The APEX (Airborne Prism Experiment) Calibration Information System

Andreas Hueni, Karim Lenhard, Andreas Baumgartner, and Michael E. Schaepman

Abstract—The calibration of remote sensing instruments is a crucial step in the generation of products tied to international reference standards. Calibrating imaging spectrometers is particularly demanding due to the high number of spatio-spectral pixels and the according large amount of data acquired during calibration sequences. Storage of these data and associated metadata in an organised manner and the provision of efficient tools for the data analysis and quick, repeatable calibration coefficient generation with provenance information are key to the provision of traceable measurements. The APEX Calibration Information System is a multi-layered information technology solution comprising a database based on the entity-attribute-value (EAV) paradigm and software written in Java and Matlab providing data access, visualisation and processing, handling the data volumes over the expected lifetime of the system. While being developed in the context of APEX, the system is rather generic and may be adapted to other pushbroom-based imagers with little effort.

Index Terms—Relational Database, Imaging Spectroscopy, Sensor Calibration

I. INTRODUCTION

Remote sensing technologies have the potential of acquiring data with a spatial coverage, temporal resolution and continuity that allow the parameterisation of Earth System Science models at regional and global scales. Such remotely sensing data are referred to as Fundamental Climate Data Records (FCDRs). These basic data are subsequently transformed into end-user products describing essential climate variables (ECVs) by data assimilation [1]. Of the 44 ECVs identified in the GCOS Second Adequacy Report [2], a total of 25 are largely dependent on satellite observations, effectively rendering remote sensing instruments one of the most important means of data collection for Earth system sciences.

Of the multitude of available sensor systems, the family of imaging spectrometers exhibits a high potential for the retrieval of ECVs from all spheres of the climate system [3, 4]. While some spaceborne imaging spectrometers do exist [5, 6] or are planned [e.g. 4, 7-9], the majority of instruments [e.g. 10, 11-13] is currently deployed on airborne platforms [3].

The calibration of imaging spectroradiometers and their data is technically demanding due to the high number of spatial/spectral pixels, also hampered by the notion that spectroradiometer measurements are still considered as one of the least reliable of all physical measurements [14, 15]. Calibration is an essential and critical step before higher product generation can be achieved with the accuracies required for the successful parameterisation of climate models in order to reduce the uncertainties of predictions [16]. Imaging spectrometers generally measure at-sensor radiances by reference to SI through either laboratory or vicarious calibration. System calibration allows defining a traceability of measurements to the SI standard and hence enables the comparison of data stemming from different sensor systems. Such traceable calibration forms the basis for the generation of consistent geophysical and biophysical products [17]. These dependencies may be conceptualised utilising an adapted DIKW (Data – Information – Knowledge – Wisdom) [18, 19] representation (Figure 1). DIKW is a model describing the building of knowledge from information based on facts or data and hence has found use in multi-disciplinary research, ranging from philosophy to systems analysis [20, 21]. Various flavours of the DIKW do exist [22], and for the purpose of this research, we add the notion of signals [23] while omitting the rather elusive tier of wisdom [24].

A. Hueni and M. E. Schaepman are with the Remote Sensing Laboratories, University of Zurich, Zurich 8057, Switzerland (e-mail: ahueni@geo.uzh.ch; schaep@geo.uzh.ch). K. Lenhard and A. Baumgartner are with the Remote Sensing Technology Institute, Oberpfaffenhofen 15236, Germany (e-mail: karim.lenhard@dlr.de; andreas.baumgartner@dlr.de).

Figure 1: DIKW applied to remote sensing system calibration and product generation
The lowest tier is formed by the physical standards, i.e. SI units, as provided by national metrology institutes. Here, instrumentation used in calibration laboratories may be calibrated and thus made traceable; these are the secondary standards, equivalent to tier two. This second tier produces optical stimuli leading to calibration sensor responses and information derived thereof. These data and information constitute the third tier and are held by a component we refer to as Calibration Information System (CAL IS). Calibrated flight data (tier four) are based on calibration information and are generated by Processing and Archiving Facilities (PAFs) [25]. Products are the output of higher-level algorithms and form the top of the pyramid, being equivalent to knowledge as it adds actionability to information [22]. We define a CAL IS as a system layer that stores raw sensor calibration and characterisation data and generates information describing the instrument’s electro-optical chain that converts signals from a continuous electromagnetic space into digital numbers within a discrete space. The CAL IS produces calibration coefficients used by the PAF to calibrate flight data and by this establishes the traceability link between airborne data and physical standards. The CAL IS holds information that leads to an enhanced understanding of the sensor properties and characteristics and as such supports the calibration scientists in developing their system knowledge. This paper defines the data sources generating raw calibration and characterisation data, lists the requirements for a CAL IS and documents the chosen implementation. The system is targeted at the APEX (Airborne Prism Experiment) system, but the general concept essentially applies to any frame based imaging system.

