A Software Framework for GIS-based Multiple Criteria Evaluation of Land Suitability

Rico Vogel[1]

Leibniz Institute of Ecological and Regional Development (IOER), Weberplatz 1, 01217 Dresden, Germany, r.vogel@ioer.de

Abstract. In this paper a software framework for the evaluation of land use suitability questions is described. To answer these questions, a multiple criteria evaluation approach is used. The criteria are expressed through a set of indicators, which are the basis for the multiple criteria evaluation with the *Compromise Programming* approach. This method expects raster datasets with corresponding weights as input for each indicator. To determine the indicator weights, the *Analytic Hierarchy Process* is used. The theoretical and mathematical principles of both approaches are described. Furthermore, this paper focuses on the development and usage of appropriate software applications, especially the logical structures as well as the information and data processing.

1 INTRODUCTION

In recent years many regions in the world have been characterised by dynamic development. Especially cities in Asia and Latin America are expanding unhindered into the surrounding areas. In most cases, political motives are crucial for the choice of locations of buildings and infrastructural facilities.

It is essential to optimise the search for appropriate locations to reduce the human influence on natural resources. The application of a multiple criteria evaluation (MCE) method seems to be useful to support decision makers during their work. MCE helps to combine the information of several criteria to get a single evaluation result. Publications like Zeleny (1982) give an overview of the principals of several MCE methods. The usage of MCE in a spatial context within Geoinformation Systems (GIS) is also described (e.g. Malczewski 1999).

In earlier publications the useability of linking the *Compromise Programming* method (CP) with the *Analytic Hierarchy Process* (AHP) has been shown by the author to evaluate land use suitability (e.g. Thinh and Vogel 2006; Vogel and Thinh 2007). To calculate the suitability in a GIS several indicators are necessary, which are represented by raster layers (e.g. binary ArcGIS data type GRID). These are stored on a geo data server (Fig. 1).

One of the main concerns for analysts is the question, which indicator of a MCE problem is more important than others and to which degree (Tabucanon 1988, p. 20). In consequence, the algorithm of the CP requires weight values for any indicator.

To get weights several approaches are available (Malczewski 1999, pp. 177-191). According to Thinh (2004, p. 138) the AHP is preferred to calculate indicator weights because there are considerable criticisms concerning most of the other approaches, especially the lack of theoretical foundations. The principle of the AHP and its implementation as a web-based application are

[1] Furthermore the author is scientific assistant at the Technische Universität Dresden, Department of Geoscience, Professorship of Geoinformation Systems, Helmholtzstraße 10, 01062 Dresden, Germany, rico.vogel@tu-dresden.de

described in section 2. The tool will be used by experts for a pairwise comparison of the indicators which will result in indicator weights calculated by the application.

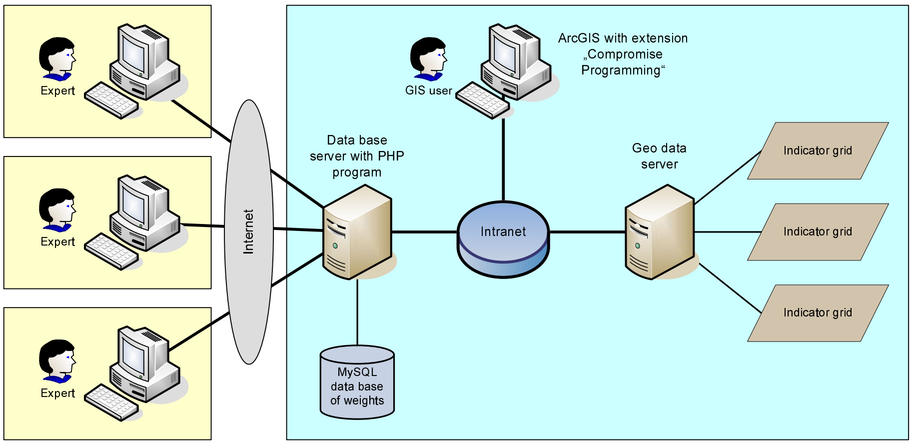


Figure 1: Principles of software and hardware usage and data storage; changed according to Vogel and Thinh (2007, p. 138)

If all input data as the indicator datasets and the indicator weights are available, the GIS analyst will use the ArcGIS extension *Compromise Programming* (section 3) to commit the MCE according to a specific land use suitability problem.

2 WEB-BASED APPLICATION FOR DETERMINATION OF WEIGHTS

This section introduces the *Analytic Hierarchy Process* (AHP) and the implementation of the method as a web-based application.

