The Course's SIB Libraries

Anna-Lena Lamprecht and Alexander Wickert

Chair for Service and Software Engineering, Potsdam University,
Potsdam, D-14482, Germany
{lamprecht,awickert}@cs.uni-potsdam.de

Abstract. This chapter gives a detailed description of the service framework underlying all the example projects that form the foundation of this book. It describes the different SIB libraries that we made available for the course "Process modeling in the natural sciences" to provide the functionality that was required for the envisaged applications. The students used these SIB libraries to realize their projects.

Keywords: scientific workflows, web services, bioinformatics, geoinformation, geovisualization.

1 Introduction and Overview

The domain-independent symbiosis of jABC [37] and jETI [27] introduced in the previous chapter [24] becomes a scientific workflow design, management, and execution system when it is enhanced by adequate domain-specific SIB libraries for the workflow's application domains. The jABC comes with a large collection of SIBs for common, frequently needed, domain-independent workflow functionality. These Common SIBs [10] are useful also for working with scientific workflows, but not sufficient. The students' projects in the scope of our course mostly concerned workflows for specific bioinformatics and geo-visualization applications, and hence required additional domain-specific SIBs. This chapter gives an overview of the services that we as the advisers of the course integrated into the jABC framework and provided as SIBs. With the created SIB repository the students were then able to model the SLGs that realize the workflow scenarios they envisioned. Note that some students also integrated the required services themselves, as described in their individual chapters.

Table 1 surveys the repository of SIBs made available for the course. It comprises existing SIB libraries from previous projects as well as new libraries that have been implemented by us during the course. The table entries are the names of the different SIB collections followed by a number in brackets that corresponds to the number of SIBs contained in the respective library. In total, more than 275 SIBs were finally made available. As the different rows show, the repository covers science-oriented bioinformatics and geoinformatics functionality, but also a number of SIBs that provide additional, more general functionality that can be useful in different domains. The columns of the table categorize the SIBs according to the technologies used for the implementation of the underlying services, which determines the way how the respective services are integrated into

A.-L. Lamprecht et al. (Eds.): Process Design for Natural Scientists, CCIS 500, pp. 30–44, 2014.
DOI: 10.1007/978-3-662-45006-2_3

Table 1. Overview of the Course's SIB Libraries

<table>
<tr><td rowspan="2">domain</td><td colspan="3">remote services</td><td rowspan="2">API-based (161)</td></tr>
<tr><td>jETI-based (7)</td><td>WSDL-based (85)</td><td>REST-based (46)</td></tr>
<tr><td>bioinformatics (25)</td><td></td><td colspan="2">pfam-sibs (6)
ebi-sibs (18)</td><td>forester-sibs (1)</td></tr>
<tr><td>geoinformatics (93)</td><td>gmt-sibs (7)</td><td>csiss-sibs (67)</td><td>geoplugin-sibs (4)
gisgraphy-sibs (6)
lbm-sibs (9)</td><td></td></tr>
<tr><td rowspan="2">both (general) (160)</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td>ftp-sibs (2)
common-sibs (119)
jeti-helper-sibs (9)
openoffice-sibs (21)
rest-sibs (5)
twitter-sibs (1)</td></tr>
<tr><td colspan="2">qrcode-sibs (3)</td></tr>
</table>

the jABC framework to become SIBs. As can be seen from the columns, four major kinds of services are used here:

- jETI-based remote services, which are especially suitable for integrating file-based command line tools,
- WSDL-based web services, for which SIBs can easily be implemented making use of client-side code generated from the WSDL documents,
- REST-based web services, for which SIBs can easily be implemented using standard JRE functionality (`java.net` package), and
- different kinds of (Java) APIs, which can naturally be called from SIBs.

Some SIB library names appear in more than one column when the respective SIB collections comprise services of different technologies. Details of what this means for the respective integration processes are given in the previous chapter of this volume [24], which introduces the principles of modeling and executing scientific workflows in the jABC. The SIBs abstract from these implementation details, so that in the following we focus only on their functionality when we describe in more detail the SIB repository available for the course.

