Exp Astron (2012) 34:65–88
DOI 10.1007/s10686-012-9299-9

ORIGINAL ARTICLE

TAPAS, a VO archive at the IRAM 30-m telescope

Stephane Leon · Victor Espigares · José Enrique Ruíz · Lourdes Verdes-Montenegro · Rainer Mauersberger · Walter Brunswig · Carsten Kramer · Juan de Dios Santander-Vela · Helmut Wiesemeyer

Received: 28 October 2011 / Accepted: 21 March 2012 / Published online: 26 April 2012

Abstract Astronomical observatories are today generating increasingly large volumes of data. For an efficient use of them, databases have been built following the standards proposed by the International Virtual Observatory Alliance (IVOA), providing a common protocol to query them and make them interoperable. The IRAM 30-m radio telescope, located in Sierra Nevada (Granada, Spain) is a millimeter wavelength telescope with a constantly renewed, extensive choice of instruments, and capable of covering the frequency range between 80 and 370 GHz. It is continuously producing a large amount of data thanks to the more than 200 scientific projects observed each year. The TAPAS archive at the IRAM 30-m telescope is aimed to provide public access to the headers describing the observations performed with the telescope, according to a defined data policy, making as well the technical data available to the IRAM staff members. A special emphasis has been made to

IRAM is supported by INSU/CNRS (France), MPG (Germany), and IGN (Spain).

S. Leon (✉) · R. Mauersberger
Joint ALMA Observatory/ESO, Av. Alonso de Córdova 3107,
Vitacura, Santiago, Chile
e-mail: sleon@alma.cl

S. Leon · V. Espigares · R. Mauersberger · W. Brunswig · C. Kramer · H. Wiesemeyer
Instituto de Radioastronomía Milimétrica (IRAM), Avenida Divina Pastora 7,
Local 20, 18012 Granada, Spain

J. E. Ruíz · L. Verdes-Montenegro · J. D. Santander-Vela
Instituto de Astrofísica de Andalucía, CSIC,
Apdo. 3004, 18080 Granada, Spain

H. Wiesemeyer
Max-Planck-Institute for Radio Astronomy,
Auf dem Hügel 69, 53121 Bonn, Germany

make it Virtual Observatory (VO) compliant, and to offer a VO compliant web interface allowing to make the information available to the scientific community. TAPAS is built using the Django Python framework on top of a relational MySQL database, and is fully integrated with the telescope control system. The TAPAS data model (DM) is based on the Radio Astronomical DAta Model for Single dish radio telescopes (RADAMS), to allow for easy integration into the VO infrastructure. A metadata modeling layer is used by the data-filler to allow an implementation free from assumptions about the control system and the underlying database. TAPAS and its public web interface (http://tapas.iram.es) provides a scalable system that can evolve with new instruments and observing modes. A meta description of the DM has been introduced in TAPAS in order to both avoid undesired coupling between the code and the DM and to provide a better management of the archive. A subset of the header data stored in TAPAS will be made available at the CDS.

Keywords Astronomical database · Virtual observatory · Radioastronomy

1 Introduction

Research in astronomy is producing an increasing amount of data thanks to new and improved instrumentation. In order to exploit those data in an optimum way it is necessary to digitally archive these data. Such an ambitious task is made possible by the use of large astronomical databases (DB) and the standardisation of the protocols through the creation of the Virtual Observatory (VO, e.g. Szalay and Gray [20]). The VO allows the distribution of heterogeneous sets of data in a standardized way. The first VO compliant archives were developed for optical data from ground facilities as well as satellite data (e.g. Simbad; Wenger et al. [24]). Very few VO compliant archives exist at radio wavelengths, apart from the Telescope Archive for Public Access System (TAPAS) project which is one of the first radio archives designed from the start to be VO compliant and described in this paper.

A key component in the implementation of an archive is the design of the Data Model (DM) which is an abstract model which describes how data are represented and accessed. Various DMs have been created in radio and for (sub-)millimeter wavelengths (e.g. the ALMA Science Data Model, ASDM, Viallefond [23]). This is the first single dish data model developed with VO compatibility at its core. so-called RADAMS (Radio Astronomical DAta Model for Single-dish telescopes; Santander-Vela et al. [17, 18]), to describe and to integrate radio observations in the VO.

This paper is structured as follows: in Section 2 we describe the main characteristics of the IRAM 30-m telescope, in Section 3 we give the content and the data policy of TAPAS, in Section 4 we describe the technical architecture of the TAPAS system, and its functionalities are explained in Section 5. Finally, a summary is given in Section 6.

2 The 30-m radiotelescope

The 30-m telescope at Pico Veleta in Sierra Nevada (Granada, Spain), is one of the two radio astronomy facilities operated by the Institute for Radio Astronomy at mm wavelengths (IRAM). It is currently one of the largest and most sensitive radio telescopes for millimeter observations. The 30-m telescope is equipped with a series of dual polarization receivers operating at $\lambda = 3, 2, 1$ and 0.8 mm, and with two cameras working at 1 mm: HERA with 9 pixels for molecular gas mapping and MAMBO, a camera with 117 pixels to map the dust emission. In December 2005 a New Control System (NCS) was installed. The NCS generates VOTables [15] in XML format during the observations. The NCS subsystems were developed in different languages appropriate to specific requirements, including the C language, Python, and FORTRAN95 for the observer's user interface "PaKo". For each observation requested by the user PaKo generates a detailed description in XML format following the VOTable, Version 1.0: 15-Apr-2002, standard. Recently, in March 2009, the A, B, C and D heterodyne receivers were replaced by the EMIR receiver. EMIR provides simultaneously a bandwidth of at least 4 GHz for two of the 3, 2, 1.3 and 0.9 mm atmospheric windows in two orthogonal linear polarizations.

3 Content and data policy of TAPAS

TAPAS is designed to record the metadata for all astronomical data taken at the IRAM 30-m antenna. In that context metadata means everything except the raw counts or intensities of the astronomical data, i.e. scan headers, meteorological data, ancillary information. The architecture also includes the possibility of linking the metadata of individual scans to the FITS [10] files actually containing the uncalibrated astronomical data. TAPAS provides a Virtual Observatory compliant facility for query and retrieval of header data.

For all projects a subset of the astronomical metadata become public with a delay of 12 months from the end of the scheduling semester in which the project was observed. Concerning the astronomical data themselves, these are in general not publicly available. For large programs, all data will be made available after a 18 months proprietary period

4 TAPAS architecture

4.1 Data model

From the beginning, VO-compliance was the most important requirement for TAPAS. Other existing radio astronomical archives were studied to see how they implemented VO protocols. Systems such as the Australia Telescope Compact Array (ATCA) archive [14] were exposing VO services only as tabular data, and relied on their existing, pre-VO data models. Other potential systems such as the ASDM include a great deal of complexity to cater for interferometry and contain VO-compatible interfaces rather than a full VO model.

Thus, we decided to create a data model for single-dish radio astronomical observations, flexible enough not to be instrument-specific, and using the existing drafts for the IVOA Observation Data Model (ObsDM) as a guideline. Therefore, the RADAMS is an actual implementation of the well defined parts of the ObsDM (such as observation "Characterisation"), and defines for the first time other ObsDM building blocks such as "Policy", "Packaging", and "Provenance".

In order to facilitate its implementation in very diverse instruments and telescopes, RADAMS is deliberately generic. Hence, an adaptation for the TAPAS project was needed. On the other hand, the New Control System (NCS), based on standards such as XML [5] files and VOTables, opens up the door to new extensions as the one we are discussing in this paper. Since the NCS lays out the perfect skeleton for the adaptation of RADAMS, the nomenclature and structure of subsystems inside the 30-m telescope can be combined with the RADAMS model in order to keep all the VO compliance it provides.

