

IMPLEMENTATION OF NON-JAVA BASED INTERFACES TO EMBED EXISTING MODELS IN SPATIAL DECISION SUPPORT SYSTEMS - CASE STUDY: INTEGRATION OF MS[®] EXCEL-MODELS IN IWEGS -

R. Laudien^a *, S. Klose^b, A. Klose^c, C. Rademacher^d, S. Brocks^a

^aUniversity of Cologne, Dept. of Geography (GIS & RS), Albertus-Magnus-Platz, 50923 Cologne, Germany - (rlaudien, sebastian.brocks,.)@uni-koeln.de

^bUniversity of Bonn, Dept. of Geology, Nussallee 8, 53115 Bonn - stklose@uni-bonn.de

^cUniversity of Bonn, Dept. of Geography, Meckenheimer Allee 166, 53115 Bonn - stklose@uni-bonn.de

^dUniversity of Cologne, Dept. of Ethnology, Albertus-Magnus-Platz, 50923 Cologne - c.rademacher@uni-koeln.de

WgS-PS: WG II/4-: Spatial Planning and Decision Support Systems

KEY WORDS: Spatial Decision Support, Software Development, Coupling Xls Based Models, Interactive Software

ABSTRACT:

This paper shows the programming and usage of computer based interfaces to embed models in modern SDSSs. The special focus is on the development and implementation of non-Java based interfaces which couple already existing Microsoft[®] (MS) Excel-based models to individual designed SDSSs. To access MS Excel files within these developed systems, Java API POI-HSSF (by Jakarta) is used. HSSF provides a way to create, modify, read and write *.xls-spreadsheets. Besides that, this library supports cell-functions and -formulas. Based on the state-of-the-art software development and design approach MVC (Model-View-Controller), the interfaces are developed as processors. To meet the given requirements of accessing MS Excel, three processors are programmed: (i) *GetXLSValueProcessor*, (ii) *SetXLSValueProcessor* and (iii) *RecalcXLSProcessor*. Each of these processors connects the SDSSs to the *.xls -file, accesses a specific spreadsheet and uses the defined cells within that sheet. With these processors, it is possible to access and run complex Excel-based models from a user-friendly SDSS-Graphical User Interfaces (GUIs) by (i) providing the given input parameters from the Excel cells, (ii) allowing to edit the parameters in the SDSS-GUI, (iii) permitting to recalculate and run the model irrespective of the SDSS, (iv) getting the processed outputs from the model and finally (v) visualize them. In addition to the description of the development, preliminary results based on the developed SDSS IWEGS are shown.

RÉSUMÉ:

L'article suivant présente la programmation et l'utilisation des interfaces informatiques pour intégrer des modèles dans des systèmes de support à la prise de décision spatiaux (SSPD). La publication focalise sur le développement et l'implémentation des interfaces fondées non – java qui couplent des modèles en Microsoft[®] (MS) Excel existant déjà aux SSPD conçus individuellement. Java API POI-HSSF (de Jakarta) est utilisé pour qu'on puisse avoir accès aux fichiers MS Excel en l'espace des systèmes développés. HSSF propose une solution pour créer, modifier, lire et écrire des fichiers *.xls. En plus cette bibliothèque soutient des fonctions et formules des cellules.

Les interfaces sont développées comme processeurs fondés sur l'approche de développement des logiciels MVC (Model-View-Controller). Trois processeurs sont programmés pour garantir la demande d'accéder MS Excel, : (i) *GetXLSValueProcessor*, (ii) *SetXLSValueProcessor* et (iii) *RecalcXLSProcessor*. Chaque processeur connecte le SSPD avec les fichiers *.xls, accède un tableau spécifique et utilise des cellules définies dans le tableau. A l'aide de ces processeurs il est possible d'accéder et exécuter des modèles complexes fondés. Les modèles sont contrôlés par l'interface graphique d'utilisateur (IGU) du SSPD, qui est confortable et facile à utiliser. L'interface (i) propose les paramètres intrants définis dans les cellules d'Excel, (ii) permet d'édifier les paramètres par l'IGU, (iii) permet de recalculer et exécuter le modèle indépendant du SSPD, (iv) reçoit les résultats traités par le modèle et finalement (v) visualise ces résultats.