2.1 Analytic Hierarchy Process

The AHP was developed by Saaty (1980) and belongs to the additive weighting methods. These are characterised by a choice of criteria, which might be meaningful for an evaluation. These criteria are classified according to their relevance. AHP can be used both for analysing problematic situations and for preparing assessments and decisions. It was described in detail in the literature (e.g. Saaty 1980, 1994; Saaty and Vargas 1991).

This paper describes how the AHP can be used to calculate the weights for a set of criteria and indicators respectively. Connected with this, the importance and the observance of the consistency condition will be underlined.

The AHP is based on the pairwise comparison of any items contained in a set of indicators. For the comparisons, Saaty (1980) suggested a scale of nine values (Malczewski 1999, p. 183). These and their related verbal definitions are listed in Table 1. If an expert prefers an indicator with extreme importance compared to another one the value 9 should be used. In the case the expert does not favour one of the two considered indicators value 1 is to be used which means equal importance. All intermediate values represent various degrees of importance (Vogel and Thinh 2007, p. 139).

The comparison values are set up the values of an evaluation matrix A. The main diagonal only contains value 1 because it represents the pairwise comparisons of any indicator with itself. The upper triangular matrix consists of the pairwise comparisons of any indicator pair. The values of the lower triangular matrix are the reciprocal values of the corresponding values of the upper one.

Table 1: Scale and definitions of pairwise comparisons according to Saaty (1980) (Malczewski 1999, p. 183)

Intensity of importance	Definition
1	Equal importance
2	Equal to moderate importance
3	Moderate importance
4	Moderate to strong importance
5	Strong importance
6	Strong to very strong importance
7	Very strong importance
8	Very to extremely strong importance
9	Extreme importance

The next step is to calculate the column sums of matrix A. They are used to normalise the values of matrix A and to get matrix B. The mean value of the normalised values of a row in matrix B is the relative weight of the corresponding indicator

After this, the overall consistency will be taken into account. Therefore, it is necessary to calculate the eigenvalues λ_i and their mean value λ_{max}. In the next step the *Consistency Index* CI can be supplied as:

$$CI = (\lambda_{max} - n) / (n - 1) \quad (2.1)$$

The final step to get the overall consistency is the calculation of the *Consistency Ratio* CR as the quotient of CI and a *Random Index* RI:

$$CR = CI / RI \quad (2.2)$$

RI depends on the number of indicators n and was published for instance by Saaty (1994, p. 84). Pairwise comparisons are consistent if

$$CR < 0.1 \quad (2.3)$$

Otherwise the pairwise comparisons have to be adjusted until the consistency condition (inequation 2.3) is accomplished. This process can require multiple cycles, which might require some time.

According to Saaty (1994) it is possible to identify those pairwise comparisons with the highest inconsistencies and to improve the appropriate assessments. It is necessary to compare each value $a_{i,j}$ $(i < j)$ with the ratio of the corresponding weights w_i/w_j. For each pairwise comparison, this results in an index value

$$PCI_{i,j} = a_{i,j}\, w_j/w_i \; (i,j = 1 \dots n) \quad (2.4)$$

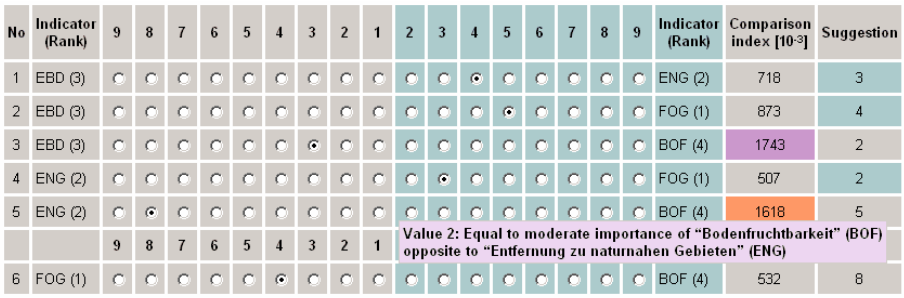
According to Vogel and Thinh (2007, p. 141) this index will be denoted as *Pairwise Comparison Index* (PCI). The higher a PCI the more intensive the pairwise comparison affects the overall inconsistency. Saaty (1994, p. 92) suggests that the pairwise comparison $a_{i,j}$ $(i < j)$ with the maximum of the $PCI_{i,j}$ values will be replaced by the appropriate weight quotient w_i/w_j. This approach optimises the overall consistency and therefore decreases CR. However, an improvement of the consistency can already be reached by approximating $a_{i,j}$ to the ratio w_i/w_j.

