Design and Implementation of JVO SkyNode

IVOA Small Projects Meeting, 2004 Oct 1

Yuji SHIRASAKI

National Astronomical Observatory of Japan

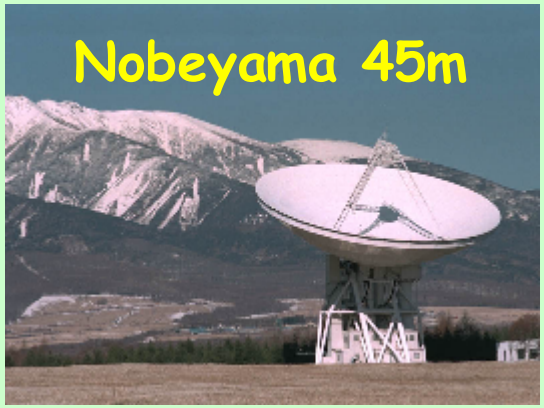
Current JVO activities

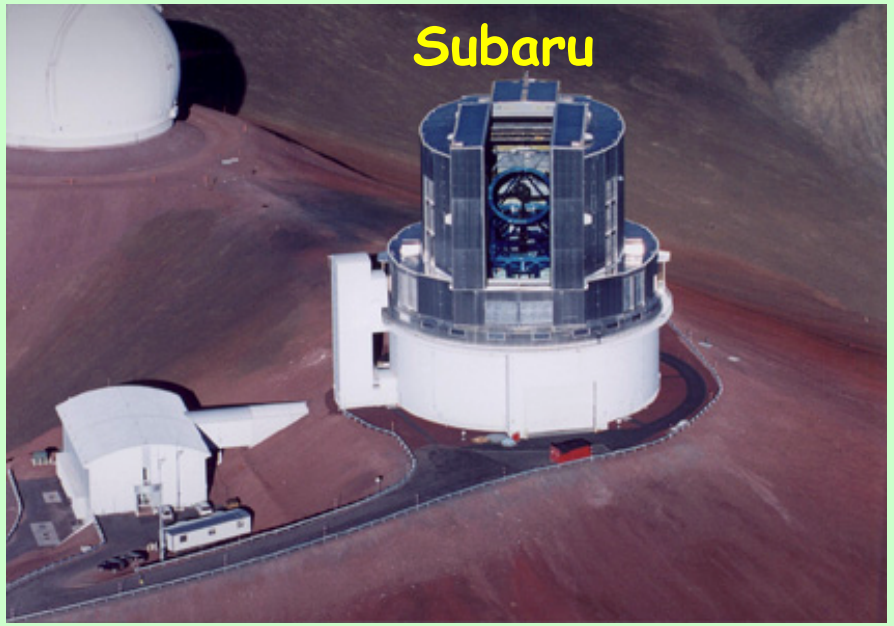
We are focusing on the following subjects:

- contribution to the IVOA activity in designing the VO query language (VOQL WG)
- development of a toolkit for making existing data services VO compliant (JVO SkyNode toolkit),
- development of a VO portal service (Masahiro Tanaka)
- VO enabled science using JVO system (Satoshi Honda)

