

# Chilean Virtual Observatory

Mauricio Solar, Mauricio Araya, Luis Arévalo  
 Department of Informatique  
 Universidad Técnica Federico Santa María, UTFSM  
 Santiago, Chile  
 {msolar, maray, larevalo}@inf.utfsm.cl

Victor Parada  
 Department of Informatique Engineering  
 University of Santiago de Chile, USACH  
 Santiago, Chile

Ricardo Contreras  
 Department of Computer Science  
 University of Concepcion  
 Concepcion, Chile

Diego Mardones  
 Department of Astronomy  
 University of Chile  
 Santiago, Chile

**Abstract**—This paper presents the challenges, architecture and current status of the Chilean Virtual Observatory (ChiVO), which is a software infrastructure for accessing and processing astronomical data generated in Chile. As ChiVO is part of the International Virtual Observatory Alliance (IVOA), we strictly follow the protocols and standards that this organization produce. However, there are always open challenges due to the new observational technologies and local requirements that motivates research on every new virtual observatory, such as the complex data models and Big Data problems that the ALMA Observatory is confronting.

The current ChiVO prototype includes IVOA compliant services as well as new solutions designed for ALMA data, all of them using modern software technologies.

**Keywords**—component; Virtual Observatory; IVOA; ICT Infrastructure

## I. INTRODUCTION

En los últimos años, el problema de la avalancha de datos se ha manifestado tanto en la ciencia como en los ámbitos empresariales, haciendo cada día más relevante el apropiado manejo y gestión de lo que se ha denominado *Big Data*.

Este concepto engloba a la investigación en el manejo grandes volúmenes de información, los cuales no son sencillos de procesar con las herramientas y procedimientos tradicionales. Cuando el volumen de datos llega a órdenes de TeraBytes (TB) a ZetaBytes (ZB), los algoritmos y procedimientos deben adaptarse para ser usados en las nuevas plataformas de computación de alto desempeño (HPC, por sus siglas en inglés), con herramientas *Cloud*, de forma distribuida y *on-line*.

Adicionalmente, no sólo se está tratando con grandes volúmenes de datos de manera estacionaria, sino que la frecuencia con la cual éstos son generados, crea nuevos desafíos para el desarrollo de soluciones, como lo son el

almacenamiento, variabilidad de formato y tiempo de respuesta.

*Big Data* no es una tecnología en sí misma, sino más bien un planteamiento de trabajo para la obtención de valor y beneficios de los grandes volúmenes de datos que se están generando hoy en día. Se deben contemplar aspectos como:

- Cómo capturar, gestionar y explotar.
- Cómo asegurar, verificar validez y fiabilidad.
- Cómo compartir para obtener mejoras y beneficios.
- Cómo comunicar para facilitar la toma de decisiones y análisis posterior.

Uno de los dominios donde el problema del *Big Data* está llegando a su punto crítico, es la astronomía. Las instalaciones de última generación en operaciones, como el *Atacama Large Millimeter/submillimeter* (ALMA), y las que están en construcción, como el *Large Synoptic Survey Telescope* (LSST) y el *Square Kilometer Array* (SKA) producen y producirán datos en gran escala, proyectándose que para el año 2020 serán más de 60 PetaBytes (PB) de información accesible para la comunidad astronómica.

Chile, siendo uno de los países con gran actividad astronómica en el mundo, no poseía hasta hace el año 2013 un Observatorio Virtual (VO por sus siglas en inglés). Por otro lado, ALMA tampoco dispone de servicios compatibles con los protocolos y estándares de otros VO, por lo que es un desafío plantear necesidades y requerimientos de este tipo de sistema.

Si bien la comunidad internacional lleva varios años trabajando y estudiando los observatorios virtuales y sus estándares, cada nuevo telescopio e instrumento impone nuevos desafíos y oportunidades de desarrollo.

Este es el caso de los datos de ALMA, que introduce el problema del *Big Data* como dilema presente, lo que requiere equipar un observatorio virtual que albergue sus datos de tecnología de punta e investigación de frontera en sus herramientas.

Este artículo tiene como objetivo mostrar la arquitectura general de Observatorios Virtuales y en particular al *Chilean Virtual Observatory* (ChiVO), desde el punto de vista de desarrollo de software. Además se presenta el estado actual y cómo se abordaron las restricciones y problemáticas que imponen los datos de ALMA.

