

VO標準規格の解説1

リソースメタデータ・VOTable・UCD

Resource Metadata

天文データのデータベース

- 様々な組織がインターネットを通じて天文データの公開をおこなっている。
 - CDS, ESAC, MAST, HEASARC, IPAC, ADAC, ISAS...
- どこにどんなデータがあるのかといったイエローページ的なものが必要になってきた。
- 天文学に関連したリソースの情報をメタデータとして記述し、データベース化しよう。

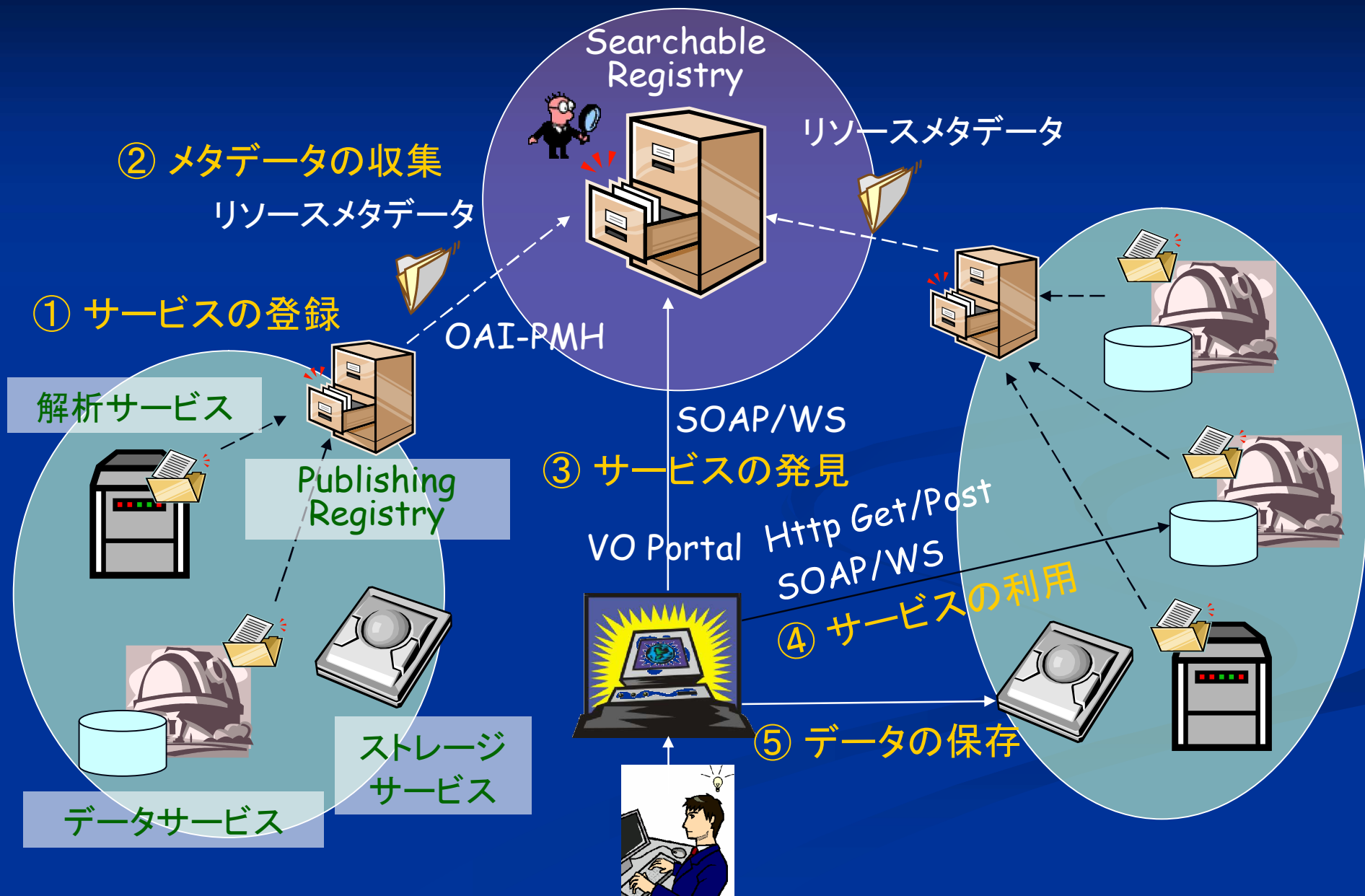
メタデータ

- Virtual Observatory におけるメタデータとは
 - 天文学に関するあらゆるリソースを記述するデータ
 - 天文カタログのメタデータ
 - 誰が作ったか。観測に使われた望遠鏡。観測波長域。観測時刻。データ取得方法。
 - 天文データサービスのメタデータ
 - どういった種類のサービスか。Registry サービス、画像検索サービス、カタログ検索サービス、天体検出サービス。
 - アクセス方法。
 - 観測装置に関するメタデータ
 - 組織に関するメタデータ
 - ...