In the Relational Model for DB management, an entity is the basic object represented. Usually, an entity is a real world object, physical or conceptual, with an independent existence. Each entity has attributes, which are the particular properties that describe it and a particular entity will have a value for each of its attributes, that characterize it. The TAPAS Data Model consists of 50 highly connected entities, holding 58 relationships between them, and encompassing 278 attributes. It is implemented using MySQL [6] as the DB engine and InnoDB as the storage engine, which provides us with transaction support and real referential integrity. Those features will assure us the consistency and integrity of the data needed for this project because of the large amount of relations present in the data model. On top of the MySQL backend, in the application layer, is running Django [11], an open source web application framework, written in Python, which follows the model-view-controller (MVC) architectural pattern [9]. Django provides a rapid development framework together with a good set of architectural practices for the web interface (Section 5). But, as we will explain in Section 4.2, it is also worth mentioning its use as an Object-relational mapping tool (ORM; [1]), which is a mechanism to convert data between incompatible type systems in relational DB and object-oriented programming languages, i.e. MySQL and Python.

The Data Model structure mirrors the different telescope subsystems, establishing the corresponding relationships between them, and closely following the NCS attribute nomenclature. But it also characterizes the telescope instrumentation with a set of static tables, and includes a meta description, or equivalently a self-description, of the full model. The meta description provides additional information on the attributes such as Universal Content Descriptor (UCDs; [16]), human readable descriptions, physical units and ontology links. This meta information, together with the NCS nomenclature and Django ORM capabilities, were extremely helpful for the Data Filler stage, as explained in the next section.

The Data Model subsystems are listed in Table 1, while a detailed description of each subsystem is presented separately in Appendix due to the complexity of the Data Model.

4.2 Data filler

The TAPAS Data Model is a structured Entity–Relationship (E/R) model (Chen [4]) with 50 entities and 278 attributes. The NCS provides all the information to be entered into TAPAS, but fragmented and distributed between many sources. Although all the generated information is stored in XML documents, it is not necessarily in the same format or with the same semantics, `nor` in the same location. In fact, each subsystem provides its information using different semantics and none of the entities in the data model has all the associated information located in a single NCS XML document.

The data filler developed for the TAPAS project is a POSIX (Portable Operating System Interface for uniX, [12]) background process responsible for filling all the data model entities with the appropriate information in real time. During an observation at the IRAM 30-m telescope, each time a scan finishes, a full range of related metadata is generated by the control system, including information as diverse as receiver temperatures, inclinometer values or meteorological conditions. The data filler compiles all the needed metadata and stores them in the archive, in real time. Thus the development is tightly coupled with the telescope control system and the data model, source and target of the information, respectively.

A high degree of coupling implies that a program relies heavily on other modules, which can lead to serious weaknesses in the system. The most significant would be that changes in one module provokes a ripple effect of changes in the rest of modules. In other words, it is neither advisable nor convenient to modify data filler's source code each time the data model or the telescope control system changes, so high degrees of coupling should be avoided.

In order to prevent most of the coupling, we have tried to make that knowledge explicit, instead of implicit. For that reason, we will have the data model meta-describe itself (see Fig. 22). This meta description, then, will not only aggregate metadata such as human readable descriptions, physical units etc., but will also provide the information needed for the data filler to do its job, specifically:

- The attributes for each entity in the data model.
- The correspondence between a specific data model attribute and its source.
- The way to fetch the information for a specific attribute.

Thereby, the data filler relies only in the data model meta description, without needing any hard coded knowledge inside the source code, and hence avoiding high degrees of coupling.

4.2.1 Source information

All information from the NCS is available as XML documents. However, the data filler needs to parse and select very specific pieces from each XML file, and we want to be able to explicitly state those pieces outside of the data filler's code, in order to avoid implicit knowledge. Addressing specific parts inside a document is precisely the purpose of XPath [7], the XML Path Language, a query language to select specific elements, also called nodes, from an XML document. The hierarchical structure of an XML document forms a tree structure that starts with an element called "root" and branches to the lowest level of the tree. The root element is *the parent* of all other nodes in the tree and it is the only element in the tree that has no ancestor. Each node can also have sub nodes, called children. The XPath language provides the ability to navigate around the tree representation of the document, selecting nodes by a set of criteria, based on their relationships, attributes, text content, etc.

For example, the XPath expression

```
//RESOURCE[@name='source']
    /PARAM[@name='sourceName']/@value
```

selects the value of a `PARAM` node which has an attribute `name` equal to `sourceName`, and which is direct child of a `RESOURCE` node with an attribute `name` equal to `source`, irrespective of where in the XML document is found this `RESOURCE`. When applying that expression, for instance, to the document snippet shown in Fig. 1, the result of the expression would be `W3OH`.

We can think of an XPath expression as a blend between a path notation and a powerful regular expression, while taking full advantage of the document structure. The XPath language has even some advanced capabilities like XPath axis, to define a node-set relative to the current node, or XPath operators, functions that can be used inside the expression to compute some value from the nodes. For TAPAS these advanced features are used for complex XML documents whose structure makes it hard to extract some values, or for XML tables as a way to avoid hardcoding indexes into the expressions to parse them.

Therefore we can fetch all the E/R model attributes just parsing the corresponding XPath expression. All the expressions for the more than 250 attributes are kept in the meta description of the model.

4.2.2 Data filler implementation

We have already outlined some of the data filler implementation details in the previous sections, for example, the ORM and MVC features of Django. The data filler implementation takes full advantage of these capabilities to establish

Table 1 Data model subsystems

Name	# Entities	# Attributes	Description
Antenna	3	29	Pointing model, pointing and focus corrections and antenna positions.
Sources	3	33	Source catalog of the observation.
Software	2	6	Software versions used during the scan observation.
Scan	3	22	General information about the project and the scan (LST, # subscans, ...).
User profiles	4	15	User information, profiles and user generated content about the observations.
Switching settings	3	10	Switching settings for the observations (position switching, wobbler switching, ...).
Scan results	3	33	Calibration, focus and pointing results.
Backends	3	11	Backend settings.
Receivers	5	28	Receiver information (bolometer, heterodyne...).
Observation settings	8	35	Settings for the observation modes (tip, focus, pointing, calibration...).
Meteo station	2	12	Information from the meteorological station.
Metadata	2	10	Metadescription of the data model.

an abstraction layer between the data source, the data model in our case, and the logic. The ORM mechanism is able to address, access and manipulate objects within the chosen programming language without having to consider how those objects relate to their data sources. On the other hand, it also allows to treat each data model entity as a Python class, without having to worry about the DB behind nor having to use any SQL to retrieve or work with the data in it.

