SPECCHIO: A WEB-ACCESSIBLE SPECTRAL DATABASE FOR THE ADMINISTRATION OF HETEROGENEOUS CAMPAIGN DATA

S. Bojinski, M. Schaepman, D. Schläpfer, and K. Itten

Remote Sensing Laboratories, RSL
University of Zurich, Winterthurerstr. 190, CH-8057 Zurich

KEY WORDS: spectral database, Web access, reference spectral collection

ABSTRACT

The administration of spectral remote sensing data is a key factor for thorough, comparative analyses which take into account the entire information available to the investigator. Today, a great variety of spectrometers operate on the field, creating a strong heterogeneity in data and meta data formats. For a holistic view on these data, an administration tool must adhere to certain principles: independence of file format, flexibility to attribute changes, establishment of relations between data, content-based search capability, common interfaces, and scalability. SPECCHIO as a Web-accessible spectral database represents a new approach that incorporates the mentioned principles and overcomes the drawbacks of file-based solutions. It is based on a relational data model and provides a prototype for an administrative tool that accounts for the complexity of heterogeneous spectral data. It may also serve as a reference database for surface targets. The proof of concept and robustness of the approach is demonstrated using spectral remote sensing data from various campaigns.

1 INTRODUCTION

The high diversity of spectral measurements in remote sensing gives rise to a great variety of spectral data as well as meta data formats, where the latter describe the former by a set of attributes. Meta data, e.g. date and location of a measurement, are frequently being stored in a file header, which is defined by some standard. There are several examples of format standards for spectral data, such as CDF, HDF (Fortner, 2000), JCAMP (IUPAC, 2000), or ENVI (RSINC, 2000). Along with a file-based organization of the data comes a file system tree that provides an logical organizational structure. However, experience shows many limitations of this approach, especially for large amounts of data, as it poorly represents the whole complexity of information prevailing in remote sensing applications. As an example, given spectral measurements from many sources (i.e. campaigns), it is a well-known phenomenon that users easily lose sight of the entirety of information available to them. Also, for a comprehensive analysis of spectra, an intercomparison of data from different sensors is desired, but hard to implement. Most of the references given in section 2 of this work, however, use the file-based way of data administration.

Six principles can be identified that are to be fulfilled by an administration tool that accounts for the heterogeneity of spectral remote sensing data: independence of file format, flexibility to attribute changes, establishment of relations between data, content-based search capability, common interfaces, and scalability.

In this paper, we present SPECCHIO (SPECtral Input Output) as a prototype of a spectral database that matches the abovementioned criteria. It consists of a relational database system and appropriate user interfaces, namely script templates and Web browser. Paragraph 3 presents the technology, paragraph 4 the conceptual side of the system. Chapter 5 provides an insight to the interfaces to the database, followed by an outlook that outlines possible applications for SPECCHIO in remote sensing.

2 OVERVIEW OF SPECTRAL REMOTE SENSING DATA RESOURCES

There exists a large number of online remote sensing data archives that can be accessed by the scientific or commercial user to query and order the data needed. For the most part, these sites are set up by the data distibutor for a particular sensor or mission (e.g. USGS1, 2000; DLR, 2000) and provide very large collections of airborne or spaceborne spectral scenes. The granularity of data is usually on scene level and does not break down the spectral information to the single spectrum.

Two publicly accessible sites could as yet be identified, that contain single spectra retrievable in ASCII format: the USGS Denver Spectroscopy Lab (Clark et al., 1993; USGS2, 2000) focuses on mineral spectra, and the ASTER Spectral Library (NASA, 2000) which includes about 2000 spectral entries from natural and man-made materials. Price (1995) created an offline collection of over 3400 spectra with reference character from various sources in ASCII format. These spectral collections are all file system-based.

MedSpec (Preissler et al., 1998) is a standalone spectral database solution that features a large number of meta data attributes for campaign data. It allows input and query of spectral data in ASCII format only, as well as of ancillary data.

3 SPECCHIO: TECHNOLOGY

In the main, there are three different technologies that can be distinguished for providing Web access to databases. To date, the still most widely used is that of access via Common Gateway Interface (CGI) programs (Gundavaram, 1996). Two other approaches based on Java are becoming increasingly popular: firstly, a Java Database Connectivity (JDBC) interface using servlets (server-sided), secondly a client-side technique that downloads Web components of the database as Java applets.