IVOA Registry Working Group

- メタデータの項目と記述フォーマットの標準化
 - XML Schema
- メタデータの交換方式の標準化
 - 公開: Publishing Registry
 - 検索: Searchable Registry

メタデータの交換方法



Resource Metadata

- <http://www.ivoa.net/Documents/REC/ResMetadata/RM-20040426.html>

リソースメタデータの内容1

- リソース識別メタデータ:
 - Title, ShortName, Identifier
- Curation メタデータ
 - Publisher, PublisherID, Creator, Creator.Logo, Contributor, Date, Version, Contact.Name, Contact.Email
- General content メタデータ
 - Subject, Description, Source, ReferenceURL, Type, ContentLevel, Relationship, RelationshipID

リソースメタデータの内容2

- **Type:** リソースの種別を以下から選択
 - Archive, Bibliography, Catalog, Journal, Library, Simulation, Survey, Education, Outreach, EPOResource, Animation, Artwork, Background, BasicData, Historical, Photographic, Press, Organization, Project, Registry, Other
- **ContentLevel:** リソースの利用対象者
 - General, Elementary Education, Middle School Education, Community College, University, Research, Amateur, Informal Education
- **RelationshipID:** 他のリソースと関連がある場合
 - Mirror-of, service-for, derived-from

リソースメタデータの内容3

■ データの内容に関するメタデータ

- Facility, Instrument, Coverage.Spatial, Coverage.RegionOfRegard, Coverage.Spectral, Coverage.Spectral.Bandpass, Coverage.Spectral.CenterWavelength, Coverage.Spectral.MinimumWavelength, Coverage.Spectral.MaximumWavelength, Coverage.Temporal.StartTime, Coverage.Temporal.StopTime, Coverage.Depth, Coverage.ObjectDensity, Coverage.ObjectCount, Coverage.SkyFraction, Resolution.Spatial, Resolution.Spectral, Resolution.Temporal, UCD, Format, Rights

リソースメタデータの内容4

■ サービスに関するメタデータ

■ Interface メタデータ

■ Service.InterfaceURL

- WSDL の URL または検索用 Web Form を含むページの URL

■ Service.BaseURL

- サービスを実行する URL. クエリパラメータが適宜必要。

■ Capability メタデータ

- Service.StandardURI, Service.StandardURL,
Service.MaxSearchRadius, Service.MaxReturnRecords

メタデータ記述フォーマット

- <http://www.ivoa.net/xml/VOResource/VOResource-v1.0.xsd>
 - <http://www.ivoa.net/xml/SIA/SIA-v1.0.xsd>
 - <http://www.ivoa.net/xml/VODataService/VODataService-v1.0.xsd>
 - <http://www.ivoa.net/xml/VORegistry/VORegistry-v1.0.xsd>
 - <http://www.ivoa.net/xml/OpenSkyNode/OpenSkyNode-v0.1.xsd>

JVO Publishing Registry

- リソースメタデータファイルの作成は JVO Publishing Registry を利用すれば容易。
 - <http://gridp.dc.nao.ac.jp:2000/regedit>
- 現在は非公開、パッケージ化して配布予定
- 試験用 Publishing Registry が以下にある。
 - <http://jvo.nao.ac.jp/p.registry.dev/>

Publishing Registry I/F

- OAI-PMH (Open Archives Initiative) (必須)
 - <http://www.nii.ac.jp/metadata/oai-pmh2.0/OpenArchivesProtocol.htm>
- Web service version of OAI interface (option)
- Operations:
 - identify, listIdentifiers, listRecords, getRecord, listSets
- Arguments:
 - from, until (listIdentifier, listRecords):
 - 更新日時による選択的メタデータの収集が可能
 - set (ListIdentifier, ListRecords)

Searchable Registry IF

- Web サービスで実装する。
- Operation:
 - Search, KeywordSearch, GetResource, GetIdentity
- Arguments:
 - Search: **condition**, from, max, identifierOnly
 - KeywordSearch: **keywords**, orValues, from, max, identifierOnly
- Output:
 - Resource Metadata **または Identifier のリスト** (Search, KeywordSearch)
 - Resource Metadata **をひとつ** (GetResource, GetIdentity)
- I/F は WSDL (Web Services Description Language) で記述される。
 - <http://www.ivoa.net/xml/RegistryInterface/RegistryInterface-v0.1.xsd>

keyword search I/F の schema 定義

```
<element name="KeywordSearch">
  <complexType>
    <sequence>
      <element name="keywords" type="string" />
      <element name="orValues" type="boolean" default="false" />
      <element minOccurs="0" name="from" type="positiveInteger" />
      <element minOccurs="0" name="to" type="positiveInteger" />
      <element minOccurs="0" name="identifiersOnly" type="boolean" />
    </sequence>
  </complexType>
</element>

<KeywordSearch>
  <keywords>HDF</keywords>
  <orValues>>false</orValues>
</KeywordSearch>
```