Fig. 1 Extract of the settings of a pointing observation

```
<RESOURCE name="source">
  <PARAM name="sourceName"
         value="W3OH"
         datatype="char">
  </PARAM>
  <PARAM name="basisSystem"
         value="equatorial"
         datatype="char">
  </PARAM>
  ...
</RESOURCE>
```

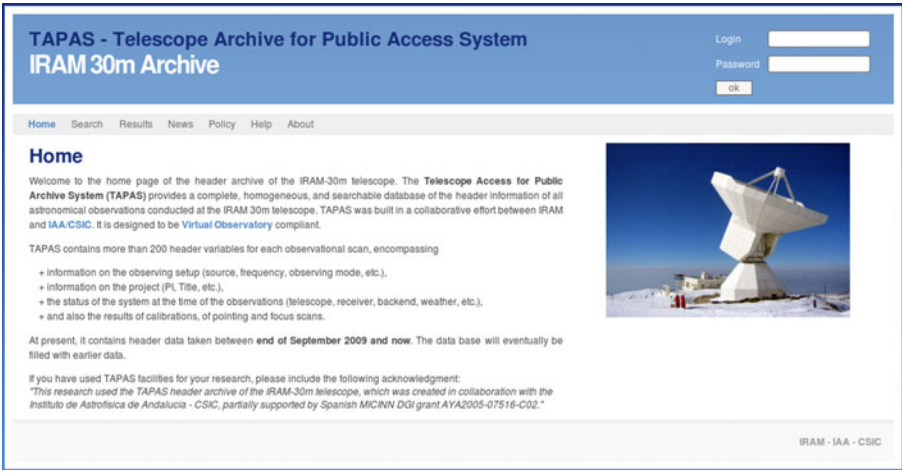

Fig. 2 Homepage for TAPAS web interface

Once we are talking about representing the entities as Object-Oriented classes and thanks to Python introspection and class inheritance, we can make each class responsible for its own data, by defining a superclass for all the entities with a set of common methods. In this manner, the data filler main task is to coordinate and to schedule the information retrieval for each class or entity, in order to satisfy the dependencies between them. But each class takes care of its own attributes, which from the point of view of cohesion is the best

Fig. 3 Search form for TAPAS web interface

we can achieve. Together with the meta description of the model, we achieve a high degree of cohesion, with a low degree of coupling.

4.2.3 Failure handling

Stability and robustness are key factors for a software of the type of the data filler. Excellent failure handling and recovery are mandatory, in order for the software to keep running under any circumstances. Besides the unavoidable risk of the software's own bugs, always present, numerous external variables can introduce errors in the system. TAPAS is an additional layer on top of an already highly complex system, so the assumption of everything working as expected cannot be taken as granted.

Several risk mitigating actions have been taken when developing the data filler, keeping always in mind that losing scan information is unacceptable for the final user. For example, besides the real time standard mode of working of the data filler, a stand alone mode is provided, where the data filler can work offline to process and insert scans from a folder hierarchy containing NCS XML documents. Thus, if anything in the data filler or in the other telescope subsystems fails, and the current observation cannot be processed in real time, we would be able to recover that information later. In case the data filler fails while processing a scan, it does not crash, but creates a report of the failure,

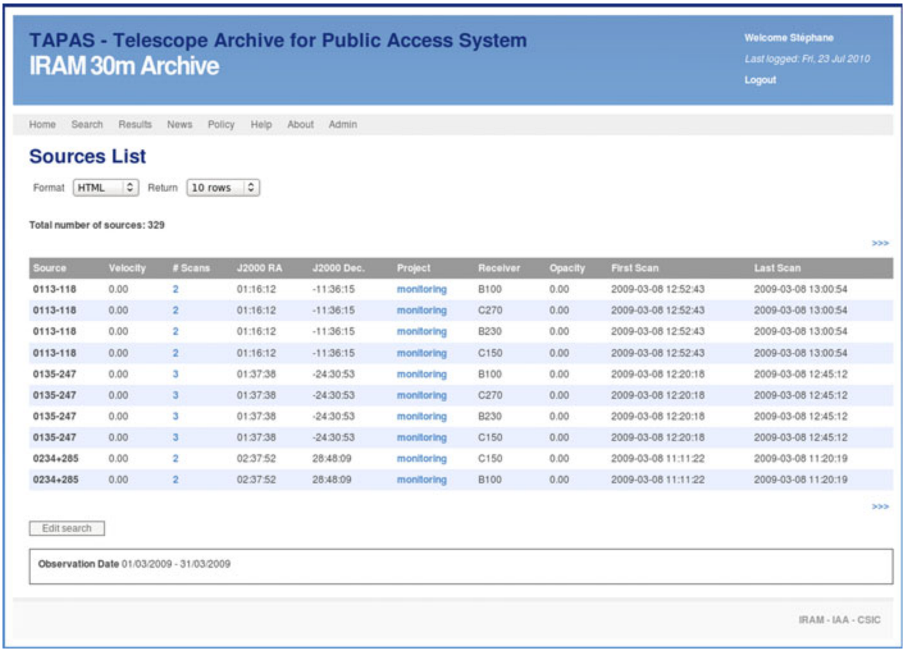

Fig. 4 Example of the first page provided by TAPAS web interface for the list of sources observed within different projects and instruments at the IRAM 30-m telescope, for a given period of time

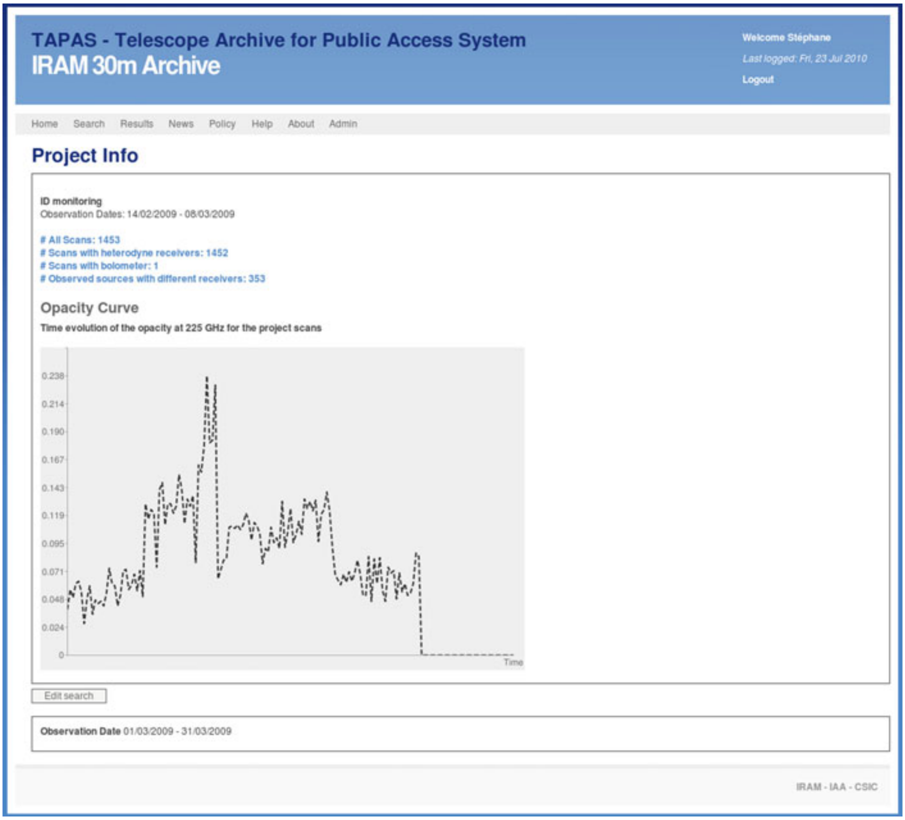

Fig. 5 Information page provided by TAPAS web interface for an observational project

skipping that scan and moving on to the next one. Later on that scan can be reinserted using the *offline* mode.

As previously introduced, each class is responsible for checking the consistency between its attributes in the data model and their meta description. Any changes in the data model not propagated to the meta description will cause the insertion of the information to fail, so this is extremely critical to check.

5 TAPAS functionalities

TAPAS provides two query interfaces: a web based query interface[1] and a VO ConeSearch service. This is convenient for querying and retrieving data related to the observations made with the telescope, and increases the visibility and operability of the data using VO software and VO data mining techniques.

[1] http://tapas.iram.es/

Fig. 6 Detailed information of one scan provided by TAPAS web interface

5.1 Public interface

The TAPAS Home screen (Fig. 2) has a login feature, that either asks for login and password, or provides last login information. From this screen users have access to a *Search* form, to the *Results* provided by the last search performed, *News* (updates to the archive, new datasets added, etc.), the data *Policy*, *Help* on the web interface and some information about the development of the archive.

