,
jsilhavy@kma.zcu.cz, smrcek@kma.zcu.cz,

Abstract

The purpose of this work is to offer tool for geomorphologic research without installing a desktop geographic information system (GIS). The main goal of this work is to find out and verify the way from existing desktop GIS application towards the server solution.

The paper concerns about application of server GIS in distributed environment. It focuses on the remote geoprocessing services. The study field is a Geomorphologic Information System (GmIS), especially the hydrological analyses. The analyses are published as web services.

The paper solves a web application that allows remote processing of hydrological analyses on the server. On the client, the users can access the analyses via the web browser. The solution has server- and client-side.

On the server-side, the technology of ArcGIS Server Advanced 9.3.1 is used. The existing hydrological tools designed for ArcGIS Desktop were modified to use in the ArcGIS Server environment. The different requirements for input and output data formats and output complex symbology were the reason for the modification. The paper describes the ways of publishing analyses on ArcGIS Server.

On the client-side, the technology of JavaScript, HTML and CSS is used to build web user interface. The creation of graphical user interface (GUI), consuming published ArcGIS Server web services, processing of user inputs and interaction between user and web client are the key parts of the client-side. The paper contains examples of source code.

In conclusion, the paper evaluates the pros and cons of distributed GmIS in comparison with desktop GmIS.

Keywords: Geoinformation technology, Geomorphologic Information System (GmIS), distributed environment, distributed geoprocessing, web services, ArcGIS Server

1. INTRODUCTION TO GmIS IN DISTRIBUTED ENVIRONMENT

The paper concerns about remote geoprocessing of geography data focused on Geomorphologic Information System (GmIS). The paper proposes the following objective:

- **Provide geomorphologist tool for his work without having to teach him to install and manage the GIS.**

The motivation is based on the geomorphologist's needs and the recent state of desktop GmIS.

The prospective user of GmIS is the geomorphologist who wants to take advantage of information technology to solve his tasks. For this purpose, the work [Jedlička 2010] implemented the desktop GmIS using the ESRI ArcGIS Desktop technology. According to conclusion of this work, there are the administration problems in the desktop implementation of GmIS. The system consists of a large number of extensions that are difficult to maintain in every installation. The moving GmIS from desktop to server is suggested in [Jedlička 2010, p. 102] to solve this problem.

The motivation can be split into two partial aims:

- **To make a proposal of moving GmIS from desktop to server with focus on using remote geoprocessing of geographic data¹.**
- **To test the proposal on the part of GmIS. The hydrological analyses were chosen as a testing area.**

The conception of GmIS was written in [Minár et. al 2005] and [Mentlík et al. 2006], the technical solution for desktop GIS was implemented in [Jedlička 2010]. How to implement GmIS in a server environment is proposed in this paper.

This proposal was tested on the selected hydrological analyses² which were run in the server environment. It results in the client/server application that allows remote geoprocessing without the need to manage the GIS for each client.

The application proposal is based on the unified process of system design, described in [Arlow, Neustadt 2007]. The proposal is divided into the following steps:

- *User requirements identification* – to find out user needs related to system. The requirements define what user expects from the system.
- *User requirements analysis* – to determine what needs to be done to meet the user requirements.
- *Proposal of solution* – to describe ways which leads to realization of aims from requirements analysis.
- *Implementation of solution* – to realize the proposal of solution. The ways which were chosen from proposal to realize the solution are described here.

2. USER REQUIREMENTS IDENTIFICATION

In [Arlow, Neustadt 2007], the dividing to functional and non-functional requirements is determined. The functional requirements appoint what the system should do by defining expected system activity. The non-functional requirements are determined by technical limits of used system.

The requirements were collected in consultation with users and the general system requirements were adopted from [Jedlička 2010].

The following requirements were defined:

Non-functional

- To replace the desktop implementation by solution that separates the system usage from its administration.
- To ensure the interoperability and availability of the system³.

Functional

- System should compute the analysis *watersheds delimitation*.
- System should compute the analysis *base surfaces creation*.

¹ The using of distributed databases was not considered in this work.

² Watersheds elimination and Base surface delimitation.

³ According to [Jedlička 2010], the interoperability requirement is the system's ability to cope with the frequent exchange of data between users. The availability requirement means, that the system is accessible to anyone and it is not necessary to protect them by security password.

The non-functional requirements meet the main goal of this work and the functional requirements are based on the aim to test the proposal.

