

A web service based approach for integrating statistics tools into an information system for experiment data

Dennis Heimann, Jens Nieschulze
Max-Planck-Institute for Biogeochemistry
Jena, Germany
{dheimann, jniesch}@bgc-jena.mpg.de

Birgitta König-Ries
Friedrich-Schiller-Universität Jena
Jena, Germany
birgitta.koenig-ries@uni-jena.de

Abstract: Data management in the life sciences has evolved from simple storage of data to complex information systems providing additional functionalities like analysis and visualization capabilities, demanding the integration of statistical tools. In many cases the used statistical tools are hard-coded within the system. That leads to expensive integration, substitution, or extension of tools because all changes have to be done in program code. Other systems use generic solutions for tool integration but adapting them to another system is mostly rather extensive work.

This paper shows a way to provide statistical functionality over a statistics web service, which can be easily integrated in any information system and set up using XML configuration files. The statistical functionality is extendable simply by adding the description of a new application to a configuration file. The service architecture as well as the data exchange process between client and service and the adding of analysis applications to the underlying service provider are described. Furthermore a practical example demonstrates the functionality of the service.

1 Introduction

Data management in the life sciences has been a very active area of research for a number of years ([JO04, LN05, RUCM07, TSJS07]). Over time, it has become obvious, that it does not suffice for tools to offer "just" integrated (centralized or virtual) data storage capabilities. Rather, users demand direct access to a diverse set of tools to access and analyze data. Of particular importance to many users are statistical analysis tools. What is needed are platforms combining integrated data storage with seamless access to data analysis tools.

The Biodiversity Exploratories¹, a large-scale and long-term biodiversity research project in Germany, are an adequate example in need of such a system. The project aims to examine the relationship between land-use intensity, biodiversity change, and ecosystem functioning for selected taxa. Within this umbrella project, a large number (currently 40, more will be added over time) of individual, independent projects from a diverse set of communities investigate different aspects of the overall problem, comprising research on botany, vertebrates, invertebrates, soil sciences, and biogeochemical processes. One of the expectations towards the umbrella project is to make data available beyond individual

¹<http://www.biodiversity-exploratories.de>

projects to allow for analysis of data across disciplines and over time, e.g., to be able to explore changes in biodiversity over a decade and relate this to changes in the soil brought on by the use of certain fertilizers.

As a technical basis for this task, we are developing the web-based Biodiversity Exploratories Information System²(BExIS) which offers central storage and management of all project data.

One of the main non-technical challenges faced by any data management system is acceptance by the user community. It is generally acknowledged that such systems should offer added value, so users have a direct benefit of their usage. We believe that seamless access to analysis tools and the ability to plug in new tools as needed is one way to provide such added value. In the long run, seamless integration of statistical methods will also enable common analyses across projects directly within the system. Statistical methods in need range from simple summary of data sets to complex analysis comprising a chain of models and include also graphical analysis. Over the last few years, a number of attempts to solve at least similar requirements have been proposed. These use different approaches to integrate their tools. One approach is to declare the input and output directory of a tool, thus enabling the host application to access the raw files [TSJS07]. Another approach is to hard-code the access to a tool within the host application code [RSL06] [RS04]. This leads to an expensive integration, substitution, or extension of tools, because all changes have to be done in the program code of the host application. Other solutions use a more generic approach to integrate external tools by using configuration files for providing definitions for tools and data types, and physical descriptions of resource locations [RUCM07]. While this approach is very promising, the integration of such a generic workflow environment into a project data management is difficult. Our proposed solution is a more lightweight yet also generic solution with focus on easy integration into existing systems. We have developed a web service for accessing diverse statistical analysis methods. Our approach is to combine all statistical methods within one web service. It provides only three operations to list all available statistical methods, to describe a method more specifically, and to invoke a method. By abstraction from the underlying applications, the web service will be easy to integrate in basically any information system.