これを SOAP message として HTTP/Post でサーバに送る。

POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

```
<Envelope
  xmlns="http://schemas.xmlsoap.org/soap/envelope/"
  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  <Header>
    <t:Transaction xmlns:t="some-URI" mustUnderstand="1">
      5
    </t:Transaction>
  </Header>
  <Body>
    ここに書く
  </Body>
</Envelope>
```

検索結果の schema 定義

```
<element name="SearchResponse">
```

```
  <complexType>
```

```
    <sequence>
```

```
      <element ref="ri:VOResources" />
```

```
    </sequence>
```

```
  </complexType>
```

```
</element>
```

```
<SearchResponse>
```

```
  <VOResource from="1" numberReturned="10" more="false">
```

```
    <Resource xsi:type="...">...</Resource>
```

```
    <Resource xsi:type="...">...</Resource>
```

```
  </VOResource>
```

```
</SearchResponse>
```

```
<xs:element name="VOResources">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element name="Resource" type="vr:Resource" minOccurs="0"
          maxOccurs="unbounded" />
        <xs:element name="identifier" type="vr:IdentifierURI" minOccurs="0"
          maxOccurs="unbounded" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="from" type="xs:positiveInteger" />
    <xs:attribute name="numberReturned" type="xs:positiveInteger" />
    <xs:attribute name="more" type="xs:boolean" />
  </xs:complexType>
</xs:element>
```

VOTable

VOTable

- VOTable はデータの受け渡しを行うための文書である。
- VOTable は XML (eXtensible Markup Language) の規約に則って書かれた文書である。
 - HTML はタグ名は勝手に決められない。
 - XML は自由に定義することができる。
- 整形形式文書 (well-formed XML document)
- 妥当な文書 (valid XML document)
 - DTD, XML Schema により文書の構成を定義
- VOTable は VOTable Schema に則った妥当な文書である。

XML 文書の例

```
<?xml version="1.0" encoding="Shift_JIS"?> XML宣言
```

```
<!-- 簡単な XML 文書 --> コメント
```

```
<cars xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="car.xsd">
```

属性

```
<car id="1001" country="日本">
```

```
<name>乗用車</name> 要素
```

```
<price>150</price>
```

```
</car>
```

```
</cars>
```

ルート
要素

<要素名 属性名1="属性値1" 属性名2="属性値2"> データ </要素名>

要素名やデータ型は XML Schema によって定義される。

XML Schema の例

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema targetNamespace="car.xsd" xmlns="car.xsd"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <element name="cars" type="carsType"/>
  <element name="car" type="carType"/>
  <element name="name" type="string"/>
  <element name="price" type="string"/>

  <complexType name="carsType">
    <element ref="car" maxOccurs="unbounded"/>
  </complexType>

  <complexType name="carType">
    <sequence>
      <element ref="name"/>
      <element ref="price"/>
    </sequence>
    <attribute name="id" type="int"/>
    <attribute name="country" type="string"/>
  </complexType>
```

VOTable の情報源

- IVOA VOTable Working Group
 - <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaVOTable>
- VOTable Schema Version 1.1
 - <http://www.ivoa.net/internal/IVOA/IvoaVOTable/VOTable-1.1.xsd>
- VOTable I/O Libraries and Tools
 - <http://www.ivoa.net/internal/IVOA/IvoaVOTable/VOTableAvailableSoftware.html>

VOTable の例

```
<?xml version="1.0"?>
<VOTABLE version="1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://vizier.u-
strasbg.fr/xml/VOTable-1.1.xsd">
<DEFINITIONS>
<COOSYS ID="J2000" equinox="2000." epoch="2000."
system="eq_FK5"/>
</DEFINITIONS>
<RESOURCE name="myFavouriteGalaxies">
<TABLE name="results">
<DESCRIPTION>Velocities and Distance
estimations</DESCRIPTION>
<PARAM name="Telescope" datatype="float"
ucd="TEL_SIZE" unit="m"
value="3.6"/>
<FIELD name="RA" ID="col1" ucd="POS_EQ_RA_MAIN"
ref="J2000" datatype="float"
width="6" precision="2" unit="deg"/>
<FIELD name="Dec" ID="col2" "POS_EQ_DEC_MAIN"
ref="J2000" datatype="float"
width="6" precision="2" unit="deg"/>
<FIELD name="Name" ID="col3" ucd="ID_MAIN"
datatype="char" arraysize="8"/>
<FIELD name="RVel" ID="col4" ucd="VELOC_HC"
datatype="int"
width="5" unit="km/s"/>
```