Identified users may have access to a larger volume of data, depending on the data policy and proprietary periods established for the observational projects. In addition, IRAM staff has access to the internal Django-powered administration interface in order to manage the content and access rights of the archive.

The Search form (Fig. 3) allows users to perform searches on the *Source Name and Coordinates*, observational *Project ID*, used *Receivers*, *Weather requirements*, *Spectral coverage* and *Date* of the observation. The results coming from searches based on multiple criteria satisfy all of them, which give the user the opportunity to filter the obtained results.

Source name or coordinates For specifying targets, users can enter either standard object names, such as those registered by NED [13] or Simbad, or the names declared in the project. If no suitable name is known, users can provide equatorial coordinates in sexagesimal format, and a cone angular size in decimal degrees. Object names, when not found in the projects, are resolved to coordinates using the CDS Sesame service [19]. A *Batch mode* exists by which users can upload a specfic formatted text file which contains a list of

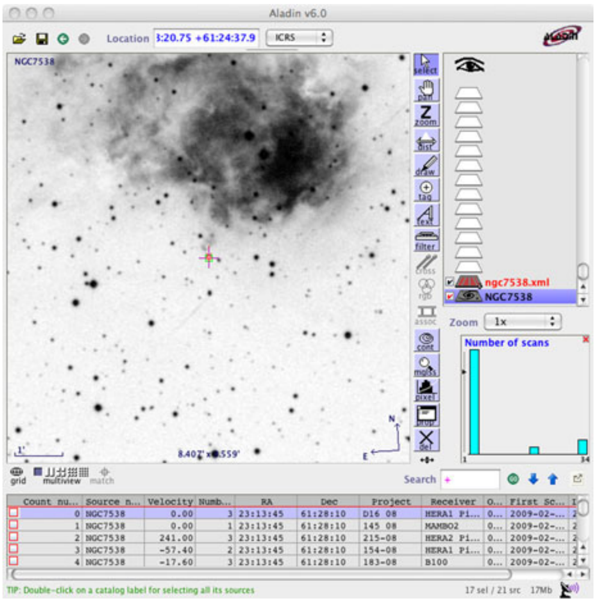

Fig. 7 Coordinates provided by a TAPAS VOTable superimposed on an optical image of NGC7538 in an Aladin window

names and/or position pair coordinates. Internal IRAM names can also be provided, as long as they are enclosed by two asterisks (e.g. *M83A*)

Project ID Users can specify, if known, the internal observational project ID. This feature allows, from one single click, the obtention of all the observations related to a single project.

Receivers Instrumental requirements include the specification of the instrument front-ends. At present, those include EMIR bands E090, E150, E230, E330, HERA heterodyne receivers, and bolometer MAMBO2 receivers.

Weather requirements A maximum *Opacity* can be imposed on searches, as it can identify those datasets which have been obtained under specific meteorological conditions.

Spectral coverage Spectral requirements are specified either in the form of frequency ranges, velocity ranges, or the name of the molecular or atomic line being observed (i.e. CO(1-0), HCN, etc.)

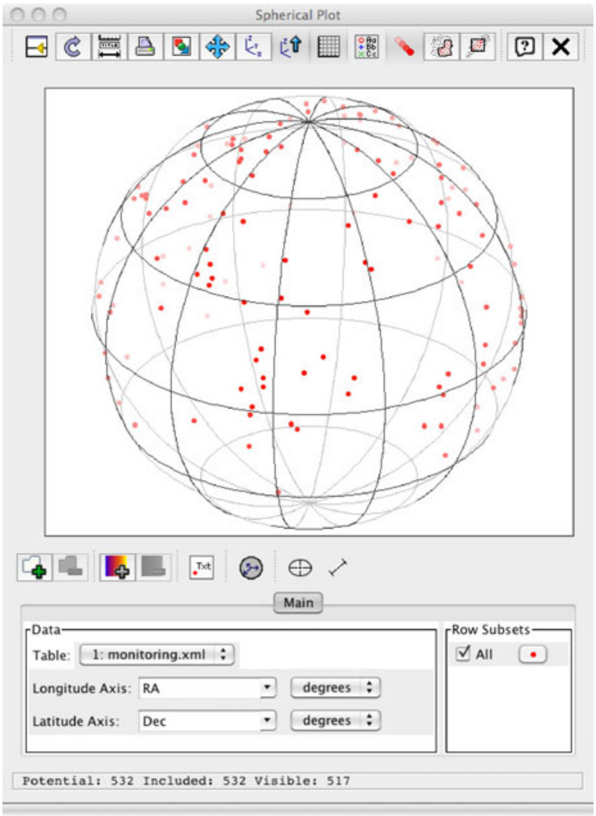

Fig. 8 Topcat window displaying all the observations related to one IRAM 30-m `monitoring` project, obtained from a TAPAS VOTable

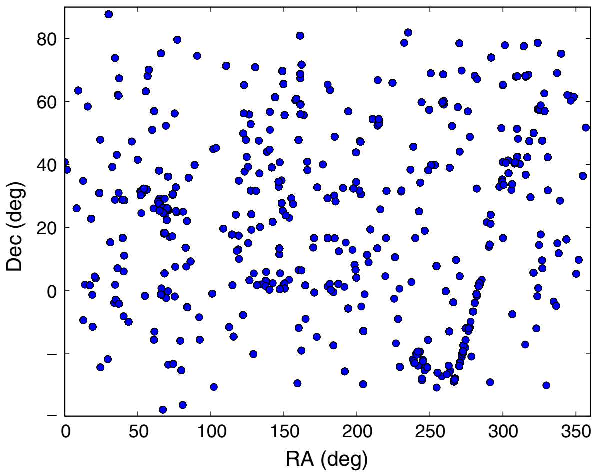

Fig. 9 Observed source positions extracted from the `CalibrationResults` table in the TAPAS DB. The data were taken from 08.02.2009 for a total of 77 days

Date of the observation A calendar-like feature in the search form allows the user to select a range for the dates in which the observations were made.

The Results for a specific search are displayed in an HTML table. The available header information are the IRAM source name and velocity, number of scans for the observation, object equatorial coordinates (J2000), project identifier (ID), used receiver, sky opacity, and time-stamps for the start of the first and the last scans for the observation (Fig. 4).

All tables provided by TAPAS can be sorted in ascending or descending order. Users can also select the number of displayed rows in every page in which the tables are split, allowing for a more comfortable navigation. All results can be displayed as HTML tables, but also Comma Separated Values (CSV) ASCII files and VOTables can be generated, and a PDF file of the results page can be downloaded.

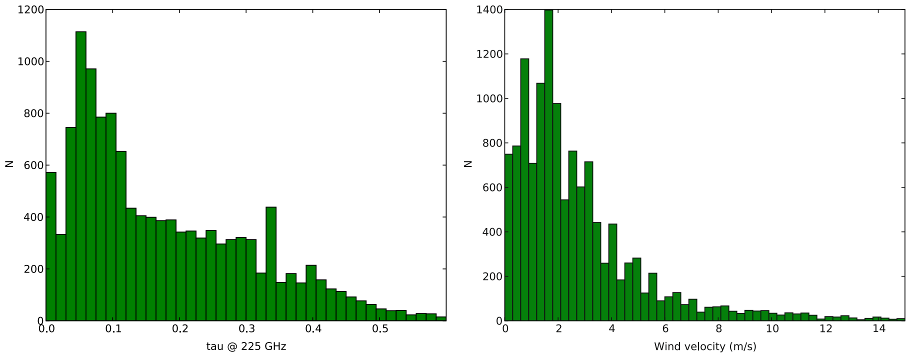

Fig. 10 *Left*: opacity distribution at 225 GHz associated with the observed scans from the `Scans` table. *Right*: wind velocity distribution from the `WeatherStation` table and provided by the weather station

When the user clicks on the Project ID, a new page appears with project details (Fig. 5), including the range in time and number of all scans in this project, the number of scans that have been performed using the MAMBO bolometer, or the heterodyne receivers, and the number of sources observed. A graph is provided where the taumeter reading is plotted for each scan, giving users a view of the evolution of opacity throughout the different project scans.