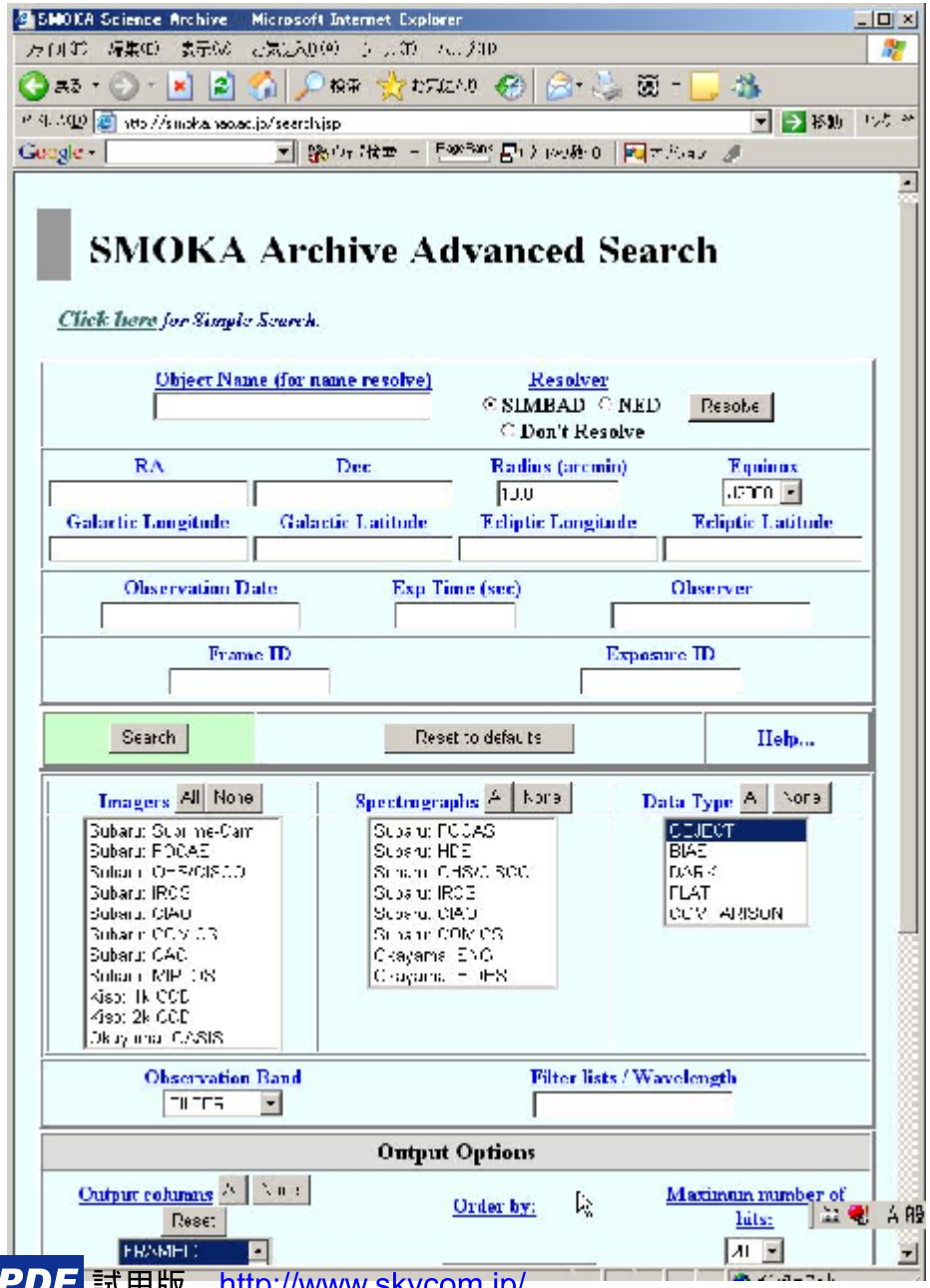
Data Resources in NAOJ

- Subaru 8.2m Optical-Infrared Telescope
- Kiso 105cm Schmidt Camera
- Okayama 188cm Optical Telescope
- Nobeyama 45m Radio Telescope
- Nobeyama Millimeter Array
- Nobeyama Radioheliograph
- VSOP
- VERA
- *ALMA*





SMOKA Archive in NAOJ



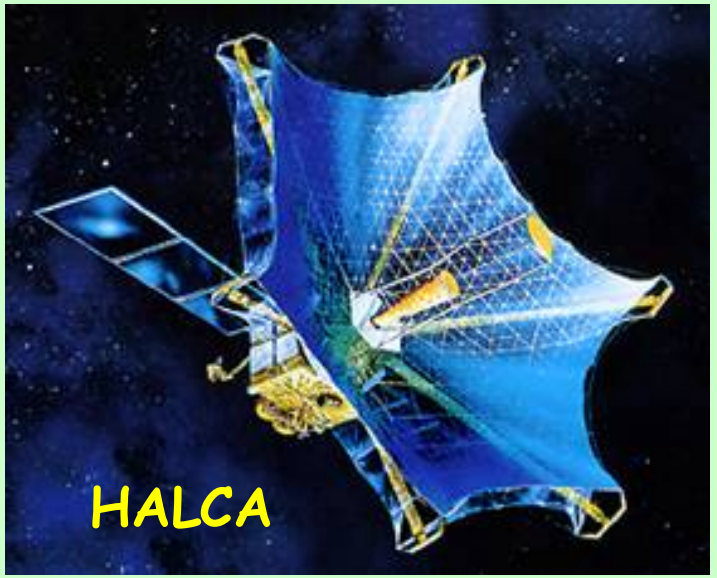
http://smoka.nao.ac.jp/

Public science archive of the
- Subaru Telescope,
- 188cm telescope at Okayama Astrophysical Observatory,
- 105cm Schmidt telescope at Kiso Observatory / University of Tokyo.

Reduced data of Subaru Suprime-Cam is now available.

Data Resources in JAXA/ISAS

- ASCA X-ray astronomy satellite
- YOHKO solar physics satellite
- Ginga X-ray astronomy satellite
- HALCA VLBI satellite
- Geotail geomagnetosphere satellite
- Akebono aurora observation satellite
- *ASTRO-F Infrared satellite*
- *ASTRO-E2 X-ray satellite*
- *SOLAR-B*



DARTS

JVO Query Language (JVOQL)

JVOQL is designed as a prototype of VO Query Language.

Characteristics of the JVOQL:

- SQL based Query Language
- Query Language for the distributed astronomical DB.
- Can search and retrieve observational data as well as catalog data
- Upward compatible with the ADQL and SIAP syntax.
- Scalable syntax structure. Very simple core syntax and extension syntax packages.

