3D DATA VISUALISATION WITHIN SPATIAL DECISION SUPPORT SYSTEMS BY USING ARCGIS ENGINE

R. Laudien[*], A. Christmann, S. Brocks

Dept. of Geography (GIS & RS), University of Cologne, Albertus-Magnus-Platz, Cologne- rlaudien@uni-koeln.de, christmann@www-dev.de, mail@sebastian-brocks.de

Commission II, WG 7

KEY WORDS: Software engineering, 3D modelling, Spatial Decision Support, ArcGIS Engine

ABSTRACT:

Based on the current state of science and technology, computer based Spatial Decision Support Systems (SDSSs) can be used for modern management and planning purposes. To fulfil the requirements of being advanced comprehensive systems, such modern SDSSs provide access to individual data and computer models. They combine tools and analysis functions of Geographical Information Systems (GIS), Decision Support Systems (DSS), Remote Sensing (RS), and numerical, expert or statistical models. In addition to that, such complex systems include expert knowledge and enable the user to interact with the SDSSs during run-time. To generate such user specific results, SDSSs need to access several different datasets with different formats. According to a given adapted logical decision tree, these systems are programmed in a modular and question-specific way.

This contribution shows a design and development of visualising three dimensional data within a decision support framework by using ArcGIS Engine and Java. Besides the description of the methodological approach, an example demonstrates the functionalities of the developed 3D-ArcGIS-Scene-Panel.

INTRODUCTION

Spatial data, mathematical models and expert knowledge are currently more and more incorporated in decision making processes (Bareth, 2009). Systems which compute these procedures need to access several different datasets with different formats. Modern Spatial Decision Support Systems (SDSSs) combine such data within complex process-based toolboxes (Leung, 1997; Wright, 1993; Sharma *et al.* 2006). According to a given logical decision tree, these systems are programmed question-specific (Hartkamp *et al.* 1999; Seppelt, 2003). Thus, detailed information about the requests, knowledge and personal needs of the potential users are essential (Keenan, 2006), and the developed systems need to be adapted based on the GIS- (Geographical Information System), RS- (Remote Sensing), and model-knowledge of the decision makers (Laudien and Bareth, 2007).

To meet the requirements of being comprehensive decision support tools, the individually generated components of a SDSS need to be connected (Bareth and Yu, 2007; Taylor *et al.* 1999; Rizzoli *et al.* 1997). This matter of fact is realised by coupling these modules with individually developed interfaces (Schneider 2003; Shaffer *et al.* 2001; Hartkamp *et al.* 1999, Seppelt, 2003). The interfaces access the system modules, e.g. data, models, GIS-visualizations or RS-analyses. By following this approach, models are independent from the SDSS body and vice versa. Only the results of each subsystem are exchanged (Longley *et al.* 1999). Thus, it is possible to develop different SDSS components at the same time independently from each other.

3D GIS are GIS systems providing data structures and operations for the presentation, management and processing of points, lines, surfaces and volumes in three dimensional spaces. (Breunig, 2001).

An increased interest in modelling 3D data like digital elevation models or city models in conjunction with new methodological approaches implementing three dimensional models and visualizations provide a high degree of relevance regarding planning functions (Henning, 2008). Therefore, this paper focus on the design and development of visualising three dimensional data within a decision support framework by using ArcGIS Engine and Java. The programming approach described in this paper was developed for the interdisciplinary research project IMPETUS (*An integrated approach to the efficient management of scarce water resources in West Africa*). One major task of the third project phase of IMPETUS (2006-2009) was the development, implementation and application of GIS-based SDSSs to support water resource management in selected river catchments of Benin and Morocco. The programmed systems were implemented in a Java/XML based framework (Enders *et al.* 2007; Enders and Diekkrüger,2009). Further information concerning IMPETUS is documented by (Speth *et al.* 2005).