## II. OBSERVATORIOS EN CHILE

Las privilegiadas condiciones atmosféricas hacen de Chile uno de los lugares más propicios para la realización de investigaciones científicas en astronomía. Existen más de una docena de instalaciones astronómicas de gran envergadura a lo largo de Chile [1], como por ejemplo, el nombrado *Atacama Large Millimeter/submillimeter Array* (ALMA), el *Very Large Telescope* (VLT), y en los próximos años el *European Extremely Large Telescope* (E-ELT), con el cual se estima que Chile contará con el 60% del poder observacional astronómico mundial.

Una de las condiciones que se establecen a nivel país, es que el 10% del tiempo de observación pertenece a la comunidad astronómica chilena, lo cual justifica la necesidad a nivel país del desarrollo de una plataforma astroinformática para una administración y análisis inteligente de los datos.

La necesidad de un sistema con estas características no es algo nuevo: desde el 2002 se viene planteando la necesidad de la creación de un Observatorio Virtual (VO, por sus siglas en inglés) a nivel nacional, como una solución al acceso público y manipulación avanzada de los datos astronómicos de gran escala.

El paradigma VO es una iniciativa internacional que permite el acceso de datos astronómicos, a cargo de centros especializados para su almacenamiento y procesamiento, a los cuales pueden acceder tanto astrónomos, como personas comunes. Con la estandarización de métodos e información, es posible estudiar los registros astronómicos sin la necesidad de solicitar nuevas observaciones, reduciendo los requerimientos físicos de instrumentos y localización, y minimizando la duplicidad de datos.

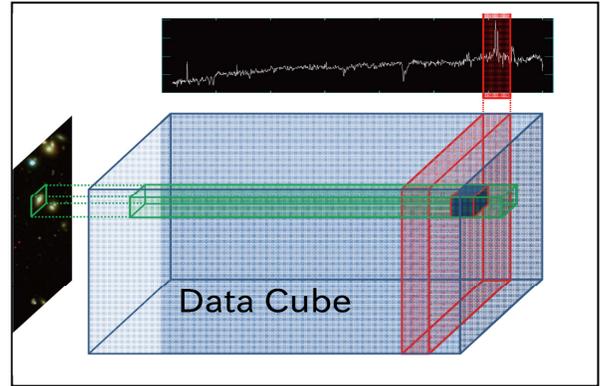
### A. ALMA y sus datos (Cubos de datos)

ALMA es un observatorio radioastronómico interferométrico, lo que en palabras simples corresponde a observar señales del orden milimétrico y sub-milimétrico usando dos o más antenas de radio-frecuencia. Al combinar las señales para analizarlas, se puede obtener información detallada de la fuente de emisión (ya sea galáctica o extragaláctica) con resoluciones sin precedentes. En su operación máxima, se podrá combinar las 66 antenas ubicadas en Chajnantor, Chile, a más de 5 mil metros sobre el nivel del mar, las cuales pueden operar en un amplio rango de frecuencias que va desde los 84 GHz hasta los 950 GHz (Banda 3 - Band 10 de ALMA).

Del punto de vista de producción de datos, este proceso genera cubos en tres dimensiones dadas por dos ejes de posición en la esfera celeste (RA/DEC) y un eje de espectro de frecuencia (ver Figura 1). La particularidad de estos cubos de datos es su tamaño, ya que la alta resolución espacial y

espectral que permite este observatorio genera cubos de datos de gran escala (orden de GB y TB).

Fig. 1. Estructura de cubos de datos de ALMA.



En cuanto a los formatos de datos en astronomía, todos se rigen por una estructura similar compuesta por metadatos (información que describe los datos) más datos binarios. ALMA posee dos modelos de datos especiales:

- **ALMA Science Data Model (ASDM):** Formato en XML diseñado para guardar los metadatos obtenidos del proceso de observación y generar un *link* hacia los datos binarios.
- **Measurement Set (MS):** Formato basado en tablas binarias (una principal y varias secundarias), el que también guarda metadata y datos binarios [2]. Este formato se usa para realizar reducción y procesamiento en el *Common Astronomy Software Applications* (CASA), software creado por NRAO<sup>1</sup> para estos fines.