When users click on the number of scans for an observation, a table showing all scans for the selected observation is provided, together with the common parameters for all scans: project ID, IRAM name of the source, velocity setting, equatorial coordinates, equinox, and receiver. If a particular scan is clicked on this page a detailed information for that scan is shown (Fig. 6): date, observation length, start time and end time, source name, coordinates, source velocity, weather conditions, antenna azimuth and elevation, pointing and focus corrections, observing mode, calibration settings, offsets, front-ends and back-ends, software version, used receivers, frequency and spectral line observed. This page may also contain a link to the astronomical raw data fits file. These data should be preferentially manipulated with the GILDAS package avaialble at the IRAM webpage.

5.2 VO services

The Virtual Observatory is changing the way on how astronomical data are exploited. The rate of data available to the community increases everyday, but also new tools allowing their exploitation are delivered continually. In this respect archives providing VO services improve dramatically their visibility and this constitutes an added value of enormous importance.

Moreover, data provided by VO services take advantage of the interoperability [8] concept that is behind the standard VOTable format. The SAMP

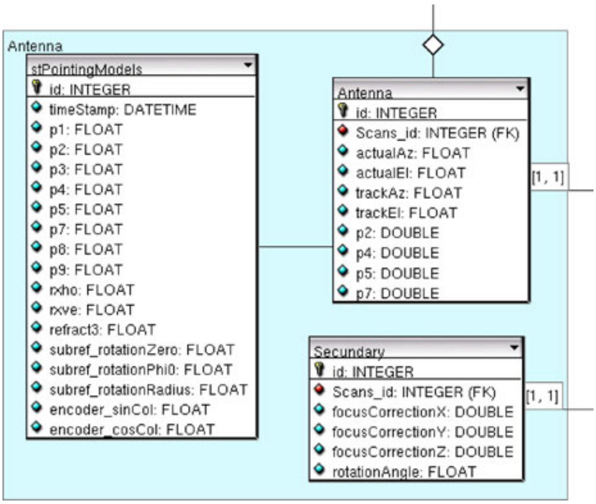

Fig. 11 Antenna subsystem

protocol [22] allows local interoperability, meaning that different VO software can communicate and share data sets. Consequently VOTables provided by TAPAS can be treated with a great diversity of VO software, everyone of them with their own specific functionalities.

TAPAS provides a *ConeSearch* [25] VO service for querying and retrieving the data with widely known VO software like Aladin [3] or Topcat [21]. Figure 7 shows an example of how Aladin can display the optical image corresponding to the region delimited by the coordinates found in a TAPAS VOTable. TAPAS VO services, once registered via VO Registry [2] web forms, will allow easy data retrieval from different astronomical client software (see also Fig. 8).

5.3 Engineering reports

Apart from its main functionalities as a VO compliant service for the astronomical community, TAPAS can be used as a powerful tool to monitor the

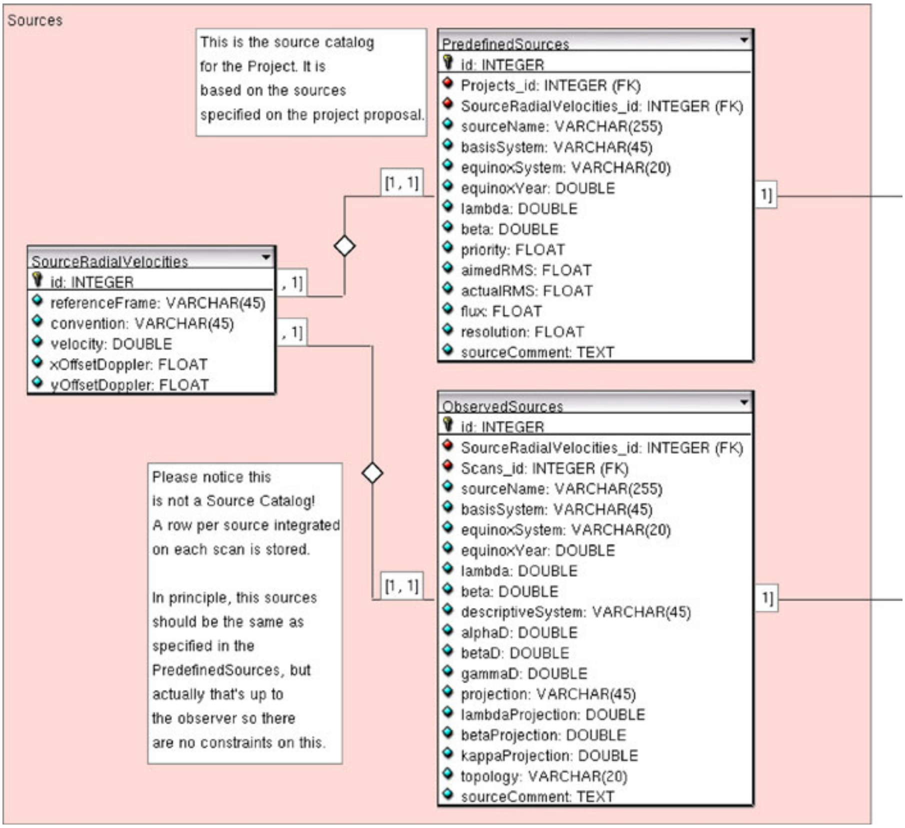

Fig. 12 Sources subsystem

performance of the IRAM 30-m. About 75% of the more than 250 attributes from TAPAS tables are actually devoted to the engineering diagnostics and are not available through the VO services. Those can play a relevant role in the continous improvement of the IRAM 30-m scientific data.

Among the numerous uses that can be given to TAPAS for the engineering work at the IRAM 30-m, we can emphasize the following:

- Analysis of the antenna parameters: pointing and focus offsets.
- Analysis of the evolution of the receiver temperature versus frequency and time.
- Project statistics: use of the backends/receivers, frequencies, observing modes.
- Weather analysis: correlation with antenna parameters, projects, etc.

Since TAPAS records the most important observational metadata, it can be used to detect fine drift or changes in some parameters. As an example, one main component of the antenna driving program is the pointing model which corrects for the deviation of the antenna position due to mechanical imperfections or deformations. The pointing model is stored in the `stPointingModels` table. This model is normally computed after a long session of pointing observations covering the whole sky. The evolution of these pointing corrections can be easily recovered as well from TAPAS through the `PointingResults` table. Another application of TAPAS is the estimation the relative drift in azimuth and elevation between different pairs of receivers.