JVOQL Syntax Requirement 1

• Unified query language for both the catalog and observation data such as image data, spectrum, 3D-cube, photon list …

- Parameter query (SIAP) can be replaced by SQL thinking that the parameters are columns of a relational table.

- Observational data or pointer (URL) to retrieve the data is also a column of the relation table.

```
http://jvo.nao.ac.jp/imageData?Pos=24,5&Size=0.2&format=VOTable
```

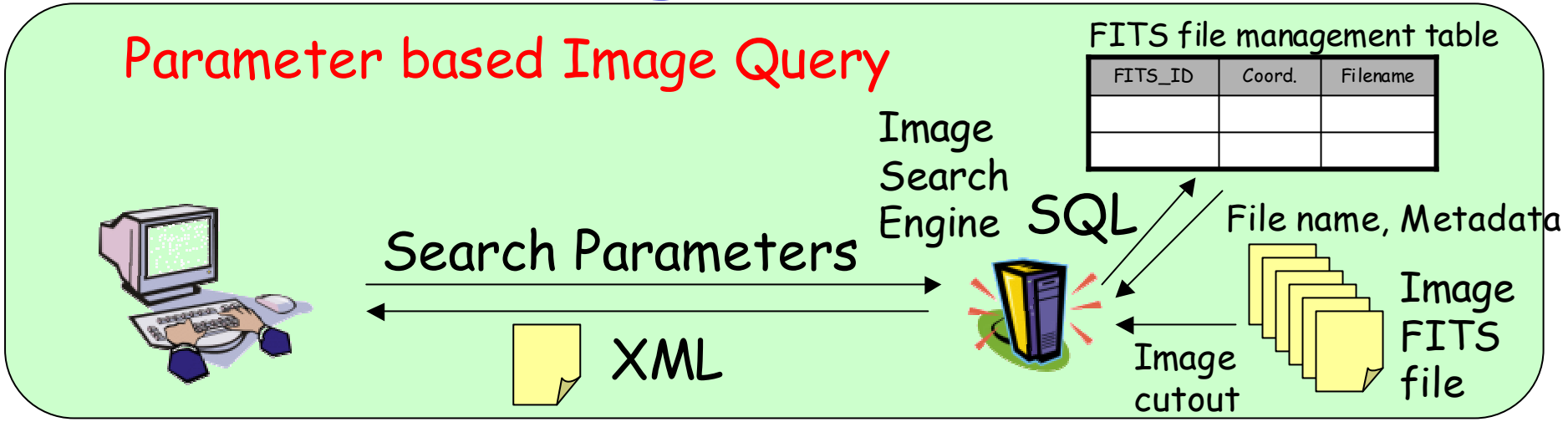
↓

```
Select      imageURL, …
From        naoj:imageData
Where       pos=Point(24,5) and size=0.2 and format='VOTable'
```

"pos" , "size", "imaegURL" are virtual columns.