```
<FIELD name="e_RVel" ID="col5" ucd="ERROR"
datatype="int"
width="3" unit="km/s"/>
<FIELD name="R" ID="col6" ucd="PHYS_DISTANCE_TRUE"
datatype="float"
width="4" precision="1" unit="Mpc">
<DESCRIPTION>Distance of Galaxy, assuming
H=75km/s/Mpc</DESCRIPTION>
</FIELD>
<DATA>
<TABLEDATA>
<TR>
<TD>010.68</TD><TD>+41.27</TD><TD>N 224</TD><TD>-
297</TD><TD>5</TD><TD>0.7</TD>
</TR>
<TR>
<TD>287.43</TD><TD>-63.85</TD><TD>N
6744</TD><TD>839</TD><TD>6</TD><TD>10.4</TD>
</TR>
<TR>
<TD>023.48</TD><TD>+30.66</TD><TD>N 598</TD><TD>-
182</TD><TD>3</TD><TD>0.7</TD>
</TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
```

VOTable (Data Structure)

VOTABLE

DESCRIPTION

COOSYS...

PARAM...

INFO...

RESOURCE...

DESCRIPTION

INFO...

COOSYS...

PARAM...

LINK...

TABLE...

RESOURCE...

TABLE

@ID, @name, @ref, @ucd, @utype, @nrows

DESCRIPTION

FIELD...

PARAM...

GROUP...

LINK...

DATA

FIELD

@ID, @unit, @datatype, @precision, @width,
@ref, @name, @ucd, @utype, @arraysize, @type

DESCRIPTION

VALUES

LINK

DATA

TABLEDATA | BINARY | FITS

属性 ID と name

大部分の要素は属性として **name** と **ID** を指定できる。

- name 属性は presentation purpose として使う。
 - VOTable 内で重複しても構わない。
- ID 属性は参照するために使う。
 - VOTable 内で重複してはならない。
 - 使える文字種に制限。(XML の ID に関する規約)

<FIELD>

属性

ID	元テーブルの列番号を推奨。e.g. ID="col3"
name	[必須] 列名
unit	Standards for Astronomical Catalogues, Version 2.0
datatype	[必須] boolean, bit, unsignedByte, short, int, long, char, unicodeChar, float, double, floatComplex, doubleComplex のいずれか。文字列は char として arraysize で文字数指定
precision	数値の精度。precision="F2" or precision="E5"
width	数値の桁数。
ref	他の要素を ID で参照する。
ucd	Unified Content Descriptors
utype	usage-specific type. 特定の目的のために定義されるデータタイプ。
arraysize	配列の大きさ。
子要素	
DESCRIPTION	説明文
VALUES	属性値の追加。値の範囲、nullの値、
LINK	関連する情報をもつURL

FIELD 要素の例

```
<FIELD name="RA" ucd="pos.eq.ra" ref="J2000" datatype="float"
width="6" precision="2" unit="deg"/>
```

```
<FIELD name="Dec" ucd="pos.eq.dec" ref="J2000"
datatype="float" width="6" precision="2" unit="deg"/>
```

```
<FIELD name="Name" ucd="meta.id" datatype="char" arraysize="8"/>
```

```
<FIELD name="RVel" ucd="phys.veloc;pos.heliocentric" datatype="float"
width="5" unit="km/s"/>
```

```
<FIELD name="R" ucd="pos.distance" datatype="float"
width="4" precision="1" unit="Mpc">
<DESCRIPTION>Distance of Galaxy, assuming
H=75km/s/Mpc</DESCRIPTION> </FIELD>
```

```
<FIELD name="references" ucd="meta.bib.bibcode" datatype="char"
arraysize="20x*" />
```

<PARAM>

用法 VOTable 内で参照されるパラメータの定義を行う。

属性・要素のデータ型

<FIELD> とほぼ同じ。唯一の違いは "value" 属性を持つこと。

例

```
<TABLE name="results">  
  <DESCRIPTION>Velocities and Distance  
    estimations</DESCRIPTION>  
  <PARAM name="TelescopeSize" datatype="float"  
    ucd="TEL_SIZE" unit="m" value="3.6"/>  
  ...  
</TABLE>
```

<GROUP>

FIELD 要素をグループ化するのに利用

例: Flux 値とその誤差を表すカラムをグループ化する。測定周波数をパラメータで定義する。