II. DATA SOURCES

A. APEX

ESA’s Airborne Imaging Spectrometer APEX (Airborne Prism Experiment) was developed under the PRODEX (PRogramme de Développement d’EXpériences scientifiques) program by a Swiss-Belgian consortium with the concept phase starting in 1998 and leading to a first test flight in 2008. APEX was formally accepted by ESA at the end of 2010 and entered the exploitation phase in 2011. It features up to 532 spectral bands in full spectral mode, ranging from 375-2500nm. Spectral programmability of the VNIR sensor enables achieving higher Signal-to-Noise-Ratios (SNR) by reducing the number of bands in a binned configuration. Data are acquired in 1000 pixels across track with a field of view (FOV) of 28°, resulting in ground pixel sizes of 1.5-2.5m at typical flight levels of 3000-5000m above ground. The main components of the APEX system are: (a) an optical sub-unit (OSU) containing the optoelectronics, (b) a control and storage unit (CSU) comprising the instrument control computer, solid state devices for the storage of the data stream and a positioning system, and (c) a temperature control unit (TCU) responsible for the regulation of the OSU optical base plate temperature to a stabilised 20°C.

Image data are stored frame-wise, where a frame consists of a combined readout of VNIR and SWIR detectors. The storage size per frame depends on the chosen binning pattern. Frames in the default binning mode amount to storage sizes of 0.62MB and to 1.1MB in unbinned mode respectively. Metadata per frame contain information on instrument settings (e.g. integration time) and readings of various auxiliary sensors (pressure, temperature and voltages) mounted in the OSU. A total of 88 metaparameters are recorded for each frame by the CSU.

B. APEX Calibration

APEX Calibration refers to the calibration and characterisation of APEX in the Calibration Home Base (CHB) with the goal of collecting data allowing the radiometric, geometric and spectral calibration of the instrument [11]. These data essentially form the base for the estimation of calibration coefficients that are applied to imaging data during data calibration. Standard system calibration runs generate around 13GB of data, while special experiments on average double the standard data volume (Figure 2). CHB calibration missions take place once to twice a year, resulting in a total raw data size of around 300GB accumulated since 2007. Data are stored on the file system of the CSU in automatically generated hierarchical folder structures with a naming convention for both folders and files. Calibration settings are sent to the CHB equipment and these parameters are also added to the metadata file associated with each frame. The total number of metaparameters per frame generated during calibration depends on the calibration experiment, comprising APEX system parameters and CHB settings as provided by the CHB interface (Table 1).

![Figure 2: Average data volumes acquired during CHB missions per calibration experiment [GB]](image-url)
C. The Calibration Home Base

The Calibration Home Base was commissioned by ESA for the calibration of APEX [26]. However, the laboratory can also be used for other sensors since imaging spectrometers used in optical remote sensing often display similar properties. The CHB provides defined light sources in the typical wavelength range of these instruments: integrating spheres for the radiometric calibration, a monochromator assembly and a slit-collimator assembly in combination with a rotating mirror on a translation stage to generate optical stimuli for the spectral and geometric calibration respectively. Since the spectral and geometric calibration procedures can consist of several hundreds or thousands of short measurements with different settings of the involved light source, the CHB was set up with automation kept in mind. The software controlling the calibration procedures was designed based on the Master/Slave pattern. The Slave controls the CHB’s devices and light sources while the Master runs calibration procedures by requesting CHB settings from the Slave and managing the data acquisition of APEX. Generally speaking, the Master provides an interface to define measurement sequences with specific routines being written for each specific calibration experiment. The communication between Master and Slave follows a synchronous client/server model, in which the slave takes the role of the server and receives requests by the Master [27]. Each request for CHB settings generates exactly one reply, which indicates that the CHB has assumed the requested state and is ready for measurements. This reply contains all the metadata generated by the CHB. These metadata stem from the laboratory devices, e.g. specifying the wavelength to which the monochromator is set, and from an environmental sensor that supplies room temperature, atmospheric pressure and humidity.

The communication between Master and Slave takes place over a TCP/IP connection, allowing operating Master and Slave on separate computers. The data are exchanged in the form of XML (eXtensible Markup Language) files, which are human-readable and well suited for the small amount of data exchanged, which is on the order of 1KB per request/reply. An additional advantage of using XML files is that their content can be checked for consistency against an XSD (XML Schema Definition) file. This includes checks for the completeness of the parameters, their data type and their valid ranges.

III. REQUIREMENTS

APEX CAL IS requirements are mainly based on the goals of generating system calibration information and enhancing the knowledge about the system in general as well as on the expected data volumes over the nominal lifetime of the system.