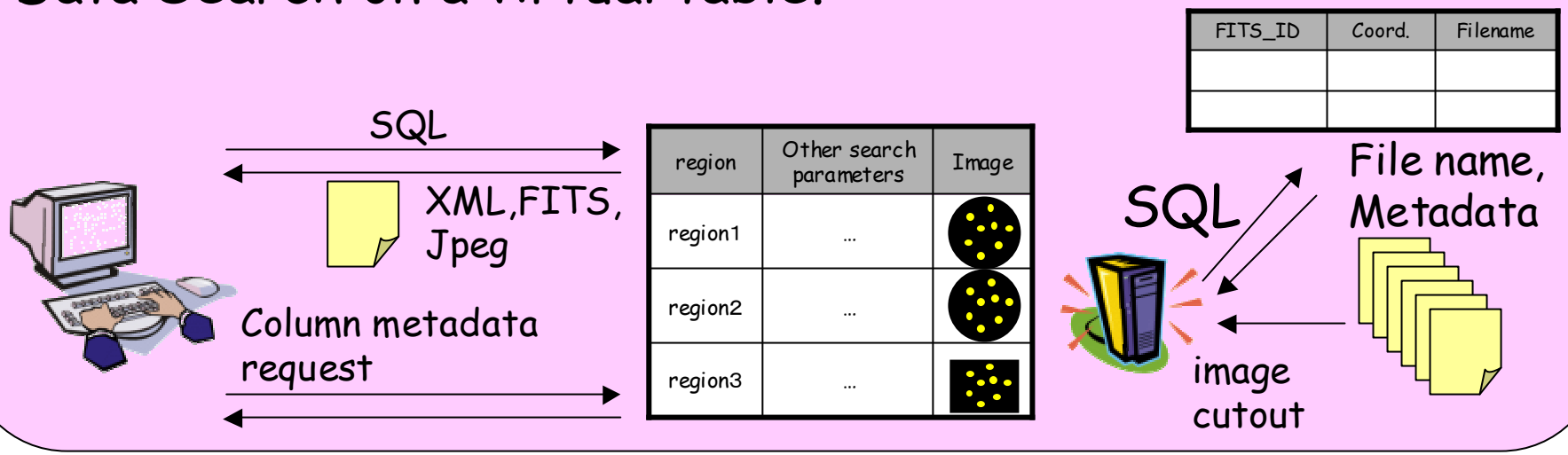
Image Search

Parameter based Image Query



SQL based Image Query

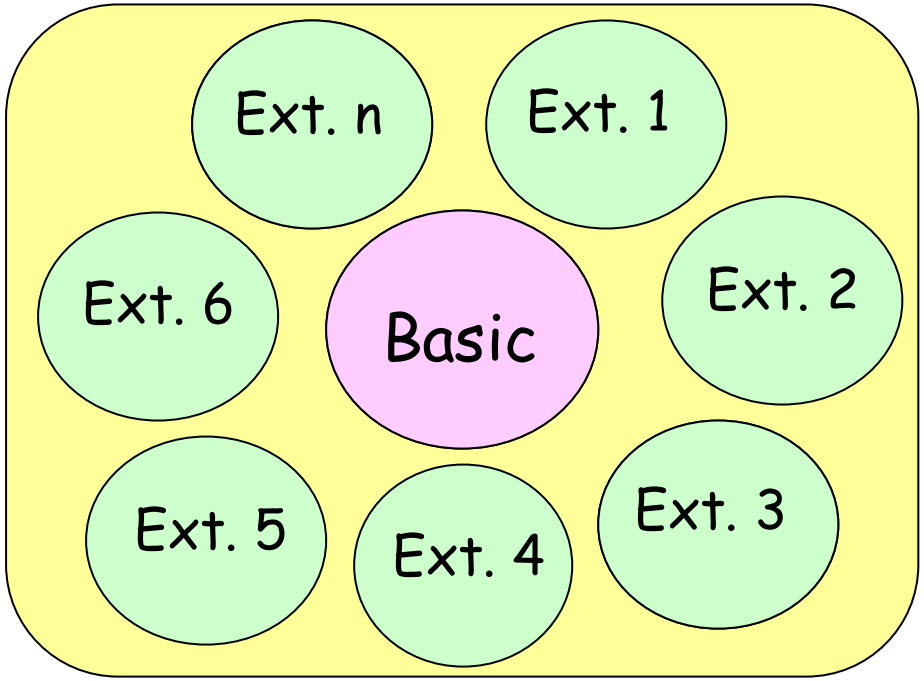
Data Search on a virtual table.



JVOQL Syntax Requirement 2

- VOQL should have scalable syntax:
  - Small size DBs → very simple syntax for easy implementation.
  - Large size DBs → sophisticated syntax for efficient data search.

  → hybrid syntax structure : a basic syntax to be implemented by all the VO data service and extension syntax packages.



Information which extensions are implemented will be registered in the registry service or data service itself returns the information through "voqlSpec" interface.