2.2 The web-based application

A web-based application was developed using the web-script language PHP (Hypertext Preprocessor) to implement the algorithm of the AHP in a dynamic manner in Hypertext Markup Language (HTML). Especially the pairwise comparisons of indicators are integrated. With reference to Thinh and Vogel (2007, p. 680) on that occasion, the following aspects were taken into account during the conception of the user interface and the application:

1. Simple useability,
2. Minimisation of inputs to a necessary number,
3. Guarantee of privacy,
4. Continuous data flow,
5. Fast availability of help information.

A clear structure of a website guarantees good useability. Because of that the main page is free of any unnecessary information. Mainly the page consists of a dynamic created list of pairwise comparisons (Fig. 2) and a table of indicators (not shown in Fig. 2). Additionally two HTML *fieldsets* complete the output. They include status and consistency information like the username, the number of indicators, and the number of pairwise comparisons as well as the current CR (Eq. 2.2) value (Thinh and Vogel 2007, p. 680).



No	Indicator (Rank)	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Indicator (Rank)	Comparison index [10-3]	Suggestion
1	EBD (3)	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	ENG (2)	718	3
2	EBD (3)	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	FOG (1)	873	4
3	EBD (3)	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	BOF (4)	1743	2
4	ENG (2)	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	FOG (1)	507	2
5	ENG (2)	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	BOF (4)	1618	5
		9	8	7	6	5	4	3	2	1											
6	FOG (1)	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	BOF (4)	532	8

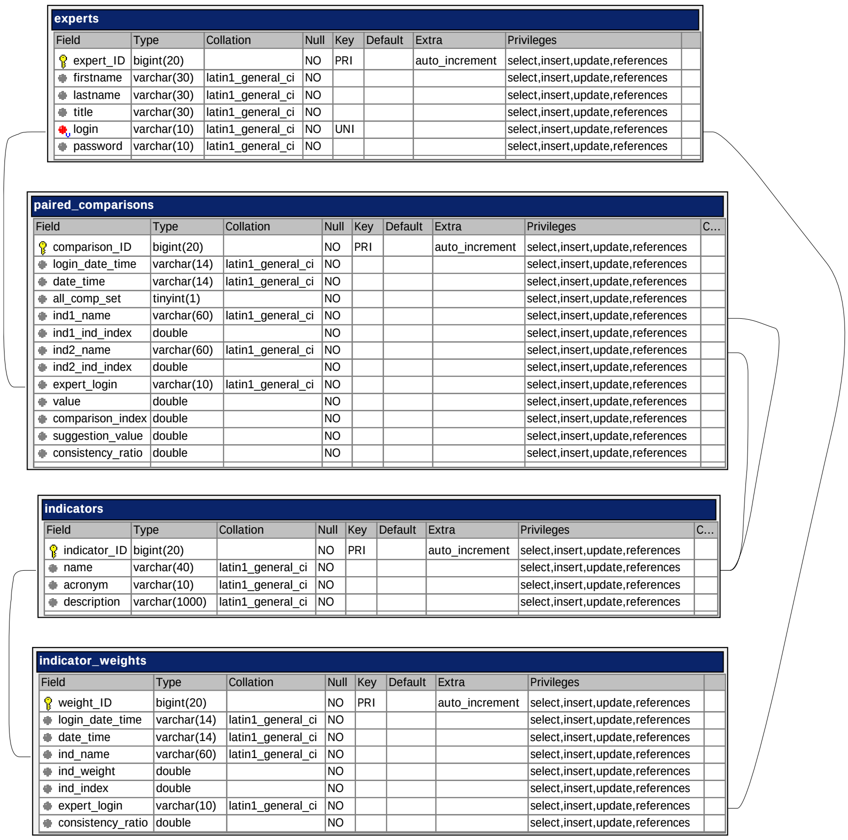
Figure 2: The web-based application provides hints about the importance of scale values and indicator information (indicator descriptions in German) (Vogel and Thinh 2007, p. 143).

To ensure a continuous data flow and to save the user inputs the PHP tool is used in combination with a MySQL database. The software package XAMPP[2] has been chosen. It includes the web server Apache, the database MySQL, the script language PHP, and the administration tool phpMyAdmin.

Experts will be provided with usernames and passwords which are saved in the *experts* table of the MySQL database *ahp-db* (Fig. 3). They shall log into the website and decide on the priority of each indicator for any possible pair of indicators. Especially in the case of a large amount of comparisons, it is necessary to minimise the number of user inputs. Therefore, 17 *radio buttons* are shown representing each scale value of each indicator preference. One mouse click is sufficient for one comparison (Thinh and Vogel 2007, pp. 680 f.). Indicator information, which are necessary to create the user interface are taken from database table *indicators*.