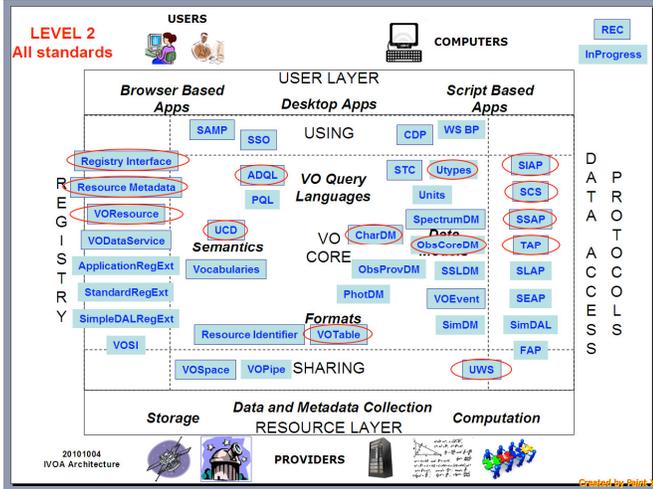
## III. CHIVO: CHILEAN VIRTUAL OBSERVATORY

ChiVO es un observatorio virtual que alojará datos de los observatorios localizados en Chile, ciñéndose a los estándares de interoperabilidad de la Alianza Internacional de Observatorios Virtuales (IVOA, por sus siglas en inglés), y utilizando tecnologías de última generación. La misión de IVOA es facilitar la coordinación y colaboración necesaria para facilitar el acceso global e integrado a los datos recogidos por los observatorios astronómicos internacionales. IVOA fue formada el 2002 y tiene actualmente 22 miembros activos, siendo Chile el más reciente. Su método de trabajo es en base a *Working Groups*, los cuales están a cargo de establecer protocolos y estándares que describen la arquitectura general de un VO (Figura 2). La arquitectura posee tres capas:

- **Usuarios:** astrónomos y científicos en general interesados por los datos publicados por los observatorios.
- **Recursos:** observatorios y centros que producen datos astronómicos (reales o simulados).
- **Capa intermedia:** esta capa define lo que es un observatorio virtual, es decir, de qué manera se comunican los usuarios y recursos usando protocolos y estándares para buscar y acceder datos.

<sup>1</sup> <http://casa.nrao.edu/>

Fig. 2. Arquitectura de IVOA.



A. Requerimientos

Para la creación de ChiVO se identificaron las necesidades actuales de la comunidad astronómica nacional, las cuales se pueden resumir en:

- **Descubrir:** Encontrar datos astronómicos de un objeto o instrumento sobre una región específica del espacio de alta dimensión, en base a parámetros de los ejes espaciales, temporales, espectrales, corrimiento al rojo, polarización, etc, ya sea por búsqueda o por exploración.
- **Obtener:** Enlace a descarga de los datos requeridos en distintos formatos, ya sea en el VO o en un servicio externo.
- **Comparar:** Cruzamiento de información de datos obtenidos entre distintas fuentes de información.

En este proceso participó un equipo multidisciplinario (astrónomos, ingenieros, científicos, expertos en datos de ALMA, etc) y duró aproximadamente 4 meses, donde la comunidad astronómica definía sus requerimientos y casos de uso, y el equipo informático contrastaba estos con las normativas internacionales.

B. Arquitectura de ChiVO

Según las necesidades de los radioastrónomos chilenos, los requerimientos, casos de uso y los modelos de datos (compatibles con los estándares de IVOA), conllevan a la creación de la arquitectura y modelo de desarrollo de la Figura 3.

Capa de abstracción: Clientes

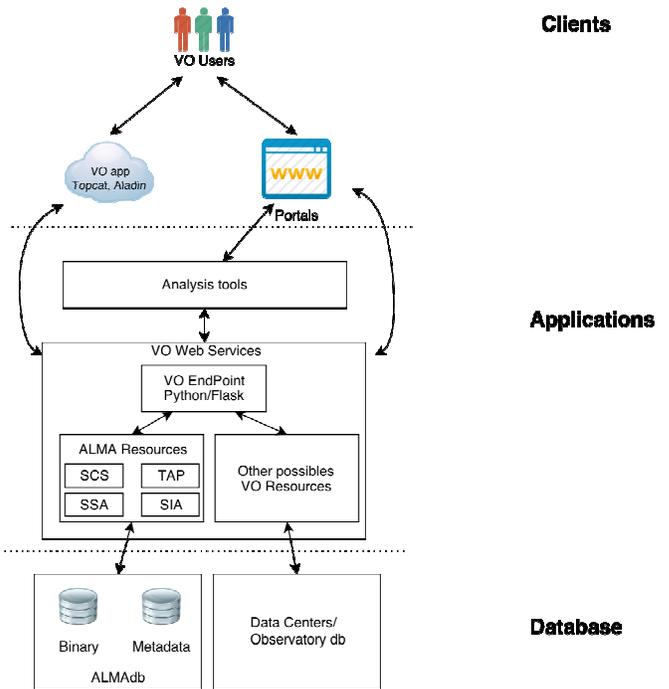
Esta capa representa al usuario final y cómo se facilita la comunicación entre el usuario y los datos. En esta capa el usuario realiza consultas a través de los protocolos de acceso ofrecidos por ChiVO o a través de un formulario avanzado, utilizando aplicaciones compatibles con VO y su portal Web. Una vez realizada la consulta, el sistema retorna al usuario una lista que describe objetos u observaciones encontrados (metadatos) y podrá acceder a ellos a través de un enlace de descarga asociado a cada resultado. Cabe destacar que gracias