Au delà de la description du développement, des résultats provisoires du SSPS IWEGS sont montrés.

KURZFASSUNG:

Der vorliegende Beitrag zeigt die Programmierung und den Einsatz von computer-basierenden Modellschnittstellen in modernen raumbezogenen Entscheidungsunterstützungssystemen (SDSS). Der spezielle Fokus liegt dabei bei der Entwicklung und Implementierung von nicht-Java-basierenden Schnittstellen, die bereits existierende MS[®] Excel Modelle mit individuell entwickelten SDSS koppeln. Um MS Excel Dateien innerhalb der entwickelten Systeme anzusprechen zu können, wird das Java API POI-HSSF (von Jakarta) genutzt. HSSF verfügt über die Möglichkeit *.xls Dateien zu erzeugen, diese zu modifizieren sowie zu lesen und zu beschreiben. Des Weiteren unterstützt diese Bibliothek Zellfunktionen und –formeln.

Basierend auf dem state-of-the-art Software Entwicklungs- und Design Ansatz „MVC“ (Model-View-Controller) sind die Schnittstellen als sog. Prozessoren entwickelt worden. Um die Vorgaben der MS Excel Kopplung zu gewährleisten, wurden drei

* Corresponding author

Prozessoren entwickelt: (i) *GetXLSValueProcessor*, (ii) *SetXLSValueProcessor* und (iii) *RecalcXLSProcessor*. Jeder dieser Prozessoren verbindet das SDSS mit *.xls-Dateien, spricht ein darin enthaltenes spezifisches Tabellenblatt an und nutzt definierte Zellen dieses Blattes. Mit den Prozessoren ist es möglich komplexe Excel-basierende Modelle von einem benutzerfreundlichen SDSS-Graphical-User-Interface (GUI) auszuführen, indem (i) voreingestellte Eingabeparameter der Excelzellen zur Verfügung gestellt werden, (ii) die Editierung dieser Parameter innerhalb des SDSS-GUIs ermöglicht wird, (iii) die Neuberechnung sowie den Modellauf als solches unabhängig von dem SDSS gewährleistet wird, (iv) die prozessierten Ergebnisse übergeben und (v) visuell dargestellt werden.

Außer der Beschreibung der Entwicklung werden auch vorläufige Ergebnisse des programmierten SDSS IWEGS aufgezeigt.

1. INTRODUCTION

Spatial data, mathematical models and expert knowledge are more and more incorporated in modern decision making processes. Spatial Decision Support Systems (SDSSs) combine such data and knowledge within process-based tools which are developed question-specific according to a given logical decision tree. Turban et al. (2005) point out that a SDSS ensures decision support and definite options for management and spatial planning actions. This corresponds with the tasks of Resource Management Information Systems which also serves decision making (McCloy 2006), and of multicriteria decision analysis (Malczewski 1999).

Detailed information about the requests, knowledge and personal needs of the potential users are essential during the development of such computer based systems. Therefore, the programming of an innovative SDSS needs to depend on the GIS-, RS-, and model-knowledge of the decision maker (Laudien & Bareth 2007).

This study presents the development and implementation of modern SDSSs interfaces to couple or embed scientific models. The programming of these modular designed systems is based on Java to guarantee platform independent software tools. By the additional use of several different Java based libraries (e. g. ArcGIS Engine) the developed systems meet the requirements of being comprehensive GI-, RS- and model based SDSSs.

In addition to the description of the used methodological approach and its implementation, this paper shows preliminary results based on a developed SDSS as a case study. The SDSS *IWEGS (Impact of Water Exploitation on Groundwater and Soil)* focuses on recommendations concerning ground water availability and soil salinisation derived from six oases in Morocco/ Africa. Based on a complex logical decision tree and because of the loose coupling of four existing models, this SDSS is predestined to be presented as an example. Three of the four used models are MS Excel based (*C.E.M. Drâa*, *Cropdem* and *BIL*). The model *C.E.M. Drâa (Consommation d'Eau Ménagère)* estimates the annual domestic water consumption per oasis considering observed demographical trends. *Cropdem* presents a look up table, which is based on the model *CropWat* (FAO). Thus, the annual water demand of the major crops is assessed for each oasis. The annual volume of available groundwater for each oasis is calculated by the model *BIL*, based on the groundwater budget estimation. By using the SDSS coupling approach, the models *C.E.M. Drâa* and *Cropdem* pass their results to *BIL* as negative items in the groundwater balance. Thereby, the crop water demand is set equal to the minimum irrigation amount, which is extracted from groundwater. In addition to the three MS Excel based models, the FORTRAN model *SahysMod* is also connected and implemented into the decision processes of IWEGS. However, as this paper only focuses on the MS Excel interfaces, the *SahysMod* coupling approach is not considered.