---

[2] http://www.apachefriends.org/de/xampp.html



experts

Field	Type	Collation	Null	Key	Default	Extra	Privileges
expert_ID	bigint(20)		NO	PRI		auto_increment	select,insert,update,references
firstname	varchar(30)	latin1_general_ci	NO				select,insert,update,references
lastname	varchar(30)	latin1_general_ci	NO				select,insert,update,references
title	varchar(30)	latin1_general_ci	NO				select,insert,update,references
login	varchar(10)	latin1_general_ci	NO	UNI			select,insert,update,references
password	varchar(10)	latin1_general_ci	NO				select,insert,update,references

paired_comparisons

Field	Type	Collation	Null	Key	Default	Extra	Privileges	C...
comparison_ID	bigint(20)		NO	PRI		auto_increment	select,insert,update,references	
login_date_time	varchar(14)	latin1_general_ci	NO				select,insert,update,references	
date_time	varchar(14)	latin1_general_ci	NO				select,insert,update,references	
all_comp_set	tinyint(1)		NO				select,insert,update,references	
ind1_name	varchar(60)	latin1_general_ci	NO				select,insert,update,references	
ind1_ind_index	double		NO				select,insert,update,references	
ind2_name	varchar(60)	latin1_general_ci	NO				select,insert,update,references	
ind2_ind_index	double		NO				select,insert,update,references	
expert_login	varchar(10)	latin1_general_ci	NO				select,insert,update,references	
value	double		NO				select,insert,update,references	
comparison_index	double		NO				select,insert,update,references	
suggestion_value	double		NO				select,insert,update,references	
consistency_ratio	double		NO				select,insert,update,references	

indicators

Field	Type	Collation	Null	Key	Default	Extra	Privileges	C...
indicator_ID	bigint(20)		NO	PRI		auto_increment	select,insert,update,references	
name	varchar(40)	latin1_general_ci	NO				select,insert,update,references	
acronym	varchar(10)	latin1_general_ci	NO				select,insert,update,references	
description	varchar(1000)	latin1_general_ci	NO				select,insert,update,references	

indicator_weights

Field	Type	Collation	Null	Key	Default	Extra	Privileges
weight_ID	bigint(20)		NO	PRI		auto_increment	select,insert,update,references
login_date_time	varchar(14)	latin1_general_ci	NO				select,insert,update,references
date_time	varchar(14)	latin1_general_ci	NO				select,insert,update,references
ind_name	varchar(60)	latin1_general_ci	NO				select,insert,update,references
ind_weight	double		NO				select,insert,update,references
ind_index	double		NO				select,insert,update,references
expert_login	varchar(10)	latin1_general_ci	NO				select,insert,update,references
consistency_ratio	double		NO				select,insert,update,references

Figure 3: Table structure of the MySQL database *ahp-db*

All results of pairwise comparisons are stored in the MySQL database *ahp-db*. Apart from pairwise comparison values the table *paired_comparisons* contains the names of the considered indicators, the login name, the current date and time as well as the current values of the PCI (Eq. 2.4) (`comparison_index`), the suggested value and the CR (Eq. 2.2) (`consistency_ratio`). The suggested value (last column in Fig. 2) is the rounded value respectively the rounded reciprocal value of the weight quotient w_i/w_j. This depends on which of the two indicators is being preferred. The background colour of the cell highlights which indicator should be favoured in the next evaluation step.

Additionally a set of PHP functions (Table 2) had been developed to calculate several matrix operations, binomial coefficients, weights, and the fast recursive quick sort algorithm (Cormen et al. 2007, 143-161). Based on these functions the weights of the indicators and the CR (Eq. 2.2) are calculated. If the value of the CR is greater than or equal to 0.1 the comparisons are inconsistent. The pairwise comparisons have to be modified.

To identify comparisons, which contribute considerable to the inconsistency, average consistency ratios for each indicator and for each pairwise comparison are calculated. High index values indicate pairwise comparisons which do only fit in a minor manner to the whole set of comparisons. Usually the experts have to adjust these comparisons, until CR < 0.1 (Thinh and Vogel 2007, p. 681).

