Publishing Scanned Plates Using DaCHS

Florian Rothmaier[1], Markus Demleitner[1], Joachim Krautter[2], Holger Mandel[2], Stephanie Schwemmer[2], Otmar Stahl[2]

[1] *Zentrum für Astronomie Heidelberg (ZAH), Astronomisches Rechen-Institut*
[2] *Zentrum für Astronomie Heidelberg (ZAH), Landessternwarte*

Abstract

Scanning plates is only the first step in preserving photographic plates. Publishing the scans is just as important. A comparatively easy way for proper, VO-compliant publication of a digitized plate archive is GAVO's Data Center Helper Suite (DaCHS). It implements all relevant VO protocols, in particular the Simple Image Access Protocol (SIAP), the Table Access Protocol (TAP), and the Observation Core data model.

Using the example of the Heidelberg Königstuhl Archives, we illustrate the publication of a large data set to the Virtual Observatory and the web with DaCHS and show how easily our software can be installed and used by other data providers to archive and publish their data collections.

Keywords: Astronomical databases, Digital plate archive, Astrometry, Virtual Observatory

1 Introduction

Since 2005, in cooperation with the German Astrophysical Virtual Observatory (GAVO), the unique collection of the Heidelberg Königstuhl Archives that encompasses more than 25,000 photographic plates dating from the late 19th century to our days has been digitized with funding provided by the Klaus Tschira Foundation.

The preservation of our photographic archives may be considered a two-step process. While the scanning of the plates ensures preservation of information despite ongoing oxidative deterioration of the photo emulsions, the subsequent archiving of the plate data in the GAVO Data Center [1] and its publication to the Virtual Observatory (VO) and the web ensure the persistence of the archives and the standardized access to them.

Data published following VO standards can be discovered by in-client standard interfaces, can be used with standard clients and without being forced to learn how to operate custom web services, and has standard metadata helping later scientific exploitation.

This contribution introduces the central components of the Data Center Helper Suite (DaCHS) and illustrates – by using the example of the Heidelberg Königstuhl archives – how this piece of software can help data providers to publish their data sets in a VO-compliant manner.

2 The DaCHS Multi-Protocol VO Server

In the course of the development of the Virtual Observatory, several key technologies have been defined aiding the dissemination of scanned plates, including the Simple Image Access Protocol (SIAP) [2] that allows image discoveries based on spatial (and serveral other) constraints, and the Table Access Protocol (TAP) [3], a service protocol for accessing general table data. For observational data, TAP is combined with a data model describing generic observations called ObsCore, yielding a powerful system letting users formulate very expressive search constraints and execute them on all compliant systems with one operation. The combination of TAP and ObsCore is referred to as ObsTAP [4].

Moreover, as a crucial part of the publication infrastructure DaCHS' ingestion component is equipped with a great number of different grammar modules ready to parse FITS images and VOTables, different types of text files and many other sources.

Finally, DaCHS contains the Stan templating system of Python's web development framework Nevow [5] allowing for the publication of HTML form-based services and documentation pages.

2.1 Resource Descriptor

The central concept of DaCHS' publication infrastructure is an XML file referred to as a Resource Descriptor (RD). Typically, all information on a data collection and the related services is collected in a single RD which imports procedures, informational and administrative web pages, and generates the actual services together with their metadata documents.

3 Steps to Publishing a Scanned Plate Archive

As a fairly typical example of a scanning project, we take the Heidelberg Königstuhl archives to illustrate the individual steps to take to get a data collection published. The finished service is called HDAP (Heidelberg Digitized Astronomical Plates).

3.1 Installing DaCHS

The preferred way to run DaCHS is on Debian or compatible systems on which the software can be easily installed from an APT repository. To use GAVO's repository, the line

```
deb http://vo.ari.uni-heidelberg.de/debian stable main
```

has to be added to the file */etc/apt/sources.list*. After updating the package cache, e.g., on the command-line via `sudo apt-get update`, DaCHS can be installed by saying `sudo apt-get install gavodachs`. All the package dependencies will be handled in an automatic way.

3.2 Metadata Handling

Our starting point is defined by a set of digitized photographic plates stored in the commonly used FITS (Flexible Image Transport System) format with minimal headers (basically, NAXISn and BITPIX).

The observation journals are digitized into a plate database containing, e.g., the observer, the observation time and so on. Additional metadata, in particular information about the instrument at which the image was taken or the photo emulsion used, are inferred based on the plate identifier. This metadata set is supplemented by the astrometric plate solution as obtained by the program SExtractor [6] that writes the objects found on a photo plate in (x, y) to a catalogue, and by the Astrometry.net tool [7] which tries to do the assignment $(x, y) \rightarrow (\alpha, \delta)$.

This process is governed by about 500 lines of python building on the processing subsystem of DaCHS, which includes code for traversing the source collection, maintaining state, and manipulating FITS headers. Most of the custom code is encoding rules for handling metadata inference for the ten different instruments that went into HDAP.

3.3 Writing an RD

In DaCHS, publishing a data collection to one or several VO protocols or a web page means writing a Resource Descriptor. In the following, we give excerpts that contain the principal elements of the HDAP Resource Descriptor. A more detailed explanation how to write such an XML file can be found in [8].

An RD starts with the root element `<resource>` followed by some pieces of meta information on the data collection, here:

```
<resource resdir="lswscans" schema="lsw">
    <meta name="creationDate">2007-11-10T12:00:00Z</meta>
    <meta name="description">Scans of plates obtained at Landessternwarte
        Heidelberg-Königstuhl, its predecessors, as well as the
        German-Spanish Astronomical Center (Calar  Alto Observatory), Spain,
        1880 through 1999.</meta>
    <meta name="title">HDAP -- Heidelberg Digitized Astronomical Plates</meta>
```

```
    ...
</resource>
```

Declaring metadata is essential for later registration of services and data. Furthermore, it is also required to generate informational pages.