A key component of the IRAM 30-m telescope are the heterodyne receivers. Apart from the calibration results, TAPAS stores the receiver temper-

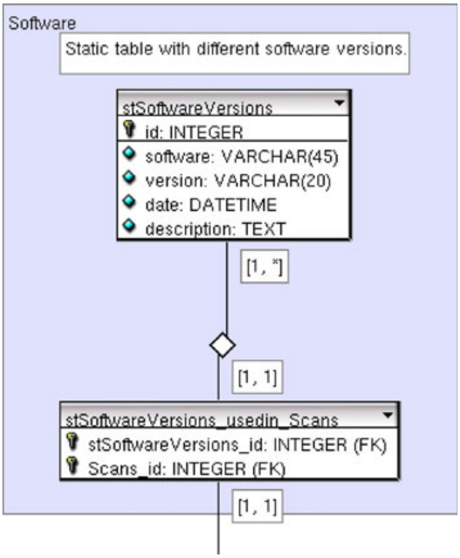

Fig. 13 Software subsystem

atures at the tuned frequencies. This allows mapping the receiver temperature $T_{\rm rec}$ as a function of time and frequency.

An aspect of the management of a telescope is the monitoring of the scientific projects, in particular recording all the settings relative to an astronomical project like the frequencies, receivers, sources or backends. An output in a telescope activity is the sky coverage achieved by the observations. Using TAPAS, we show in Fig. 9 the source positions observed during a time interval of 77 days. We note that given the small beam size of the IRAM 30-m (less than 29 arcsec at 88 GHz), the actual sky coverage is much smaller than shown on that figure.

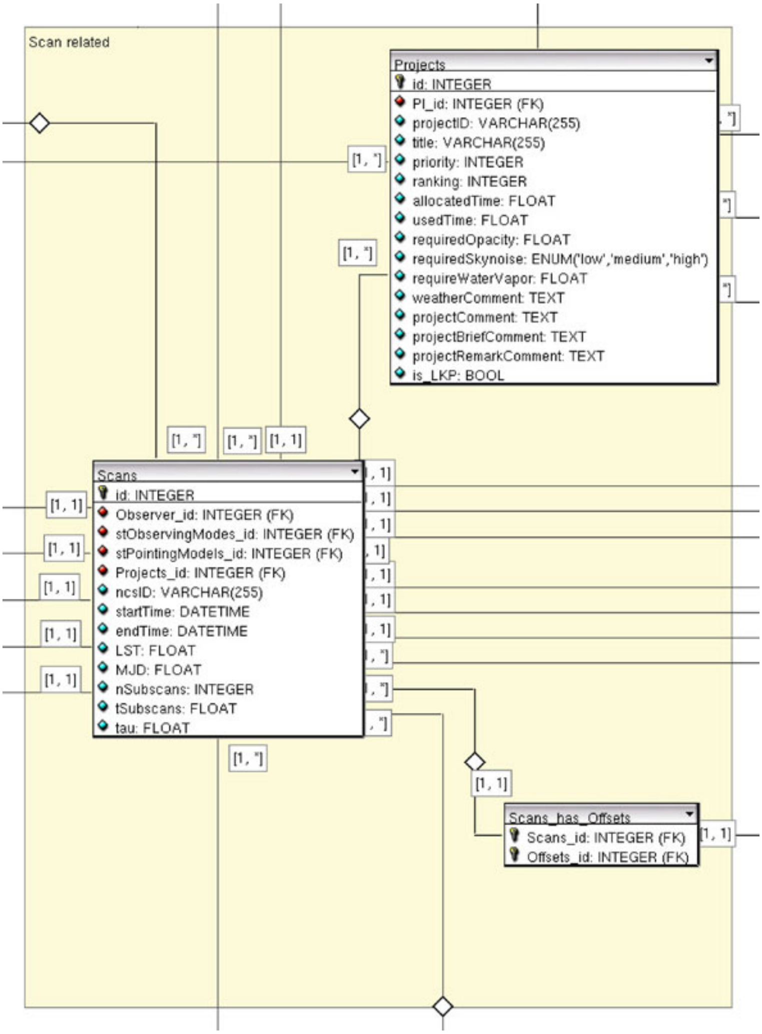

Fig. 14 Scan subsystem

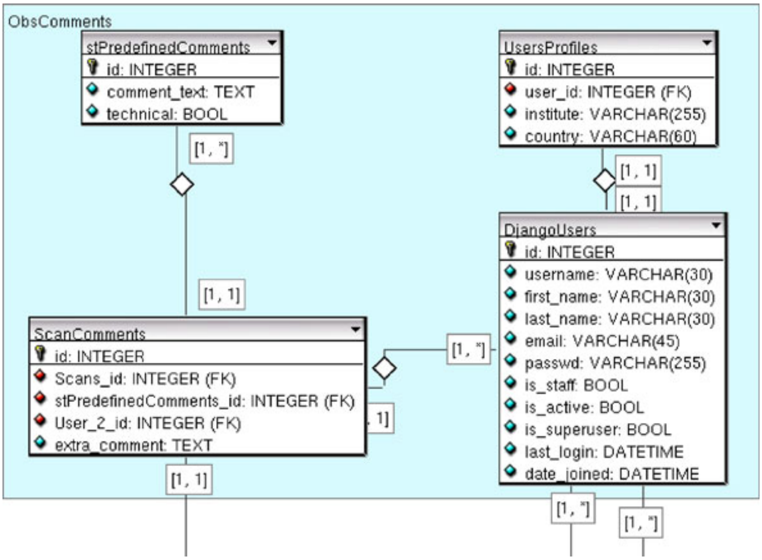

Fig. 15 Users profile subsystem

Weather information is also part of the diagnostics of the antenna parameters. For example the pointing accuracy may be affected by the wind strength and direction, and the ambient relative humidity. TAPAS continuously stores in an independent table, `WeatherStation`, the information from the meteorological station, but stores as well some weather information related directly to each scan. Figure 10 shows two examples using the weather information in