a la separación por capas de abstracción se logra la flexibilidad y escalabilidad en el sistema, para que independientemente de la capa, nuevas aplicaciones puedan interactuar con ChiVO, así como la adición de nuevas fuentes de información aparte de ALMA.

Las consultas son recibidas por ChiVO a través de su Endpoint de datos, que recibe consultas en HTTP, GET o POST, ante lo cual el Endpoint retorna la lista de resultados en una tabla en el formato XML VOTable [3].

Para el caso del portal web, el VOTable es mostrado al usuario a través de una herramienta Web que permite la manipulación simple y eficiente de VOTables, llamada VOView.

Fig. 3. Arquitectura de ChiVO



Capa de abstracción: Aplicaciones

En esta capa se encuentran los programas que procesan las consultas entre los usuarios y los datos. Cada estándar de IVOA requiere un mínimo de su propia implementación para ser compatible con el VO. En el caso de estos protocolos de acceso sólo es necesaria la recepción de consultas HTTP básicas junto a los parámetros de búsqueda requeridos.

El elemento que representa a las herramientas de análisis es fundamental en la eficiencia de ChiVO. Esto debido a que los datos a analizar por los astrónomos suelen tener un gran tamaño y es costosa su transferencia. Este problema se resuelve acercando las herramientas de análisis y procesamiento al lugar donde están almacenados los datos a procesar, que es una solución básica para Big Data.

Considerando que en el futuro será necesario ofrecer búsquedas basadas en otros datos, no sólo provenientes de

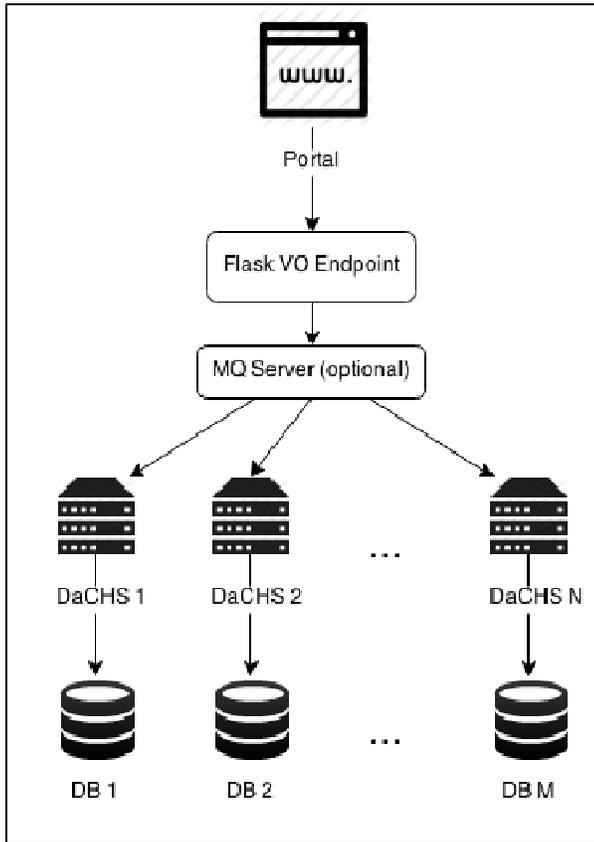
ALMA, es necesaria cierta abstracción al momento de implementar esta capa, ya que debe permitir la interacción de nuevas fuentes de información, siempre y cuando se mantenga la compatibilidad de IVOA.

En esta capa también está en desarrollo un sistema capaz de resolver nombres (tipo *Sesame* [4]) pero para datos de ALMA y el registro de ChiVO.