Section 2 describes the available bioinformatics-specific SIBs, Section 3 deals with the geoinformatics-specific SIBs, and finally Section 4 presents the SIBs for the other, more general services. Apart from the gmt-sibs, which are directly generated from our jETI server on request, and the Common SIBs, which are included in the jABC standard distribution, all SIBs listed here are available at the web site of the jabc-sibs community project at `https://projekte.itmc.tu-dortmund.de/projects/jabc-sibs` (registration required).

2 The Bioinformatics SIBs Collections

In this section we describe the SIBs for access to specific bioinformatics services and libraries that were used in students' projects. These libraries had been

available in a smaller version already before the course and were used in previous projects. They were now extended on-demand according to the upcoming requirements.

2.1 ebi-sibs Library (18): SIBs for EBI Web Services

The main mission of the European Bioinformatics Institute (EBI) is to build, maintain and provide biological databases and information services. Via a steadily growing number of web and web service interfaces the EBI provides access to several of its data resources (for instance the EMBL Nucleotide Database, UniProt, Ensembl, and IntAct), to a plethora of analysis tools, including the EMBOSS suite [32], and specific tools for similarity searches, multiple alignment, structural analysis and for exploring literature and ontologies [29,23,18].

For the students' projects, we implemented SIBs for 18 of the EBI's SOAP web services, covering different bioinformatics areas. As there is no specific taxonomic organization, we list them alphabetically:

- `ArrayExpressExperiments` searches the ArrayExpress database (a database of functional genomics experiments) for experiments.
- `ArrayExpressFiles` searches ArrayExpress for files.
- `ClustalW2Phylogeny` creates neighbor-joining or UPGMA phylogenic trees based on ClustalW sequence alignments.
- `Emma` computes a multiple sequence alignment using ClustalW.
- `Garnier` predicts secondary structures of proteins.
- `Getorf` finds and extracts open reading frames (ORFs).
- `Makenucseq` creates random nucleotide sequences.
- `Makeprotseq` creates random protein sequences.
- `NCBIBlast` searches sequence databases using the NCBI BLAST algorithm.
- `Needle` computes a Needleman-Wunsch global sequence alignment.
- `Pepinfo` plots amino acid properties.
- `PsiSearch` combines the Smith-Waterman search algorithm with the PSI Blast profile construction strategy to find distantly related proteins.
- `Scopparse` generates a DCF file from raw SCOP files.
- `Ssematch` searches a DCF file for secondary structure matches.
- `Transeq` translates nucleotide sequence into the corresponding peptide sequences.
- `WSDBFetch` fetches database entries based on IDs or accession numbers.
- `Water` computes a Smith-Waterman local alignment.
- `WUBlast` searches sequence databases using the WU BLAST algorithm.

SIBs from the ebi-sibs library were used in the following student projects:

- Protein classification workflow (Judith Reso) [30]
- Workflow for rapid metagenome analysis (Gunnar Schulze) [34]
- Workflow for phylogenetic tree construction (Monika Lis) [25]

Note that also Leif Blaese used EBI web services in his project "Data mining for unidentified protein sequences" [16], but accessed them by using a command line client via the ExecuteCommand SIB from the Common SIBs library. While this introduces undesirable platform dependency, it makes it possible to influence the order in which the numerous web service queries in this workflow are submitted and the results collected (which is not possible with the predefined ebi-sibs), possibly leading to a faster overall workflow execution time.

2.2 pfam-sibs Library (6): SIBs for Pfam Web Services

The Pfam protein families database [17] provides a number of REST-style web services for data retrieval and database searches [11]. The following SIBs have been implemented for use in a project on functional annotation of protein sequences:

- `Accession2Id` converts a Pfam accession number into a Pfam ID.
- `Id2Accession` converts a Pfam ID into a Pfam accession number.
- `PfamAAnnotations` retrieves the annotations from a Pfam-A family page.
- `PfamAFamilyList` retrieves a list of all Pfam-A families in the latest Pfam release.
- `ProteinSequenceData` retrieves the protein sequence data from a Pfam-A family page.
- `SequenceSearch` searches for Pfam domains matching the input protein sequence.