Conventionally, the next items in the RD are the table definitions:

```
<table id="plates" onDisk="True" adql="True" mixin="//siap#pgs">
    <mixin collectionName="'HDAP'"
        targetName="object"
        expTime="exposure"
        tMin="(to_char(startTime, 'J')::double precision-2400000.5)"
        tMax="(to_char(endTime, 'J')::double precision-2400000.5)"
    >//obscore#publishSIAP</mixin>
    <column name="exposure"
        tablehead="Exp. time" unit="s" ucd="time.duration;obs.exposure"
        description="Effective exposure time" verbLevel="15"/>
    ...
</table>
```

The `<column>` elements set the fields of the database table. A more complicated feature are the `<mixin>` elements, a technique used by DaCHS to ensure a certain functionality on a table. In this case, there are two mixins, the first one to support the requirements for an SIAP service, the second one to endow the table with everything needed for ObsTAP. While SIAP allows to search HDAP for images matching a certain region of the sky, ObsTAP also enables a user to browse the archive for uncalibrated plates or non-stationary objects like comets.

The ingestion component of DaCHS is controlled using data elements:

```
<data id="import" updating="True">
    <sources recurse="True">
        <pattern>data/part1/*.fits</pattern>
        <pattern>data/part2/*.fits</pattern>
    </sources>
    <fitsProdGrammar qnd="True" id="impGrammar">
        ...
    </fitsProdGrammar>
    <make table="plates" rowmaker="make_plates"/>
</data>
```

The input data sets are declared within the `<sources>` elements. The grammar chosen here returns FITS headers as dictionaries, i.e., a sequence of string-to-string mappings. These are then turned into proper database rows in `<rowmaker>` elements.

Finally, our RD has to define the services exposing the data:

```
<service id="siap" core="qhsiap" allowed="siap.xml">
    <publish render="siap.xml" sets="ivo_managed"/>
    <publish render="form" sets="ivo_managed" service="q"/>
    <meta name="shortName">hdap_siap</meta>
    ...
</service>
```

A service is a combination of a core, the element which performs the actual computations for the service, and one or more renderers setting the interface, in this case a web form or an SIAP interface. Service-specific metadata may override any piece of global metadata given in the beginning of the Resource Descriptor.

3.4 Data Access

In addition to the VO protocols, HDAP is also available through two form-based web services, also driven by DaCHS. While[9] allows for retrieving images based on positions and thus can only be used to find astrometrically calibrated plates, [10] also gives access to uncalibrated images and is the appropriate tool for object name-based searches.

The popular Virtual Observatory application TOPCAT [11] has both a built-in SIAP and a TAP client. Our archives can be accessed via SIAP by the URL

```
http://dc.zah.uni-heidelberg.de/lswscans/res/positions/siap/siap.xml?
```

and by TAP giving the URL

```
http://dc.zah.uni-heidelberg.de/tap
```

to TOPCAT's TAP client. The table name is `lsw.plates`.

Like any tool speaking SIAP, Aladin 12 can be used to visualize the Königstuhl archives. Catalog data or images from other sources can be overlaid on the photographic plates endowed with an astrometric calibration.

4 Conclusion

DaCHS is in operation in various data centers all over the world, with services serving everything from giga-record-size catalogs to collections of historical astrophotographic plates. Due to its easy installability and its powerful publication infrastructure supporting to Virtual Observatory standards, DaCHS provides a reasonably smooth path for publishing scanned plates. Technical support for doing so is – within reasonable limits – available from the authors.

5 Acknowledgements

The Heidelberg Digitized Astronomical Plates have been produced at Landessternwarte Heidelberg-Königstuhl under grant No. 00.071.2005 of the Klaus Tschira Foundation. The development of DaCHS is supported under BMBF grant 05A08VHA.

References

[1] The GAVO Data Center, http://dc.zah.uni-heidelberg.de/

[2] Tody, D.; Plante, R. (2009), Simple Image Access Specification, IVOA Standards, http://www.ivoa.net/documents/SIA/20091116/

[3] Dowler, P.; Rixon, G.; Tody, D. (2010), Table Access Protocol Version 1.0, IVOA Standards, http://www.ivoa.net/documents/TAP/

[4] Louys, M. et al. (2011), Observation Data Model Core Components and its Implementation in the Table Access Protocol, IVOA Standards, http://www.ivoa.net/documents/ObsCore/index.html

[5] Nevow, a web templating framework, https://launchpad.net/nevow

[6] SExtractor – Astromatic.net, http://www.astromatic.net/software/sextractor

[7] Astrometry.net, http://astrometry.net/

[8] Demleitner, M. (2013), The DaCHS tutorial, http://docs.g-vo.org/DaCHS/tutorial.pdf

[9] HDAP – Heidelberg Digitized Astronomical Plates, http://dc.zah.uni-heidelberg.de/lswscans/res/positions/q/form

[10] Full plate access to Heidelberg Digitized Astronomical Plates, http://dc.zah.uni-heidelberg.de/lswscans/res/positions/fullplates/form

[11] TOPCAT (Tool for Operations on Catalogues and Tables), http://www.star.bris.ac.uk/ mbt/topcat/

[12] Aladin Sky Atlas, http://aladin.u-strasbg.fr