3. USER REQUIREMENTS ANALYSIS

Although the geomorphologists are satisfied with using desktop implementation of GmIS, the non-functional analysis shows that they need to separate the system usage from its administration. The detailed analysis of current GIS architectures in [Šilhavý 2010, p. 19] shows that the Client/Server GIS architecture best fits this requirement. The Client/Server GIS solution provides central system administration on the server-side and the system usage by multiple users on the client-side.

The desktop GmIS is built on ESRI ArcGIS technology [Jedlička 2010]. The server solution of GmIS should use the same technology to meet the openness requirement. The server solution from ESRI which can process and analyze geospatial data is ArcGIS Server.

The availability requirement needs the client to be accessible without installing any program on the client-side. The usage of the web client meets this requirement.

The functional analysis is described in detail in [Šilhavý 2010, p. 46]. The desktop implementation of GmIS contains algorithms to compute both analyses that are required. These algorithms must be modified before they are used in the server environment.

4. PROPOSAL AND IMPLEMENTATION OF THE SOLUTION OF DISTRIBUTED GmIS

The detailed proposal is described in [Šilhavý 2010]. The proposal describes the ways to fulfill the needs from requirements analysis. The implementation consists of the proposal validation at the selected hydrological analyses. In this chapter just the key features of proposal are mentioned followed by its implementation. The both server and client side are described.

4.1. Implemented architecture of server solution

The implemented architecture of proposed solution is shown in Fig. 1. It has two tiers – the presentation logic is supplied by the web client and the tasks of application and data logic are fulfilled by ArcGIS Server. The architecture has server and client sides.

The client-side of application consists of the web client based on JavaScript which uses REST API to communicate with ArcGIS Server web services. Through the web client it is possible to download the analyses results to the user file system on the client-side. The web client is the graphical user interface for published analyses which provides interactivity between the user and tools on the server.

The server-side consists of both the web server and the application server. The web server Microsoft Internet Information Services (IIS) provides communication through HTTP protocol. The application server ArcGIS Server is used to publish GIS resources like web services. Both servers are running on the Windows 2008 Server operating system⁴.

Server-side

4.2. The desktop tools modification

Although ArcGIS Server and ArcGIS Desktop share the same functional base – ArcObject libraries, they have several libraries different. This fact allows using desktop tool in server environment but the tool's modification is necessary. The differences are related to input and output data types and displaying the complex symbology. [ESRI 2004, p. 28-34] [ESRI 2010a]

⁴ The server installation and setup were performed by technicians from ARCDATA Praha s.r.o. company, who recommended us Windows rather than Linux system due to better stability of ArcGIS Server.

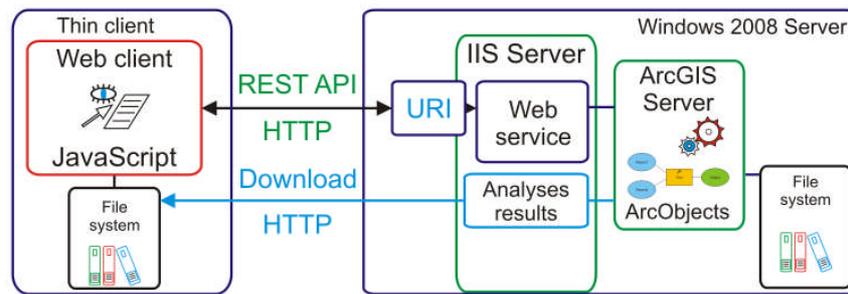


Fig. 1. Realized architecture of distributed GmIS

Design of ArcGIS Server described in the previous paragraph influences data types, which can be used on the input and output of analyses. The thin clients (like a web browser), which mostly communicates with ArcGIS Server, are not able to handle with complex data types like Feature Class, Raster Dataset and Table because they do not contain the necessary ArcObject libraries. That means the thin client cannot use the complex data type like input or output of the geoprocessing service. [ESRI 2010a]

The shapefiles and rasters are very usual inputs and outputs of hydrological analyses thus the three ways to solve the complex data input from the web client (see Fig. 2) were proposed (according to [ESRI 2010a]).

