IVIP – A Scientific Workflow System to Support Experts in Spatial Planning of Crop Production

Christopher J. Tuot, Michael Sintek, Andreas R. Dengel

German Research Center for Artificial Intelligence DFKI GmbH,
Trippstadter Straße 122, D-67663 Kaiserslautern, Germany
{christopher.tuot,michael.sintek,andreas.dengel}@dfki.de
http://www.dfki.de/km

Abstract. Decision making for crop production planning is essentially driven by location-based or more precisely by space-oriented information. Therefore, farmers and regional experts in the field mostly rely on new spatial-data-oriented decision making tools. IVIP[1] is a prototype for a Web-based Spatial Decision Support System (WSDSS) demonstrating the benefits of location-based decision making using digitalized geographic information about ground allocation and soil quality. We present how the library of potential models for the IVIP WSDSS has been realized by extending the Scientific Workflow Management System Kepler that assists the collaboration of agricultural experts and computer scientists during model development. We first describe the requirements of our WSDSS, and then give a short introduction to the Kepler platform and explain in detail which extensions have been realized: cascading client-server architecture, spatial operations support, and WSDL interface. Finally, we illustrate how the biomass yield model has been modeled in our system.

Key words: Scientific workflow models, Scientific data integration, Spatial Decision Support System (SDSS), Kepler, Workflow Management System (WMS), Web Service, WSDL, GIS, Forecast, Agriculture

1 Motivation

Decision making for crop production planning is essentially driven by location-based or more precisely by space-oriented information. The required Global Positioning System (GPS) and sensor technology are becoming standard for agricultural machinery and most of the current Farm Management Information Systems (FMIS) already provide interfaces to import the acquired sensory information. With the standards for geospatial content and services developed by the Open Geospatial Consortium (OGC) the interoperability for geospatial technology has been highly facilitated. Nevertheless, the amount and the complexity of the spatial information becoming available is dramatically increasing and so is

[1] The IVIP project is funded by the Ministry for Economy, Transport, Agriculture and Viniculture (MWVLW) and is part of the EU program: “Ziel 2 Programm RLP.”

the demand for tools capable of handling such information. Therefore, farmers and regional experts in the field mostly rely on new spatial-data-oriented decision making tools. IVIP is a prototype (see Fig. 1 (a)) for a Web-based Spatial Decision Support System (WSDSS) demonstrating the benefits of location-based decision making using digitalized geographic information about ground allocation and soil quality.

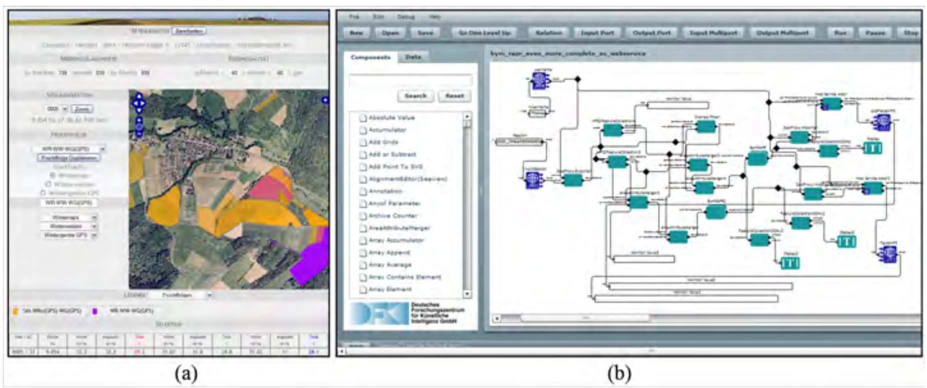

Fig. 1. Screenshots: (a) IVIP Spatial Decision Support System for Crop Production; (b) the bym Model in Kepler Flex GUI

2 WSDSS Requirements

Sprague [9] defines a Spatial Decision Support System (SDSS) as a standard decision support system plus some semi-structured spatial problem. A SDSS usually consists of three parts. First there must be a *Data-Base Management System* (DBMS) capable of handling geographical data like, e.g., a Geographical Information System (GIS). Then there must be a *library of potential models* that can be used to forecast the possible outcomes of decisions. Finally, an *interface* should aid the users to interact with the computer and should assist them in analyzing the outcomes. This defines the architecture of our WSDSS. In this paper, we focus on the library of models.

In the GIS world, geospatial information is based on *features*. A *feature* is an entity with a geographic location and some additional meta-information stored as key-value properties. The geographic location of a *feature* is described by a geometry based, e.g., on points, arcs, or polygons. Therefore, our WSDSS must be able to deal efficiently with **feature-data oriented models**.

Current SDSS solutions like, e.g., CommunityViz, an extension for ArcGIS, are definitively meant for GIS specialists. In IVIP, domain experts have **none or relatively limited programming skills and experience with GIS**. Nevertheless, those experts must be able to work with the WSDSS and in particular to develop new models.