In this paper, we are going to discuss our approach of seamless integration of such tools into a data management platform. We will use the integration of the R statistics package into BExIS as our running example.

2 Overview of the Solution

2.1 Requirements

In order to achieve seamless integration of external tools, in our running example statistical analysis tools, into BExIS the whole system needs to meet a number of requirements, most importantly abstraction and scalability.

²<http://www.exploratories.bgc-jena.mpg.de>

The latter can be realized by making addition and substitution of methods possible with as little effort as possible. Abstracting from integrated methods is really important to allow a transparent view to all tools by the users. Today, the most common approach – and the one chosen by us, too – to fulfill these requirements is via a service-oriented architecture (SOA). SOA allows for the seamless, platform independent integration of different tools. It makes it possible to change dynamically the set of offered tools and enables integration of tools without the need of their modification. All that is needed is the realization and description of an appropriate interface. Such an interface can be added to a tool without the need to alter anything in the tool itself - even without "knowledge" of the tool.

On close inspection of tools boxes such as R, it becomes obvious that they offer a large number of different functions that a user may want to use. The naive approach would be to offer each of these functions as a separate web service (cf. [LMF⁺06]) or at least as a separate operation within a common web service. It should be obvious, that such an approach would result in a lot of description and implementation effort. Thus, a more lightweight approach is needed to make the development of a flexible solution feasible. We opted for the Open GIS Web Processing Service (WPS). The WPS specifies the interface of a general purpose web service that can be used to encode the offering of any desired GIS functionality [Sch08]. Developed for the geographic data realm we adapted its principal functionality to cover our needs.

2.2 Architecture and Process

In order to keep our statistics web service as simple as possible, we did not want a full-fledged implementation of a WPS but adopted only those parts of its concept that we needed. In general, to abstract from the underlying applications the statistics web service provides only three operations analogous to the WPS:

- *getMethods* returns a list of all statistical methods provided by the service.
- *getMethod* returns a description of a specific method including its inputs and outputs
- *runMethod* runs a method and returns its outputs.

By abstracting from the underlying applications the statistics web service is easy to extend, for Java applications only the configuration files have to be changed.

As shown in Figure 1, the *StatisticsService* contains a central *Controller* handling all service requests and routing them over an *ApplicationConnector* to the corresponding underlying applications³ providing the statistical functionality. The routing is based on two configuration files, *methods.xml* and *serverConfig.xml*. All available methods provided by the web service as well as their access information are described within the *methods.xml* file. For description, the Web Service Description Language (WSDL) is used. The underlying applications are not necessarily web services, but WSDL offers all needed constructs to describe the provided methods [W3C04], including their interface and data type descriptions. In addition to *methods.xml* the *serverConfig.xml* file contains some general information about the server, for example the input and output paths of the server applica-

³Currently the applications are limited to Java but others can be easily integrated by wrapping them in a Java application.

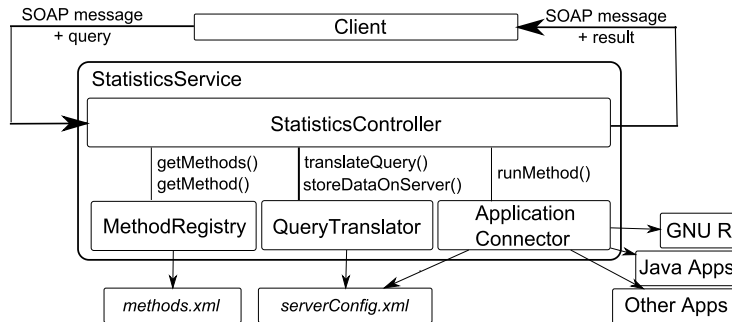


Figure 1: Statistical Web Service Architecture

tions.