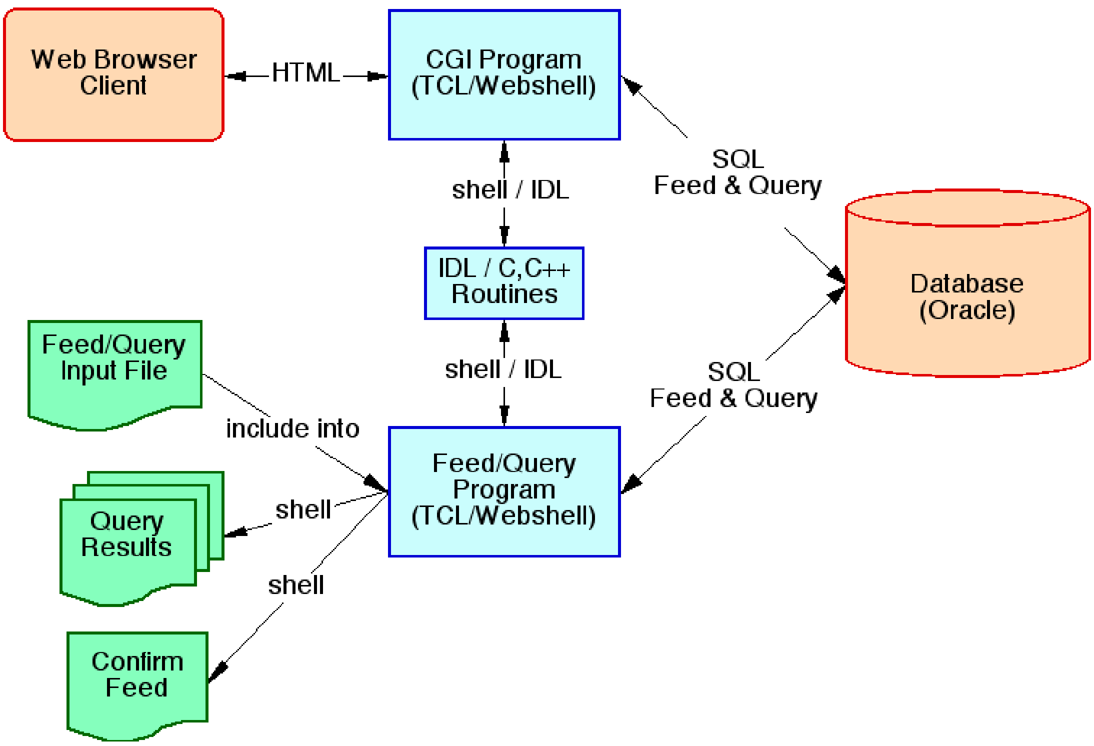


Figure 1: Connectivity schema of SPECCHIO. Two alternatives to access the database are available: the Web-based interface in the upper half, the script-based way in the lower half of the diagram.

We use the CGI-based approach for reasons of implementation speed and ease of maintenance. It uses the standard HTTP protocol for all communication between client and server. Users supply query or update parameters by means of HTML forms, and these parameters are sent in HTTP to the Web server, which calls a CGI program to interpret the request and establish a connection to the SPECCHIO database. The query results provided by the database are encoded in HTML documents and returned to the client. Afterwards, it is possible to select individual datasets for run-time generated graphics display in the browser window, technically realized by a background process that generates GIF files.

In many scientific applications, automated processing tasks are called for, performed by executable scripts. This gives rise to a script-based feed and query option to access the database. These scripts directly connect to the database and execute the requested transactions. Output is written into files on the local file system. An overview of the SPECCHIO connectivity is given in Fig. 1.

Dynamic Web interface generation and database connectivity has been implemented with the scripting languages TCL (TCL, 2000) and Webshell (WEBSH, 2000), the latter being an Open Source programming environment with specific CGI features and built-in TCL interpreter. The relational database system Oracle 8i (ORA, 2000), accessed by transactions coded in embedded SQL, is particularly suited for the consistent and reliable storage and backup of large amounts of data. Processing and display of spectral data has been implemented with IDL and IDL On the Net (ION) (RSINC, 2000). Currently, SPECCHIO runs on a UNIX/Sun Solaris system with an Apache Web server.