### Capa de abstracción: Datos

En esta capa se encuentran los recursos que tienen los datos y metadatos. El trabajo está asociado a una base de datos relacional para almacenar los metadatos asociados al modelo de datos recomendado por IVOA *Observation Core Data Model (ObsCore)* [5], usando un *framework* desarrollado por el VO alemán *DaCHS* [6, 7]. Con respecto a rendimiento, esta es la sección que consume más recursos, tanto en tiempo de computación (resuelve las consultas hechas a las bases de datos) y almacenamiento físico de los datos. A modo de verificación momentánea, la primera versión de la implementación de ChiVO trabaja con un conjunto de datos, con un tamaño de 1 TB, los que provienen de la reducción del ciclo 0 de ALMA. Debido a esta limitación se propone el esquema de funcionamiento de la Figura 4, en donde pueden haber  $N$  servidores con *DaCHS* apuntando a  $M$  bases de datos distribuidas o replicadas [6, 7].

Fig. 4. Estructura de cubos de datos de ALMA.



### C. Metadatos de los datos de ALMA

Para poder construir la base de datos relacional con el modelo de datos *ObsCore* fue necesario mapear campos desde el *ALMA Science Data Model (ASDM)* [8]. Un ASDM contiene una serie de tablas XML con la metadata de la observación, y en este caso se usaron: *main*, *Source*, *ExecBlock*, *spectralWindow*.

En la Table I se muestra el resultado de la investigación. A la izquierda se despliegan las columnas de la clase *Observation*, la segunda columna indica de donde se obtienen los datos para asignarlos a los campos de la primera columna para el caso de los ASDM.

Para poder llenar los campos de la clase *Observation* es necesario escribir una rutina capaz de operar sobre las tablas del ASDM (XML).

Actualmente existen múltiples herramientas en el Paquete de Aplicaciones de Software Comunes de Astronomía (CASA, debido a sus siglas en inglés) [2].

TABLE I. CAMPOS DEL *ObsCore* Y ORIGEN DESDE ASDM

Campo ObsCore	ASDM
dataproduct_type	visibility
calib_level	1
obs_collection	ALMA
obs_id	[ExecBlock.execBlockUID]
obs_publisher_did	[Cycle ID]
access_url	[URL de ChiVO]
access_format	application/x-asdm
access_estsize	[main.dataSize]
target_name	[Source.sourceName]
s_ra	[Source.direction]
s_dec	[Source.direction]
s_fov	[1.2 * lambda / Diametro antena]
s_region	circle
s_resolution	[1.2*lambda/(ExecBlock.baseRangeMax)]
t_min	[ExecBlock.startTime]
t_max	[ExecBlock.endTime]
t_exptime	[main.interval]
t_resolution	[mainTable.interval]
em_min	[ExecBlock.baseRangeMin]
em_max	[ExecBlock.baseRangeMax]
em_res_power	[spectralWindow.resolution]
o_uct	em.mm
pol_states	[Source.stokesParameter[numStokes]]
facility_name	ALMA
instrument_name	ALMA

### D. Tecnologías utilizadas

Para el desarrollo de ChiVO se evaluaron distintas herramientas posibles de las cuales se concluyó en cada capa:

#### Capa de abstracción: Aplicación

Los *framework* que se evaluaron para la implementación del *Endpoint* fueron:

- *Ruby on Rails: Framework* de desarrollo Web ampliamente utilizado el día de hoy. Se basa en el concepto Modelo-Vista-Controlador (MVC). Sin embargo, esta herramienta no será utilizada debido a que

muchas funcionalidades no son necesarias para el presente prototipo.

- **Python/Flask:** Flask es un *microframework* diseñado especialmente para servicios y herramientas Web pequeñas. La presente solución provee un marco de trabajo para la creación de aplicaciones Web que pueden ser accedidas mediante distintos métodos HTTP. Existe mucha documentación y comunidad activa que permite implementar y solucionar problemas de forma rápida.