A typical query process is as follows: A client sends a request to the web service using the SOAP message protocol [W3C03]. Within the SOAP message body the query (a simple message schema based on xml) is defined. On the server-side the *StatisticsController* class receives the query and checks its type using the *QueryTranslator* class. Depending on the query type either the *MethodRegistry* class or the *ApplicationConnector* class are used to answer the request or rather to invoke an application. The result of a query is send back to the client by the *StatisticsController* class also via a SOAP message.

3 Example

As explained in Section 1 BExIS wants to provide statistical tools to the user. The integration of methods based upon the statistics package GNU R⁴ is provided as an example in the this section. We have chosen R because it is widely used by our user community. Its integration thus offers a considerable benefit to our users.

3.1 Method Integration

A Java application was developed which uses Rserve⁵ and its Java client to implement the different methods. Rserve is a TCP/IP server which allows other programs to use facilities of R from various languages without the need to initialize R [Urb03].

The application that is integrated in the statistical web service is based on a simple design. Currently, there is only one class which implements all methods and connects to Rserve, but there is no restriction to modularize or extend it. Each method implementation is based on an R script that is executed using the Rserve connection.

⁴<http://www.r-project.org/>

⁵<http://www.rforge.net/Rserve/>

For the integration of this method into the statistics web service only two steps are required: Firstly, a WSDL description of the desired function has to be created and put into the *methods.xml* file. Secondly, the corresponding class files have to be put into the statistical web service classes folder. After a servlet container restart, the method will be accessible.

3.2 Statistics Web Service Used by BExIS

The usage of the statistics web service and its provided methods is illustrated by a scatterplot analysis within BExIS. The web service is accessible through BExIS by a separate statistics application.

To analyze a data set using a method from the statistics web service the procedure is as

Table 1: Example data set: Measurement of dbh.

observation	species	dbh	treeage
384301	beech	31.31	75
384302	beech	33.88	140
384303	beech	28.93	110
384304	beech	20.38	84
...			

Table 2: Description of the ScatterPlot method.

Name	Type	Descr	Direction
inData	string	...	IN
variable1	string	...	IN
variable2	string	...	IN
return	xs:binary	...	OUT

follows: **Firstly**, the user has to select a data set stored in the system. An example of this step is shown in Table 1. It shows an observation of trees including the measurement of their diameter at breast height (dbh) and their age in years.

Secondly, the user has to choose a statistical method she wants to apply to the data set. The list of methods is obtained by the use of the *getMethods* operation of the statistics web service. For the sample data set the linearity between age and girth of trees is a common question, suitably addressed by the *ScatterPlot* method. After the user has chosen the method, she uses the *getMethod* operation from the web service to obtain detailed information about this function, in particular about the required inputs and expected outputs. The description of the ScatterPlot example is shown in Table 2.

Thirdly, the user has to assign a value to each IN parameter in the description over a web form. Depending on the parameter description, the value can be a text or a number, or a name of a column of the selected data set. For example, the *inData* parameter is assigned an arbitrary text. This text specifies the name of the data file to be stored on the server by the *QueryTranslator* class. The assignments for *variable1* and *variable2* are column names of the sample data set, namely *treeage* and *dbh*. They specify the values of the columns of the data set to be used by the ScatterPlot method.

The data access specification has to be determined by the statistics web service client, that is BExIS. BExIS has to specify the format (inQuery, URI or JDBC), the position of the Parameter containing the value for the data description, the column delimiter, and the decimal sign. Additionally BExIS has to prepare the data depending on the transfer format.

Fourthly, the *runMethod* operation can be invoked and the result will be displayed on the page. Depending on the return type of a statistical method the displayed result can vary

from text to image data. The ScatterPlot example returns an image as a byte array, so BExIS has to transform the byte array to an image and displays it on the page.