Although networking technologies are constantly being improved, data-transfer speed is becoming a bottle-neck since the resolution of spatial information increases considerably at the same time. Therefore, our WSDSS must allow to **realize data-intensive operations** at the places where the data is actually located (in the network). Moreover, it is often the case that a model relies on other already existing models in the WSDSS which again can rely on other models. Consequently, a **cascading architecture for GIS operations** is needed which would not only solve the problem of efficient data-transfer but which would also define a good basis for further issues data providers might be concerned with like data-control access or billing aspects.

3 The KEPLER Platform

KEPLER[2] [1] is an open-source **scientific workflow system** that allows scientists to design scientific workflows and execute them efficiently. A *scientific workflow* is a **high level description** of a data-oriented process which can be used to solve a scientific problem. The dataflow consists of data being processed though parameterizable modules called *actors*. KEPLER is based on the PTOLEMY II [3] system for heterogeneous, concurrent modeling and design developed at UC Berkeley. Designing a scientific workflow in KEPLER is done following a visual programming principle and a simple workflow using existing actors can be realized with only a few mouse clicks. Depending on the size and the complexity (in terms of internal logic) of the workflows, basic programming skills can be helpful but are not mandatory. Advanced programming skills are only required for implementing new actors.

4 New Extensions for KEPLER

Spatial operations support: Possibilities to handle geospatial data in KEPLER have already been discussed in [5, 8]. However, most of the propositions rely on externalizing the spatial operations, for example in Web Services or dedicated applications like the Geographic Resources Analysis Support System[4] (GRASS). Although delegating those complex operations to specialized applications makes sense, this has for the user enormous drawbacks in terms of workflow transparency. Therefore, we decided to handle geospatial operations directly in KEPLER. This tends to go back to the original idea of PTOLEMY however, high level workflows can still be designed on the feature level.

GeoTools[5] is an open-source Java library which provides OGC standards compliant methods for manipulating geospatial data. The **GeoTools** provide an efficient internal representation for features and all necessary basic spatial operations. We have implemented additional KEPLER actors using the GeoTools API.

[2] `http://kepler-project.org/`
[3] `http://ptolemy.berkeley.edu/ptolemyII/`
[4] `http://grass.itc.it/`
[5] `http://geotools.codehaus.org/`

Spatial information can now directly be imported in KEPLER using either the **WFS** actor, implementing the OGC Web Feature Service [10] protocol (WFS), or by using the GMLReader actor which imports data in the OGC Geography Markup Language [4] format (**GML**). We have also implemented several actors handling all common operations at the **feature level** (add, remove, update, merge). Finally, storing the resulting information can be done either using our **WFS-T** actor implementing the WFS transactional (WFS-T) protocol or by using the GMLWriter actor which exports GML data.

Client-Server architecture: Although KEPLER is currently distributed as a standalone application, its design actually made it possible to separate the GUI from the workflow engine itself. Our modified version of KEPLER can be deployed on a **standard application server** and be remotely controlled through the HTTP protocol. Besides defining the first step towards building a **cascaded architecture** to process spatial information, the client-server architecture also plays a major role in knowledge sharing.

One key feature of KEPLER is to allow scientists to exchange their workflows. Scientific workflows in KEPLER are stored using PTOLEMY's own Modeling Markup Language (MOML). MOML files only contain high level information about the workflow and the actors used are only referenced. Hence the definitions of the KEPLER actors themselves are not contained in the workflow but are literally part of the workflow engine. Therefore, if a workflow is using non-standard actors, it cannot be directly executed on another installation of KEPLER. Unfortunately, KEPLER still misses a proper management system to manage actors for tasks like adding, deleting, or updating actors. Those tasks remain too difficult for users without solid Java programming skills and really affect the possibilities of exchanging workflows. This is actually one of the issues the new KEPLER CORE initiative[6] will try to solve in the next years. Our idea is to further take advantage of our client-server architecture.

An application server is per definition capable of handling multiple-user requests. Let A and B be two scientists both running different instances of KEPLER (client-server architecture). A sharing a workflow with B could be done on the KEPLER installation of A. This insures that all necessary actors to run the workflow provided by A are present in the workflow engine. So the real question now is whether or not the standalone KEPLER GUI can **remotely execute a workflow** on a distant KEPLER. Unfortunately, although we managed to separate the GUI and the workflow engine, the GUI could in this state no more be used without spending a considerable amount of time to reinterface it with the KEPLER server. Alternatively, we went for a simplified version of the GUI supporting basic functionalities to create and execute workflows. We decided to develop a web-based KEPLER client using the new Adobe Flex 3[7] technology. Adobe Flex 3 is a cross platform open-source framework for creating rich applications. Flex applications are platform independent and are a good compromise

[6] http://www.kepler-project.org/Wiki.jsp?page=KeplerCORE
[7] http://labs.adobe.com/technologies/flex/

between standard rich versus thin clients. Behind the pure software engineering performance of implementing this client, downloading and installing the current 100 Megabytes KEPLER package is now no longer required to start working with scientific workflows. A single client-server KEPLER version can provide several scientists with the most core functionalities of KEPLER and allow them to efficiently share their workflows.