4 SPECCHIO: DATABASE SCHEMA

It is an important goal of SPECCHIO to manage all spectral remote sensing data and meta data within one single relational database instance. In doing so, data security issues can be delegated to the database management system, such as the integrity of data, access control, consistency, and the recovery of data in case of hardware failure. Furthermore, the stored information can be efficiently be retrieved by the standardized query language SQL.

In every relational database, the entire information is stored in two-dimensional tables that consist of columns and rows. Tables represent semantic entities, columns semantic attributes of an entity, and rows the actual datasets. Data cells that are defined by column name and row number contain atomic values. The 'spectral remote sensing data' aspect of reality is broken down to entities, that are specified by attributes and logically related by primary and foreign keys, constituting an entity-relationship diagram (Fig. 2). The relations are in the 3rd normal form to preclude update anomalies (Schlageter & Stucky, 1983).

Grouped around the main entity 'Spectrum' are four other entities that contain meta data, 'Sensor', 'Measurement General', 'Measurement Position', and 'Target Type', that are related to 'Spectrum' by 1:n -relations. This means that each 'Spectrum' dataset is uniquely related to exactly one entry in the 'Sensor' table, 'Measurement General' table, etc. Vice versa, an entry in the 'Sensor' table can be related to one or many entries in the 'Spectrum' table, and accordingly. As a result, a new entry into SPECCHIO is equivalent to $m \geq 1$ entries or updates in the 'Spectrum' table, associated with $m \leq n$ entries or updates in each meta data table.

Within each entity, a distinction is made between active and passive attributes. Active attributes can be queried in the query interfaces, passive ones come along as the result of a query transaction, but cannot be queried in their own right. As a quality control measure for the database, a number of attributes has been defined as not null, i.e., they must be stated at all times.

SPECCHIO stores meta data independent of sensor types and file formats. The actual data defining the physical quantity of the spectrum physically remains outside the database, addressed online by a file location pointer, or offline by an external medium identifier. This concept makes the database server a 'thin' server. Access to the database is provided by scripts as well as by standard Web browsers, which are shown in paragraph 5.