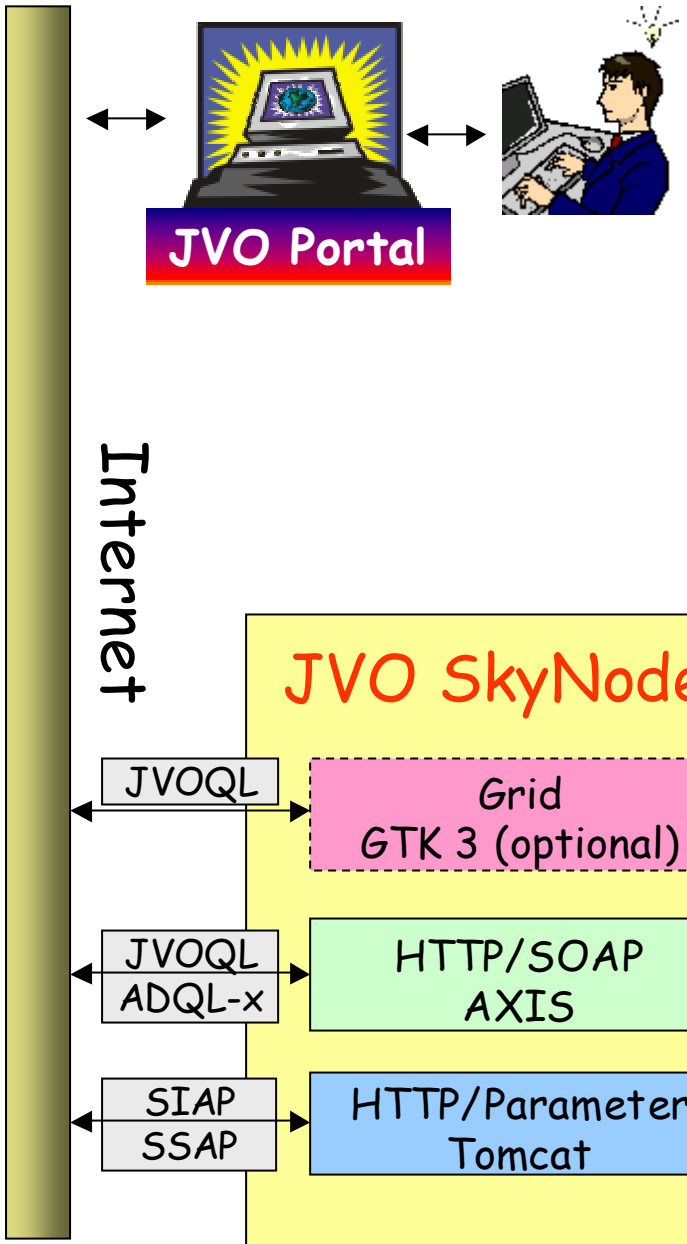
Development of JVO Skynode Toolkit

JVO Skynode :

- can accept ADQL-x over HTTP/SOAP, SIAP over HTTP/Parameters → VO compliant,
- can accept JVOQL over HTTP/SOAP and Grid (experimental) . → functionality test of JVOQL,
- returns VOTable, CSV file and FITS file,

The JVO Skynode toolkit is intended to be used as an wrapper for existing data services to become VO compliant → easy and quick implementation of the skynode interface on the existing system.

JVO SkyNode Architecture

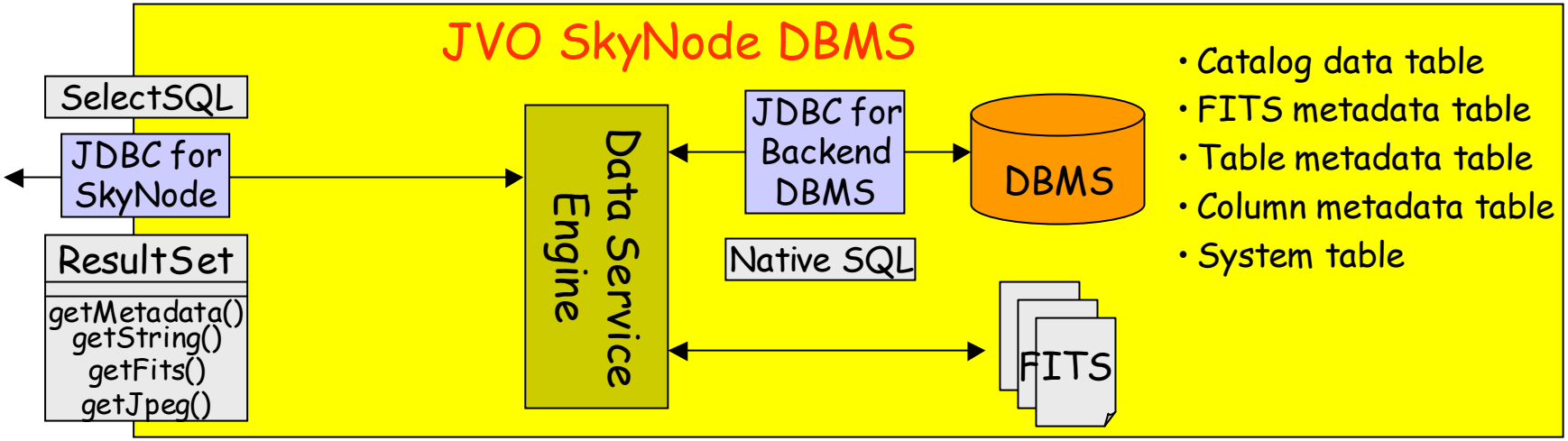


- Apache Axis and Tomcat are used for implementing the VO standard interface, and Globus Toolkit is used for an optional Grid interface.
- Four kinds of query languages are converted to java class SelectSQL and query is executed through JDBC-like interface.
- Query results is obtained as ResultSet, and table data is formatted to VOTable or CSV file.

JVO SkyNode DBMS

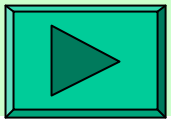
JVO SkyNode DBMS :

- is an astronomical database system which accept JVOQL syntax and return observation data as well as tabular data ,
- includes DBMS which is used to store catalog data, FITS file metadata, and system information,
- can access to observational data of FITS files which are managed by unix file system,
- implemnts a JDBC-like interface, search request can be specified by SelectSQL java class and result is returned as ResultSet.



Free software used in JVO Skynode

- Java (J2SE) : Generally used in the development.
- JAXB 1.0: used for generating Java class files from VO standard schema, ADQL, VOTable, VOResource etc…
- JavaCC : used for parsing JVOQL and constructing SelectSQL java object.
- PostgreSQL : Backend DBMS.
- HTM library : developed by JHU, used for region search.
- Apache AXIS, Tomcat : Web service and servlet.
- Globus Toolkit : Grid service.
- etc…



JVOQL ←→ ADQL-x

```
Select    ra, dec, mag
From      spcam
Where     (ra, dec) within
          Point((20, 30), 1.0)
```

JVOQL2Parser parses JVOQL and constructs SelectSQL object. SelectSQL object has a method to return a SelectType object which is a data model defined in ADQL schema.