Table 2: Main functions of PHP file functions.php

PHP function name	Function
`getScaleValueDefinition`	Delivers scale value definitions (Table 1)
`getBinomialCoefficient`	Delivers the binomial coefficient
`getVectorSum`	Delivers the value sum of a vector
`getColumnSum`	Delivers the column sum of a matrix
`getRowSum`	Delivers the row sum of a matrix
`getMatrixMultVector`	Delivers the result of the multiplication of a matrix with a vector
`getWeights`	Delivers the weight vector
`getLambda`	Delivers the lambda vector
`getConsistencyIndex`	Delivers the CI value (Eq. 2.1)
`getRandomIndex`	Delivers the RI value
`quickSortExchange`	(Quick sort sub function)
`quickSortDivide`	(Quick sort sub function)
`quickSort`	Sorts a list with the fast quick sort algorithm

Afterwards the calculated weights are saved in the database table *indicator weights*. As shown in Figure 3 the current date and time as well as the indicator name, the username and the CR are also stored in this table. These additional information can be used to recapitulate the complete user input history and for statistical as well as sensitivity analyses.

According to Thinh and Vogel (2007, p. 681) the average values of the indicator weights of experts, which accomplished overall consistency, serve as input for the CP (section 3).

3 ARCGIS EXTENSION COMPROMISE PROGRAMMING

In this section an introduction to the theory of the method *Compromise Programming* is described. Furthermore, the software implementation of the method algorithm is presented.

3.1 The Compromise Programming Method

Zeleny (1982, pp. 314-383) originally developed the method *Compromise Programming* (CP). As described by Thinh and Vogel (2006, pp. 136 f.) it is the basic idea of CP to identify solutions, which are closest to the ideal point as determined by some measure of distance. The basic postulate is accepted which means that decision makers prefer solutions as close as possible with respect to the ideal. It is known as *Zeleny's axiom of choice*. It is necessary to look for compromise solutions since the ideal solution is infeasible. For that, distances between each solution and the ideal point are calculated. To avoid a meaningless summation of values with different units and solutions biased towards those indicators that can achieve larger values, the degrees have to be normalised. Thus the degree of closeness d_j between the j^{th} indicator and its ideal is formulated by

$$d_j = \frac{\left|z_j^* - z_j(x)\right|}{\left|z_j^* - z_{*j}\right|} \qquad (3.1)$$

In this equation x is a vector, z^*_j is the ideal value for the j^{th} indicator, and z_{*j} is the anti-ideal or nadir point for this indicator. To measure the distance between each solution and the ideal point CP uses the family of L_p metrics and attempts to minimise the distance from the ideal solution (e.g. Zeleny 1982, p. 322; Ehrgott 2000, p. 90)

$$L_p(w) = \left[\sum_{j=1}^{n} w_j^p \left| \frac{z_j^* - z_j(x)}{z_j^* - z_{*j}} \right|^p \right]^{\frac{1}{p}} \rightarrow \text{Min!} \qquad (3.2)$$

That means that any point $z(x) = (z_1(x), z_2(x), ..., z_n(x))^T$ is a compromise solution if it minimises $L_p(w)$ for some choice of weights $w_i > 0$, $w_1 + w_2 + ... + w_n = 1$, and $p \geq 1$. The value p is a natural number. Ehrgott (2000, p. 92) denoted $L_p(w) \rightarrow$ min as the weighted compromise programming problems. To solve these problems it is general practice to use following values for the parameter p (Thinh 2004, p. 120):

- $p = 1$ (the city-block norm),
- $p = 2$ (the Euclidean norm), and
- $p = 10$ in the case of $p = \infty$ (the maximum norm).

The parameter p reflects the importance of the maximal deviation from the ideal point. For $p = 1$ all deviations are weighted equally. In the case of $p = 2$ each deviation is weighted in proportion to its magnitude. The greater the deviation the greater the weight will be. In case of $p = \infty$ we have a min-max-problem, that is the compromise solution minimises the maximum difference between the ideal point and the solution with respect to all indicators.

3.2 Prototype of the ArcGIS extension Compromise Programming

The aim of the development of an extension is the integration of the algorithm of the CP in the GIS ArcGIS. There are several ways to achieve this. Customisations within ArcGIS Desktop can be realised for instance by user interface customisations, VBA (Visual Basic for Applications) macros and extensions (Cameron et al. 2004, p. 47).

ArcGIS extensions provide additional GIS functionality like license and management functions. It is not possible to develop full extensions using VBA. As development environment and programming language Visual Basic 6 (VB6), Visual C++ and Visual Studio .NET are suitable. To develop the prototype of the ArcGIS extension, VB6 has been chosen because it performs with comparable performance like Visual C++ and it is a productive programming language, especially for the task of the development of appropriate user interfaces (Cameron et al. 2004, p. 67).