SPATIAL DECISION SUPPORT SYSTEMS

As soon as spatial data is embedded in a Decision Support System, GIS functionalities get an important role. These functions support the user making spatially diverse decisions. In this context, the term Spatial DSS was established in the mid 1980s/1990s (Armstrong *et al.* 1986; Armstrong *et al.* 1990; Densham 1991; Goodchild *et al.* 1993; Wilson 1994; Crossland *et al.* 1995). Mennecke (1997) sees a SDSS as an easy-to-use subset of a GIS, which incorporates facilities for manipulating and analysing spatial data. In addition to that, Malczewski

(2006) addresses that a SDSS also includes functionalities of multicriteria decision analysis. Furthermore, SDSSs provide the opportunity of integrating, visualizing and evaluating various analytical models, and therefore can be used to develop management strategies (Muller, 1993; Keenan, 1996; Leung, 1997; Malczewski, 1999; Manoli *et al.* 2001; Yeh 1999). The structure of such systems is introduced considering latest technology developments and can be considered as the geo-data infrastructure for decision support related to specific spatial based questions and problems (Bareth, 2009). The SDSSs are connected to extensive geo-databases which include (i) all data for the system modelling, (ii) the models itself in a Model base Management System (MBMS), (iii) the interfaces between data, models and user knowledge, as well as (iv) the functions for data generation and data mining (Leung, 1997)

DESIGN AND DEVELOPMENT

In the interdisciplinary research project IMPETUS (Speth *et al.* 2005), the object oriented programming language Java was used to develop platform independent SDSSs. Java allows the modelling of numerous scientific questions. It reduces the complexity of a program source code in terms of abstraction, encapsulation, and defined interfaces. Object oriented programming donates an appropriate representation of the relationships between classes and objects, and increases the developer efficiency regarding the reuse of software code (Krüger, 2006). Within this programming environment, the source code is translated into byte code and then executed in a special environment, the so called Java Runtime Environment (JRE). The major part of the JRE is the Java Virtual Machine (Java-VM) which interprets and executes the byte code. Because Java-VMs exists for multiple different Operating Systems, the major advantage of programming in Java consists in the fact that all developments can run on different computers and different operating systems. Therefore, software which is developed with Java is nearly platform independent (Herter and Koos, 2006).

To fulfil the given requests of guaranteeing GIS- and RS-functionalities within the SDSSs, the ESRI developer library ArcGIS Engine 9.2 was used. With ArcGIS Engine, the software developer gets the opportunity to implement spatial analysis functions. The full version of ArcGIS Engine, which is accessible by C++, VB.NET, C#, or Java, comes with several different extensions which can be integrated, based on the users' needs. To execute the SDSSs, the JRE and the ArcGIS Engine Runtime Environment need to be installed on the computer. The IMPETUS SDSSs are developed using the programming environment Eclipse SDK which contains the Eclipse platform (Eclipse 3.2, (http://www.eclipse.org/), tools for Java programming and the environment to develop Eclipse plug-ins. In addition to that, the Subclipse plug-in is used to store the source code in a subversion repository (SVN). Such a repository allows several software developers to work on the same project independently without causing backup or versioning errors. Besides the standard Java GUI components, specific GIS- and RS-components are used for the programming. These include functionality available through the ArcGIS Engine library.

The (geo-) data (e.g. rasters, vectors, and alphanumerical data) of the single SDSSs are stored in ESRI 9.2 file-based geo-databases.

For the 3-D visualisation special framework processors and functionalities were developed. The following selection of code

excerpts illustrate configuration of the *Multi Document Desktop* as well as some of the dependent processors within the client configuration. All objects instantiated during run time are stored in a global parameter map, thus instantiated objects and data residing in memory are accessible by object names.