SIBs from the pfam-sibs library were used in the following student projects:

- Protein classification workflow (Judith Reso) [30]
- Workflow for rapid metagenome analysis (Gunnar Schulze) [34]

2.3 forester-sibs Library (1): SIBs for the Forester API

The forester libraries provide a rich collection of functionality for phylogenomics and and evolutionary biology research [41]. They are, for instance, the basis for the Archaeopteryx phylogenic tree viewer (formerly ATV [42]). The forester-sibs use the Java library of the Archaeopteryx project to create graphical representations (e.g. PNGs) from textual phylogenetic tree formats. While there is potential for more, currently the forester-sibs library consists only of one SIB:

- `TreeFile2GraphicsFile` creates a graphics file from a phylogenetic tree file.

This SIB was used by Monika Lis in her project "Workflow for phylogenetic tree construction" [25] for the visualization of the created phylogenetic tree.

3 The Geoinformatics SIBs Collections

In this section we describe the SIBs to access specific geoinformatics services and libraries that were used in students' projects. Only the lbm-sibs library had been available before the course and used in previous projects. All the other SIBs have come into being in the scope of our course.

3.1 gmt-sibs Library (7): SIBs for the Generic Mapping Tools (GMT)

The Generic Mapping Tools (GMT) [40,39] are "an open source collection of 65 tools for manipulating geographic and Cartesian data sets (including filtering, trend fitting, gridding, projecting, etc.) and producing Encapsulated PostScript File (EPS) illustrations ranging from simple x-y plots via contour maps to artificially illuminated surfaces and 3-D perspective views"[39]. Being file-based command line tools, they were most easily integrated into the jABC framework using the **jETI technology** (cf. previous chapter).

We did not integrate the entire GMT with all possible arguments, but only those that were required for the students' projects. Concretely, the following GMT-SIBs have been created (see [39] for comprehensive documentation):

- `grd2xyz` converts one or more 2D grid files to ASCII or binary format.
- `grdclip` clips the range of data values in a 2D grid file.
- `ps2raster` converts one or more postscript files to raster format.
- `pscoast` plots land masses, water masses, coastlines, borders and rivers of a given region into a postscript file.
- `psscale` plots a grey scale or a color scale on a map in postscript format.
- `pstext` plots text on a map in postscript format.
- **psxyz** plots 3D lines, polygons and symbols in a postscript file.

SIBs from the gmt-sibs library were used in the following student projects:

- Location Analysis for Placing Artifical Reefs (Lasse Scheele) [33]
- Creation of Topographic Maps (Josephine Kind) [21]

With the jETI server up and running and the experiences with the GMT that we have obtained during the project work, integrating more of these tools is now a straightforward process and can be done on demand when the need arises in the scope of future projects.

3.2 csiss-sibs Library (67): SIBs for the CSISS Web Services

The geospatial web services of the Center for Spatial Information Science and Systems (CSISS) [3] "have been developed to provide geospatial processing and analysis based on existing software or geosciences modules" [3]. Based on the open source Geographic Resources Analysis Support System (GRASS) [8] and some existing web services and geoscience analysis modules, the CSISS web services provide functionality for working with raster, vector and satellite image

data. The 67 SIBs implemented for this collection cover all six categories of services (for more elaborate documentation please refer to the CSISS web site [3]):