IWEGS as well as several other Information Systems and SDSSs are carried out for the interdisciplinary research project IMPETUS (*An integrated approach to the efficient management of scarce water resources in West Africa*). Besides others, one major task of the third project phase (2006-2009, <http://www.impetus.uni-koeln.de>) is the development, implementation and application of GIS-based SDSSs for the water resource management in selected catchments of Benin and Morocco. The programmed systems will be implemented in a Java/XML based framework (Enders et al. 2007).

2. MATERIAL AND METHODS

The following sections show the used materials and methods to follow up the selected modular approach.

2.1 Development environment and data storage

Java, an object oriented programming language, is used to develop the SDSSs and their interfaces. The software which is developed with Java is nearly platform independent (Krüger 2006).

Eclipse SDK which contains the Eclipse platform (Eclipse 3.2) was chosen as the programming environment to develop the SDSSs. Eclipse already comes with tools for Java programming and the environment to develop Eclipse plug-ins.

Besides the established Java beans, specific GIS- and RS-components are used for the programming. These are implemented in ArcGIS Engine (by ESRI) and can easily be integrated into the source code. Furthermore, coding of additional Java libraries completes the developed SDSS. Apart from that, other non-Java libraries, e. g. POI-HSSF components (Java API by Jakarta), are used to couple numerical models (in this context: MS[®] Excel based models).

The (geo-) data (e. g. raster, vector, and alphanumerical data) of the single SDSSs are stored in ArcGIS Engine file-based geodatabases. These databases show a high performance when large amounts of data are involved, and they are almost not inherent in size limit (the limit is that of the file-system itself) (ESRI 2007).

2.2 Description of the used models for the case study

The following three sections describe briefly the used models of the case study. The three models *C.E.M. Drâa*, *Cropdem* and *BIL* are embedded in a MS Excel spreadsheet and are connected to each other by mutual parameterization. *SahysMod* is developed in FORTRAN. A short introduction of *SahysMod* is

also given as this fourth model gets core input parameters of the three xls-based models and therefore is also part of *IWEGS*.

2.2.1 C.E.M. Drâa

C.E.M. Drâa is an empirical model designed to calculate the annual domestic rural and urban water consumption for each of the six Drâa oases. Future trends of domestic water consumption are simulated by considering demographic trends for rural and urban population development (Haut Commissariat au Plan 2005). Users of the model are invited to change the parameters of rural and urban growth rates in order to simulate different possible future trends.

2.2.2 Cropdem

CropDem is a lookup – table containing the water demand of six main crops as well as date palms. Water demand is calculated with the Penman-Monteith equation (Allen et al. 1998) in combination with crop coefficients published especially for the Drâa valley (Ministère des travaux publics 1998). The water demand is calculated per square meter and crop and summed per oasis depending on the spatial distribution of crops.

2.2.3 BIL

BIL is a lumped parameter model based on the groundwater balance approach. It was specifically developed for the Middle Drâa Valley (South Morocco) and realised in a MS Excel spreadsheet. In order to represent the groundwater budget completely, all natural and artificial affluxes and extractions are considered. BIL outputs the average annual groundwater volume for each aquifer within an aquifer cascade. Therefore, the aquifers are dynamically coupled by one dimension groundwater flow which is formalised by the Darcy equation (Darcy 1856).

2.2.4 SahysMod

SahysMod is a numerical model to predict the salinity of soil and groundwater under irrigation agriculture. Thereby different (geo)hydrologic conditions and varying water management options can be represented including the combined use of surface- and groundwater for irrigation. A variable network of polygons allows accounting for spatial variation; crops are defined by their water demand. The model is made up by three main components: an agronomic water balance model calculating the vertical water fluxes in the soil profile, a ground water model calculating the vertical and horizontal fluxes in the aquifer and a salt balance model. The former two interact, while the latter runs parallel to the water balance components (ILRI 2005).