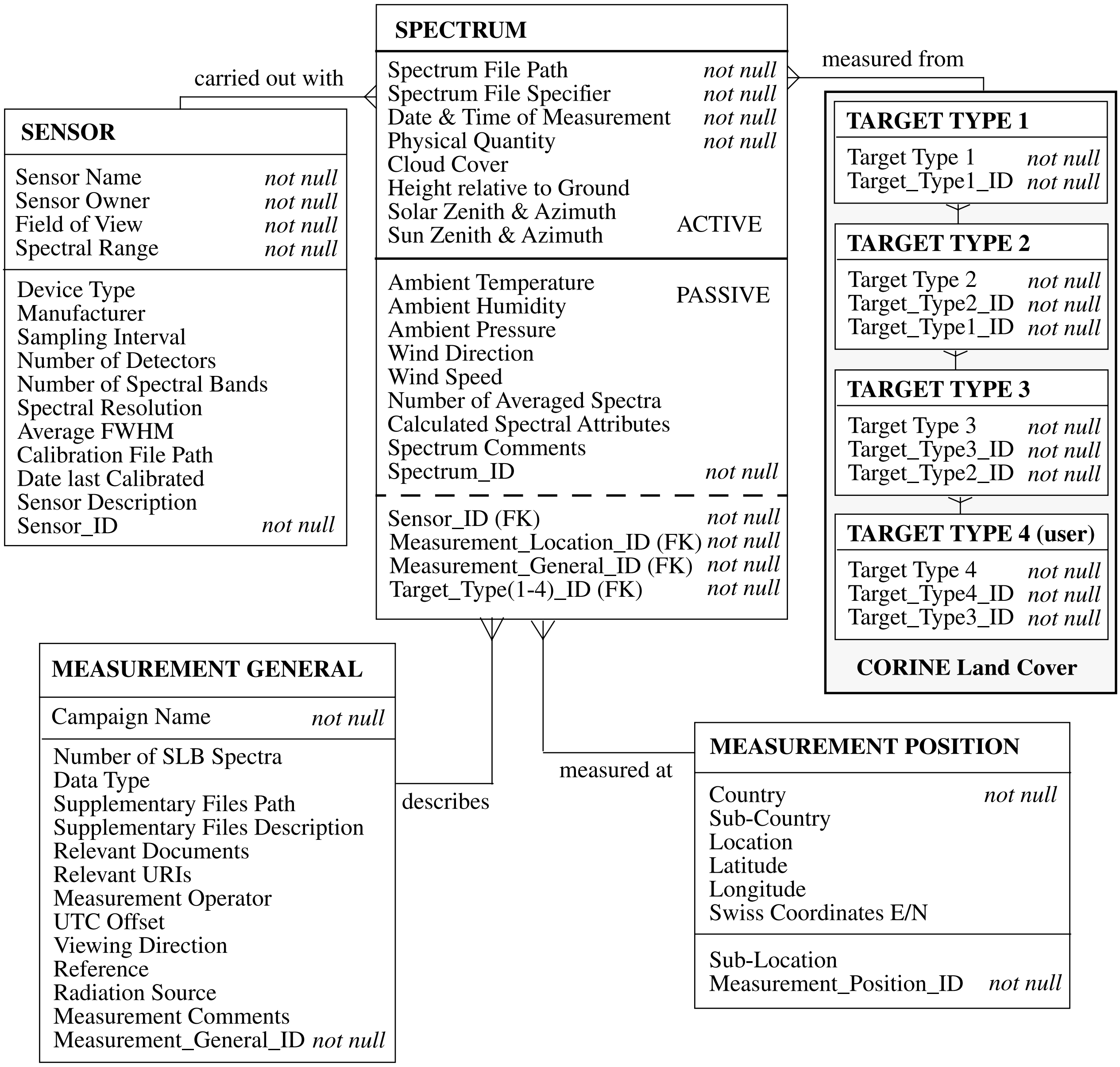


Figure 2: Entity-relationship diagram reflecting the relational database schema, which describes the 'spectral remote sensing data' aspect of reality. The principal 'Spectrum' table is related to the surrounding tables by foreign keys (FK) defined on the respective IDs. 'Active' attributes can be queried as opposed to 'Passive' attributes. *not null* tags denote compulsory attributes. The hierarchical target type structure (Target Type 1-3) represents the CORINE Land Cover Schema (European Commission, 1993).

5 INTERFACES TO SPECCHIO

Interaction with the SPECCHIO database instance is intended to be easy to implement and widely usable. For remote and interactive access, dynamically created Web forms provide a universal way to communicate with the database via a standard browser. Alternatively, for automated tasks, predefined script templates can be employed.

.

5.1 Feed data

The Web-based feed site grants a user-friendly possibility to enter information into the database. Returned dynamically by the server, the feed site shows the current range of values already present in the database for attributes with relatively small semantic variation. This feature reduces semantic redundancies, i.e. different concepts with the same meaning are prevented from being entered and conceived as different by the database. For data quality reasons, it is advisory to give as many attributes as possible, though some are compulsory and denoted as such. Currently, the implementation is mainly intended for small numbers of spectra to put in. The issue of large spectral sets in which individual spectra differ only in one or two attributes (e.g. sensor angles) is still to be considered closely. The feed site may as well serve as an database view to support the script-based alternative in terms of semantic redundancy.

The script-based feed is mainly intended for automated or large-scale input to the database (Fig. 3). It is currently optimized for spectra in the ENVI spectral library format. In principle though, any set of spectra stored in a number-distinguishable way can be entered, as they are automatically allocated a spectral specifier.