The following code excerpt from the configuration xml demonstrates how feature classes are read from a file based geo database into memory to be further processed. The processor is capable to load several feature classes at once.

```
<processor
class="de.isdss.ext.arcgis.processors.ArcGISLoadFeature
ClassesProcessor">

<param key="readFile">${_CACHEDIR_}/farmadam/BeL4.gdb
</param>
<param key="outWorkspace">geodb.workspace</param>
<param key="featureclasses">14</param>
<!-- General Purpose Feature classes -->
<param key="inFeatureclass.1">main_cities</param>
<param key="outFeatureclass.1">fc.cty</param>
<param key="inFeatureclass.2">streets</param>
<param key="outFeatureclass.2">fc.str</param>
<param key="inFeatureclass.3">regions</param>
<param key="outFeatureclass.3">fc.reg</param>
<param key="inFeatureclass.4">departments</param>
<param key="outFeatureclass.4">fc.dep</param>
<param key="inFeatureclass.5">gemeinden</param>
<param key="outFeatureclass.5">featureclass.com</param>
<param key="inFeatureclass.6">AEZ</param>
<param key="outFeatureclass.6">fc.aez</param>
<param key="inFeatureclass.7">regions_Line</param>
<param key="outFeatureclass.7">fc.reg.line</param>
<param key="inFeatureclass.8">departments_Line</param>
<param key="outFeatureclass.8">fc.dep.line</param>
<param key="inFeatureclass.9">gemeinden_Line</param>
<param key="outFeatureclass.9">fc.com.line</param>
<param key="inFeatureclass.10">AEZ_Line</param>
<param key="outFeatureclass.10">fc.aez.line</param>
<!-- Point Feature classes to be cloned later -->
<param key="inFeatureclass.11">gem_for</param>
<param key="outFeatureclass.11">fc.com.ori</param>
<param key="inFeatureclass.12">dep_for</param>
<param key="outFeatureclass.12">fcdep.ori</param>
<param key="inFeatureclass.13">reg_for</param>
<param key="outFeatureclass.13">fc.reg.ori</param>
<param key="inFeatureclass.14">aez_for</param>
<param key="outFeatureclass.14">fc.aez.ori</param>
</processor>


<!-- Fields for Land Use -->
<processor
class="de.isdss.ext.processors.NewArrayProcessor">
<param key="name">array.attributes.landuse</param>
<param key="element.0">crops</param>
<param key="element.1">forest</param>
<param key="element.2">fallow</param>
<param key="element.3">pasture</param>
<param key="element.4">savanna</param>
<param key="element.5">other</param>
<param key="elements">6</param>
</processor>
```

To visualize the model data results, the values must be read from the data source. In this case a MS Excel sheet file containing research data and model calculation rules, that are processed each time this processor, is called providing a set of results in memory during run-time.

```
<processor
class="de.isdss.ext.processors.GetXlsArrayProcessor">
<param key="file">
${_CACHEDIR_}/farmadam/BeL4_data.xls
</param>
<param key="sheet">FARMADAM</param>
<param key="rows">10</param>
<param key="cols">
E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB,AC,AD
,AE,AF,AG,AH,AI,AJ,AK,AL,AM,AN,AO,AP,AQ,AR,AS,AT,AU,AV,
AW,AX,AY,AZ,BA,BB,BC,BD,BE,BF,BG,BH,BI,BJ,BK,BL,BM,BN
</param>
<param key="name">array.com.ids</param>
```

```
</processor>
```

After loading and calculating model data and feature data into memory, those two data sources must be merged together to create a visualization to be used later. For this purpose processors were developed to manage and manipulate resources during execution of the application.

```
<processor
class="de.isdss.ext.arcgis.processors.ArcGISAddFeatureL
ayerAttributesProcessor">
<param key="featureclass">layer.communes</param>
<param key="order">rows.com</param>
<param key="ids">ids.com</param>
<param key="type">lu.types</param>
<param key="attribute">lu.attr</param>
<param key="data">lu.communes</param>
</processor>
```

To prepare visualizations to be used in ArcGIS Scene or Map components, feature classes must be converted into feature layers. A feature layer is a relational, georeferenced model containing rules on how this data is to be visualized. In this case a point feature class is demonstrated having its result shown as extruded symbols on the scene it will be used in.

```
<!-- Communes -->
<!-- Land Use -->
<!-- LUC Crops -->
<processor
class="de.isdss.ext.arcgis.processors.ArcGISGet3DUnique
PointLayerProcessor">
<param key="featureclass">fc.com.point</param>
<param key="symbol.angle">0</param>
<param key="symbol.width">3.0</param>
<param key="symbol.depth">1.0</param>
<param key="symbol.size">1</param>
<param key="symbol.offset.x">-3750.0</param>
<param key="symbol.offset.y">-3750.0</param>
<param key="symbol.offset.z">0</param>
<param key="symbol.shape">character</param>
<param key="symbol.font">
ESRI Enviro Hazard Analysis
</param>
<param key="symbol.index">92</param>
<param key="color.red">255</param>
<param key="color.green">255</param>
<param key="color.blue">0</param>
<param key="scale.min">2000000</param>
<param key="scale.max">500000</param>
<param key="name">layer.com.crops</param>
<param key="by.field">NAME</param>
<param key="with.field">crops</param>
<param key="apply.on">depth</param>
<param key="value.scale">20.0</param>
</processor>
```