2.3 Development of specific HSSF Processors

The Java API POI-HSSF 3.0.2 (by Jakarta) is used as a non-Java based library to access MS Excel formatted files. HSSF provides a way to read spreadsheets, create, modify, read and write XLS spreadsheets. Besides that, this library supports cell-functions and -formulas.

Based on the state-of-the-art software development design approach MVC (Model-View-Controller), the interfaces are developed as processors. To meet the given requirements of accessing MS Excel, three processors are programmed: (i) *getXLSValueProcessor*, (ii) *setXLSValueProcessor* and (iii) *RecalcXLSProcessor*.

Each of these processors is able to connect a specific file (*.xls), accesses its Excel spreadsheet and uses the defined cells within

that sheet. With these processors, running complex Excel-based models from a SDSS-Graphical User Interfaces (GUI) is enabled.

2.4 Integration into the methodological approach

In order to assess the impact of water exploitation for domestic and agricultural purposes on groundwater availability and soil salinity, the coupling of different disciplinary methods and existing models was realised within the development of *IWEGS*. This interdisciplinary approach leads to parameter optimisation of the input for the models *BIL* and *SahysMod*. By coupling the models, *C.E.M. Drâa* and *Cropdem* pass their results to *BIL* to optimise the negative items in the groundwater balance, such as domestic extraction and irrigation (see Fig.1) withdrawal. In respect to scenario projections, the coupling of the models needed to be dynamically. Consequently the interfaces between the MS Excel-based models and the SDSS are essential and need to be developed modular.

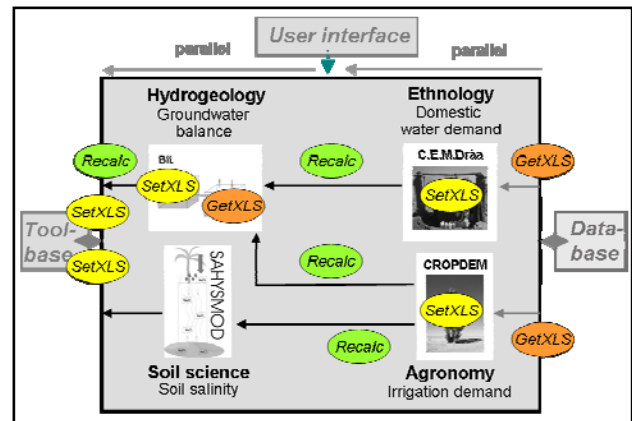


Figure 1. Flowchart of the IWEGS methodological approach (modified from Klose et al. 2008).

In addition to this modular model coupling approach, Fig.1 shows also where the interfaces in terms of pipeline processors are implemented into the overall SDSS development approach. Based on the existing models which are stored in the IWEGS geo-database, the *getXLSValueProcessor* provides the model input parameters of the Excel cells. With a developed SDSS-GUI the user is able

- (i) to edit these parameters and rewrite them back into the sheet (*setXLSValueProcessor*),
- (ii) recalculate and run the model irrespective of the SDSS (*RecalcXLSProcessor*),
- (iii) get the processed outputs from the MS Excel sheet and
- (iv) visualize them at a following SDSS-GUI (*getXLSValueProcessor*).

By having the option to modify pre-defined parameters (or boundary conditions), the IWEGS users are able to create scenarios on their own. Thus the GUIs enable a convenient handling of the complex system and guarantee error-free model-runs.

3. RESULTS

The result chapter is divided into two sections: The interface implementation describes how the interfaces are implemented

into pipeline processors. The case study shows the application and usage of the developed processors and gives a brief introduction into IWEGS.

3.1 Interface implementation

As already mentioned above, the interface development is realised by using the Java object oriented programming language. Thereby, developed abstract Java classes form the basement of the SDSS interfaces. These classes are implemented into an overall framework in terms of pipeline processors (Enders et al. 2007).

The advantage of these processors within pipelining is their inside organisation into stages or steps which can semi-independently work on separate jobs. Each stage is organised and linked into a chain so each stage's output is inputted to another stage until the job is done. Software pipelines, consisting of multiple processes arranged so that the output stream of one process is automatically and promptly fed as the input stream of the next one (Wikipedia 2008).