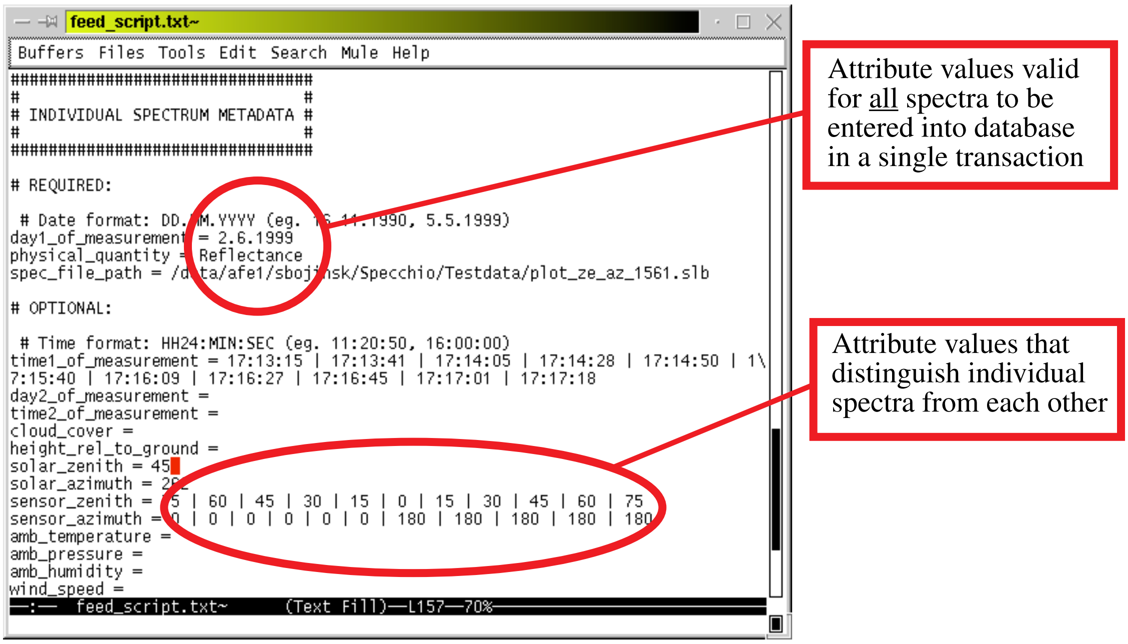


Figure 3: Feed input file syntax.

5.2 Query data

The majority of SPECCHIO users may have questions directed to the database, that are translated into a combination of selected attributes. For example,

- Who measured GER 3700 barley spectra in 1997, and how exactly were they measured?
- Show me all the grass spectra measured at sensor angle 45 degrees!
- Compare all bare soil spectra measured with the ASD device during the Barrax campaign!

For fast and possibly automated query of the database, a script solution exists, which is syntactically similar to the feed script. The output consists of a text file containing a list of datasets that match the search criteria, including all the associated meta data. The Web site allows an interactive one-step query set-up for a selection of attributes which are considered to be most important (see Fig. 4). Due to procedural modular internal programming, the appearance of the query site can be altered easily. Some multiple attribute lists are dynamically generated and form an up-to-date and distinct selection of attribute values, enabling more precise searching,