```
<FIELD id="col1" name="Flux" datatype="float" unit="mJy"/>
```

```
<FIELD id="col2" name="e_Flux" datatype="float" width="4" unit="mJy"/>
```

```
<GROUPE name="Flux">
```

```
  <FIELDref ref="col1"/>
```

```
  <FIELDref ref="col2"/>
```

```
  <PARAM name="Freq" unit="MHz" datatype="float" value="352"/>
```

```
  <DESCRIPTION>Flux measured at 352MHz</DESCRIPTION>
```

```
</GROUPE>
```

<TABLEDATA>

<TABLE>

<FIELD ID="aString" datatype="char" arraysize="10"/>

<FIELD ID="Floats" datatype="float" arraysize="3"/>

<FIELD ID="varComplex" datatype="floatComplex" arraysize="*" />

<DATA×TABLEDATA>

<TR>

<TD>Apple</TD>×<TD>1.62 4.56 3.44</TD>

<TD>67 1.57 4 3.14 77 -1.57</TD>

</TR>×<TR>

<TD>Orange</TD>×<TD>2.33 4.66 9.53</TD>

<TD>39 0 46 3.14</TD>

</TR>

</TABLEDATA>×</DATA>

</TABLE>

aString	Floats	varComplex
Apple	1.62 4.56 3.44	67+1.57i 4+3.14i 77-1.57i
Orange	2.33 4.66 9.53	39+0i 46+3.14i

XML パーサの種類

- DOM : メモリロード型
- SAX : シーケンシャル読み込み
- XML Pull : シーケンシャル読み込み

VOTable の読み書きソフト

Java Parsers

<http://www.ivoa.net/internal/IVOA/IvoaVOTable/VOTableAvailableSoftware.html>

Sr. Num.	Name of parser	Description	コメント
1	JAVOT (NVO)	Supports reading, writing	メモリロード型。 2002年以降更新されていない。
2	SAVOT (European VO)	Support reading, writing and editing.	メモリロード型とシーケンシャル読み込み型。 書き込みをメモリロードが必要。
3	VOTable Java Streaming Writer (VO-India)	Supports writing.	シーケンシャル書き込み型。 大規模データの書き出しに有効。
4	VOTable Java Parser based on XML Schema (NVO)	Supports reading and writing.	メモリロード型。 JAVOTの後継版？
5	Starlink Tables Infrastructure Library (Starlink)	Supports reading and writing.	メモリロード型とシーケンシャル読み込み型。 多機能。

UCD

UCD unified content descriptors

- <http://www.ivoa.net/Documents/latest/UCD.html>
- 限られた単語セットでデータ意味を表す指標
- ひとつの Word あるいは複数の Words を“;” でつなげた語
- Word ひとつの Atom または複数の Atoms を“.” でつなげた語
- 基本単語は Word である。(Atom ではない)。
- UCD Word は UCD Boards によって維持・管理され、追加・削除は UCD Boards が決定する。
- UCD Word のリスト (合計 443 words)
<http://www.ivoa.net/Documents/latest/UCDlist.html>
 - P(54), S(104), Q(248), E(29), C(1), V(6)

Atoms

Word は12のカテゴリに分けられ、以下の12種のAtomで始まる

- Arith: 他のカテゴリのWord 間の演算によって得られる物理量。(Arith.diff, arith.ratio, arith.zp)。
- Em: 波長に関連した語。(em.IR.3-4um, em.IR.J, em.X-ray.hard, em.opt.R)
- Instr: 観測装置に関連した語 (instr.filter, instr.tel, instr.obsty)。
- Meta: 測定量ではなく、その他のカテゴリにも属さない語 (meta.id)。
- Obs: 観測に関連した語 (obs.airMass, obs.observer, abs.atmos.extinction)。
- Phot: 測光に関連した語 (phot.flux, phot.mag, phot.count)
- Phys: 原子・分子データや基本的な物理量(温度、質量、...)に関連した語 (phys.abund.Fe, phys.mass, phys.temperature)
- Pos: 天球上での位置に関連した語 (pos.eq.ra, pos.eq.dec, pos.galactic.lat, pos.galactic.lon)
- Spect: 分光に関連した語 (spect.index, spect.line.eqWidth)