The so prepared set of feature layers can be incorporated into the *Multi Document Desktop* component. Since all data in memory is relational, several different perspectives can be created by projecting the loaded feature layers based on sets of fields. Each of those projections can be presented in a different way. In this case a 3-D Scene as well as a pie chart diagram is rendered using the same type of data source.

```
<function id="multipanel"
class="de.isdss.ext.arcgis.functionalities.ArcGISMultiP
anelFunctionality">

<param key="panels">5</param>

<param key="panel.1.title">scene.title</param>
<param key="panel.1.type">scene</param>
<param key="panel.1.zfactor">25.0</param>
<param key="panel.1.baseheight">layer.dgm</param>
<param key="panel.1.layers">5</param>
<param key="panel.1.layer.1">layer.cities</param>
<param key="panel.1.layer.2">layer.communes</param>
<param key="panel.1.layer.3">layer.regions</param>
<param key="panel.1.layer.4">layer.streets</param>
<param key="panel.1.layer.5">layer.luc</param>
<!-- ... -->
<param key="panel.4.title">com.luc</param>
<param key="panel.4.type">piechart</param>
<param key="panel.4.featureclass">com.luc</param>
<param key="panel.4.title">label.luc</param>
<param key="panel.4.fields">luc.fields</param>
<param key="panel.4.legendfield">NAME</param>
<param key="panel.4.order">columns</param>
<!-- ... -->
</function>
```

3-D DATA VISUALISATION WITHIN THE MULTI DOCUMENT DESKTOP ENVIRONMENT

The *Scientific Model Integration pipeLine Engine* (Enders and Diekkrüger, 2009) was used as the software framework for IMPETUS. *SMILE* is a pipeline based framework architecture providing processor interfaces written in Java and configured using a XML client configuration. Hence, complex applications could be formed by interconnecting processors in a linear way that were configured using processor specific parameters within the XML client configuration. To communicate between each other, processors share one common parameter map in memory which is organized as a hash, associating parameter names to values. As the pipeline is processed, values are stored and read from the parameter map. The parameter map is a collection of any Java based object.

Since numerous field research results and simulation models of IMPETUS were provided as spreadsheet files, the processors for reading, writing and calculating values had to be implemented as *SMILE* processors. Jakarta's POI-HSSF API was used to meet the requirements of accessing MS Excel spreadsheets. In addition to that, other processors had to be created to visualize data on screen (e.g. JFreeChart API, ESRI ArcGIS Engine components like the SceneBean).