A. Data Volumes

The CAL IS must be able to handle the estimated total data volume over the expected lifetime of the system. The data volume includes calibration data acquired during the operational stage, currently set at 10 years, plus data acquired during the system acceptance test phase. RAW data volumes range between 290-410 GB under two different scenarios: (a) per annum volumes remain identical to the current average, i.e. special experiments constitute half of the volume, and (b) special experiment volumes diminish exponentially over time, leading to per annum volumes mainly governed by the standard data calibration runs. These estimated volumes will roughly double when data are processed to level-1 in the CAL IS, i.e. leading to total sizes in the order of 0.6-0.8 TB.

B. Generic Frame Support

The number of spectral pixels of an APEX frame depends on the binning patterns applied to the VNIR detector. The CAL IS must be able to seamlessly handle frames of differing binning patterns, including the frame size independent storage and the generation of calibration coefficients for various binning patterns.

C. Flexible Metaparameters

A flexible handling of the number of metaparameters per frames is required as (a) the number of metaparameters is dependant on the particular calibration experiment, (b) the CAL IS can add new metaparameter generated from both metadata and frame data during the forming of information, and (c) upgrades in the CHB may lead to different or additional metaparameters over time.
D. Data Ingestion, Data Structuring and Quality Control

Data ingestion must be an automated process, retrieving frame and metadata from the files generated during calibration. Near real time data control during calibration missions requires loading data into the CAL IS at various points in time, i.e. allowing data ingestion with a delta data loading capability. The data hierarchy generated during the calibration campaign on the CSU reflects the experimental structure and provides an easy way for the users to interactively navigate through the wealth of calibration data. The CAL IS must retain such structures and replicate them within the database. Automated generation of quality flags is important for two reasons (a) detection of problems in near-real time during calibration runs, and (b) exclusion of unsuitable data from calibration coefficient generation processes. Examples include saturation detection or thresholds for system temperatures and pressures.

E. Support for processing levels

The concept of processing levels is identical to the one commonly implemented in airborne and space based instrument PAFs. The levels reflect various stages of processing in the system and allow the efficient generation of higher-level products without a complete reprocessing starting from raw sensor data. Storage of several processing levels allows the easy study of effects caused by transforming processes, helping the debugging of according algorithms. Table 2 lists the required processing levels of the APEX CAL IS.

F. Interactive data exploration

Developing a sound knowledge about the sensor system and controlling the generation of calibration coefficients requires the ability to graphically explore data in an interactive fashion. The dimensions to be explored are (a) the spatio-spectral pixel values of a frame, (b) the time domain as the system response changes due to modification of or noises in the external stimuli or due to system inherent drifts or noises, (c) the metadata space [28], and (d) combined metadata – spatio-spectral domains where the spectral response at any pixel may be mapped versus parameters of the metadata space.

G. Provenance

Provenance describes the origin and evolution of data [29]. This information is important for the APEX CAL IS as it allows tracing effects found at any processing stage to its original cause. Provenance data forms a graph consisting of data sources, data sinks and processes. Such a topology is also highly useful when the definition of uncertainty budgets is required as all the contributing sources of noise are essentially given by the provenance graph. An example is the level-0 to level-1 processing, i.e. the dark current correction: each level-1 frame must have an associated creating process description and links to both level-0 frame and dark current frame used for the correction.

IV. Concepts

A. Overall Architecture

The APEX CAL IS is designed as a multi-layered system (Figure 4).

A relational database management system (RDBMS) serves as storage solution, implemented by a MySQL server (Version 5.5). Data are stored in physical database tables within the RDBMS, using a mixture of traditional relational database model and of a meta-system architecture known as EAV (Entity-Attribute-Value) paradigm [30]. The EAV meta-layer is the representation of the metadata known to the APEX CAL IS by according entries in physical database tables. The database connection and representation layer is written in Java and handles all communications with the database, offering functionality for data insert, querying and deletion, essentially mapping the EAV information to an object oriented representation and representing frame data as matrices. The application layer holds routines for the analysis and processing of data, including the graphical representation. This layer is implemented in Matlab (Release 2010a) utilising Java components for the communication with the database and for some graphical data representations.
Figure 5: APEX CAL IS System Architecture and Dataflow

Figure 5 illustrates the dataflow and the overall system architecture. Frame and metadata files are transferred from the APEX CSU to a workstation by FTP. A pure Java based application is used to ingest these files into the APEX CAL IS database, which is hosted by a database server. Higher-level processing, visualisation and analysis are carried out in a Matlab environment, relying on Java components for database communication, i.e. on lower-level data services as implemented in the EAV database connection and representation layer. The illustrated setup reflects the most common one, but installations where a laptop takes the role of a database server and processing computer at the same time are also feasible, e.g. within the CHB where a direct feed into a remote database server may not be as performant as a locally hosted database instance. The centrally hosted database allows the simultaneous data access by several researchers.