By following this approach, source code based on abstract encapsulated object oriented programming can be generated to meet the given requirements of coupling SDSS modules, in this context existing MS Excel models.

Source code excerpt showing the *getXLSValueProcessor* as an example:

```
public class GetXLSValueProcessor extends Processor {
    protected boolean process() throws PipeLineException
    {
        FileInputStream tFis = null;
        String tOutput = "";
        String tFileIn = getStringInParameter("file");
        try
        {
            tFis = new FileInputStream(tFileIn);
            HSSFWorkbook tWorkbook = new HSSFWorkbook(tFis);
            tFis.close();
            HSSFSheet tSheet =
                tWorkbook.getSheet(StringHelper.replaceParamTokens(
                    getInParameterMap(),
                    getStringInParameter("sheet"),
                    getStringInParameter("sheet")));
            for (Object tConfigElement :
                getConfiguration().getChildren())
            {
                Element tElement = (Element)tConfigElement;
                String tValue =
                    StringHelper.replaceParamTokens(getInParameterMap(),
                        tElement.getValue(), tElement.getValue());
                CellReference tCellRef = new CellReference(tValue);
                HSSFRow tRow = tSheet.getRow(tCellRef.getRow());
                HSSFCell tCell = tRow.getCell(tCellRef.getCol());
                if (tCell != null)
                {
                    if (tCell.getCellType() == HSSFCell.CELL_TYPE_NUMERIC)
                    {
                        tOutput =
                            Double.toString(tCell.getNumericCellValue());
                    }
                    if (tCell.getCellType() == HSSFCell.CELL_TYPE_FORMULA)
                    {
                        HSSFFormulaEvaluator tEvaluator = new
                            HSSFFormulaEvaluator(tSheet, tWorkbook);
                        tEvaluator.setCurrentRow(tRow);
                        HSSFFormulaEvaluator.CellValue tCellValue =
                            tEvaluator.evaluate(tCell);
                        switch (tCellValue.getCellType())
                        {
                            case HSSFCell.CELL_TYPE_BOOLEAN:
                                tOutput =
                                    Boolean.toString(tCellValue.getBooleanValue());
                                break;
                            case HSSFCell.CELL_TYPE_NUMERIC:
                                tOutput =
                                    Double.toString(tCellValue.getNumberValue());
                                break;
                            case HSSFCell.CELL_TYPE_STRING:
                                tOutput =
                                    tCellValue.getRichTextStringValue().getString();
                                break;
                            case HSSFCell.CELL_TYPE_BLANK:
                                break;
                            case HSSFCell.CELL_TYPE_ERROR:
                                break;
                        }
                    }
                }
            }
        }
        catch (Exception e)
        {
            return false;
        }
        return true;
    }
}
```

```
        }
        break;
    }
}
if (tCell.getCellType() == HSSFCell.CELL_TYPE_BOOLEAN)
    tOutput =
        Boolean.toString(tCell.getBooleanCellValue());
if (tCell.getCellType() == HSSFCell.CELL_TYPE_STRING)
    tOutput = tCell.getRichStringCellValue().getString();
}
setOutParameter(tElement.getAttributeValue("key"),
    tOutput);
}
return true;
}
```

With concern to the Excel-coupling approach, the interfaces between the Excel files and the SDSS are implemented in the processor source codes by

- (i) importing the POI HSSF usermodel in a first step,
- (ii) instantiating the specific classes and finally,
- (iii) by declaring the variables and using the POI HSSF specific properties and methods (see source code excerpt as an example).

3.2 The interactive SDSS IWEGS

The regional scaled SDSS IWEGS allows the estimate groundwater availability and soil salinity in an annual resolution based on data of six Drâa oases.