**Capa de abstracción: Datos**

Dentro de los *toolkits* de *Data Access Layer* (DAL) recomendados por IVOA para el acceso a datos mediante protocolos se tiene: *Simple Cone Search* (SCS) [9]; *Simple Image Access* (SIA) [10]; *Simple Spectral Access* (SSA) [11]; y *Table Access Protocol* (TAP) [12] a través de *Astronomical Data Query Language* (ADQL) [13] y *Universal Worker Service* (UWS) [14], se probaron y verificaron los siguientes (Table II):

- **SAADA:** Desarrollado por el VO francés. Es una herramienta bastante útil del punto de vista del usuario del sistema. Posee excelente documentación y un conveniente proceso de instalación. Está desarrollado en *Java* y su correspondiente despliegue se lleva a cabo mediante **Tomcat**. Es posible configurar servicios SCS/SIA/SSA/TAP y no es un proyecto *Open Source*.
- **VO-Dance:** Desarrollado por el VO Italiano, es una herramienta en *Java* en su *Backend*, y *Python* en su *Frontend* (*Framework Django*). Lo destacable de esta herramienta es que trabaja usando *MySQL* como motor de base de datos principal, y de acuerdo a las últimas noticias relacionadas a su desarrollo, podría existir un soporte para *PostgreSQL* en el futuro. La herramienta no es *Open Source* y la documentación es precaria debido a que aún está en desarrollo. Compatible con servicios SCS/SIA/SSA/TAP.
- **openCADC:** Desarrollado por el VO canadiense, es una herramienta *Open Source* escrita en *Java*, utilizada actualmente en el *ALMA Science Archive*. Este *toolkit* es uno de los más robustos, contiene distintos paquetes con servicios a ser utilizados en el *webservice*. Sin embargo, posee una documentación precaria, lo que es compensado por una comunidad abierta de desarrollo. Es posible configurar servicios TAP y UWS.
- **DaCHS:** Desarrollado por el VO alemán. Es una herramienta *Open Source* escrita en *Python*. Es uno de los *toolkits* DAL más usados por los VO, ya que posee una amplia documentación de instalación y configuración. Es posible configurar servicios SCS/SIA/SSA/TAP.

**Capa de abstracción: Clientes**

Inicialmente la interfaz de usuario o Frontend poseería solo vistas, por lo que el desarrollo podría ser en prácticamente cualquier lenguaje o *framework*, como por ejemplo PHP, *Django* o *Ruby on Rails*. Sin embargo, con los requerimientos de la plataforma, especialmente el de capa de

usuarios, se decidió escoger un *framework* MVC que fuese lo suficientemente ágil y compatible con el resto de servicios, *Ruby on Rails*.

TABLE II. RESUMEN DE LOS *TOOLKITS* EN TABLA COMPARATIVA

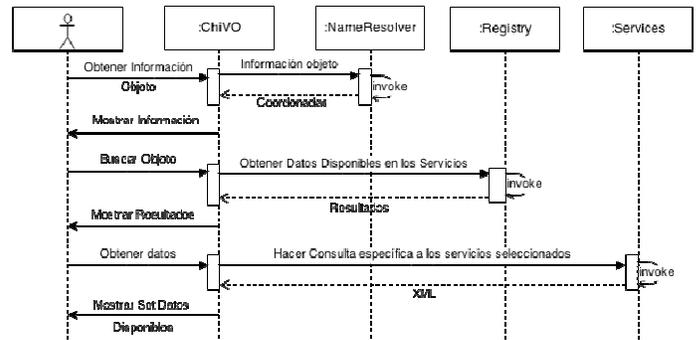
Toolkits	Leng.	Open Source	Documentado	Servicios	Última actualización
SAADA	Java	No	Si	SCS/SIA/SSA/TAP	Mayo 2012
VO-Dance	Java/Python	No	No	SCS/SIA/SSA/TAP	Diciembre 2012
openCADC	Java	Si	No	TAP	---
DaCHS	Python	Si	Si	SCS/SIA/SSA/TAP	Junio 2013

IV. IMPLEMENTACIÓN DE ChiVO

Para comenzar el desarrollo de ChiVO fue necesario identificar aparte de los requerimientos y casos de uso, las interacciones que los usuarios realizarán con el sistema. El diagrama de secuencia de interacción entre el usuario y ChiVO se puede ver en la Figura 5.

En base a este diagrama, el estado de avance de los requerimientos y tecnologías utilizadas, se especificará por cada capa de abstracción.

Fig. 5. Diagrama de secuencia entre Usuario y ChiVO.



La arquitectura de software de ChiVO está basada en el uso de protocolos y estándares de IVOA. Estos protocolos y estándares están agrupados en capas, de las cuales se destaca la capa de aplicación y la de datos, ya que sus estándares básicos definen la operación mínima que un VO debe realizar. A continuación, se detallan los elementos que ChiVO ha implementado en estas capas.