B. Database Schema

The APEX CAL IS schema (Figure 6) is implementing the EAV paradigm, but uses some traditional relational modelling as well, as suggested by Dinu and Nadkarni [31]. The frame table represents the primary data, i.e. the entities. Most of the entries in this table are APEX frames, but as data are stored as binary large objects, data of other dimensionality may be stored as well. Frame data are serialised objects of matrix classes belonging to the UJMP package [32] and as such may assume dimensionalities between one and three, referring to single spectra, two dimensional frames and imaging cubes respectively.

Frames can be associated with multiple metaparameters, which in turn may be referenced by multiple frames. This is achieved by a cross-relational table (frame_x_value) and a value table holding the actual values. A tuple within the value table may assume the data types of integer, floating point, string, binary object or date/time, storing them in the applicable fields, i.e. adopting one possible representation of the values within an EAV schema [31]. Value tuples can refer to other value tuples by the way of the value_x_value cross-relational table. This is heavily used in the modelling of hierarchical folder structures in the system while the representation of these relations is part of the system software. Value tuples refer to both attribute and unit table entries. The APEX CAL IS handles attributes and units in a flexible way, allowing for values of a certain attribute to have differing units. To support the use of the EAV related Java classes in other projects where a more strict approach is needed, namely the SPECCHIO spectral database project [33], the option to define standard units and default storage fields was added to the attribute table.

Provenance is modelled by a provenance table, representing instances of transformations. A transformation comprises a processing module of a certain version, stored in the process table, and a number of input and output frames, cross-linked via the provenance_x_frame table and the input/output node type given by entries in the node_type table.

Data integrity is ensured by foreign keys and according constraints in the database schema.

Figure 6: Entity Relationship Diagram of the APEX CAL IS schema

C. Data Insert

Data insert is dealing with the ingestion of data stored on the file system and with the insert of processing output. Data loading from the file system is utilising concepts developed in the SPECCHIO project [33]. It is based on the assumption that data are organised by campaign on the file system, each campaign folder forming the top of a hierarchical file/folder structure. Data are ingested into the database campaign by loader processes that parse the campaign folder structure, replicate the hierarchy in the database and insert all spectral files. Data loaders are aware of the existing content of the database and thus will only insert new data found during the parsing. This functionality allows the continuous update of a campaign while data are being collected in the CHB and is referred to as the delta loading capability.

Metadata are highly redundant between frames, especially within the same calibration experiments and a redundancy minimisation is required to reduce the number of inserted metadata parameters per frame and improve the query responsiveness of the system. This is accomplished by retaining a dynamic list of already inserted metaparameters in the loader processes. In case a metaparameter matches an existing entry in the list, only a reference is inserted into the frame_x_value table.
Metadata inserts are carried out as bulk inserts, i.e. all rows that are inserted for a frame into the values table and frame_x_values table respectively are combined into one SQL statement, thus minimising the database statement overhead and leading to an optimised loading speed. Attributes and units defined in the EAV meta-layer are automatically updated when new entities are encountered during insert procedures, i.e. the EAV layer is built on the fly while files are ingested or higher-level processing generates new attributes.

Frames generated by processing existing frames in the database, e.g. dark current correction, are inserted into the database by adding a new row in the frames table and linking with existing metadata of the input frames, thus avoid data redundancy. Metaparameters that are not applying to the higher-level frames are removed or updated, thus either omitting a link between new frames and original metaparameters or by adding a new metaparameters with updated values.

D. Data Retrieval

Generally speaking, all data retrieval is based on metadata subspace projections, i.e. frames are selected by defining metaparameter restrictions and the frames complying with these are contained within a subspace [28]. In practice, two variations for the definition of such projections exist: (a) selection via browsing of the data hierarchy where frames are identified by either their filenames or their containing directories, and (b) the programmatic definition of SQL queries that convert the restrictions to actual statements. For many instances of data processing or analysis, data selection involves both methods by first selecting a set of frames in the interactive browser and then projecting that set to a subspace by additional restrictions. Figure 7 illustrates such a combined use on the example of the radiometric calibration in the form of a sequence diagram.

The EAV database connection and representation layer offers a number of methods to select data, refine subsets and group data by multiple attributes. The following table lists some of the main methods/classes and examples of their practical application.

E. Data Processing

Data processing relates to the transformation of data, either for analysis purposes or for the generation of calibration products. Data volumes are rather big and efficient procedures are required to select and load data from the database and insert possible results. In this respect, the number of database statements must be minimised while ensuring that the memory allocation is sufficient to hold the data to be processed. To meet these needs, data processors are utilising the optimised methods offered by the EAV database connection and representation layer to make use of tuned functions, such as the data bulk-loading feature. Data that may be used multiple times during a processing run are ideally cached, such as dark current frames applied in the dark current correction procedure.