- Src: 天体の属性に関する語 (src.class, src.density, src.redshift)
- Stat: 統計量に関する語 (stat.mean, stat.max)
- Time: 時間に関する語 (time.obs.start, time.obs.end, time.expo)

使い方

- 最初の方に記述された Word ほど、重要度が高いとみなす。
 - `<word1>;<word2>;<word3>;...`
- 最初の Word で一次のオーダーの表現
- 例:
 - Magnitude: `photo.mag`
 - A flag on this magnitude: `meta.code;phot.mag`
 - An error on this magnitude: `stat.error;phot.mag`
 - 装置の最大温度: `phys.temperature;instr;stat.max`
 - V-band の明るさの誤差: `stat.error;phot.mag;em.opt.V`

複数の Words が必要な例

- Phot.mag, phot.flux, phot.count は em で始まる Word が後に続く必要がある。
 - phot.mag;em.IR.J
- Phot.color は em で始まる Word が二つ後に続く必要がある。
 - Phot.color;em.opt.B;em.opt.V : B-V
- Meta.main, meta.modelled, stat.mean, stat.median, stat.min, stat.max は修飾語。値を表す word の後に書く。

UCD 選択ツール

- <http://cdsweb.u-strasbg.fr/UCD/>

VO標準規格の解説 2

データサービスの標準規格
SIAP/SSAP, SkyNode

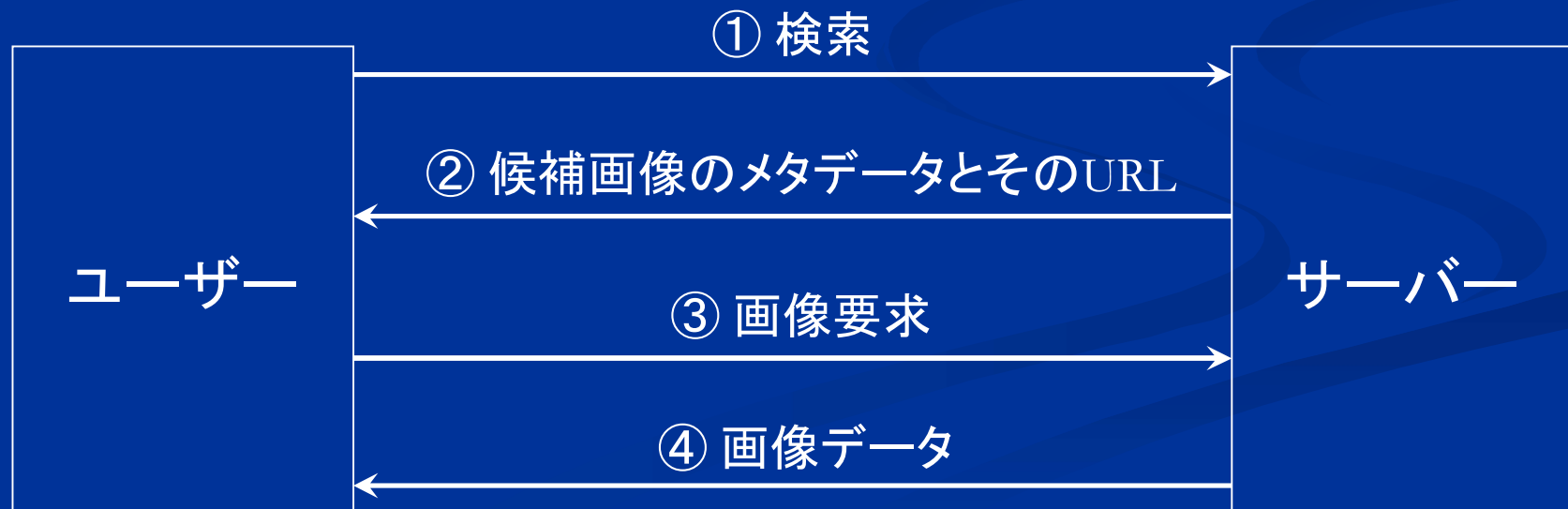
SIAP/SSAP/SLAP

- Simple Image Access Protocol
 - <http://www.ivoa.net/Documents/latest/SIA.html>
- Simple Spectrum Access Protocol
 - <http://www.aoc.nrao.edu/~dtody/ssa/ssa-v095.pdf>
- Simple Line Access Protocol
 - http://www.ivoa.net/internal/IVOA/SpectralLineListsDOcs/SLAP_0.5.pdf

SIAP

SIAP : Simple Image Access Protocol

- 画像データを取得するプロトコル
- Input: HTTP/GET name-value
- Output: VOTable (XML)
- 画像データ取得手順



SIAP V1.0 の仕様書

- <http://www.ivoa.net/Documents/WD/SIA/sia-20040524.html>
- 画像取得のアーキテクチャ定義
- 検索パラメータの標準化
- 検索結果として返す画像メタデータの標準化
- サービスメタデータの標準化
 - サポートするパラメータ名
 - 取得可能な画像メタデータ
 - サービスの種類、利用制限(検索数の上限)など

サービスの分類

■ サービスの分類

- 画像切り出しサービス (Image Cutout Service)
- モザイクサービス (Image Mosaicing Service)
- サーベイ観測データサービス (Astral Image Archive)
- ポインティング観測データサービス (Pointed Image Archive)

■ 画像データはキャリブレーション済みのもの

- Fits, gif, jpeg, ...

検索方法

■ HTTP GET

- Base URL : `http://<server-address>/<path> ?`
`[<extraGetArg>&[...]]`
- Constraints : `<name>=<value>` を `&` でつなぐ。
- Query = BaseURL + Constraints

検索パラメータ 1

■ 検索領域の指定

- **POS**: 中心座標を赤経・赤緯で指定。ICRS。POS=12.821,-33.4
- **SIZE**: 領域サイズ。
 - 値をひとつ与えた場合、赤経・赤緯で同じ離角サイズ、
 - 二つ与えた場合、最初の値は赤経方向の離角、2番目の値は赤緯方向。赤経方向の離角は $d(RA) \cdot \cos(DEC)$ であることに注意。」
 - SIZE=0 の場合、サーベイ・ポインティング観測の場合指定した座標を含む画像のリストを返す。カットアウト・モザイクサービスの場合はデフォルトサイズの画像を返す。
- **INTERSECT**: 指定した領域と返す画像範囲の関係
 - **COVERS**: 返す画像は指定した領域を完全に含む
 - **ENCLOSED**: 返す画像は指定した領域に完全に含まれる。
 - **CENTER**: 返す画像は指定した中心座標を含む
 - **OVERLAPS**: 返す画像は指定した領域と部分的に重なる。(デフォルト)
 - 切り出し・モザイクサービスの場合、返される画像のうち valid な値をもつ領域に適用される。

検索パラメータ 2

■ 画像生成パラメータ

- 切り出し・モザイクサービス用
- 取得する画像の解像度や向きの指定
- サポートしない場合は無視される、
- NAXIS, CFRAME, EQUINOX, CRPIX, CRVAL, CDELTA, ROTANG, PROJ

■ 画像フォーマット

- Image/fits, image/png, image/jpeg, text/html
- ALL: all formats supported by the service (デフォルト)