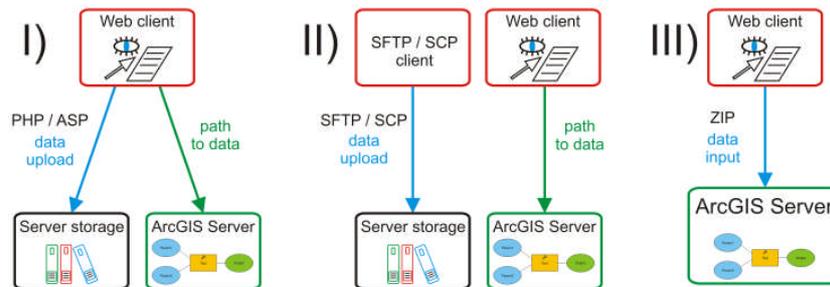


Fig. 2. The ways to upload data to a server

- *server-side scripting* – first, the files are uploaded to a server storage using the *server-side scripting* technology (like PHP or ASP). The complex data types stored on the server-side can be used as the input to ArcGIS Server analyses. The input parameter is the absolute path to the uploaded data.
- *protocol SFTP or SCP* – using another application which communicates with a server by SFTP or SCP protocol the data is uploaded to the server storage. As with the previous solution, the input parameter is the absolute path to the uploaded data.
- *data type File* – the input data is compressed to a ZIP file and posted to ArcGIS Server by a thin client. The data type file is supported by ArcGIS Server on the input from the thin clients. The ArcGIS Server provides the tool to uncompressing the ZIP files to specific location from where they can be used like an input.

The implemented application works with sample data saved on the server-side. Based on the proposal, the SCP method was chosen to uploading the sample data to the server (more in [Šilhavý 2010, p. 63]). The absolute path to the input directory is used like the input parameter for the server tools. The data type String is supported by ArcGIS Server on the input from thin clients.

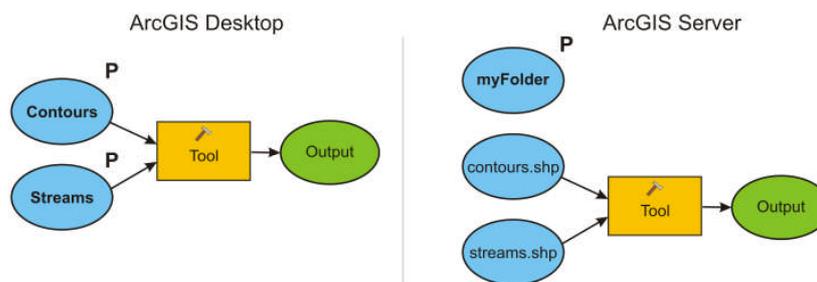


Fig. 3. Two ways of data input to the ArcGIS tools

In the Fig. 3, there are the two models, which perform the same function, one for ArcGIS Desktop and one for ArcGIS Server. The ArcGIS Desktop model has two input parameters (**P**), (both for data type Feature Class) which are represented by files *contours.shp* and *streams.shp*. The Feature Class is unsupported data type for solution composed of ArcGIS Server and thin client, thus the string input parameter *myFolder* (**P**) is used. The *myFolder* parameter links to directory which contains the input files *contours.shp* and *streams.shp*. The path used in the model seems like `%myFolder%\contours.shp`.

For the input directory, it needs to be specified which files are related to which input parameters and which attributes required in the tool. E.g. the tool *Create Hydrologically Correct DEM*⁵ needs to know that file *contours.shp* is data input *Contour* with the attribute field *ELEVATION* used to assign the contour's elevation. The ways to solve the specification of user data input from a web application are proposed in [Šilhavý 2010]. The following paragraphs introduce the implemented way.

- *using of standard names of input files and attributes* – for the each data input is hard typed the name of the file. In the same way, the input attributes names are defined (see table 1). This solution is easy to implement and requires the user preprocessing to be done.

Table 1. The naming standard for input files

File	Input type	Attribute
contours.shp	Contour	elevation
rivers.shp	Stream	
points.shp	Pointelevation	elevation

All the tools used in this work require standard names of input files and their parameters in the input directory.⁶ The list of the conventional names for each tool and other proposals how to solve this problem are written in [Šilhavý 2010].

The outputs of the desktop tools are mostly unsupported data types by ArcGIS Server. Above that, the output complex symbology cannot be displayed by thin clients. Both problems can be solved using Result Map Service (see [Šilhavý 2010]). The server renders the results and a raster image is returned to the client instead of the complex data type. All server tools used in this work were published as Result Map Service with defined symbology.

According to concurrent use of tools published by ArcGIS Server, the output data must be written to scratch folder or geodatabase using the relative paths.⁷ The path seems like `%scratchworkspace%\outputName`.

⁵ One of the tools described in [Šilhavý 2010].

⁶ The project is at the beginning, thus the easiest solution was implemented. Other solutions of user input are described in [Šilhavý 2010] in detail. There is a space for future improvement.