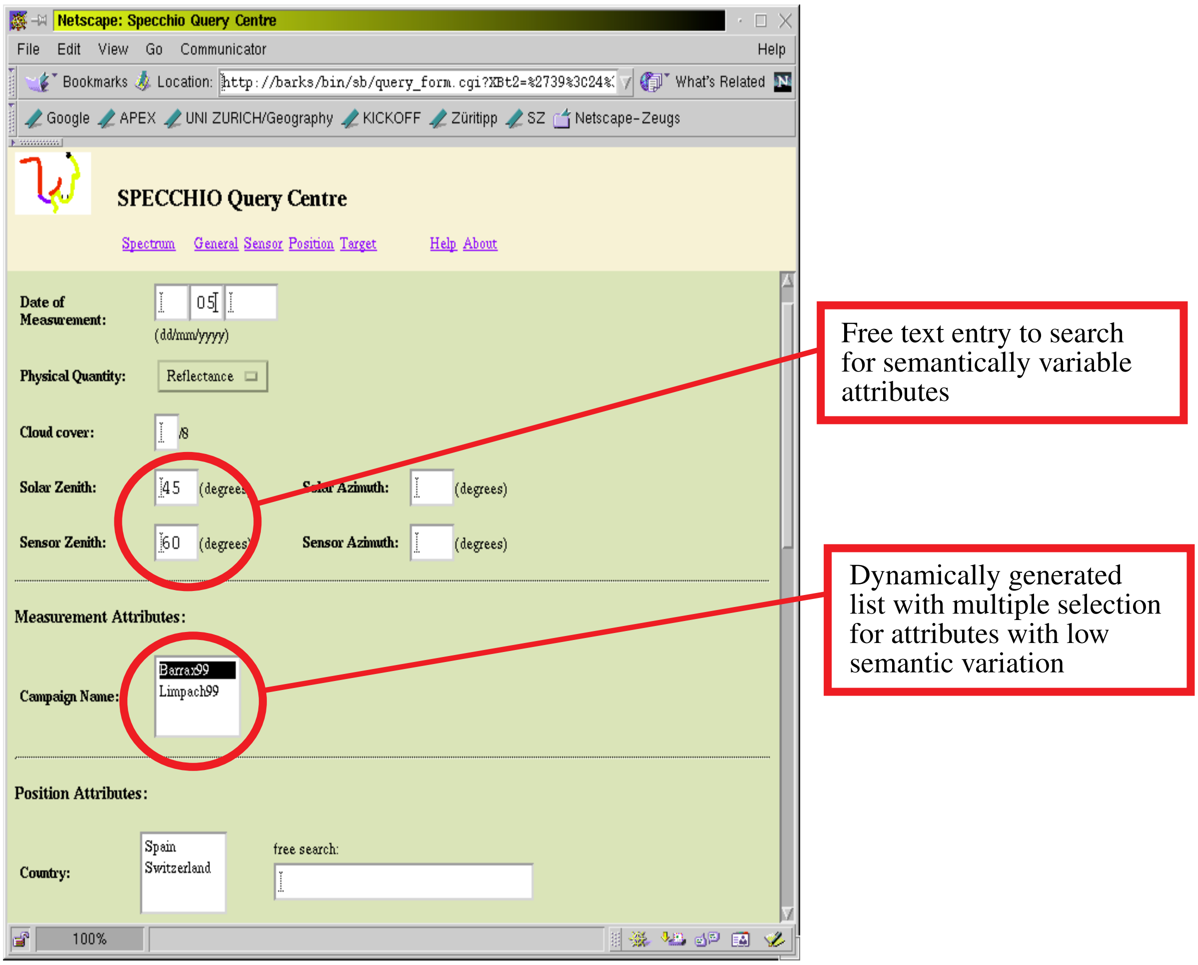
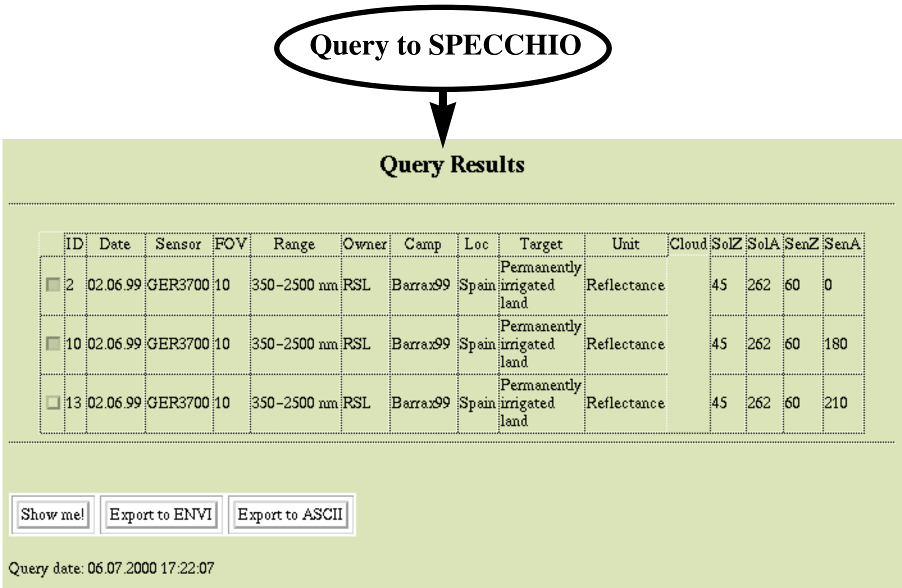


Figure 4: Query Web site with different attribute input modes.