To provide a wide variety of data representation and to show input and output at the same time, a *Multi Document Desktop* was implemented, giving the user the opportunity to choose between several visuals giving several perspectives to selected and projected data simultaneously (Fig. 1).
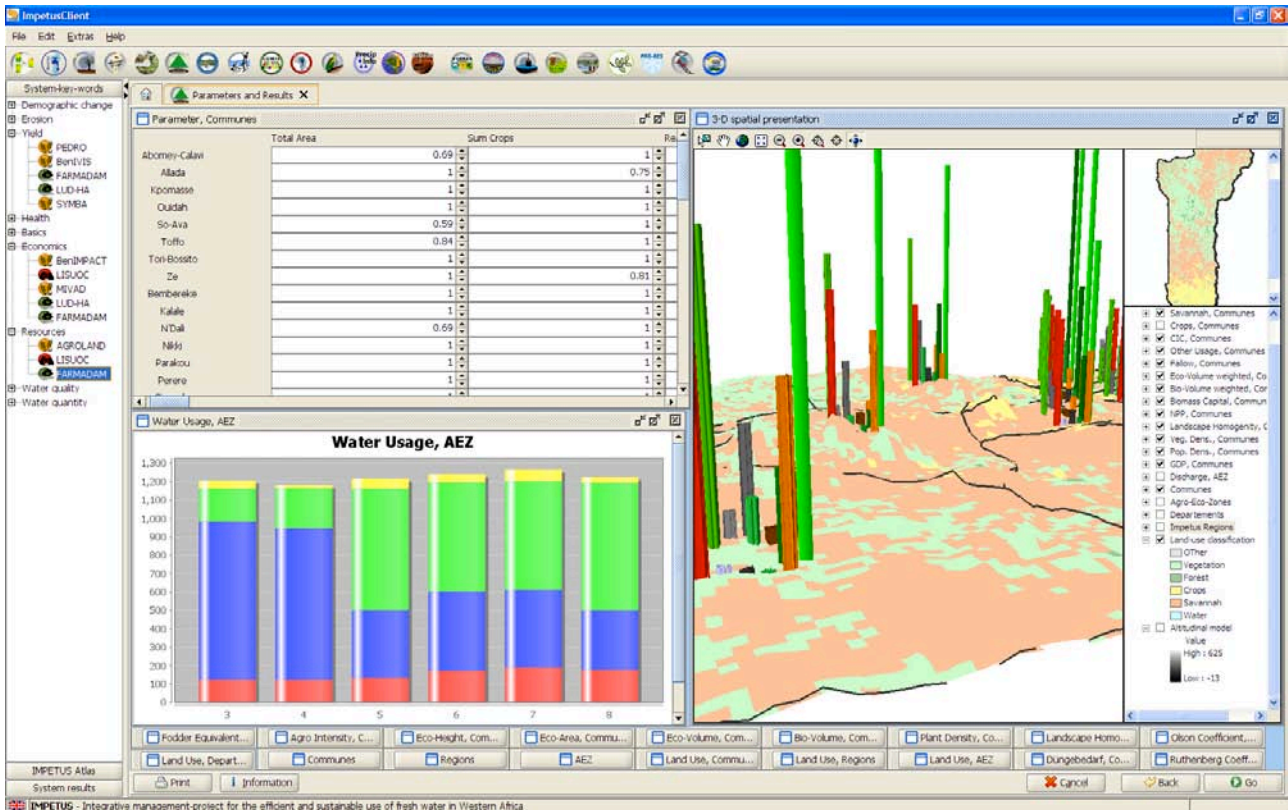
Figure 1. Screenshot of the 3-D Multi Document Desktop.

By using the *Multi Document Desktop* within the Client each component is rendered as a single window. Windows can be opened or closed, changed in size and moved freely on the screen. All windows that are currently closed are shown as buttons on the bottom of the *Multi Document Desktop*. Depending on the configuration each window renders a visualization of specific data (simulation results, research data) as either tables or charts. Additionally, a simulation parameter input panel (Fig. 1 upper left) allows the user to change simulation parameters at runtime as well as the visualisation output in 2-D or 3-D. Within maps and scenes rectified simulation results and parameters can be visualized.

Within the 3-D spatial presentation the user is able to navigate freely through space and select specific features. The land use classification, provided as raster data is projected into the third dimension by using a DEM. The resulting TIN could be overlaid with other vector data from the geo-database (e. g. major settlements, or borders of administrative zones).

Each zone of interest contains several sets of rendered bars, which are visualizing the simulation results. This way the user can compare different areas in regard to specific simulation results. Each value is visualized using an extruded coloured symbol.

On the right side of the 3-D spatial presentation, a configuration panel is placed in which each layer can be turned on and off.