⁷ This workflow is recommended for Desktop and necessary for Server.

The directory which contains the results in their original data type (not image only) is available to download as a zip file. For this purpose, the *zip python script*⁸ is published as geoprocessing service. [ESRI 2010a]

Client-side

4.3. The development environments

The ArcGIS Server API for JavaScript was chosen in [Šilhavý 2010] based on the requirements for openness and accessibility of system. The development environment used in this work was Aptana Studio with Aptana Code Assist Plugin⁹ which helps with completing JavaScript code and provides access to ArcGIS Server JS API documentation.

The programming code of the web client is divided to three parts. The HTML page `WebGmIS.html` defines the structure of GUI, the CSS style sheet `style.css` modifies the appearance of the client GUI and the JavaScript file `functions.js` contains all the function used in the client application.

The JS API programming is based on *Dojo JavaScript Toolkit* libraries (<http://www.dojotoolkit.org/>). Dojo is an open source set of functions which provides using preprogrammed components *Dojo widgets* (also called *dijits*) in own application. The client GUI can be composed from several *Dojo widgets*, e.g. forms with input validation, buttons, tables, graphs or web page layouts. The JS API 1.6, which is built on Dojo 1.4.1.¹⁰, is used in this work. [ESRI 2010b]

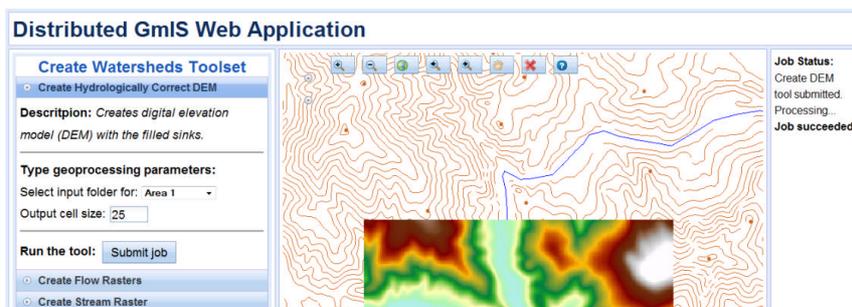


Fig. 4. Realized GUI

The GUI is implemented using *dijit.layout* components. The basic layout is inspired by layout of desktop GIS software. On the left pane, the toolbar and table of contents are situated. The dominant part of GUI is consisted of the map container which is placed in center region. The right pane is used for displaying informative messages and help. The GUI is shown in Fig. 4.

⁸ The tool created by ESRI, available at http://webhelp.esri.com/arcgisserver/9.3/dotNet/index.htm#geoprocessing/zip_python_script.htm

⁹ http://resources.esri.com/help/9.3/arcgisserver/apis/javascript/arcgis/help/jsapi/api_codeassist.htm

¹⁰ The latest version of JS API for ArcGIS Server 9.3.1. was 1.6 in May 2010.

4.4. Interaction and client functionality

Interaction of the GUI consists in client's and user's mutual communication. The summary of interaction activities is shown in Fig. 5.

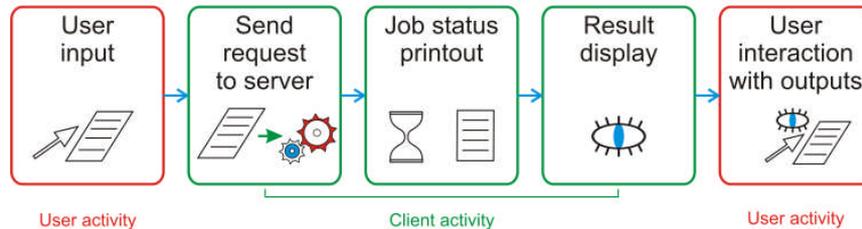


Fig. 5. User and client activity

The first step is the user input which is realized by forms with validation. The client's simple application logic processes the user input into server-readable structure and uses JS API to send them to the tool on the server. During the processing of data, the user is being informed by the job status printouts displayed in the GUI.

When the processing is complete, the result data is received, processed and displayed in map and table of contents by the client. The user interaction with the results consists in toggling visibility of displayed map layer and map interaction (pan and zoom the map). The implementation of the client activities is described in the following paragraphs.