Based on the predefined structure of an IMPETUS SDSS, after accessing IWEGS from the IMPETUS Client, the start screen and the flowchart provide first information about the system and show its structure (Fig2. a & b). Once the climatic scenario is chosen, the first model (*C.E.M. Drâa*) is initialised using the *getXLSValueProcessor* which fills the default model input parameters into the given GUI table (Fig.2 c). By using the IMPETUS Framework form components, the user is able to edit these default values directly at the GUI. In a next pipeline step, the *setXLSValueProcessor* writes these values back into the Excel spreadsheet and therefore provides *C.E.M. Drâa* user defined model input parameters. Finally, the *RecalcXLSProcessor* runs the model and again provides the output cell values to the *setXLSValueProcessor* for setting model connecting parameters to other models. In addition to that, the output cell values feed the *getXLSValueProcessor* for visualizing the output at the GUI (e.g. Fig.2 d showing the GUI of *Cropdem*).

By using this modular software development approach, *C.E.M. Drâa* calculates the household annual water demand per oases, *Cropdem* estimates the average annual water demand of the crops per oases and *BIL* computes the average annual groundwater deposit per oases using the results of *C.E.M. Drâa* and *Cropdem* as negative balance items. The water demand of the crops is being estimated by the volume of the water withdrawal used for irrigation (corrected by the irrigation volume implemented by lâchers = Episodically released outlets from the upstream located reservoir Mansour Eddahbi). Specific result parameters of *Cropdem* are implemented into the model *SahysMod*, which determines the averaged annual soil salinity (as electrical conductivity) per oases.

To generate the spatial decision support in terms of an interactive GIS, predefined thresholds are used to approximate the groundwater stock and soil salinity results and generate classified thematic maps. Consequently, the GIS results (soil salinity and groundwater volume) are displayed (Fig.2 e), including the possibility of displaying time series (histograms), by using ArcGIS Engine. In addition to the GIS output, the table output forms (Fig.2 f) provides the annual household and

agricultural water demands, the groundwater salinity, - efficiency and -sufficiency.

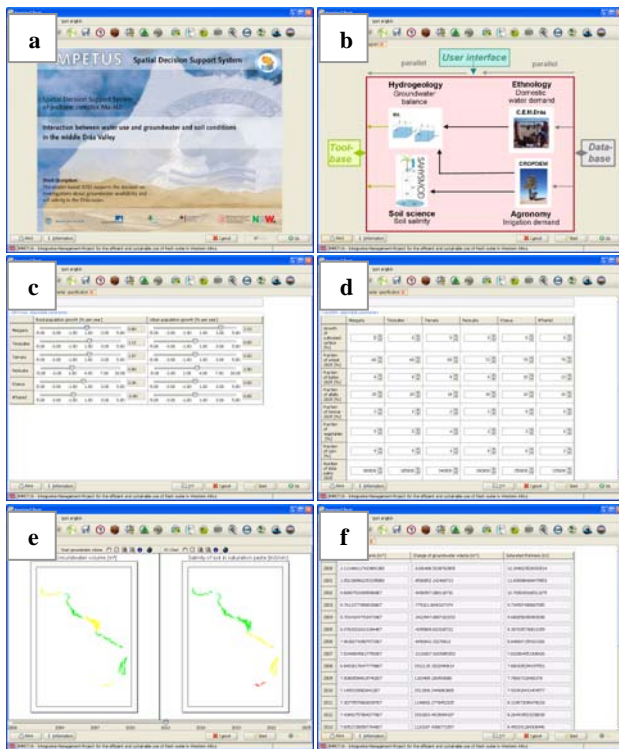


Figure 2. IWEGS screenshots of preliminary SDSS development results: (a) start screen, (b) flowchart, (c) *C.E.M Drâa* GUI, (d) *Cropdem* GUI, (e) GIS output, (f) table output.

To summarize, the SDSS IWEGS with its implemented POI HSSF connecting and accessing processors can be used as a comprehensive spatial decision support tool for specific stakeholders in Morocco.

4. DISCUSSION & CONCLUSIONS

This paper shows the implementation of non-Java based interfaces to embed existing models in computer based Spatial Decision Support Systems. Besides the description of the used materials and of the methodological approach, a case study is presented which points out the integration of MS[®] Excel-models in a developed SDSS as an example.

The major advantage of developing individual SDSS-interfaces is the connection of its single modules (e. g. spatial geo-data or computer based models) without re-programming them. Developed object-oriented processors help to access and run these SDSS modules in the background of the system. Consequently, the SDSS-parts are loosely coupled by the programmed interfaces. This matter of fact points out the overall benefit of the followed approach: By developing comprehensive SDSSs and their interfaces in a modular way, potential decision makers (users) are still able to maintain their implemented modules (e.g. source code of an already developed model or MS Excel spreadsheets containing cell formulas or functions) without the need to access the SDSS frame source code.