- **Capa Aplicación:** Un Servicio Web compatible con VO necesita al menos un *Table Access Protocol* (TAP) [12] para acceder al modelo de datos de ChiVO usando el *Astronomical Data Query Language* [13]. Además, para el cumplimiento de los requerimientos del sistema, se ha implementado el protocolo para realizar búsquedas cónicas *Simple Cone Search* (SCS) [9], el protocolo para acceder a datos espectrales *Simple Spectral Access* (SSA) [11] y el protocolo de acceso a imágenes *Simple Image Access* (SIA) [10].
- **Capa de datos:** En esta capa se requiere la configuración de la base de datos relacional con un modelo de datos

recomendado por IVOA llamado *Observation Core Data Model (ObsCore DM)* [5]. Este modelo de datos permite que los VO sean interoperables, ya que definen una cantidad mínima de atributos en las tablas con cierto nombre y tipo de dato, de forma que acceder a diferentes servicios mediante *TAP + Obscore* es estándar. Además el formato de transferencia de información (metadata) es con el formato estándar *XML VOTable*.

#### A. Capa de abstracción: Clientes

Dentro del diagrama de secuencia (Fig. 5), el *Frontend* es el *VO-Client*, es decir, está a cargo de generar la interacción entre los usuarios y los demás elementos dentro del sistema. Actualmente se generó un *Frontend* en *Ruby on Rails* que permite interactuar con:

- **Name Resolver:** a partir de un nombre (*String*) retorna la posición de un objeto en coordenadas celestiales usando el servicio *SESAME* de *Astrogrid* [4].
- **Registry:** busca servicios dentro del *Endpoint*, y luego según lo que necesite el usuario elige sobre cuales trabajar para hacer consultas.
- **Services:** consulta a diferentes servicios web que proveen datos mediante cierto protocolo y unifica los resultados en un *XML VOTable* para ser desplegados en forma ordenada usando una biblioteca de *javascript VOView*.

#### B. Capa de abstracción: Aplicaciones

El *Endpoint* está a cargo de generar una interacción transparente entre los clientes de VO y los posibles recursos que se configuren. En este caso, el *Endpoint* genera una interfaz para los servicios configurados por el *Backend* y los disponibles a través del registro de *VO-Paris*. *VO-Paris* posee una Web API en REST que permite consultar por recursos y retorna un archivo *JSON* con resultados.

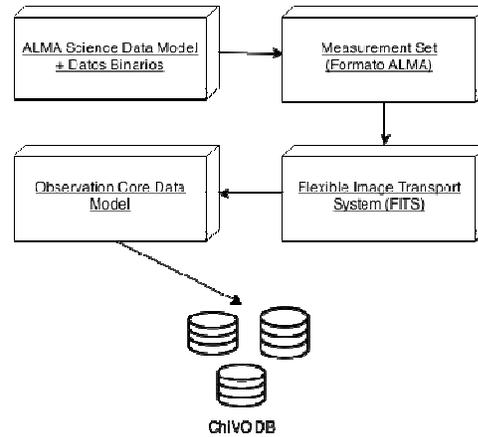
#### C. Capa de abstracción: Datos

Actualmente ALMA le facilita a ChiVO datos públicos, los cuales incluyen *ASDM*, *Measurement* [2] y *FITS* [15], de los cuales es necesario extraer los metadatos del *ObsCoreDM* para ser ingresados a la base de datos (Figura 6).

El procedimiento se hace a través de un programa, el cual usando el *framework* de *Backend DaCHS*, permite configurar recursos y servicios mediante archivos de configuración *Resource Descriptor* [7].

Una vez creadas las entradas en la base de datos y configurados los servicios *SCS*, *SSA*, *SIA*, *TAP* se pueden acceder mediante las consultas definidas en cada protocolo.

Fig. 6. Proceso transformación desde Frontend de ALMA hasta la base de datos de ChiVO.



## V. CONCLUSIÓN Y TRABAJO FUTURO

La implementación de ChiVO presentó varios desafíos, partiendo con entender las necesidades de los astrónomos y cómo éstas se traducen a sistemas de datos y servicios usando los estándares y protocolos internacionales de IVOA.

El enfoque de desarrollo estuvo orientado a prototipos incrementales, de tal forma que se generan entregables frecuentemente a los usuarios del sistema (Astrónomos), los cuales se ven satisfechos por los avances realizados. Sin embargo, siempre quedan cosas por implementar o aparecen nuevas ideas.