4.5. Introduction to JavaScript API

ArcGIS JavaScript API is available online at <http://serverapi.arcgisonline.com/jsapi/arcgis/?v=1.6>, where the number 1.6 means release version of API. The functions of JS API are arranged to thematic packages. The basic package `esri.map` is used to work with map. It allows displaying the map services and the results of geoprocessing services on a map. [ESRI 2010b]

The code in the example 1 demonstrates how to load `esri.map` package, to create a new instance of the map object (`map`) and to associate the map with HTML container (`mapContainer`).

```

dojo.require("esri.map");
var map = new esri.Map("mapContainer");
...
<div id="mapContainer"></div>
  
```

Example 1. Define a new map instance

To display a published map service on the map, the new layer has to be defined. The service URI (Uniform Resource Identifier) is used to link the layer and existing service (see example 2).

```

bgMap = new
esri.layers.ArcGISDynamicMapServiceLayer("http://eugen.ff.zcu.cz/ArcGIS/rest/services/WebGmIS/backgroundMap/MapServer");
map.addLayer(bgMap);
  
```

Example 2. Add a map service to the map

The geoprocessing services are accessible in the similar way. The example 3 shows how to create a new `Geoprocessor` instance that works with this type of services. The input parameters from the user forms are collected to a variable `params`, which is forwarded to the server tool as input parameter. The tool is

submitted by the `submitJob()` function that specifies two callback functions `gpStatusCallback()` and `gpCompleted()`.

```
gp = new
esri.tasks.Geoprocessor("http://eugen.ff.zcu.cz/ArcGIS/rest/services/WebGmIS/GPS
erver/CreateHydrologicallyCorrectDEM");
params = {
    "Output_cell_size": 25,
    "myFolder": "c:\\WebGmIS\\ToolData\\"};
gp.submitJob(params, gpCompleted, gpStatusCallback);
...
gp.getResultImageLayer(jobId, outputParam, imgParams, displayResult);
```

Example 3. Create and submit the geoprocessing service

After sending data to the server, the function `gpStatusCallback(jobInfo)` is repeatedly run to check the status of the job at a regular interval. The content of `jobInfo` variable is being written to informative pane of the web client.

When the job is successful the function `gpCompleted(jobInfo)` is called to process the results of the geoprocessing service. To return the result layer by a function `gp.getResultImageLayer`, the unique identification of the service (`jobId`) and name of the output parameter (`outputParam`) must be specified. The output is returned by a server as a map layer which can be displayed on the map.

4.6. Interactive client

The user data input is supplied by the sample data (see 4.2) stored in five folders. The folders contain data of different study areas (see Fig. 6). The first user input is to select which area should be processed. The other tool's inputs are collected using various forms.

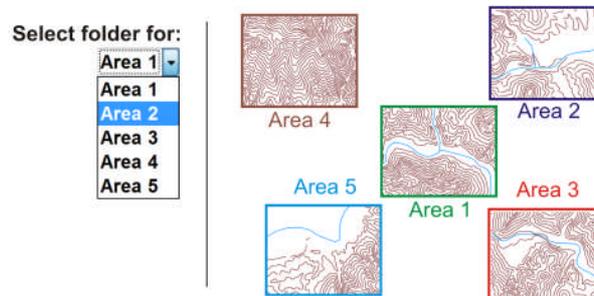


Fig. 6. The area selection for the analyses input

After the *send button* is clicked, the client start to process the inputs and `calculateTool('toolName')` function is called. According to parameter `toolName`, the client recognizes which tool should be run. The input parameters are read from correct form using the unique identifications. The example 4 shows calling the *Create Hydrologically Correct DEM* tool (parameter `createDEM`) with reading user inputs from forms. The *output cell size* parameter is accessible using the ID `oCellSize` and the selected study area has ID `inputFolder`. The variable `params` is filled by these values and send to server (see example 3).

```
onClick="calculateTool('createDEM') "
...
switch (toolName) {
case "createDEM":
var oCellSize = dojo.byId("oCellSize").value;
var myFolder = dojo.byId("inputFolder").value;
```

Example 4. Process the user input

The result of each tool is stored in logical structure on the client. The variable `jobLists` is the associative array where the key is the tool ID (`toolName`) and the value is the linked list (see Fig. 7). One field in the array represents list of result for one tool. One node of the linked list contains instance of `MyJob` object which represents one result of specific tool. The object attributes describe all the important information about the processed tool. E.g. for downloading the results, the parameter `zipURL`, which links to zip file stored on the server in a public place (see 4.2), is used.