The ArcGIS system is built using ArcObjects software components (e.g. Burke 2003; Chang 2004) which aggregate comprehensive GIS functionality for developers. It is possible to use these components to extend ArcGIS and to develop extensions as well. The compilation of ArcObjects functionality into a Component Object Model (COM) for instance includes (Cameron et al. 2004, p. 9):

1. Creation of a project,
2. Referencing of required ArcObjects type libraries,
3. Implementation of ArcObjects interfaces,
4. Adding of source code,
5. Compilation of the source code into a binary file.

The creation of custom components has several advantages. The source code can be hidden in a binary file, ArcGIS technology can be extended virtually and the custom component can be easily delivered to end users via custom setup routines (Cameron et al. 2004, p. 9).

To realise a suitable extension user interface a form is used which includes three frames (Fig. 4). The first one includes a ListBox for the parameters p (section 3.1) and a CommandButton, which can be used to add additional parameter values. The second frame includes information about the indicators, especially the layer names currently available in ArcMap and the corresponding indicator weights. Two OptionButtons are used to decide if equal weights are used or if manually weighted inputs are expected. This is useful if the analyst use the weights determined by the web-based application (section 2.2). The last frame is used to specify the prefix of the output raster dataset names.

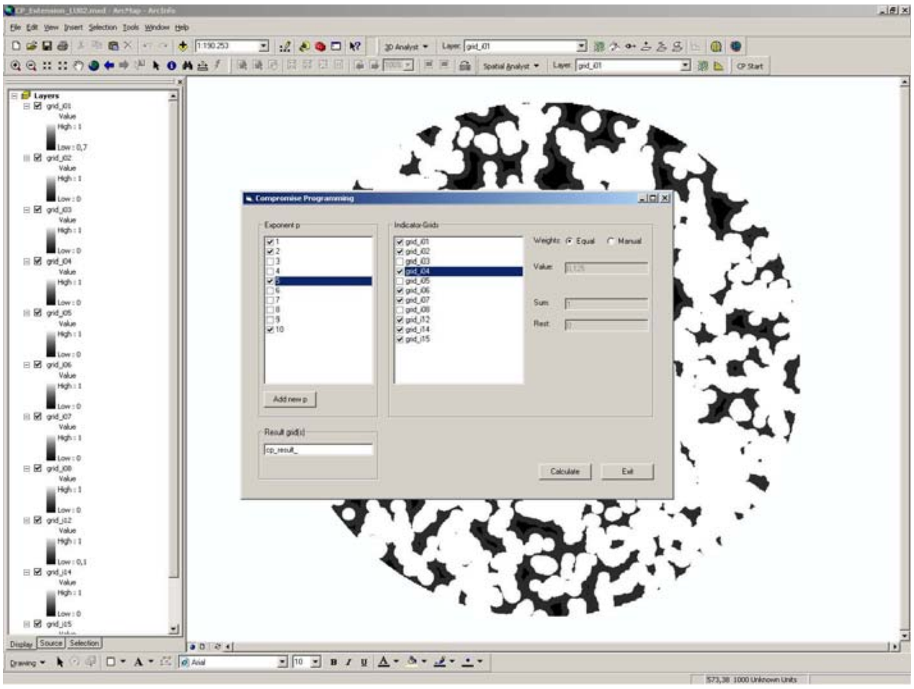


Figure 4: ArcMap with user interface of the prototype of the extension Compromise Programming

The background functionality is realised through several VB6 Sub-directives. The main functionality is fulfilled by the Sub-directives `SetWeights()` and `Calculation()`. `SetWeights()`serves as calculation module in the case of equal weighs and starts by execution of Sub-directive `Form_Initialize()`. Sub-directive `Calculation()` includes the algorithm of the CP (section 3.1).

First each selected indicator's ideal and anti ideal values are determined by the functions `RasterMinimumValue(…)` and `RasterMaximumValue(…)` via raster statistic property values. Next for each selected indicator a Map Algebra (Burrough 1998, pp. 184 f.) expression is created, which includes summand j for indicator j in equation 3.2. This serves as input for the method `Execute(…)` of object `pMapAlgebraOp`.