This is an example of translation from JVOQL to ADQL-x.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Select xmlns:ns1="urn:vo-coord" xmlns:region="urn:vo-region"
        xmlns="http://www.ivoa.net/xml/ADQL/v0.8">
  <SelectionList>
    <Item Table="spcam" Name="ra" xsi:type="columnReferenceType"/>
    <Item Table="spcam" Name="dec" xsi:type="columnReferenceType"/>
    <Item Table="spcam" Name="mag" xsi:type="columnReferenceType"/>
  </SelectionList>
  <From>
    <Table Name="spcam" Alias="" xsi:type="tableType"/>
  </From>
  <Where>
    <Condition xsi:type="regionSearchType">
      <Region> <region:Circle unit="deg" coord_system_id="IRCS">
        <region:Center>20.0 30.0</region:Center> <region:Radius>1.0</region:Radius>
        </region:Circle>
      </Region>
    </Condition>
  </Where>
</Select>
```

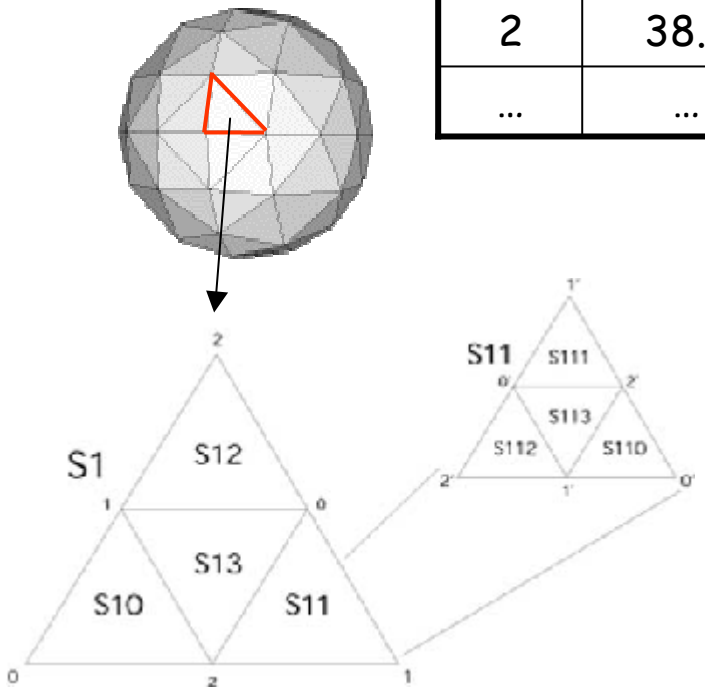
JVOQL ←→ ADQL-x

```
Select   opt.ra as ra, opt.dec as dec, (opt.R - opt.B) as color, opt.mag, x.flux
From     spcam opt, xmm x
Where    (opt.ra, opt.dec) within ((20, 30), 1.0)
         and Distance((opt.ra, opt.dec), (x.ra, x.dec)) < 1
```


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <Select xmlns:ns1="urn:vo-coord"
xmlns:region="urn:vo-region" xmlns="http://www.ivoa.net/xml/ADQL/v0.8"> <SelectionList> <Item
As="ra" xsi:type="aliasSelectionItemType"> <Expression Table="opt" Name="ra"
xsi:type="columnReferenceType"/> </Item> <Item As="dec" xsi:type="aliasSelectionItemType">
<Expression Table="opt" Name="dec" xsi:type="columnReferenceType"/> </Item> <Item As="color"
xsi:type="aliasSelectionItemType"> <Expression xsi:type="closedExprType"> <Arg Oper="-"
xsi:type="binaryExprType"> <Arg Table="opt" Name="R" xsi:type="columnReferenceType"/> <Arg
Table="opt" Name="B" xsi:type="columnReferenceType"/> </Arg> </Expression> </Item> <Item
Table="opt" Name="mag" xsi:type="columnReferenceType"/> <Item Table="x" Name="flux"
xsi:type="columnReferenceType"/> </SelectionList> <From> <Table Name="spcam" Alias="opt"
xsi:type="tableType"/> <Table Name="xmm" Alias="x" xsi:type="tableType"/> </From> <Where>
<Condition xsi:type="intersectionSearchType"> <Condition xsi:type="regionSearchType"> <Region>
<region:Circle unit="deg" coord_system_id="IRCS"> <region:Center>20.0 30.0</region:Center>
<region:Radius>1.0</region:Radius> </region:Circle> </Region> </Condition> <Condition Comparison="&lt;"
xsi:type="comparisonPredType"> <Arg xsi:type="userDefinedFunctionType">
<Name>GCdistance</Name> <Params xsi:type="userDefinedFunctionType"> <Name>Point</Name>
<Params Table="opt" Name="ra" xsi:type="columnReferenceType"/> <Params Table="opt" Name="dec"
xsi:type="columnReferenceType"/> </Params> <Params xsi:type="userDefinedFunctionType">
<Name>Point</Name> <Params Table="x" Name="ra" xsi:type="columnReferenceType"/> <Params
Table="x" Name="dec" xsi:type="columnReferenceType"/> </Params> </Arg> <Arg
xsi:type="atomType"> <Literal Value="1.0"/> </Arg> </Condition> </Condition> </Where> </Select>
```