- GRAPHIC: jpeg, png, gif のいずれか。
 - GRAPHIC-ALL, GRAPHIC-jpeg,png,gif
- METADATA: メタデータを返す。

検索パラメータ 3

■ Verbosity

■ 画像メタデータの詳細度を指定

■ VERB=0 ~ 3

- 0: 最低限のメタデータのみ返す。
- 1: 返す画像をユニークに指定するのに十分なデータを返す。
- 2: サポートされているパラメータの値をすべて返す。
- 3: 最大限のデータを返す。

■ サービス独自のパラメータも OK

検索結果の仕様 1

- 検索結果は VOTable FORMAT で返される。
- VOTable は type 属性に“results” が設定されている RESOURCE 要素をひとつ含むこと。(必須)
- その RESOURCE 要素は結果を含むTABLE element ひとつ含むこと。(必須)
- その RESOURCE 要素は name 属性が“QUERY_STATUS”である INFO 要素を含むこと。検索が正常に行われた場合 その INFO 要素の value 属性には“OK”を設定すること。(推奨)
- テーブルの各行は各画像のメタデータ。

検索結果の仕様 2

- 必須のメタデータ。
 - タイトル、中心座標の赤経と赤緯、画像データの次数 (NAXES)、ピクセル数 (赤経・赤緯方向) (NAXIS, int の配列)、ピクセルサイズ (double の配列)、フォーマット、画像のURL
- 推奨されるメタデータ
 - 観測装置、観測の平均時刻 (MJD)、座標の基準フレーム (ICRS, FK5, FK4, ECL, GAL, SGAL)、投影方法 (TAN, ARC, SIN, ...)、基準ピクセル (CRPIX)、基準ピクセルの座標 (CRVAL)、WCS 回転行列 (CD)、バンドパスID、バンドパス単位、バンドパスの基準値、バンドパスの下限、上限、ピクセルフラグ、画像サイズ
- オプションなメタデータ
 - Equinox、画像データの有効期限、
- すべての行で同じ値をもつカラムは PARAMETER 要素であらわしてもよい。
- 検索エラー
 - INFO 要素の value 属性に "ERROR", "OVERFLOW" を設定

サービスメタデータ

- FORMAT=METADATA
 - サポートされているパラメータのリスト
 - PARAM 要素
 - 結果として返されるカラム
 - 検索結果と同じ内容を含む VOTable を返す。ただし、結果テーブルがはいってない。

サービスの登録

- Title, Publisher, Contact.Name, Contact.Email, ReferenceURL, Type.ImageService (Cutout, Mosaic, Atlas, Pointed), Description, Coverage.Spectral, Coverage.Temporal, Coverage.Spatial
- MaxQueryRegionSize, MaxImageExtent, MaxImageSize, MaxImageFileSize, MaxRecords, ImageFormats, ImageQueryBaseURL, ImageStagingBaseURL

SSAP

SSAP: Simple Spectrum Access Protocol

- 1D spectra, time series, SED
- 基本的は SIAP と同じ
- 範囲を表す方法: “/” を使用
 - $5E-7/8E-7$, $/5$, $5/$
- データ種別: Atlas、Pointed、Modeled、Observed、Composite、Simulated
- サービス種別: static, cutout, resampling

SSA Parameter

- **必須:** POS, SIZE, BAND, TIME, FORMAT
- **FORMAT:** ALL, COMPLIANT, EXTERNAL, GRAPHIC, votable, fits, xml
- **推獎:** APERTURE, SPECRES, SPATRES, TIMERES, SNR, REDSHIFT, TARGETNAME, TARGETCLASS, PUBID, CREATORID, TOP, INDEX, COMPRESS, SINZE, RUNID

SSA output

- **必須:** Dataset.Type, Dataset.SSA.NSamples, Coverage.Location.Spatial, Coverage.Location.Time, Coverage.Location.Spectral, Coverage.Bounds.Time, Coverage.Bounds.Spectral, AccessReference, Access.Format,
- **推奨:** Query.Score, Dataset.DataModel, Dataset.Title, Dataset.SSA.Aperture, Dataset.SSA.CreationType, Curation.Collection, Curation.CreatorID, Coverage.Location.Spectral.BandID, Coverage.Bounds.Flux, Accuracy.Spatial.Calibrated, Accuracy.Spatial.Resolution, Accuracy.Time.Calibrated, Accuracy.Time.Resolution, Accuracy.Spectral.Calibrated, Accuracy.Spectral.Resolution, Accuracy.Flux.Calibrated, Frame.Time.SIDim, Frame.Spectral.SIDim, Frame.Flux.SIDim, Frame.Flux.UCD, Access.Encoding, Access.Size