1. Geospatial web services for satellite image processing (12):
   - `Raster_EdgeDetection` finds the edges in an image.
   - `Raster_FFT` processes the image based on the FFT algorithm.
   - `Raster_FusionBrovey` performs a Brovey transformation.
   - `Raster_HIS2RGB` transforms an HIS image to RGB color space.
   - `Raster_IFFT` transforms the output of `Raster_FFT` into a normal image.
   - `Raster_Mosaic` mosaics adjacent images.
   - `Raster_OIF` calculates the optimal index factor.
   - `Raster_PCA` performs a principal components analysis (PCA).
   - `Raster_RGB2HIS` transforms and RGB image to HIS color space.
   - `Raster_SupervisedClassifcationService` can be used to reclassify multispectral satellite data with supervised classification methods.
   - `Raster_TasseledCap` performs Tasseled Cap (Kauth Thomas) transformation.
   - `Raster_UnsupervisedClassifcationService` can be used to reclassify multispectral satellite data with unsupervised classification methods.
2. Geospatial web services for raster map processing (28):
   - `Raster_Aspect` generates a raster map layer of aspect derivates.
   - `Raster_BBoxClip` clips a raster map by a bounding box.
   - `Raster_Buffer` creates buffer zones in a map.
   - `Raster_ChangeColortable_Copy`, `Raster_ChangeColortable_Predefined` and `Raster_ChangeColortable_Userdefined` change the color table of an image by copying a table from another image, or by using predefined or user-defined tables, respectively.
   - `Raster_CreateContour` produces a contour map.
   - `Raster_GeoparameterCalculation` extracts terrain parameters.
   - `Raster_GreyScale` converts the map to greyscale.
   - `Raster_ImageAlgebra` performs arithmetics on raster map layers.
   - `Raster_LatLonBBoxClip` clips a raster map by a bounding box.
   - `Raster_MatrixFilter` applies a matrix filter.
   - `Raster_NDVI` calculates the normalized differenced vegetation index.
   - `Raster_PatchMultiBand` mosaics RGB channels of adjacent images.
   - `Raster_PatchSingleBand` patches together adjacent map layers.
   - `Raster_PolygonClip` clips a raster map by a polygon.
   - `Raster_Profile` identifies raster map values on user-defined lines.
   - `Raster_ProfileCurvature` computes profile curvatures.
   - `Raster_Rescale` changes the image scale.
   - `Raster_RGBcomposite` combines RGB map layers into a color image.
   - `Raster_RGBextract` extracts the RGB components.
   - `Raster_Slope` computes slopes.
   - `Raster_SurfaceGeneration` creates a raster elevation map.
   - `Raster_SurfaceInterpolation` performs raster data interpolation.
   - `Raster_TangentialCurvature` computes tangential curvature.

   - `Raster_TopographicIndex` creates a topographic index.
   - `Raster_TopographicShading` creates a shaded relief map.
   - `Raster_Vectorization` converts a raster map to vector format.
3. Geospatial web services for raster map statistics (6):
   - `Raster_AreaStatistics` creates a histogram of the training areas.
   - `Raster_ClassificationStatistics` calculates classification statistics for the cells of the map.
   - `Raster_CovarianceCorrelation` gives a covariation/correlation matrix.
   - `Raster_DefinedIntervalStatistics`, `Raster_EqualIntervalStatistics` and `Raster_ManualIntervalStatistics` compute equal interval classification statistics based on the specified interval size, class numbers and class ranges, respectively.
4. Geospatial web services for vector map processing (15):
   - `Vector_AttributeColumn` prints the types and names of the attributes.
   - `Vector_Buffer` creates a buffer around selected features.
   - `Vector_BuildPolylines` builds polylines from lines or boundaries.
   - `Vector_BuildTopology` creates topologies.
   - `Vector_CleanTopology` automatically fixes vector topologies.
   - `Vector_FeatureExtraction` extracts vector objects for selected features.
   - `Vector_FeatureSelection` supports selection of features from a vector.
   - `Vector_GML2SHP` converts GML to shape file format.
   - `Vector_Overlay` overlay two vector maps.
   - `Vector_Patch` patches together several map layers.
   - `Vector_QueryInformation` reports basic information about the map.
   - `Vector_Rasterization` transforms a vector map layer to raster format.
   - `Vector_SHP2GML` converts shape files to GML format.
   - `Vector_ShortestPath` performs shortest-path analysis.
   - `Vector_ValueExtraction` extracts raster values.
5. Web services for hydrological analysis based on raster maps (5):
   - `Raster_DrainageBasin` computes drainage directions and watershed basins.
   - `Raster_FlowAccumulation` computes flow accumulations.
   - `Raster_FlowDirection` computes flow direction.
   - `Raster_StreamExtraction` computes stream networks.
   - `Raster_OpennessCalculation` computes surface openness.
6. Web service for fire-spread simulation (1):
   - `Fire_SpreadSimulation` simulates the spread of wildfires.

SIBs from the csiss-sibs library were used in the following student projects:

- Web-based Map Generalization Tools Put to the Test (Henriette Sens) [35]
- CREADED: Coloured-Relief Application for Digital Elevation Data (Franziska Noack) [28]
- Location Analysis for Potential Areas for Wind Turbines (Tobias Respondeck) [31]

3.3 geoplugin-sibs Library (4): SIBs for the Geoplugin Web Services

GeoPlugin [6] is a REST-style web service that provides operations to determine the location of an IP address and retrieves additional information for a given geolocation. The geoplugin-sibs library comprise four SIBs:

- `Geolocation` returns information (such as city, region, countryName, ...) for a given IP address.
- `Location` returns the closest location (if possible) for a given latitude-longitude pair.
- `NearbyPlaces` returns nearby places for a given latitude-longitude pair.
- `Postalcode` returns the postal code for a given latitude-longitude pair.