Figure 2 shows the result for our example; evidently, the relationship between age and

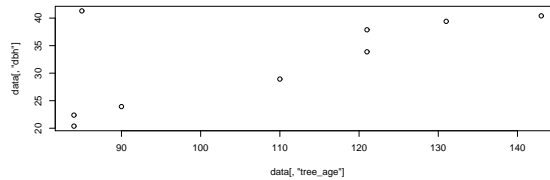


Figure 2: Result of ScatterPlot method.

girth of trees is approximately linear, bar the outlier.

4 Conclusion

In this paper, we described a web service-based approach for integrating statistical analysis methods in an information system for experiment data.

With respect to the need for the integration of external tools, we explained the design of our statistics web service. This service allows the usage of any methods provided by the R statistics tool set from within our information system. We have described the architecture as well as the encapsulation of statistical methods by configuration files. The use of configuration files enables the simple and low effort extension or substitution of methods without the need to touch the underlying statistics program or to code anything within our information system. To abstract from the underlying applications providing the statistical methods we have followed the example of OGC's WPS and introduced a simple query message providing three operations to access information about offered methods, details about the individual methods' interfaces and the possibility to execute any of these methods.

For data exchange between client and web service we implemented three alternatives, namely to transfer data within the query, to pass a URI describing the data location, or to pass a JDBC connection string. We have run a number of tests to determine the performance of the system. For most cases⁶, the response times are well within acceptable ranges⁷. Exceptions are the transfer of a larger amount of data comprising well more than 50000 rows as well as more complex calculations.

The proposed web service has been implemented and successfully tested within our project. Up to now, service implementation is in development and we plan to deploy it within the operative BExIS system for use by the different subprojects of the Biodiversity Exploratories in the near future.

⁶common calculations comprising data sets with up to 50.000 rows

⁷under two seconds

References

- [JO04] HV Jagadish and F. Olken. Database Management for Life Sciences Research. *ACM SIGMOD Record*, 33(2):15–20, 2004.
- [LMF⁺06] Nianhua Li, Martin T. Morgan, Seth Falcon, Robert Gentleman, and Duncan Temple Lang. From R to Java: the TypeInfo and RWebServices paradigm. Technical report, BioConductor, 2006.
- [LN05] Ulf Leser and Felix Naumann. (Almost) hands-off information integration for the life sciences. In *Second Biennial Conf. on Innovative Database Research (CIDR)*, pages 131–143, Asilomar, CA, January 2005.
- [RS04] M. Rampp and T. Soddemann. A Work Flow Engine for Microbial Genome Research. *Forschung und wissenschaftliches Rechnen*, 68:30–53, 2004.
- [RSL06] M. Rampp, T. Soddemann, and H. Lederer. The MIGenAS integrated bioinformatics toolkit for web-based sequence analysis. *Nucleic Acids Research*, 34(Web Server issue):W15–W19, 2006.
- [RUCM07] R. Rifaieh, R. Unwin, J. Carver, and M.A. Miller. SWAMI: Integrating Biological Databases and Analysis Tools Within User Friendly Environment. *LECTURE NOTES IN COMPUTER SCIENCE*, 4544:48–58, 2007.
- [Sch08] P. Schut. OpenGIS Web Processing Services. OGC Publicly Available Standard OGC 05-007r7, Open Geospatial Consortium, Inc., June 2008. Version 1.0. 0.
- [TSJS07] C. Türker, E. Stolte, D. Joho, and R. Schlapbach. B-Fabric: A Data and Application Integration Framework for Life Sciences Research. In *Data Integration in the Life Sciences 4th International Workshop (DILS)*, pages 37–47, Philadelphia, PA, USA, June 2007. Springer.
- [Urb03] S. Urbanek. Rserve—A Fast Way to Provide R Functionality to Applications. In *Workshop on Distributed Statistical Computing (DSC)*, pages 20–22, Vienna, Austria, March 2003.
- [W3C03] W3C. SOAP Version 1.2 Part 0: Primer. *W3C Recommendation*, 24, 2003.
- [W3C04] W3C. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *W3C Working Draft*, 26, 2004.