The case study "IWEGS" is predestined to demonstrate the implementation of the programmed processors into the used overall SDSS approach. Three MS[®] Excel-based models were available from the beginning of the development. Hence the coupling of these existing tools allowed a quick development of the system. With the implementation of specific XLS interfaces in terms of pipeline processors, these three modules did not need to be embedded into the Java source code of the SDSS. This matter of fact demonstrates the high degree of transparency for scientists and potential users who are not familiar with programming, SDSS development and/ or databases. The single modules are accessible, editable and therefore customizable at any time. As the models are stored irrespectively from the SDSS, even overwriting is possible without destroying the system. Hence, the followed methodological approach allows easily adjusting the models without touching the SDSS development.

REFERENCES

- Allen, R.G., Pereira, L.S., Raes, D. and Smith, M., 1998. Crop evapotranspiration - Guidelines for computing crop water requirements. UN-FAO Irrigation and drainage paper 56, Rome, Italy.
- Darcy, H., 1856. Les fontaines publiques de la ville de Dijon. Victor Dalmont. Paris.
- Enders, A., Laudien, R. & Hoffmann, R., 2007. Spatial Decision Support Systems.- A. H. Fink, and B. Reichert (Eds), Integratives Management-Projekt fuer einen Effizienten und Tragfaehigen Umgang mit Suedwasser in Westafrika: Fallstudien fuer ausgewaehlte Flusseinzugsgebiete in unterschiedlichen Klimazonen, Siebter Zwischenbericht, pp. 7-21, http://www.impetus.uni-koeln.de/content/download/ZB2006/IMPETUS_Zwischenbericht_2006.pdf.
- ESRI: What's New in ArcGIS 9.2., 2007. http://webhelp.esri.com/arcgisdesktop/9.2/pdf/Whats_New_In_ArcGIS_92.pdf.
- HAUT COMMISSARIAT AU PLAN, 2005. Recensement Général de la Population et de l'Habitat 2004. Rabat.
- ILRI (International Institute for Land Reclamation and Improvement), 2005. SahysMod Spatial Agro-Hydro-Salinity Model. Version1.7, March 2005.
- Klose, S., Rademacher, C., Klose, A., & Roth, A., 2008. Wechselwirkungen zwischen Wassernutzungsstrategien und den Grundwasser- und Bodenverhältnissen im mittleren Drâa-Tal.- A. H. Fink, and B. Reichert (Eds), Integratives Management-Projekt fuer einen Effizienten und Tragfaehigen Umgang mit Suedwasser in Westafrika: Fallstudien fuer ausgewaehlte Flusseinzugsgebiete in unterschiedlichen Klimazonen, Achter Zwischenbericht, pp. 233.
- Krüger, G., 2006. Handbuch der Java-Programmierung.- 4. Auflage, Addison-Wesley, ISBN 3-8273-2201-4.
- Laudien, R. & Bareth, G., 2007. Entwicklung und Programmierung von räumlichen Entscheidungsunterstützungssystemen mit ArcGIS Engine und Java.- GIS-Geoinformationssysteme, 16-21.

Malczewski, J., 1999. *GIS and multicriteria decision analysis*. Wiley, New York.

McCloy, K. R., 2006. Resource management information systems: remote sensing. *GIS and modeling*, Taylor & Francis: Boca Raton, 575 p.

Ministère des travaux publics, 1998. Etude du plan directeur de l'aménagement des eaux des bassins sud-atlasiques, Mission 3: Etude des schemas d'aménagement, Volume 4: Unités du Draa.

Turban, E., Aronson, J. E., Linag, T. P., 2005. *Decision support systems and intelligent systems*. Prentice Hall, New York.

ACKNOWLEDGEMENTS

This study is part of the interdisciplinary scientific project and is supported by the Federal German Ministry of Education and Research (BMBF) under grant No. 01 LW 06001A and 01 LW 06001B as well as by the Ministry of Innovation, Science, Research and Technology of the federal state of Northrhine-Westfalia under grant No. 313-21200200.