SIBs from the geoplugin-sibs library were used in the following student projects:

- Visualization of Data Transfer Paths (Christian Kuntzsch) [22]

3.4 gisgraphy-sibs Library (6): SIBs for the Gisgraphy Web Services

Gisgraphy introduces itself as "a free, open source framework that offers the possibility to do geolocalisation and geocoding via Java APIs or REST webservices" [4]. The gisgraphy-sibs cover all six web services that gisgraphy provides:

- `AddressParser` parses a text address and returns its components (street name, house number, street type, etc.).
- `Fulltext search` does a fulltext search and returns all associated information of the found item.
- `Geocoding` returns information (such as city, zipCode, country, ...) for a given address with country code.
- `Geolocalisation` finds nearby places, streets, etc. of a given latitude-longitude pair and a radius.
- `Reverse geocoding` returns street name and other information of a given latitude-longitude pair.
- `Street search` returns information of a given street.

SIBs from the gisgraphy-sibs library were used in the following student project:

- GraffDok: A Graffiti Documentation Application (Robin Holler) [20]

3.5 lbm-sibs Library (9): SIBs for Location-Based Mapping Services

The lbm-sibs have originally been implemented for projects preceding our course, but were updated to be useful again. Making use of different location-based mapping services (OpenStreetMap API [2], Google Maps API [7], InstaMapper [9] and Wikipedia's GeoNames database [5]), they comprise the following SIBs:

- `AddWaypoint` draws a waypoint for a given location into a map using the OpenStreetMap API.

- `CenterMap` centers the map around a given location using the OpenStreetMap API.
- `GetLocationFromGoogle` retrieves a geoposition matching a given search string using the Google Maps API.
- `GetLocationFromInstaMapper` gets the most recent geolocation for a given device using InstaMapper GPS tracking.
- `GetLocationFromGeonamesByPostalCode` retrieves a list of geolocations for a given postal code using the GeoNames database.
- `GetLoactionsFromInstaMapper` gets the last $amount geolocations for a given device using InstaMapper GPS tracking.
- `GetSurroundingLocationsFromGeonames` gets a list of surrounding geolocations for a given geolocations using the GeoNames database.
- `SetZoomLevel` sets a new zoom level for the given map using the OpenStreetMap API.
- `ShowOpenStreetMap` initializes and opens a new map window with OpenStreetMap's rendered tiles.

While these SIBs had been used for tutorial exercises preceding the project work, they have not appeared in any of the final projects. Instead, the students used services with similar but more application-specific functionality, like the latest functions provided the Google Maps API, or the geoplugin and gisgraphy SIBs described above.

4 SIBs Collections for Other Functionality

In this section we describe the SIBs to access general, domain-independent services and libraries that we prepared for the students' projects. Although not all these libraries were actually used in the projects, for instance when more suitable alternatives were finally found, these libraries provide useful functionality and are likely to be used in other projects, so we include them here.

4.1 ftp-sibs Library (2): SIBs for FTP Operations

The ftp-sibs provide basic file transfer services:

- `FtpDownload` downloads a file from a remote FTP server.
- `FtpUpload` uploads a file to a remote FTP server.

In addition to the local and remote paths, user name and password must be specified in order to establish the connection to the server. Note that no external library was necessary for the implementation of these SIBs, the standard Java libraries provide all required functionality.