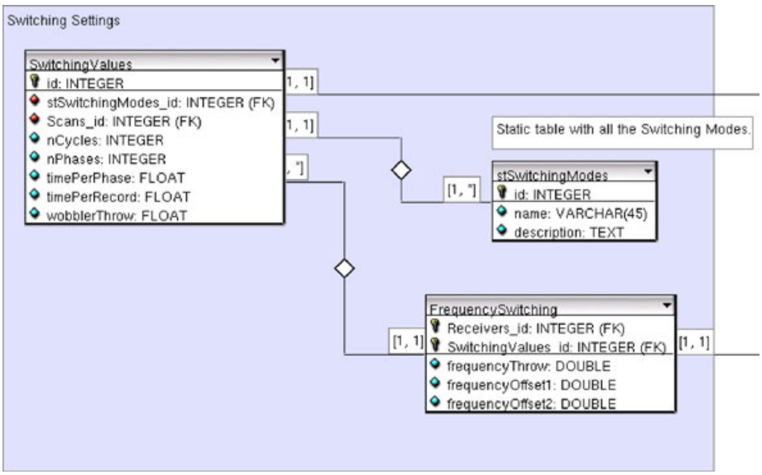

Fig. 16 Switching settings subsystem

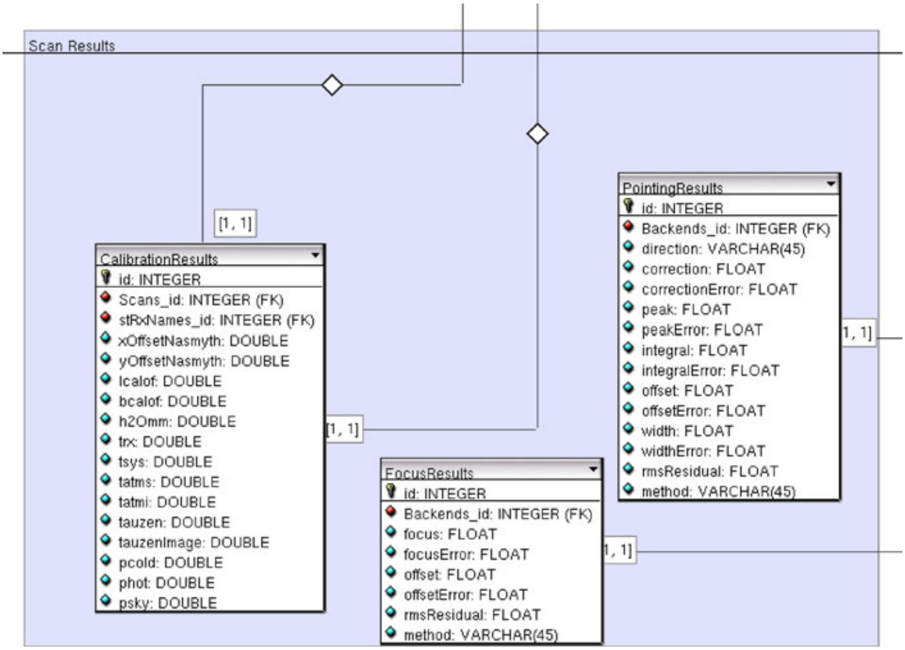

Fig. 17 Scan results subsystem

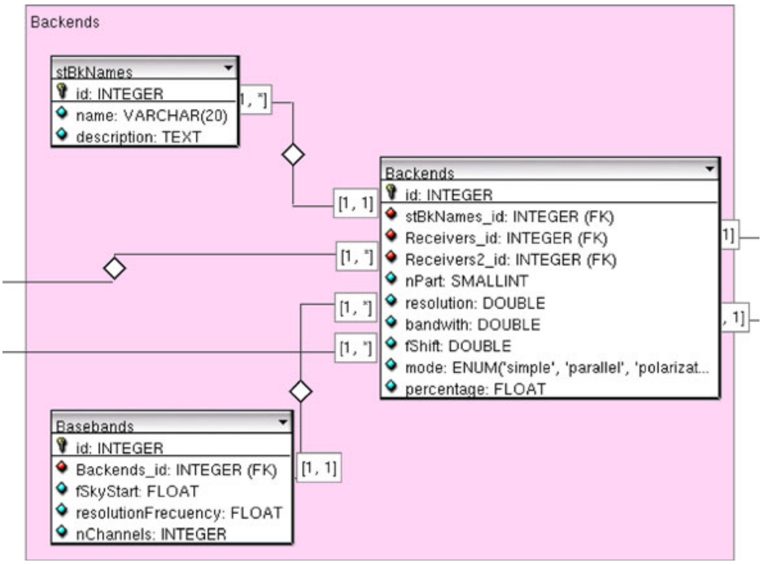

Fig. 18 Backends subsystem

TAPAS. More complex analysis is allowed by TAPAS thanks to its design and its implementation in the Django framework.

5.4 Future developments

TAPAS is still in development and will be improved continuously in the future. We intend to make the calibration results of MAMBO observations available in TAPAS. It is planned as well to make available to the community observations taken before September 2009. A further important planned upgrade will be the possibility of the observer to add standardized and free comments, thus allowing to create full observations logsheets. Another example is the production of calibration or trending plots directly from the web interface and not through the direct query of the TAPAS database as for the Figs. 9 and 10. And finally, and thanks to the flexibility of TAPAS, for all the data of the new instruments to be installed at the IRAM 30-m (bolometer, heterodyne) an effort will be made to make them accessible via TAPAS. We may need to add quality flags for some of the parameters like pointing offsets.

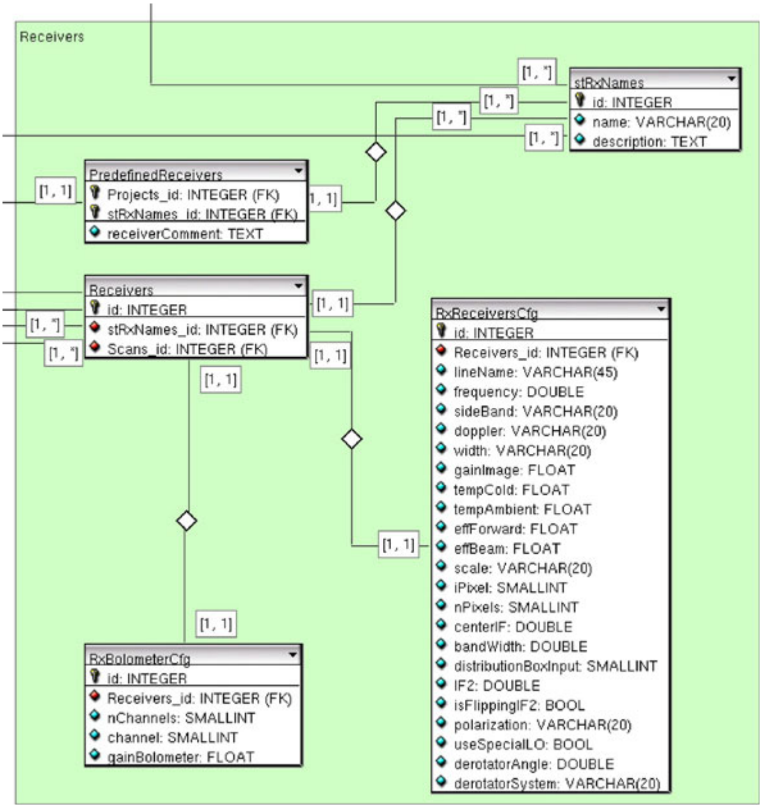

Fig. 19 Receivers subsystem

6 Summary

Recent efforts in the domain of VO for radio astronomy have provided new tools to facilitate the implementation of VO compliant archives. In that context, a joint project between the AMIGA group of the IAA and IRAM Granada has served as framework to build an archive of the IRAM 30-m observations, TAPAS. The archive is VO compliant and has its own web interface available for the astronomical community. The main characterics of TAPAS can be summarized as follows:

- The data model is based on RADAMS and the XML description of the NCS at the IRAM 30-m telescope. To avoid an excessive direct dependency between the data model and the code, a meta description of the DB is included.
- TAPAS is built using the Python Django web framework on top of a MySQL DB.