Como resumen se puede decir que se han capturado los requerimientos de los astrónomos, se han estudiado los protocolos y estándares para abordar éstos, creando así, los servicios de acceso a datos (*SCS*, *SIA*, *SSA*, *TAP*), los que acceden a una base de datos que usa un modelo de datos relacional (*ObsCore*), sujetos a la arquitectura propia de ChiVO, la cual permite que los servicios sean interoperables.

Hasta el momento el prototipo cuenta con los datos públicos del ciclo 0 de ALMA. Esto permitirá que más usuarios usen el sistema, por lo que se podrá testear en terreno las funcionalidades y limitaciones actuales de las implementaciones. Para las versiones mejoradas se tiene pensado implementar otros protocolos de la arquitectura de IVOA, como la sección de Registros y estándares de acceso a recursos, como *VOSpace* [16]. En cuanto a modelos y acceso de datos multidimensionales como los cubos de ALMA, será necesario trabajar en la creación o adaptación de estándares de IVOA.

## ACKNOWLEDGMENT

Este trabajo ha sido financiado parcialmente por el Proyecto de Investigación FONDEF D11I1060 y ICCHA 79130008.

## REFERENCES

- [1] CONICYT. Astronomy, technology, industry. roadmap for the fostering of technology development and innovation in the field of astronomy in Chile. 2012.
- [2] Dirk Petry et al. Analysing alma data with casa. arXivpreprint arXiv:1201.3454, 2012.
- [3] Francois Ochsenbein, Roy Williams, Clive Davenhall, Daniel Durand, Pierre Fernique, David Giaretta, Robert Hanisch, Tom McGlynn, Alex Szalay, Mark B Taylor, et al. Ivoa recommendation: Votable format definition version 1.2. arXiv preprint arXiv:1110.0524, 2011.
- [4] Sesame. <http://cds.u-strasbg.fr/cgi-bin/Sesame>. Online; accessed 04-24-2014.
- [5] Mireille Louys, Francois Bonnarel, David Schade, Patrick Dowler, Alberto Micol, Daniel Durand, Doug Tody, Laurent Michel, Jesus Salgado, Igor Chilingarian, et al. Ivoa recommendation: Observation data model core components and its implementation in the table access protocol version 1.0. arXiv preprint arXiv:1111.1758, 2011.
- [6] Gavo dachs. <http://vo.ari.uni-heidelberg.de/soft/>. Online; accessed 04-24-2014.
- [7] Gavo dachs. <http://docs.g-vo.org/DaCHS/opguide.html>. Online; accessed 04-24-2014.
- [8] F Viallefond and R Lucas. Sdm tables short description. Technical report, Tech. Rep. COMP-70.75. 00.00-007-A-DSN, Atacama Large Millimeter/submillimeter Array (apr 2010), 2009.
- [9] Roy Williams, Robert Hanisch, Alex Szalay, and Ray Plante. Simple cone search version 1.03. IVOA Data Access Layer WG Recommendation, 2008.
- [10] Doug Tody, Ray Plante, and Paul Harrison. Simple image access specification version 1.0. IVOA Recommendation, 20091111, 2004.
- [11] D Tody, M Dolensky, J McDowell, F Bonnarel, et al. Simple spectral access protocol. <http://www.ivoa.net/Documents/REC/DAL/SSA-20080201.pdf>. IVOA Standards, 2008.
- [12] Patrick Dowler, Guy Rixon, Doug Tody, K Andrews, J Good, R Hanisch, G Lemson, T McGlynn, K Noddle, F Ochsenbein, et al. Table access protocol version 1.0. IVOA Recommendation, March, 2010.
- [13] Naoki Yasuda, Yoshihiko Mizumoto, Masatoshi Ohishi, William O'Mullane, Tamas Budavari, Vivek Haridas, Nolan Li, Tanu Malik, AS Szalay, Martin Hill, et al. Astronomical data query language: simple query protocol for the virtual observatory. In *Astronomical Data Analysis Software and Systems (ADASS) XIII*, volume 314, page 293, 2004.
- [14] Guy Rixon and Paul Harrison. Universal worker service version 0.5, 2008.
- [15] DC Wells, EW Greisen, and RH Harten. Fits-a flexible image transport system. *Astronomy and Astrophysics Supplement Series*, 44:363, 1981.
- [16] Matthew Graham, Paul Harrison, Dave Morris, and Guy Rixon. Vospace service specification, version 1.01, 2007.