A further strategy is the partitioning of larger datasets during processing by loading only subsets into memory. This division into data collections is e.g. applied when processing spectral calibration data, where a full calibration dataset may be several GB in size.
Figure 7: Sequence diagram illustrating data selection, loading and product/provenance inserts for the radiometric calibration

Data processors written in Java are subclasses of a Data Processor class, which implements the support of provenance generation. Provenance data are compiled during processing, adding timestamps and input/output frames for each atomic operation. These accumulated provenance data are inserted into the database as one statement once the processor finished, thus minimising the database communication overhead. Processors written in Matlab can use an instance of the Data Processor class to handle the provenance generation in a seamless manner (see Figure 7).

F. Data Representation

The graphical representation to the user is key to the efficient handling of these multidimensional data. Developing a graphical user interface (GUI) in Matlab while using Java Swing components allows for the seamless integration of functionality offered by the EAV database connection and representation layer with the advanced plotting tools of Matlab. A Java key component is the hierarchical data browser that graphically represents the structure of the data as stored in the database as recursive attribute-value entries. The integration of Swing components in the Matlab GUI is accomplished by the JControl package [34].

Frame data within the Matlab environment are represented as matrices. These are populated by first loading the frames from the database into Java where they are de-serialised and exist as UJMP instances. In a second step, data are transferred into Matlab matrices by using a UJMP to standard Java double array conversion.

V. RESULTS

This section demonstrates the capabilities of the system by documenting the loading speed for data retrievals and the graphical data representation on the example of the main application user interface. A detailed description of the individual calibration and analysis modules is beyond the scope of this article and will be treated in dedicated, future publications. Readers interested in a practical usage example of the APEX CAL IS are referred to the case study presented in section VI.

Data loading speed was tested by loading frame collections selected in the spectral data browser and ranging between 1-600 items into Matlab, using three different setups to give indicative speeds for the most likely configurations: (a) database server and application running on the same machine, i.e. localhost, (b) database server hosted by a machine in the same Ethernet network as the workstation running the application, and (c) database hosted by a server at our APEX partner institution VITO and the application running on a workstation at the University of Zurich, with a network connection established using VPN tunnelling. The latter setup being the one used for shared database access for both the operations and science teams of the APEX project. At the time of the testing, calibration data for the years 2010-2012 was loaded into the system, consisting of 190'000 frames and 10.2 million metaparameter entries. Figure 8 shows the resulting loading speeds for the three setups: the total loading time refers to the time needed to load frames from the database into Matlab (top-left), the Java loading time is equivalent to the time spent in the EAV database connection and representation layer to load the frame data from the database into the memory of the workstation (top-right), the total loading time per frame is the amount of time required to load one frame into Matlab under the scenario
of different collection sizes (bottom-left), and the Java loading time per frame is the time spent within Java per frame for the different collection sizes (bottom-right).

The results show clearly that the loading speed is a function of the number frames and largely governed by the time spent in the EAV database connection and representation layer, loading the frame data from the database into Java allocated memory. The loading times are also governed by the type of database connection with the localhost being fastest as it the connection uses a Unix socket file, while the most overhead and delays occur for the tunnelled connection to a server in a different physical network.

The loading times per frame are a function of both the number of frames and the database server hosting location. Data loading involves a certain overhead such as the sending of statements and the compilation of metadata on the frame sizes. Hence smaller frame collections show a notable overhead per frame, which gets minimised as collection sizes increase. Actual speeds of the system can vary as they are influenced by the database server configurations such as the memory allocated for the caching of query results, the overall network traffic, the number of other processes running on the application workstation, the amount of RAM available and the fragmentation of the current free memory, which impacts the speed of memory allocation within Matlab. Hence, the shown loading speeds hold only true when enough RAM to hold the whole virtual cube is available and the performance will drop significantly when the operating system is forced to use virtual memory.

The data sizes involved in the loading speed test are shown in Figure 9, illustrating the increase in data volume as frames represented by 16 bit integer matrices in Java are moved into Matlab matrices of 64bit floating point data type. Frame collection sizes bigger than 150 frames are very rare under usual system usage, with the most common number of frames being loaded ranging between 1-50, i.e. the loading speed remains in an agreeable range for the users.