By clicking the Go button on the bottom right of the application, the currently set simulation parameters are used to recalculate new simulation results based on the current state of the simulation and the components are refreshed. To realize these kinds of iterative simulations that use accumulating decision effects for different simulation cycles, the last processor in the 3.D visualisation processing pipeline points

back to the first one forming a simulation loop using the pipeline based framework (Fig. 2).
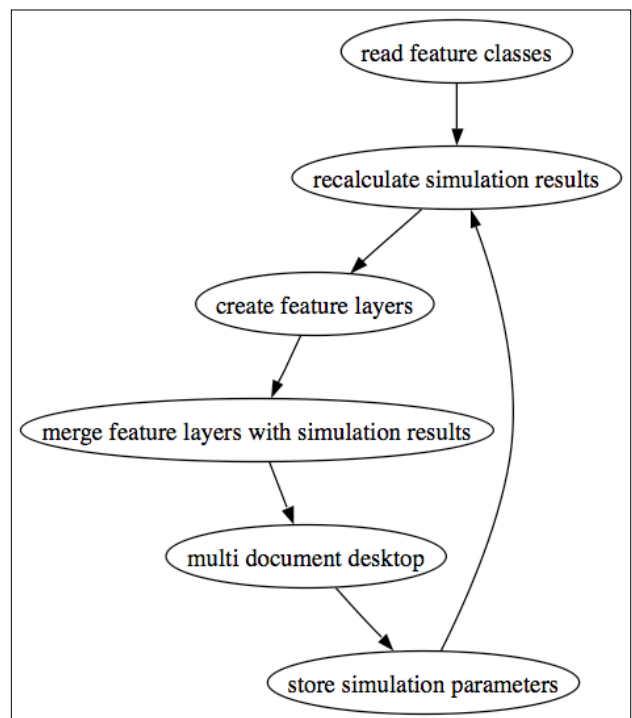

Figure 2. Farmadam pipeline structure.

Each visualized location is composed of a set of extruded symbols visualizing the simulation results and used parameters. For each commune a different set of symbols is used to allow visualization of decision effects in a three dimensional scene. Figure 3 illustrates effects on eco and bio volume of crops,

forest, grassland, savannah, available water resources, farming, fertilizer requirements, urbanization, and population density as well as economic and additional parameters as an example. By using these sets of 3D symbols, different areas of interest become comparable in regard to each other showing effects of decisions. Continued execution of the simulation model visualizes long time decision effects, allowing the user to identify decisions producing unwanted results as well as the possibility to test possible countermeasures.
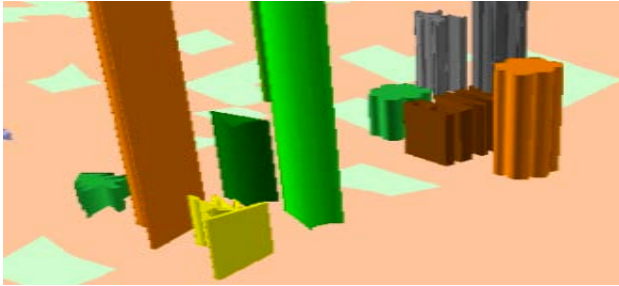

Figure 3. Extruded symbols.

Other components are used to present decision effects and parameters in more detail like different types of charts and diagrams as well as tables, allowing import and export of all relevant parameters and results. This way the current state of the simulation can be stored and imported as initial simulation state for subsequent alternative simulations.

CONCLUSION

The usage of spreadsheet files to describe simulation models and field research data reduces the necessary programming effort, especially with regard to the problem specific point of view. However, this comes at the cost of performance, since loading, recalculating and writing spreadsheet files is a time consuming process. Another drawback are the limitations of spreadsheet files, for example limiting the number of records depending on the used file format as well as the capabilities of the HSSF spreadsheet API for Java, which is used in this system. On the other hand, the usage of spreadsheet files reduced the required development time since model changes and data incorporation can be done by researchers directly without requiring special programming knowledge. By developing single generic processors with a high variability by using parameter based configuration within the XML-client configuration, development was highly flexible and changing requirements during the phase of development (often a time consuming task) could be matched very quickly. Three dimensional representations of spatial topology in conjunction with spatially represented data provide a user friendly, easy to navigate interface having an improved visibility compared to common approaches using two dimensional representations. Additionally, three dimensional visualization techniques simplify understanding of complex situations from the users point of view, especially when working with several different types of data sets in conjunction to examined locations encompassing multiple fields of interest simultaneously. Since 3D capabilities in GIS systems are increasing recently, 3D SDSS applications will become more common in the near future. Currently 3D capabilities of GIS and therefore SDSS applications are not yet fully supported by available frameworks. Hence, more research is required to match upcoming requirements to fully use the potential of modern information technology.