SIBs from the ftp-sibs library were used in the following student projects:

- CREADED: Coloured-Relief Application for Digital Elevation Data (Franziska Noack) [28]
- Location Analysis for Potential Areas for Wind Turbines (Tobias Respondeck) [31]

4.2 jeti-helper-sibs Library (9): SIBs for Working with jETI Services

The jeti-helper-sibs come with the jETI plugin for the jABC framework. As jETI works with a separate execution context, the so-called TransferHandler that transparently manages the transfer of data between client and server(s), all data that is to be transferred to/from a jETI service has to go via this context. The jeti-helper-sibs provide different ways for preparing data to be used by jETI services, and to process data that is returned from jETI service executions:

- `ETIErrorSIB` displays jETI error messages.
- `ReadFile` loads the specified file to the TransferHandler.
- `ReadFromContext` loads a file name from the context and loads the corresponding file to the TransferHandler.
- `ReadFromURL` reads a file name from an URL and loads the corresponding file to the TransferHandler.
- `Viewer` shows a file from the TransferHandler in a specified program.
- `ViewerWin32` is the same as `Viewer`, but for Windows platforms.
- `WriteFile` writes a file from the TransferHandler to the specified file.
- `WriteFileToURL` writes a file from the TransferHandler to a specified URL.
- `WriteFileToContext` copies a file from the TransferHandler to the temporary directory and writes the file name into the jABC's standard execution context.

SIBs from the jeti-helper-sibs library were used in the following student projects, which are exactly those that also used the jETI-based gmt-sibs library:

- Location Analysis for Placing Artifical Reefs (Lasse Scheele) [33]
- Creation of Topographic Maps (Josephine Kind) [21]

4.3 openoffice-sibs Library (21): SIBs for Accessing OpenOffice Functionality

The openoffice-sibs have also been created in the scope of a previous project [10] to make OpenOffice [1] functionality available for use within jABC workflows. With these SIBs, it is possible to create basic OpenOffice documents (like simple text files and spreadsheets), and to perform basic manipulation operations on them. Some selected examples from this SIB collection are given in the following:

- `CalculateColumnSum` calculates the sum of a table column.
- `CloseDocument` closes the active document.
- `InsertNewSheet` inserted a new sheet (tab) into the spreadsheet container.
- `InsertTable` inserts a table into a text document.
- `MoveCursor` moves the cursor within the current text document
- `NewDocument` creates a new document or opens an existing one.
- `PrintDocument` prints the active document.
- `ReplaceText` replaces all occurrences of a specific text phrase in the the document by another text.

- `SaveDocument` saves the current document to a file.
- `SetCellValue` sets the value/formula/text of a specific cell in a spreadsheet.
- `SetCursorProperties` sets properties of the cursor (like, e.g., font and color) in the text document.
- `SetCurrentSheet` activates a sheet in the spreadsheet container.
- `WriteText` writes a text to the active text document at the current cursor position.

SIBs from the openoffice-sibs library had initially been used to generate report files in the "GraffDok" project of Robin Holler [20], which are assembled from the textual and graphical data that is collected about the graffiti. They were however later replaced by LaTeX commands (executed by the `ExecuteCommand` SIB), simply because the use of LaTeX templates allowed for a better customization of the document layout.

4.4 rest-sibs Library (5): SIBs for General REST Web Service Access

Unlike some of the SIBs described above, which provide access to specific REST-style web services (like, e.g., the pfam-sibs or the gisgraphy-sibs), the rest-sibs are designed to act as generic REST service clients: they simply read the content from a given URL. This way, they can be used to access arbitrary REST services. As the concrete requirements of workflow and the service interfaces vary, we have finally implemented different versions of REST-accessing SIBs:

- `FetchDataURL2File` fetches the data behind the given URL as it is and writes it into a file.
- `FetchImageURL2BufferedImage` fetches the data behind the given URL, interprets it as image and puts it into the ExecutionContext.
- `FetchImageURL2File` fetches the data behind the given URL, interprets it as image and writes it into a file.
- `FetchTextualURL` fetches the data behind the given URL, interprets it as text and puts in into the ExecutionContext.
- `FetchTextualURL2File` fetches the data behind the given URL, interprets it as text and writes it into a file.

SIBs from the rest-sibs library were used in the following student projects:

- Visualization of Data Transfer Paths (Christian Kuntzsch) [22]
- Geocoder Accuracy Ranking (Daniel Teske) [38]
- CREADED: Coloured-Relief Application for Digital Elevation Data (Franziska Noack) [28]
- Location Analysis for Potential Areas for Wind Turbines (Tobias Respondeck) [31]
- Spotlocator - Guess where the Photo was taken! (Marcel Hibbe) [19]

4.5 twitter-sibs Library (1): SIBs for Accessing Twitter

The twitter-sibs library currently contains only one SIB for sending ("tweeting") a message via Twitter [13]:

- `TwitterText` tweets a text message.