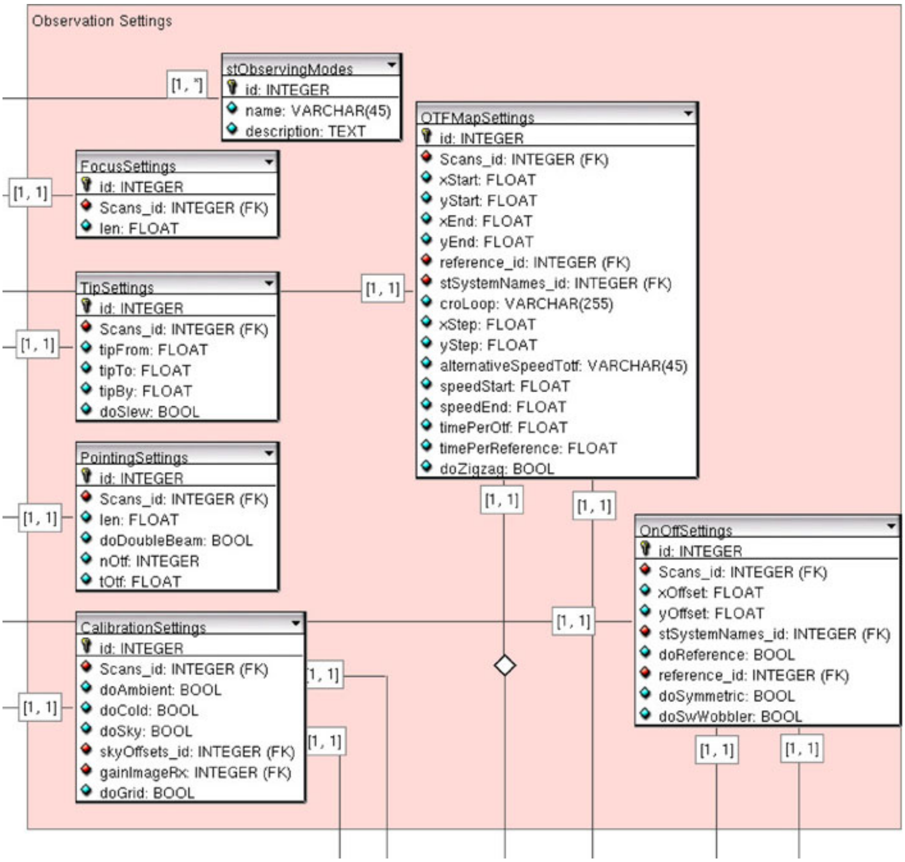

Fig. 20 Observation settings subsystem

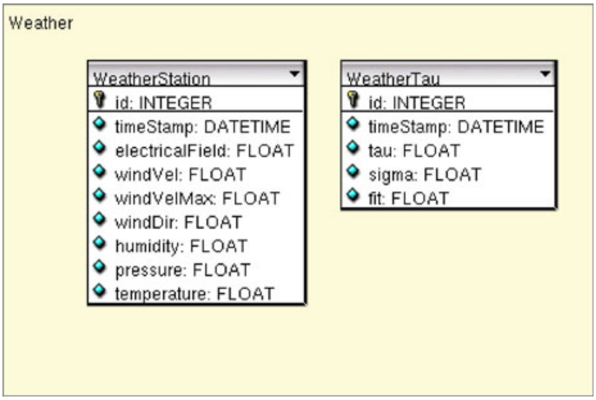

Fig. 21 Meteo station subsystem

- The system is fully included in the New Control System and is capable of handling possible failure modes.
- The web interface is user oriented and grants access at different levels: public, PI project, IRAM staff.
- The conesearch VO-protocol is implemented.
- The web interface allows to combine various criteria (source, project ID, receiver, weather, etc) to look for the relevant observations in the archive.

Acknowledgements We would like to thank the anonymous referee for the very helpful comments. The authors acknowledge support from Spanish DGI Grants AYA 2005-07516-C02-01, AYA 2005-07516-C02-02, AYA2008-06181-C02, and Junta de Andalucía grant P08-FQM-4205-PEX.

Appendix: Data model subsystems

In this section we present (Fig. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22) a detailed description of each subsystem of the Data Model listed in Table 1.

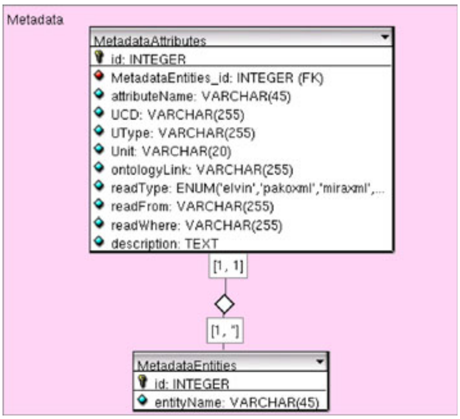

Fig. 22 Metadata subsystem

References

1. Barnes, J.M.: Object-relational mapping as a persistence mechanism for object-oriented applications. Honors Projects. Paper 6 (2007)
2. Benson, K., Plante, R., Auden, E., et al.: IVOA Registry Interfaces version 1.0. IVOA Recommendation, IVOA. http://www.ivoa.net/Documents/RegistryInterface/ (2009)
3. Bonnarel, F., Fernique, P., Bienaymé, O. et al.: The ALADIN interactive sky atlas. A reference tool for identification of astronomical sources. A&AS **143**, 33 (2000)
4. Chen, P.P.: The entity–relationship model: toward a unified view of data. ACM Trans. Database Syst. **1**, 9–36 (1976)
5. Cowan, J., Tobin, R.: XML information set, 2nd edn. W3C Recommendation, W3C. www.w3c.org/TR/2004/REC-xml-infoset-20040204 (2004)
6. DuBois, P.: MySQL, 4th edn, p. 1200. Addison-Wesley. ISBN 0-672-32938-7 (2008)
7. Fernández, M., Marsh, J., Malhotra, A., Nagy, M., Walsh, N.: XQuery 1.0 and XPath 2.0 data model. W3C working draft, W3C. www.w3.org/TR/2003/WD-path-datamodel-20031112 (2003)
8. Genova, F.: Interoperability. In: Astronomical Data Analysis Software and Systems XI, ASP Conf. Ser., vol. 281 (2002)
9. GuangChun, L., Lu, W., Hanhong, X.: A novel web application frame developed by MVC. SIGSOFT Softw. Eng. Notes **28**, 2, 7 (2003). doi:10.1145/638750.638779
10. Hanisch, R.J., Farris, A., Greisen, E.W., et al.: Definition of the flexible image transport system (FITS). A&A **376**, 359 (2001)
11. Holovaty, A., Kaplan-Moss, J.: The Definitive Guide to Django: Web Development Done Right, p. 447. Apress, ISBN 978-1-59059-725-5 (2007)
12. IEEE Standard 1003.1: Standard for information technology - portable operating system interface (POSIX). Shell and utilities (2004)
13. Mazzarella, J.M.: The NED team: NED for a New Era. In: Astronomical Data Analysis Software and Systems XVI ASP Conference Series, vol. 376 (2007)
14. Murphy, T., Lamb, P., Owen, C., Marquarding, M.: PASA **23**, 25–32 (2006)
15. Ochsenbein, F., Williams, R., Davenhall, C., Durand, D., Fernique, P., Giaretta, D., Hanisch, R., McGlynn, T., Szalay, A., Taylor, M.B., Wicenec, A.: VOTable format definition version 1.2. IVOA recommendation, IVOA. http://www.ivoa.net/Documents/VOTable/20091130/REC-VOTable-1.2.html (2009)
16. Preite Martinez, A., Derriere, S., Delmotte, N., et al.: The UCD1+ controlled vocabulary version 1.23. IVOA Recommendation, IVOA (2007)
17. Santander-Vela, J.D., et al.: arXiv:0810.0385 (2007)
18. Santander-Vela, JdD.: PhD Thesis, University of Granada (Spain) (2012)
19. Schaaff, A.: Web services and related works at CDS. In: Astronomical Data Analysis Software and Systems XIII ASP Conference Series, vol. 314 (2004)
20. Szalay, A., Gray, J.: The world-wide telescope . Science **293**, 2037 (2001)
21. Taylor, M.: TOPCAT & STIL: starlink table / VOTable processing software. In: Astronomical Data Analysis Software and Systems XIV ASP Conference Series, vol. 347 (2005)
22. Taylor, M., Boch, T., Fitzpatrick, M., Allan, A., Paioro, L., Taylor, J., Tody, D.: SAMP simple application messaging protocol version 1.11. IVOA Recommendation. http://www.ivoa.net/Documents/latest/SAMP.html (2009)
23. Viallefond, F.: ADASS XV. PASP **351**, 627 (2006)
24. Wenger, M., Ochsenbein, F., Egret, D., Dubois, P., Bonnarel, F., Borde, S., Genova, F., Jasniewicz, G., Laloë, S., Lesteven, S., Monier, R.: The SIMBAD astronomical database. The CDS reference database for astronomical objects. A&AS **143**, 9 (2000)
25. Williams, R., Hanisch, R., Szalay, A. Plante, R.: IVOA Recommendation, IVOA. http://www.ivoa.
net/Documents/REC/DAL/ConeSearch-20080222.html (2009)