REFERENCES

Armstrong, A. P. and Densham, P. J., 1990. *Database organization strategies for spatial decision support systems.* International Journal of Geographical Information Systems, 4 (1), pp. 3-20.

Armstrong, M.P., Densham, P.J. and Rushton, G., 1986. Architecture for a microcomputer Based Spatial Decision Support System. In: *Proceedings of the 2nd International Symposium of Spatial Data Handling.* Williamsville: International Geographical Union), pp. 120-131.

Bareth, G. and Yu, Z., 2007. Interfacing GIS with a process based agro-ecosystem model - case study North China Plain. In: X. Tang, Y.Liu, J Zhang, and W. Kainz, eds. *Advances in spatio-temporal analysis.* London: Taylor & Francis.

Bareth, G., 2009 (in print). GIS- and RS-based Spatial Decision Support: Structure of a Spatial Environmental Information System (SEIS), *International Journal of Digital Earth.*

Breunig, M. 2001: *On the Way to Component-Based 3D/4D Geoinformation Systems.* Springer, Berlin ISBN 978-3540678069

Crossland, M.D., Wynne, B.E. and Perkins, W.C., 1995. Spatial Decision Support Systems: An Overview of technology and a test of efficacy. *Decision Support Systems*, 14 (3), pp. 219-235.

Densham, P.J., 1991. Spatial Decision Support Systems. *In* D.J. Maguire, M.F. Goodchild and D.W. Rhind, eds. *Geographical Information Systems.- Vol. 1: Principles.* Harlow: Longman Scientific & Technical, pp. 403-412.
Enders, A. and Diekkrüger, B. (2009): Development of a Spatial Decision Support Framework for IMPETUS project in West Africa.- In: Ioannis N. Athanasiadis . Pericles A. Mitkas , Andrea E. Rizzoli . Jorge Marx Gómez (Eds.): Information Technologies in Environmental Engineering, Proceedings of the 4th International ICSC Symposium, Thessaloniki, Greece, May 28-29, 2009, pp. 132-148.

Enders, A., Laudien, R. and Hoffmann, R., 2007. Spatial Decision Support Systems. In Interaktives Management-Projekt für einen Effizienten und tragfähigen Umgang mit Süßwasser in Westafrika: Fallstudien für ausgewählte Flusseinzugsgebiete in unterschiedlichen Klimazonen, Siebter Zwischenbericht, pp. 7-21 [online], Available from: http://www.impetus.uni-koeln.de/content/download/ZB2006/IMPETUS_Zwischenbericht_2006.pdf, [accessed 25 February 2009

ESRI, 2008. *ArcGIS 9.2 Desktop Help, Geo-databases and ArcSDE - Types of geo-databases* [online]. Available online at: http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Types_of_geodatabases [accessed December 2009]

Goodchild, M.F. and Densham, P., 1993. *Initiative 6: Spatial decision support systems (1990-1992)* (Santa Barbara, CA: National Center for Geographic Information and Analysis).

Hartkamp A .D., White J. W. and Hoogenboom, G., 1999. Interfacing GIS with agronomic modeling: A Review. *Agronomy Journal*, 91, pp. 761-772.

Hartkamp A .D., White J. W. and Hoogenboom, G., 1999. Interfacing GIS with agronomic modeling: A Review. *Agronomy Journal*, 91, pp. 761-772.

Henning S.D. 2008: Prozessierung von Laserscanndaten zur Erstellung eines 3D-Stadtmodells: CampusGIS-3D pp. 1. Diplomarbeit an der Universität zu Köln, Geographie (GIS / RS).

Herter, M., Koos, B. 2006: *Java und GIS. Programmierung – Beispiele – Lösungen*, Wichmann, ISBN 978-3-87907-379-5, 318p.