Region Search using HTM index

Region search is a common search criterion for an astronomical database. For efficient search data should be properly indexed on the object coordinate.



http://www.sdss.jhu.edu/htm/

Catalog table

id	ra	dec	mag
1	12.3	-23.4	18.4
2	38.5	+34.2	16.5
...	...	...	...

HTM Index table

id	htm
1	16522516
2	16754765
...	...

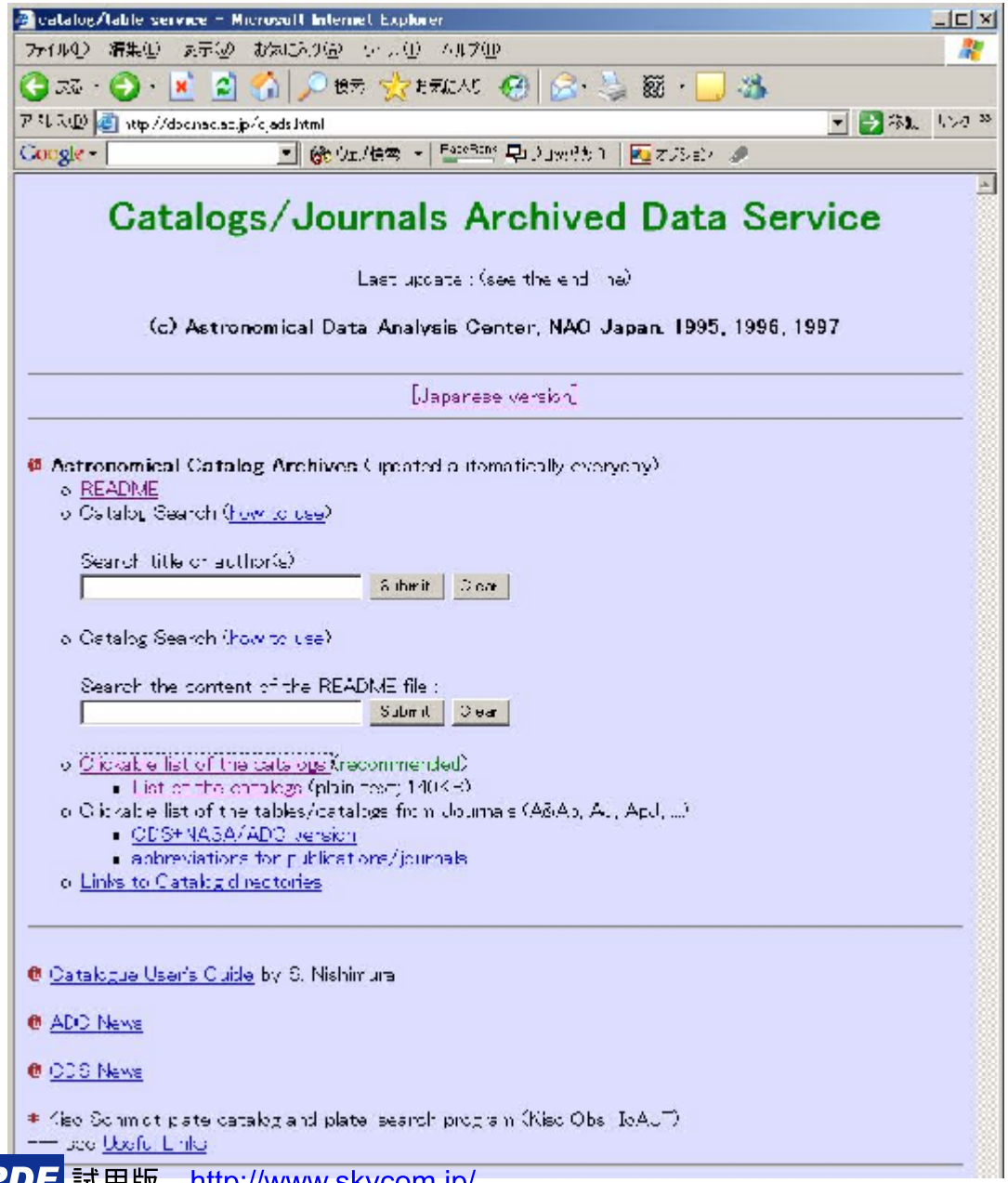
```
Select   ra, dec, mag
From     Catalog
Where    Point(ra,dec) within Box((20,+15), 1.0)
```

```
Select   c.ra, c.dec, c.mag
From     Catalog as c
         Natural Left Join htmIndex as i
Where    i.htm between 16522500 and 16522512
         OR
         i.htm between 16522500 and 16522512
         ...
```

Progress

- We have started Skynode implementation a few weeks ago.
- First release will complete in December and will be applied to the data of Subaru Deep survey.
- We plan to apply the JVO SkyNode toolkit to the Subaru Sprime-Cam reduced database next year.