Figure 10 shows the main interface to APEX CAL IS written in Matlab. The Spectral Data Browser Java component is featured on the left of the window, showing the data held by the database organised by campaigns and the directory structure below each campaign reflecting the original storage on disc. Selected data are visualised in the four displays to the right of the data browser, showing a frame view with 1000 pixels across track and spectral pixels depending on the binning pattern (top left), a spectral profile of the selected spatial position (top right), an across track profile for the selected spectral position (bottom left), and an along track profile for the selected across track and spectral position (bottom right). Positions within this virtual cube are selected with three Java based scrollbars placed at the edges of the frame display and along track display respectively. Java radio controls (bottom left) allow focusing the display on VNIR, SWIR or both detectors. Two smaller data displays (middle right) show the number of saturated pixels per frame of the virtual cube split into SWIR and VNIR detectors, basing on saturation data compiled during initial data loading. All further functions of the APEX CAL IS, such as calibration and analysis functions are accessed via menu entries provided by the main window.
Figure 9: Frame data volumes when loading frames into Java and Matlab

The virtual cube shown in Figure 10 comprises 150 level-1 radiometric calibration frames acquired with 5 different radiance settings on a small integrating sphere illuminating the centre of the field of view. The intensity steps can be easily discerned in the along-track profile (bottom-right plot). The two last intensities exhibit saturation in both detectors due to a too high integration time for these radiance levels. These saturations are indicated in the GUI by the red bars in the two smaller displays (middle right).

Figure 10: Matlab APEX CAL IS main interface

VI. CASE STUDY

This section exemplifies the practical application of the APEX CAL IS by alluding to the spectral system calibration, which is rather data intensive as well as particularly interesting from an algorithmic point of view. Goal of the spectral calibration is the definition of centre wavelength and FWHM (Full width at half maximum) parameters for all spatio-spectral pixels of the system’s detectors. The spectral calibration module must primarily deal with three problems: (a) processing of level-1 frames, amounting to 5.8GB and 5.3GB of standard spectral calibration data for VNIR and SWIR sensors respectively, (b) extracting spectral response functions (SRFs) from these data at the sampled points and providing values for all pixels by suitable inter/extrapolation algorithms, and (c) enabling the user to interactively inspect each stage of the calibration layer generation. The spectral calibration is implemented generically and operates automated and independent of the selected spatio-spectral sampling pattern, spectral binning pattern and chosen monochromator sampling step size by relying on the detailed metadata, minutely describing the acquisition and calibration instrument settings.

The overall processing time is largely governed by the loading of the data into the main memory of the client machine and the generation of spectral calibration layers requires around a quarter of an hour per detector, which is acceptable as operationally only performed once to twice a year. Data loading is partitioned based on the sampling pattern and thus only a subset of the calibration data is loaded into memory at a time. Once extracted, the actually required data vectors and associated metadata may be saved in intermediate files, typically taking a mere 300-500KB of memory per detector.

Figure 11 presents the GUI of the spectral calibration module. The displays show the following interactive information, essentially representing the extracted data for the VNIR detector (from left to right and top to bottom): (a) DN response of a selected, illuminated pixel plotted versus the changing monochromator wavelength, (b) Gaussian curve
fitted to the data points, used for the determination of centre wavelength and FWHM, (c) spatio-spectral sampling pattern with currently visualised sampling point indicated by red crosshairs, (d) centre wavelength across-track profiles, i.e. equivalent to the spectral misregistration, for the measured spatial pixels at the currently selected spectral position, (e) extracted centre wavelengths for selected spatial position, (f) first and second order statistics of centre wavelength across-track profiles, (g) inter/extrapolating curve fitted to centre wavelengths for the selected spatial position, and (h-k) similar data as in (d-g) but for the FWHM parameter.

Centre wavelength and FWHM values are approximated for the whole detector by applying a spatial interpolation to the data points already interpolated in spectral dimension. These final layers are compiled into spectral calibration cubes per detector and stored in the database as calibration products, annotated with metadata describing the parameterisation of the calibration module as well as time and date of data acquisition and layer generation. At this point, it is readily available to the CAL IS to parameterise operations such as spectral convolution, required during the radiometric calibration of the system.

Table 2: APEX CAL IS processing levels

<table>
<thead>
<tr>
<th>Designator</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level-0</td>
<td>RAW</td>
<td>Raw frame data as generated by APEX</td>
</tr>
<tr>
<td>Level-1</td>
<td>DC Corrected</td>
<td>Frame data corrected for the dark current, taken from the closest dark current frame in the acquisition time line</td>
</tr>
<tr>
<td>Level-2</td>
<td>Desmeared</td>
<td>Frame data corrected for image smear caused by the readout mechanism (applies only to the VNIR part of the frame)</td>
</tr>
<tr>
<td>Level-3</td>
<td>Intermediate System Coefficients</td>
<td>Coefficients describing the electro-optical chain, but not yet integrated for all spatio-spectral pixels, e.g. gains/offsets for the nadir pixels.</td>
</tr>
<tr>
<td>Level-4</td>
<td>System Calibration Layers</td>
<td>Coefficients for each spatio-spectral pixel, i.e. a matrix of the same dimension as the imaging</td>
</tr>
</tbody>
</table>