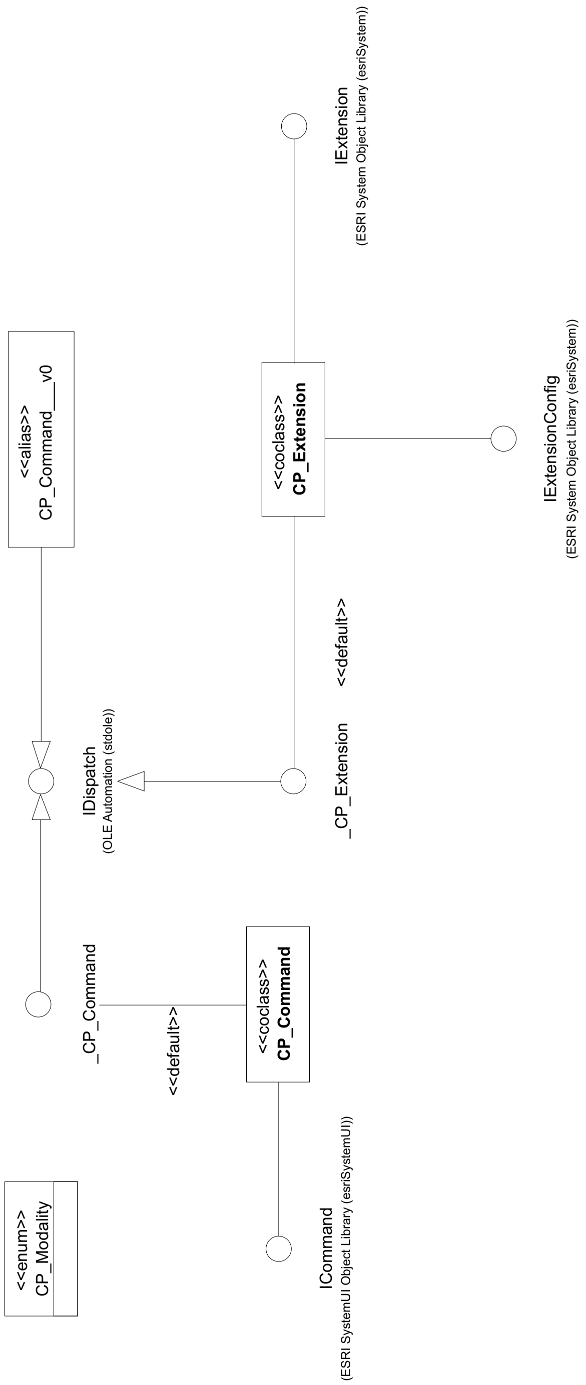


Figure 5: UML class diagram of the VB6 project of the prototype of the extension Compromise Programming

After this for each indicator, a new Map Algebra expression is built which includes the summary of all summands and the power of this value with the reciprocal value of the parameter p. This algorithm is being repeated for each selected parameter p. Finally, all calculated raster datasets are added to the active ArcMap data frame.

The development of an extension requires the implementation of further components. These are the ClassModules *CP_Extension* and *CP_Command*. As illustrated in Figure 5, the ClassModule *CP_Extension* implements the interfaces *IExtension* and *IExtensionConfig*, which are part of the ESRI System Object Library *esriSystem*.

The ClassModule *CP_Command* serves to reference the extension as a new command in the toolbar of ArcMap. It implements the *ICommand* interface. To get access to the application in VB6, the *OnCreate* event has to be used to pass in a parameter called *hook*. It is equivalent to the preset *Application* variable, which refers to ArcMap (Cameron et al. 2004, p. 70).

4 CONCLUDING REMARKS AND OUTLOOK

The paper shows that the combination of the raster-based method *Compromise Programming* with the *Analytic Hierarchy Process*, which is used to obtain indicator weights, offers a useful approach to solve MCE problems for decision makers.

Through the implementation of the CP algorithm as an extension, it is possible to evaluate suitability questions of landscapes directly within ArcGIS. Calculation results for several exponents p are available immediately as new raster layers in ArcGIS for further geoprocessing analysis.

The developed web-based tool allows easy application of AHP to estimate the relative weights for a set of indicators. The participation of experts in the evaluation process is guaranteed. The provided application does not only support the comparison process in an effective way, it also helps to shorten this process by delivering consistency information (Thinh and Vogel 2007 p. 681).

Furthermore, the analyst needs no effort to evaluate the pairwise comparisons of the experts because the weights are calculated automatically, and are stored in a database. Accordingly, calculation errors are avoided.

There are numerous potentials for further development of the software applications described above. The ArcGIS extension could be enhanced in several ways. On one hand, the access to the MySQL database could be realised automatically to get the indicator weights by continuous data flow. On the other hand, the extension could be extended in order to predefine cartographic representations of calculated suitability datasets.

Another improvement is the enhancement of the extension functionality by components, which enable the usage of distributed spatial thematic data sources, and provides a query filtering mechanism that returns only the relevant objects (Essid et al. 2007, p. 304). Interface specifications for geoinformation (GI) services published by the Open Geospatial Consortium (OGC)[3] allow for example publishing and searching spatial data and GI services (e.g. Web Catalogue Service – CSW), access to spatial data via simple spatial queries (e.g. Web Feature Service – WFS) and visualising spatial data in maps (e.g. Web Map Service – WMS) (Bernard and Ostländer 2007, p. 5).