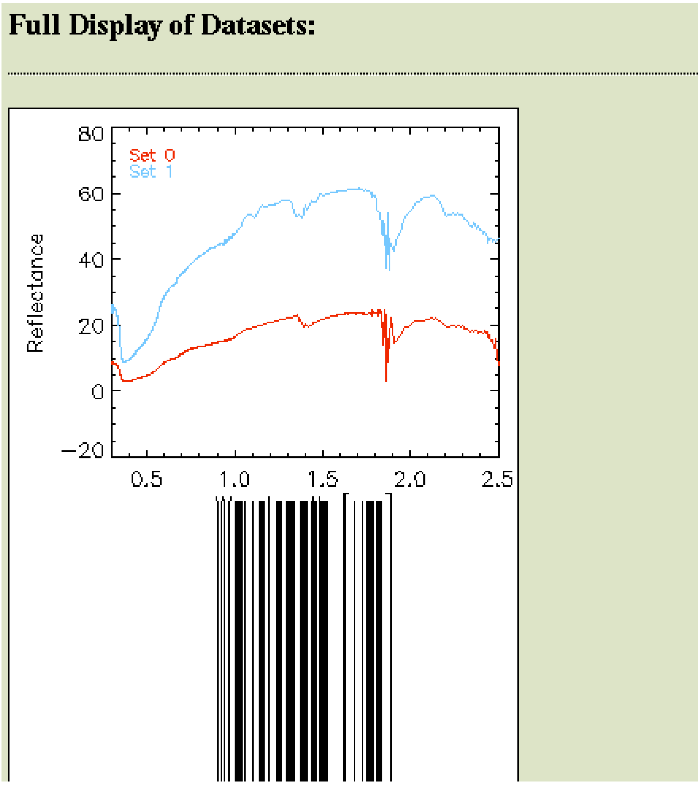
as opposed to free text entries. For both query interfaces, a specification of different attributes is logically equivalent to a Boolean AND operator, a multiple statement within a single attribute corresponds to an OR operator. If the Web site query yields datasets that match the search criteria, a clickable table shows up with columns corresponding to the searchable attributes (cf. Fig. 5). It is then possible to select datasets for graphical display and full statement of all associated meta data. Spectral plots can be chosen as single curves or overplots of all the selected spectra. The display feature as well as the export of spectral files to a file system is currently enabled for ENVI spectral library and GER 3700 ASCII data formats.

5.3 Advanced features

For the moment, query and feed interfaces to the database are implemented on a prototype level. In many applications of hyperspectral data, a demand for information on spectral features or the statistics of spectral ensembles arises. Therefore, a low level of processing functionality is going to be added to SPECCHIO, aimed at the retrieval of spectral values for predefined spectral channels, or the calculation of statistical key values. The latter can as well be stored in the database to save processing time in subsequent applications. Accordingly, biophysical parameters, such as LAI, can be calculated for a selection of spectra and subsequently stored. SPECCHIO truly serves as a reference database, once the full range of statistical functions exists. Another planned feature is the extension of display options after the selection of a query result, which include scaling and resampling of spectra. Lastly, the semi-automated input and output of selected data formats is realized. The routines called by the database to perform calculations are written in IDL or, if time-critical, in C/C++.



	ID	Date	Sensor	FOV	Range	Owner	Camp	Loc	Target	Unit	Cloud	SolZ	SolA	SenZ	SenA
	2	02.06.99	GER3700	10	350-2500 nm	RSL	Barrax99	Spain	Permanently irrigated land	Reflectance		45	262	60	0
	10	02.06.99	GER3700	10	350-2500 nm	RSL	Barrax99	Spain	Permanently irrigated land	Reflectance		45	262	60	180
	13	02.06.99	GER3700	10	350-2500 nm	RSL	Barrax99	Spain	Permanently irrigated land	Reflectance		45	262	60	210



6 OUTLOOK

We have presented a prototype for a comprehensive administration tool for spectral remote sensing data. After conceptual and technological considerations, the database schema has been shown which, to our knowledge, comprehensively represents the field of spectral remote sensing. Large amounts of data with a great variety of