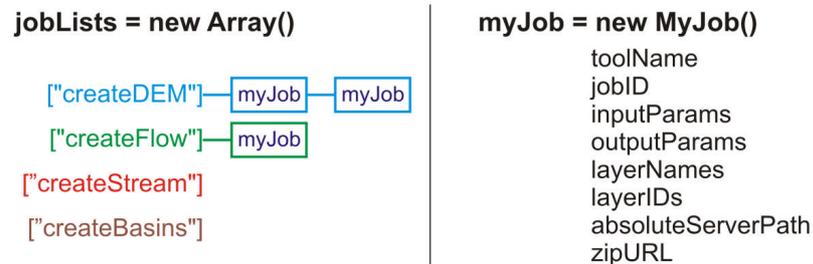


Fig. 7. The list of tools results

The table of contents generating uses the logical structure stored in `jobLists` variable. The client goes through all lists in the structure and finds out the names, IDs and URLs of returned result layers (`layerNames`, `layerIDs`, `zipURL`). The layer names are displayed in the table of contents with hyperlink to data download. The layer IDs are used to layers visibility toggling (see example 5).

```
layer = map.getLayer(layerID);
if(layer.visible)
layer.hide();
else
layer.show();
```

Example 5. Toggle visibility of map layers

5. DISCUSSION

The implemented solution was focused on the distributed geoprocessing of the hydrological analyses with accent to own programming of web client. The hydrological analyses played the role of testing area to proposal verifying.

The work [Pávková 2010] introduces remote processing of advanced hydrological analyses focused on hydrological modeling. The distributed environment of ArcGIS Server is used like a tool to serve the developed analyses through the internet.

The using of ArcGIS Server can be conceived in many ways – use it like a tool or use the platform for a custom server solution.

The actual information about the project is available on <http://git.zcu.cz/index.php/WebGmIS>.

6. CONCLUSION

This work is based on known research of Geomorphological information system [Minár et. al 2005] and [Mentlík et al. 2006]. The hydrological analyses were realized in the previous work [Jedlička 2010] for desktop environment. The server implementation of GmIS, introduced in this paper, brings the advantage of central administration and wide accessibility through the web browser without special extensions and plugins needed.

The quality of analyses processing in distributed GmIS is at the same level as desktop GmIS but the GUI is less user friendly in compare to desktop GmIS. The web client does not substitute the interaction of desktop applications like ArcMap and ArcCatalog.

The introduced solution meets the main objective – it provides the tool for geomorphological research without installing and administrating the GmIS on each client machine. The proposal of moving GmIS from desktop to server environment described in this work was successfully tested on the selected hydrological analyses.

7. REFERENCES

- [1] ARLOW, J., NEUSTADT, I. (2007). UML 2 a unifikovaný proces vývoje aplikací. Objektově orientovaná analýza a návrh prakticky. Translated by Bogdan Kiszka. 2. release. Brno: Computer Press, 2007. 567 s. ISBN 978-80-251-1503-9.
- [2] ESRI (2010a). ArcGIS Server for the Microsoft .NET Framework 9.3 Help. Redlands (California): ESRI, 2010. [on-line] <<http://webhelp.esri.com/arcgisserver/9.3/dotNet>>
- [3] ESRI (2010b). ArcGIS Server Resource Center. Redlands (California): ESRI, 2010. [on-line] <<http://resources.esri.com/arcgisserver/>>
- [4] ESRI (2004). ArcGIS Server Administrator and Developer Guide. Redlands (California): ESRI Press, 2004. 725 p. ISBN 978-1589480971.
- [5] JEDLIČKA, K. (2010). Geomorfologický informační systém [PhD thesis]. Ostrava: Institut geoinformatiky, Vysoká škola báňská – Technická univerzita Ostrava, 2010. [on-line] <http://gis.zcu.cz/projekty/GmIS/Jedlicka_DSP>
- [6] MENTLÍK, P., JEDLIČKA, K., MINÁR, J., BARKA, I. (2006). Geomorphological information system: physical model and options of geomorphological analysis. In *Geografie*. Year 111. no. 1. pages 15-32.
- [7] MINÁR, J., et al. (2005). Geomorphological information system: idea and options for practical implementation. In *Geografický časopis*. Year 57. no. 3. pages 247-266. ISSN 0016-7193.
- [8] PÁVKOVÁ, K. *Modelování povrchového odtoku v prostředí ArcGIS Serveru*. [thesis]. Olomouc. Univerzita Palackého v Olomouci, 2010.
- [9] ŠILHAVÝ, J. *Hydrologické analýzy v distribuovaném prostředí*. [thesis]. Plzeň. Západočeská univerzita v Plzni, 2010.