Table 1: Number of metaparameters per frame for the most common calibration experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Metaparameters per frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute radiometric calibration</td>
<td>157</td>
</tr>
<tr>
<td>Relative radiometric calibration (flat fielding)</td>
<td>158</td>
</tr>
<tr>
<td>Across-track geometric calibration</td>
<td>204</td>
</tr>
<tr>
<td>Along-track geometric calibration</td>
<td>204</td>
</tr>
</tbody>
</table>

Figure 11: Interface of the APEX CAL IS Spectral Calibration module, showing VNIR data
Table 3: Main methods/classes for data retrieval, loading and grouping

<table>
<thead>
<tr>
<th>Method/Class</th>
<th>Use/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_list_of_metaparameter_vals</td>
<td>Return a list of values for a set of frames. E.g. list of integration time</td>
</tr>
<tr>
<td>frames, attribute</td>
<td>(frames, attribute)</td>
</tr>
<tr>
<td>get_eav_ids(frame, attributes)</td>
<td>Return all eav_ids of a frame within an attribute set. Used to identify</td>
</tr>
<tr>
<td>frames, attribute</td>
<td>spectral calibrations to apply in convolution radicometric operations.</td>
</tr>
<tr>
<td>get_eav_ids(frames, attribute)</td>
<td>Get metadata subset of frames. E.g. get EAV IDs holding the neutral density</td>
</tr>
<tr>
<td>frames, attribute</td>
<td>filter value used in absolute calibration.</td>
</tr>
<tr>
<td>get_closest_product_frames(frame,</td>
<td>Returns frames that are products (level-3 data) and are closest on a time</td>
</tr>
<tr>
<td>product)</td>
<td>line to the supplied frame. E.g. identify spectral calibration frames by</td>
</tr>
<tr>
<td></td>
<td>neutral density filter values.</td>
</tr>
<tr>
<td>get_frames_by_attribute(frame,</td>
<td>Returns the frames of a set of frames that have an attribute with a</td>
</tr>
<tr>
<td>attribute, value)</td>
<td>value matching the supplied one. E.g. get EAV frames belonging to a</td>
</tr>
<tr>
<td></td>
<td>certain neutral density filter value. Used to identify spectral calibration</td>
</tr>
<tr>
<td></td>
<td>frames by neutral density filter value.</td>
</tr>
<tr>
<td>frame_bulk_reading</td>
<td>Read frames into memory using one database statement for speed reasons. Used</td>
</tr>
<tr>
<td></td>
<td>to build a dark current system calibration frames by neutral density filter</td>
</tr>
<tr>
<td>AV_MatchingListCollection</td>
<td>Class to group a set of frames by values that are similar or are close on</td>
</tr>
<tr>
<td></td>
<td>a time line to the supplied frame. E.g. used to group dark current frames</td>
</tr>
<tr>
<td></td>
<td>by integration time to build a dark current system calibration frames by</td>
</tr>
<tr>
<td></td>
<td>neutral density filter as required for radiometric calibration.</td>
</tr>
</tbody>
</table>

VII. DISCUSSION

In the following we will discuss the selected architectures, namely the database schema and the interface software organised as layers using two programming languages, as these initial choices during system design have an impact on the scalability, flexibility, speed of the final system as well as on the implementation effort.

The choice of a flexible data schema, mainly based on the EAV paradigm, over a traditional database schema is a critical one, as it allows a very flexible approach to the handling of metadata but on the other hand database performance may drop considerably as data sizes increase, the latter being one of the main criticisms of the EAV scheme. The flexibility of the system regarding frame related metadata could be proven, but careful analysis was needed when considering the modelling of the provenance. Including the provenance information into the EAV concept was initially considered, but the graph structure with processor and specific processing information would not easily fit into the EAV without a massive overhead of logic implemented in the data representation layer. Hence it was decided to add provenance as traditional relational structure, keeping the EAV representation as simple as possible. Data retrieval speed is a matter of the origin of data selection queries. Metadata are queried extremely fast when starting from the primary datasets (entity-centric operation), while selecting frames based on their metadata (attribute-centric operation) is more time consuming; this effect is in fact a well known property of EAV databases [31]. For the attribute-centric operation, a careful consideration of the SQL statements is required. Rewriting of queries during implementation to optimise them for the EAV case proved to be essential, sometimes improving the speed by a factor of 10 or more. It was also found that overly complex queries were more likely to be optimised for the EAV databases as EAV queries have intermediate metadata subsets of a frame. E.g. get EAV IDs holding the neutral density filter value used in absolute calibration.