The SIB needs to be pointed to an existing Twitter account, which also has to be set up for that functionality. The implementation has been done using the Twitter4J-Library [14], an unofficial but convenient Java library for the original Twitter API. The SIB was used in the "Spotlocator" project of Marcel Hibbe[19].

4.6 qrcode-sibs Library (3): SIBs for Reading and Writing QR Codes

Finally, we have provided the students with SIBs for reading and creating Quick Response (QR) codes:

- `CreateQRCode` creates a QR code from the provided data.
- `DecodeQRCodeFromBufferedImage` decodes a QR code that is available as buffered image.
- `DecodeQRCodeFromURL` decodes the QR code located at the given URL.

In the overview table, this SIB collection is classified both as REST-based and as API-based implementation. This is because creating the QR codes is done via the REST-based web service of the QR-Server API [12], while reading the codes is done via the Java library ZXing [15]. Note that the creation of QR codes allows many configuration options, such as the color of the data squares, the color of the background squares, the border width, and the size of the whole QR code. Although developed according to some specific project ideas, this SIB library has finally not been used in any of the projects.

5 Conclusion

More than 275 SIBs have finally been made available for the student projects. We created around half of them during the course, to a large extent specifically according to the upcoming requirements of the projects. A large part of them was also used in the next editions of the course in the following years, where the SIB libraries were again extended according to the concrete projects' requirements. Although the technical details of individual tools, services and APIs are sometimes challenging, this is generally a swift and easy business. As described in [26], a single component or service does typically offer a number of functionalities. In the *servification* process, the useful functionalities are identified (often ad-hoc for the current use) and transformed into a SIB collection. Further functionalities can be added later on, in a form of *incremental* formalization [36] of the domain. In fact, several of the SIB libraries do still not yet cover the full range of functionality provided by the underlying services, but it will be a straightforward process to extend them accordingly if there is need in the future.

References

1. Apache OpenOffice -The Free and Open Productivity Suite, http://www.openoffice.org/de/ (last accessed September 10, 2013)
2. API - OpenStreetMap Wiki, http://wiki.openstreetmap.org/wiki/API (last accessed September 9, 2013)
3. CSISS/GMU Geospatial Web Services, http://geobrain.laits.gmu.edu/grassweb/manuals/index.html (last accessed September 9, 2013)
4. Free opensource geocoder and webservices for geonames and openstreetmap data, http://www.gisgraphy.com/ (last accessed September 9, 2013)
5. GeoNames, http://www.geonames.org (last accessed September 9, 2013)
6. geoPlugin to geolocate your visitors, http://www.geoplugin.com/ (last accessed September 9, 2013)
7. Google Maps API - Google Developers, https://developers.google.com/maps/ (last accessed March 21, 2013)
8. GRASS GIS - The world's leading Free GIS software, http://grass.osgeo.org (last accessed September 9, 2013)
9. InstaMapper, http://www.instamapper.com (shut down on December 13, 2012)
10. jABC SIBs - Introduction, http://jabc.cs.tu-dortmund.de/sib/ (last accessed September 10, 2013)
11. Pfam: RESTful interface, http://pfam.sanger.ac.uk/help#tabview=tab10 (last accessed September 9, 2013)
12. QR-Code-API - Documentation, command "create-qr-code" - QR-Server, http://qrserver.com/en-us/api/documentation/create-qr-code/ (last accessed September 10, 2013)
13. Twitter, https://twitter.com (last accessed September 10, 2013)
14. Twitter4J - A Java Library for the Twitter API, http://twitter4j.org/ (last accessed September 10, 2013)
15. zxing - Multi-format 1D/2D barcode image processing library with clients for Android, Java, http://code.google.com/p/zxing/ (last accessed September 10, 2013)
16. Blaese, L.: Data Mining for Unidentified Protein Sequences. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 73–87. Springer, Heidelberg (2014)
17. Finn, R.D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J.E., Gavin, O.L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E.L.L., Eddy, S.R., Bateman, A.: The Pfam protein families database. Nucleic Acids Research 38(suppl 1), D211–D222 (2010)
18. Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J., Lopez, R.: A new bioinformatics analysis tools framework at EMBL-EBI. Nucleic Acids Research 38(Web Server issue), W695–W699 (2010)
19. Hibbe, M.: Spotlocator Project Documentation. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 149–158. Springer, Heidelberg (2014)
20. Holler, R.: GraffDok: A Graffiti Documentation Application. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 235–247. Springer, Heidelberg (2014)
21. Kind, J.: Creation of Topographic Maps. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 225–234. Springer, Heidelberg (2014)