WSDL interface: Our client-server architecture of KEPLER was just the first step towards a cascaded architecture to process spatial information. Indeed the HTTP interface to remotely control KEPLER workflows is not suitable to allow two KEPLER instances to communicate with each other. The Web Service Description Language [3] (**WSDL**) is an XML-based language providing a model for describing Web services. Whereas Web Services can already be used in KEPLER workflows[6], our idea was to publish a KEPLER workflow itself as a Web service allowing any application supporting WSDL to use this workflow. Therefore we introduce two new actors: *WebServiceInput* and *webServiceOutput*. Those actors are only necessary for prototyping the input/output parameters of the workflow which unfortunately cannot yet be done automatically using the rest of the information contained in the workflow. A specific file folder is used as Web service pool and any workflow (properly designed with the WebServiceInput/Output actors) is directly available under the url:
http://serverIp:port/KFlexServer/wsdl/workflow=workflowName.

5 Example: The Biomass Yield Model in KEPLER

The Biomass Yield Model (**bym**) was developed at the University of applied sciences Eberswalde [7, 2]. The original model was developed in Visual Basic using the software ArcGIS[8] from ESRI. Thanks to our custom GIS actors, we managed to design a KEPLER workflow computing the bym. The top level workflow in Fig. 1 (b) benefits from the composite actors architecture since the bym KEPLER workflow actually consists of eight encapsulated sub-workflows representing together more than 130 actors. Thus, the whole bym workflow has been broken into atomic operations which facilitates the work of the experts aiming at improving or optimizing this workflow.

The bym workflow can compute the yield for 16 different crops depending on the soil quality and the precipitation levels. Different kinds of scenarios are taken into consideration, e.g., conventional vs. ecological farming and three different levels of precipitation (low, normal, high). On the whole, six different scenarios are computed at once. The input of the workflow is a map containing the fields of a farmer, usually a GML file. The soil quality and the precipitation levels are on two remote GIS which are accessed using the WFS protocol. The results of the workflow are stored in a local GIS, i.e., in the same network. The storage is done using the WFS-T protocol. The data can then be visualized either directly

[8] `http://www.esri.com/software/arcgis/`

in our GIS or with any mapping service application, e.g., OpenLayers[9] or Google Maps.

The results delivered by this workflow represent the key information for our IVIP prototype, a Decision Support System to optimize crop production. All necessary parameters for the workflow can be entered using our GUI using the Web-Service interface of the bym KEPLER workflow to compute the results.

6 Conclusions and Future Work

In this paper we describe how we extended the Scientific Workflow Management System KEPLER to define the core of IVIP, a Web-based Spatial Decision Support System for crop production planning. The resulting cascading system for GIS operations enables non-GIS experts to design workflows for their spatial models. Next steps are to develop more GIS functionalities for KEPLER allowing to design more complex spatial models and to evaluate the benefits of our WSDSS.

References

1. I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423–424, 2004.
2. S. Brozio, H.-P. Piorr, D. Müller, and F. Torkler. Potenziell nutzbare Biomasse – Modellierung mit GIS, 2006.
3. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL). *W3C Web Site*, 2001.
4. S. Cox, P. Daisey, R. Lake, C. Portele, and A. Whiteside. OpenGIS Geography Markup Language (GML) Implementation Specification. *OpenGIS project document reference number OGC*, 2003.
5. E. Jaeger, I. Altintas, J. Zhang, B. Ludäscher, D. Pennington, and W. Michener. A Scientific Workflow Approach to Distributed Geospatial Data Processing using Web Services. *17th Intl. Conference on Scientific and Statistical Database Management*, 2005.
6. S. Perera and D. Gannon. Enabling Web Service Extensions for Scientific Workflows, 2006.
7. H.-P. Pior, K. C. Kersebaum, and A. Koch. Die Bedeutung von Extensivierung und ökologischem Landbau für Strukturwandel, Umweltentlastung und Ressourcenschonung in der Agrarlandschaft. *Eberswalder Wissenschaftliche Schriften*, 3:99–114, 1999.
8. C. Rueda, M. Gertz, B. Ludäscher, and B. Hamann. An Extensible Infrastructure for Processing Distributed Geospatial Data Streams. *Proceedings of the 18th International Conference on Scientific and Statistical Database Management*, pages 285–290, 2006.
9. R. Sprague Jr and E. Carlson. *Building Effective Decision Support Systems*. Prentice Hall Professional Technical Reference, 1982.
10. P. Vretanos. Web Feature Service Implementation Specification. *OpenGIS project document: OGC*, pages 02–058, 2002.

[9] `http://www.openlayers.org/`