- **オプション:** Query.LName, Query.LNameKey, Dataset.SSA.TimeAxis, Dataset.SSA.SpectralAxis, Dataset.SSA.FluxAxis, Dataset.Derivation, Target.Name, Target.Class, Target.SpectralClass, Derived.Redshift, Derived.VarAmpl, Derived.SNR, Curation.Creator, Curation.PublisherID, Curation.Data, Curation.Version, Coverage.Bounds.Spatial, Coverage.Fill.Spatial, Coverage.Fill.Time, Coverage.Fill.Spectral, Accuracy.Spatial.SysErr, Accuracy.Spatial.SgtatErr, Accuracy.Time.SysErr, Accuracy.Time.StatErr, Accuracy.Spectral.SysErr, Accuracy.Spectral.StatErr, Accuracy.Flux.SysErr, Accuracy.Flux.StatErr, Frame.Sky.Type, Frame.Sky.Equinox, Frame.Time.System, Instrument.Name, Instrument.Exposure, Instrument

SSA data format

- Application/x-votable+xml
- Application/fits
- Application/xml
- Image/jpeg
- Text/plain

SkyNode

SkyNode によるデータ公開

- データはテーブル単位で公開する。
- どういうデータを公開しているのかという情報はテーブルとカラムのメタデータから得られる。
- テーブル・カラムのメタデータ公開法は2通りある。
 - Web サービスのオペレーション tables, columns でそれぞれのメタデータを返す。
 - tables, columns という名前のテーブルを用意し、そこにメタデータを登録しておく。

SkyNode の仕様 1

- 検索言語 (ADQL)

<http://www.ivoa.net/Documents/latest/ADQL.html>

- インターフェイス (SkyNode)

<http://www.ivoa.net/Documents/latest/SNI.html>

ADQL の仕様

- ADQL は2種類のフォーマットがある
 - ADQL-s (SQL), ADQL-x (XML)
 - <http://www.ivoa.net/xml/ADQL/ADQL-v1.0.xsd>
- 構文は標準 SQL にほぼ準拠
 - 制限:
 - テーブルアリアス名必須、...
 - 拡張:
 - TOP, INTO, REGION(), XMATCH(), ...

サポートすべき SQL の構文

- ADQL には Core 仕様と Extension 仕様がある。
- Core 仕様はすべての SkyNode がサポートしなくてはならない。
- SELECT 文のごく一部をサポートすればよい。
 - セレクトできるのはカラムまたは `count(*)` のみでよい。
 - 一度に検索できるテーブルはひとつだけでよい。
 - WHERE 節
 - 比較文 ($A=B$, $A>B\dots$), LIKE, BETWEEN, IN, AND, OR
 - Region('Circle FK5 35.0 10.0 1.0')
- SQL の XML 表現 (ADQL-x)
 - サービス側で SQL パーサを持たなくてよい。

SkyNode の仕様 (SkyNode)

- Web サービスで実装する。
- BASIC & FULL
- BASIC
 - `performQuery(ADQL[, format]), Tables(), Columns(), Column(table), Formats(), Functions()`
- FULL
 - `queryCost(ADQL), xmatch(ADQL, votable), executePlan(plan), footprint(region)`

サービスの公開

- Publishing Registry にメタデータを登録するだけ。
- Searchable Registry により
- Publishing Registry は現在約6個
- Searchable Registry は4個？
- 独自につくってもよい。
- Publishing Registry 登録は Registry of Registry に登録することになる？（現在は Searchable Registry の管理者にメールでしらせている）

SkyNode サービスを立ち上げてみよう

- SkyNode をつくるためのソフトが公開されている。
 - <http://jvo.nao.ac.jp/download/skynode-toolkit/>
 - <http://us-vo.org/summer-school/2005/proceedings/index.html>
- 必要なソフト (JVO Skynode の場合) のインストール
 - Java, tomcat, DBMS(PostgreSQL or MySQL), SkyNode toolkit
- サービスの立ち上げ
 - Instruction.txt
- サービスの登録 (試験用)
 - <http://jvo.nao.ac.jp/p.registry.dev/>
- JVO ポータルから検索してみる。