22. Kuntzsch, C.: Visualization of Data Transfer Paths. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 140–148. Springer, Heidelberg (2014)
23. Labarga, A., Valentin, F., Anderson, M., Lopez, R.: Web services at the European bioinformatics institute. Nucleic Acids Research 35(Web Server issue), W6–W11 (2007)
24. Lamprecht, A.-L., Margaria, T., Steffen, B.: Modeling and Execution of Scientific Workflows in the jABC Framework. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 14–29. Springer, Heidelberg (2014)
25. Lis, M.: Constructing a Phylogenetic Tree. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 101–109. Springer, Heidelberg (2014)
26. Margaria, T., Boßelmann, S., Doedt, M., Floyd, B.D., Steffen, B.: Customer-Oriented Business Process Management: Visions and Obstacles. In: Hinchey, M., Coyle, L. (eds.) Conquering Complexity, pp. 407–429. Springer, London (2012)
27. Margaria, T., Nagel, R., Steffen, B.: jETI: A tool for remote tool integration. In: Halbwachs, N., Zuck, L.D. (eds.) TACAS 2005. LNCS, vol. 3440, pp. 557–562. Springer, Heidelberg (2005)
28. Noack, F.: CREADED: Coloured-Relief Application for Digital Elevation Data. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 182–195. Springer, Heidelberg (2014)
29. Pillai, S., Silventoinen, V., Kallio, K., Senger, M., Sobhany, S., Tate, J., Velankar, S., Golovin, A., Henrick, K., Rice, P., Stoehr, P., Lopez, R.: SOAP-based services provided by the European Bioinformatics Institute. Nucleic Acids Research 33(Web Server issue), W25–W28 (2005)
30. Reso, J.: Protein Classification Workflow. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 65–72. Springer, Heidelberg (2014)
31. Respondeck, T.: A Workflow for Computing Potential Areas for Wind Turbines. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 196–211. Springer, Heidelberg (2014)
32. Rice, P., Longden, I., Bleasby, A.: EMBOSS: the European Molecular Biology Open Software Suite. Trends in Genetics 16(6), 276–277 (2000)
33. Scheele, L.: Location Analysis for Placing Artificial Reefs. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 222–234. Springer, Heidelberg (2014)
34. Schulze, G.: Workflow for Rapid Metagenome Analysis. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 88–100. Springer, Heidelberg (2014)
35. Sens, H.: Web-Based Map Generalization Tools Put to the Test: A jABC Workflow. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 171–181. Springer, Heidelberg (2014)
36. Steffen, B., Margaria, T., Claßen, A., Braun, V.: Incremental Formalization: A Key to Industrial Success. Software - Concepts and Tools 17(2), 78–95 (1996)
37. Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C.: Model-driven development with the jABC. In: Bin, E., Ziv, A., Ur, S. (eds.) HVC 2006. LNCS, vol. 4383, pp. 92–108. Springer, Heidelberg (2007)
38. Teske, D.: Geocoder Accuracy Ranking. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 159–170. Springer, Heidelberg (2014)

39. Wessel, P.: The GMT Home Page, http://gmt.soest.hawaii.edu (last accessed January 3, 2013)
40. Wessel, P., Smith, W.H.F.: Free software helps map and display data. EOS Trans. Amer. Geophys. U. 72(41) (1991)
41. Zmasek, C.M.: forester | software libraries for evolutionary biology and comparative genomics research (November 2012), http://www.phylosoft.org/forester/ (last accessed December 6, 2012)
42. Zmasek, C.M., Eddy, S.R.: ATV: display and manipulation of annotated phylogenetic trees. Bioinformatics (Oxford, England) 17(4), 383–384 (2001)