In hindsight, choosing the APEX PAF resulted in a massive implementation effort in the connection and representation layer, thus easing and accelerating the data access in the application layer. This approach resulted in a massive implementation effort in the backend in order to develop the database project and in turn add the new generic EAV capabilities to it with little overhead involved. The integration of Java within Matlab proved to be a well working concept. Implementing the EAV database connection and representation layer in Java allowed the reuse of existing classes from the SPECCHIO spectral database project and in turn add the new generic EAV capabilities to it with little overhead involved. The integration of Java within Matlab was relatively flawless, given the good support of Java by Matlab and the combination of Java and Matlab graphical user interface elements with the help of the JControl package in particular, as the latter allowed the development of much more flexible user interfaces than pure
Matlab would have offered. The overhead involved in the loading of frame data represented by serialized Java class instances stored in the database to in-memory instances via database queries and deserialization and the subsequent transfer of data as matrices into Matlab turned out to be no hindrance as is indicated by the loading time results presented, essentially not compromising the required interactive data exploration for databases hosted within the same physical network. Only the loading speed of data from databases via VPN connections may be prohibitive of interactive exploration, mainly caused by the massive amount of data that need transferring for larger frame collections. For these instances, running the application remotely on the server may be the preferred option. The frame data representation in Matlab was chosen as the default 64 bit floating point to avoid any conversions during subsequent floating point computations, but a different Java to Matlab casting, e.g. to 16 bit integer for level-0 and level-1, could be easily added if an increase in the number of frames in memory would be critical. However, all analysis and calibration algorithms written to date have not met such limitations as data sub-setting is applied when frame collections grow too big.

VIII. CONCLUSIONS

The APEX CAL IS is an effective system for the generic storage of imaging spectrometer calibration frame data and associated metadata including provenance information. It provides a system layer for the database connection and data representation, allowing efficient data access for higher-level application programs, such as various calibration and analysis tools whose description is beyond the scope of this article. Main system components are a MySQL database with an EAV paradigm enabled schema, a system layer implemented in Java and interactive interfaces written in Matlab but using Java based graphical and system layer components. The design has been proven to cope very well with the expected amount of data and its introduction into the APEX data processing environment has resulted in a boost of sensor understanding, calibration to product cycle time, quality control and repeatability of calibration coefficient estimation. The rather generic nature of the system suggests that an adaptation to other pushbroom-based systems would be of little effort, requiring only the implementation of appropriate file reading routines for the data insert and the writing of higher-level routines for the specific calibration routines required by the target system.

ACKNOWLEDGMENT

We wish to acknowledge the financial support provided by the Swiss Space Office during system design and implementation and the continuing support by EURAMET EMRP ENV-04. The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

REFERENCES


Andreas Hueni received a B.Sc. degree in computer science from the University of Applied Science Brugg-Windisch, Switzerland, in 1997, a PGDip in geographic information systems and an MPhil(Sc) degree in earth science from Massey University, Palmerston North, New Zealand in 2005 and 2006 respectively, and a Ph.D. degree in geography from the University of Zurich (UZH), Zurich, Switzerland in 2011. He has worked on the APEX project since 2007 and is currently a postdoctoral researcher with the Remote Sensing Laboratories, University of Zurich, responsible for APEX sensor and data calibration. His research interests are the calibration of spectrometers and the design of combined database and software systems such as the spectral database SPECCHIO.
Karim Lenhard has obtained a diploma in physics from the University of Bonn, Germany in 2008 and is since then a researcher and Ph.D. student at the German Aerospace Center (DLR). At DLR, he is jointly responsible for operating and improving the APEX calibration home base, an optical laboratory for calibration and characterization of airborne hyperspectral imagers. His research interests are the spectral and radiometric calibration of hyperspectral instruments.

Andreas Baumgartner received a B.Eng. degree in mechatronics with focus on optical engineering in 2008 and an M.Eng. in electrical engineering and information technology in 2010 from the University of Applied Sciences Deggendorf (HDU), Germany. Since then he has worked at the German Aerospace Center (DLR) and is jointly responsible for the calibration home base of APEX. His research interests are improving the setup, methods and software for the spectral, geometric and radiometric calibration of imaging spectrometers.

Michael E. Schaepman (M’05–SM’07) received the M.Sc. degree and the Ph.D. degree in geography from the University of Zürich (UZH), Zürich, Switzerland, in 1993 and 1998, respectively. In 1999, he spent his postdoctoral time at the Optical Sciences Center, The University of Arizona, Tucson. In 2000, he was appointed Project Manager of the European Space Agency Airborne Prism Experiment spectrometer. In 2003, he accepted a position of Full Chair of geoinformation science and remote sensing at Wageningen University, Wageningen, The Netherlands. In 2009, he was appointed Full Chair of remote sensing at UZH, where he is currently heading the Remote Sensing Laboratories, Department of Geography. His interests are in computational Earth sciences using remote sensing and physical models, with particular focus on the land-atmosphere interface using imaging spectroscopy.