Keenan, P., 1996. Using a GIS as a DSS Generator. In: J. Darzentas, J. S. Darzentas, and T. Spyrou, eds. *Perspectives on DSS*. University of the Aegean, Greece, pp. 33-40.

Keenan, P.B., 2006. Spatial Decision Support Systems: A coming of age. *Control and Cybernetics*, 35(1), pp. 9-27.

Krüger, G., 2006. *Handbuch der Java-Programmierung.- 4. Auflage*. Munich: Addison-Wesley.

Laudien, R. and Bareth, G., 2007. Entwicklung und Programmierung von räumlichen Entscheidungsunterstützungssystemen mit ArcGIS Engine und Java. In: *GIS-Geoinformationssysteme*, 4, pp. 16-21.

Leung, Y., 1997. *Intelligent Spatial Decision Support Systems*. Berlin: Springer.

Longley P.A, Goodchild D.J. Maguire and D.W. Rhind, 1999. Data quality – introduction. In: P.A. Longley M.F. Goodchild D.J. Maguire and D.W. Rhind, eds. *GIS, Vol. 1*. New York: Wiley, pp. 175-176.
Malczewski, J., 1999. *GIS and multicriteria decision analysis*. New York: Wiley.

Malczewski, J., 2006. GIS-based multicriteria decision analysis: a survey of literature. I*nternational Journal of Geographical Information Science*, 20 (7), pp. 703-726.

Manoli, E., et al., 2001. Water demand and supply analysis using a spatial decision support system. *Global NEST: The International Journal*, 3 (3), pp. 199-209.

Mennecke, B.E., 1997. Understanding the Role of Geographic Information Technologies in Business: Applications and Research Directions. *Journal of Geographic Information and Decision Analyis*, 1, pp. 44-68.

Muller, J.-C., 1993. Latest developments in GIS/LIS. *International Journal of Geographical Information Systems*, 7 (4), pp. 293-303.

Rizzoli, A.E. and Young, W.J., 1997. Delivering environmental decision support systems: software tools and techniques. *Environmental Modelling & Software*, 12 (2-3), pp. 237-249.

Schneider K., 2003. Assimilating remote sensing data into a land surface process model. International Journal of Remote Sensing, 24, pp. 2959-2980.

Seppelt K., 2003. *Computer-based environmental management. Weinheim*: Wiley-VCH.

Shaffer M.J., Ma L. and Hansen, S., 2001. Introduction to simulation of carbon and nitrogen dynamics in soils. In: M.J. Shaffer., L. Ma and S. Hansen M.J. Shaffer., L. Ma and S. Hansen, eds. *Modeling carbon and nitrogen dynamics for soil management*, Boca Raton: Lewis Publishers, pp. 1-10.

Sharma T., Carmichael J. and Klinkenberg, B., 2006. Integrated modeling for exploring sustainable agriculture futures. *Futures*, 38 (1), pp. 93-113.

Speth, P., et al., 2005. IMPETUS-West Africa- An integrated approach to the efficient management of scarce water resources in West Africa – Case studies for selected river catchments in different climate zones. In: *DLR– Projektträger im DLR*, eds. *GLOWA – German Programme on Global Change in the Hydrological Cycle, Status Report 2005*, pp. 86-94.

Taylor, K., Walker, G. and Abel, D., 1999. A framework for model integration in spatial decision support systems. *International Journal of Geographical Information Science,* 13 (6), pp. 533-555.

Wilson, R.D., 1994. GIS & Decision Support Systems. *Journal of Systems Management*, 45 (11), pp. 36-40.

Wright J. R. and Buehler K. A., 1993. Probabilistic Inferencing and Spatial Decision Support Systems. In: J. R. Wright, L. L. Wiggins, R. K. Jain and T. J. Kim, eds. *Expert Systems in Environmental Planning*. Berlin: Springer, pp. 119-144.

Yeh, A., 1999. Urban planning and GIS. *In:* P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind, eds. *Geographic information system.- Vol.2*. New York: John Willey and Sons. August 2001