Catalog Data Service in NAOJ



Basic Specification

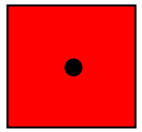
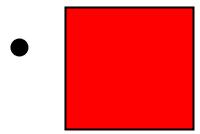
```
Select   ColumnName [[AS] AliasName]、…  | *
From     TableName [[AS] AliasName]
Where    Condition [AND Condition]
```

- Only column name or "*" is specified in the selection list.
- Don't support an algebraic expression.
- Only one table is specified at "From" part.
- Table name and Column name can have alias name.
- Comparison operators: =, <, >, >=, <=, <>, LIKE, BETWEEN
- Region Comparison operator: =, within, contains, overlaps
- Logical operator: AND and NOT (OR is not supported)
- Functions: Distance(), Point(), Circle(), Box()

Region Comparison

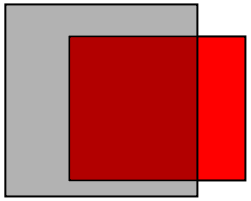
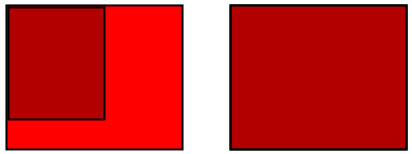
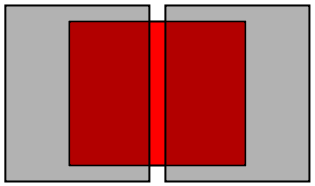
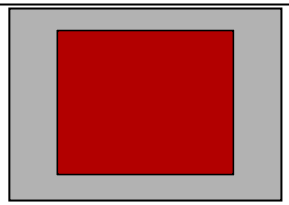
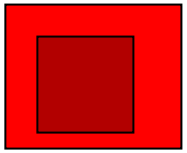
[<SpacePoint> <RegionCompOper>] <SpaceRegion>

<SpaceRegion> [<RegionCompOper> <SpacePoint>]

Region Comparison	Meaning	Image Atlas Data Service
A within B	Point A is within Region B.	
A contains B	Region A contains Point B.	
A outside B	Point A is outside Region B.	
A excludes B	Region A excludes Point B.	

Region Comparison

<SpaceRegion> <RegionCompOper> <SpaceRegion>

Region Comparison	Meaning	Image Atlas Data Service	Image Cutout Service
A = B	Region A is the smallest region which overlaps the largest part of B.		
A overlaps B	Region A is the smallest region which overlaps B.		Same as A = B
A contains B	Region A is the smallest region which contains B.		
A within B	Region A is the largest region which is contained in B.		Same as A = B

Example for basic spec. JVOQL

Catalog Query for the specified region

```
Select   ra, dec, mag_r
From     galaxy
Where    Point(ra, dec)  within  Circle((24.3, +5.0), 2.0))
          and mag_r < 24
```

Image Query for the specified region.

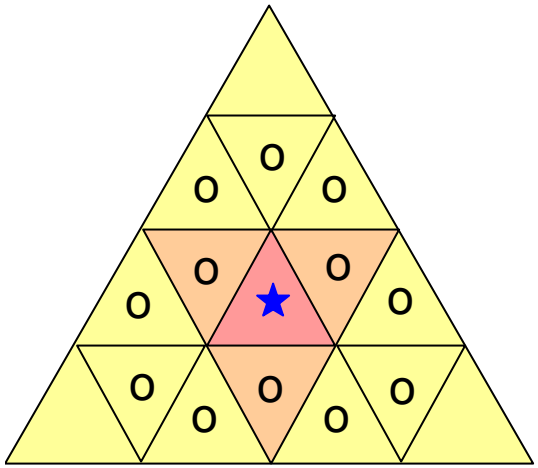
```
Select   filter, image
From     imageData
Where    region = Box((24.3, +5.0), 0.2))
```

pos = Point(24.2, 5.0) and size = 0.2

c.f. http://jvo.nao.ac.jp/imageData?POS=24.2,5.0&SIZE=0.2

Simple quick cross match procedure

- Cross identification of the object in two catalogs are fundamental procedure in the astronomical analysis.
- The simplest way is doing NxM sequential search, which is a time consuming process if N and M is large.
- Alternatively, dividing the data into n partitions according to the htm index, we can reduce the time by a factor n if data access time is ignored.



1. Determine the size and level of a triangle from the specified cross match precision.
2. From table A, select data of the same htm index.
3. From table B, select data of the neighbor htm index (o and star in the figure).
4. Try N'xM' sequential search.
5. Repeat 2~4 for all object in table A.