Several user interface languages can enhance both applications. In addition a simple administration tool is useful for the web-based AHP application to create and handle connected MySQL databases.

[3] http://www.opengeospatial.org

Acknowledgements

The author would like to thank PD Dr. habil. Nguyen Xuan Thinh (Leibniz Institute of Ecological and Regional Development, Dresden), Prof. Dr. Ralf Bill (Universität Rostock, Professorship for Geodesy and Geoinformatics) and Prof. Dr. Lars Bernard (Technische Universität Dresden, Professorship of Geoinformation Systems) for their support and valuable suggestions.

REFERENCES

Bernard L, Ostländer N (2007) Assessment of Climate Change Vulnerability in the Arctic using Geographic information Services in Spatial Data Infrastructures. Climatic Change: submitted.

Burke R (2003) Getting to know ArcObjects: Programming ArcGIS with VBA. Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA USA 92373

Burrough PA, McDonnell RA (1998) Principles of Geographical Information Systems. New York: Oxford University Press (Spatial Information Systems and Geostatistics)

Cameron E, Elkins R, Gill S., Jones S, Laframboise A, Meister G, Esch S van (2004) ArcGIS Desktop Developer Guide. ArcGIS 9.0. Environmental Systems Research Institute, Inc. (ed), 380 New York Street, Redlands, CA USA 92373

Chang KT (2004) Programming ArcObjects with VBA: A Task-Oriented Approach. Boca Raton, London, New York, Washington: CRC Press LLC

Cormen T, Leiserson CE; Rivest R, Stein C (2007) Algorithmen – Eine Einführung. 2., korr. Aufl. München, Wien: Oldenbourg Wissenschaftsverlag

Ehrgott, M (2000) Multicriteria Optimization. Lecture Notes in Economic and Mathematical Systems 491, Springer-Verlag Berlin

Essid M, Lassoued Y, Boucelma, O (2007) Processing Mediated Geographic Queries: a Space Partitioning Approach. In: Fabrikant SI, Wachowicz M (Eds.) The European Information Society: Leading the way with Geo-information. (Lecture Notes in Geoinformation and Cartography) Berlin, Heidelberg: Springer-Verlag, pp. 303–315

Malczewski J (1999) GIS and Multicriteria Decision Analysis. New York, Chichester, Weinheim: John Wiley & Sons

Saaty TL (1980) The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. New York, St. Louis, San Francisco: McGraw-Hill.

Saaty TL (1994) Fundamentals of Decision Making and Priority Therory with the Analytic Hierarchy Process. Pitsburgh: RWS Publications

Saaty TL, Vargas LG (1991) Prediction, Projection and Forecasting. Boston, Dordrecht, London: Kluwer Academic Publishers

Tabucanon MT (1988) Multiple criteria decision making in industry. (Studies in production and engineering economics; 8) Amsterdam, Oxford, New York, Tokyo: Elsevier

Thinh NX (2004) Entwicklung von mathematisch-geoinformatischen Methoden und Modellen zur Analyse, Bewertung, Simulation und Entscheidungsunterstützung in Städtebau und Stadtökologie. Universität Rostock, Habilitationsschrift

Thinh NX, Vogel R (2006) Further Development of a Methodology for the Ecological Evaluation of Land-Use Options and Application to the Dresden Elbe-Region. In: Tochtermann K, Scharl A (Eds.) Managing Environmental Knowledge. EnviroInfo 2006, Graz. Aachen: Shaker Verlag, (20th International Conference on Informatics for Environmental Protection) pp. 135–143

Thinh NX, Vogel R (2007) Application of the Analytic Hierarchy Process in the Multiple Criteria Decision Analysis of Retention Areas for Flood Risk Management. In: Hryniewicz O, Studzinski J, Romaniuk M (Eds.) Environmental Informatics and Systems Research. EnviroInfo Warsaw 2007. Aachen: Shaker Verlag, pp. 675–682

Vogel R, Thinh NX (2007) Web-gestützte Anwendung des AHP zur Gewichtung von Indikatoren für die Eignungsbewertung von Retentionsflächen. In: Wittmann J, Wohlgemuth V (Eds.) Simulation in Umwelt- und Geowissenschaften. Workshop Berlin 2007. Aachen: Shaker Verlag, (ASIM-Mitteilung 110), pp. 137–146

Zeleny, M. (1982) Multiple criteria decision making. New York, St. Louis, San Francisco: McGraw-Hill.