VOTable (1.1)

François Ochsenbein
CDS, Strasbourg

http://www.ivoa.net/Documents/latest/VOT.html

C'est quoi, VOTable ?

- VOTable est un format XML pour *échanger* des données tabulaires, dans le cadre de l'Observatoire Virtuel
  - doit permettre l'échange de tables volumineuses (plusieurs millions de lignes)
- VOTable doit rester compatible avec les tables FITS *(accès au contenu des tables FITS sans réécriture de données)*

C'est quoi, une table ?

- Structure de données universelle (bases relationnelles), bien adaptée aux catalogues et gros ensembles de données, qui comprend 2 parties:
  - une description de la *structure* de la table (vue en tant que classe d'objet, colonnes = attributs). On parle aussi de schéma de la table, ou bien des métadonnées
  - une collection (importante) de ces objets (tuples) qui partagent cette même structure.

Une structure qui semble bien pauvre...

... comparativement aux structures des systèmes objets d'aujourd'hui ... et pourtant!

- c'est très compact (par rapport aux objets)
- ça possède des propriétés algébriques exploitées dans les bases de données relationnelles (algèbre relationnelle, SQL)
- nombreux traitements génériques

⇒ça peut être rendu très performant

Qu'apporte XML ?

- XML permet simplement de mélanger du texte avec des informations cachées dans des "balises" dûment définies par un schema (anciennement dans un DTD). Rien de génial!
  - le texte = les données
  - le contenu des balises = les métadonnées, qui suivent une structure strictement *hiérarchique*.

Un exemple XML

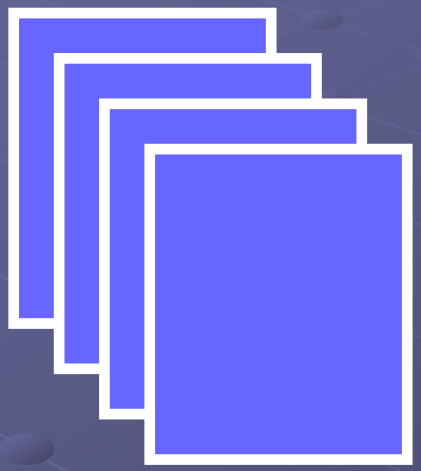
```
<BOOK ISBN="9782212090819" LANG="fr" SUBJECT="applications">
    <!-- Ceci est un commentaire -->
    <AUTHOR>
        <FIRSTNAME>Jean-Christophe</FIRSTNAME>
        <LASTNAME>Bernadac</LASTNAME>
    </AUTHOR>
    <AUTHOR>
        <FIRSTNAME>François</FIRSTNAME>
        <LASTNAME>Knab</LASTNAME>
    </AUTHOR>
    <TITLE>Construire une application XML</TITLE>
    <PUBLISHER>
        <NAME>Eyrolles</NAME>
        <PLACE>Paris</PLACE>
    </PUBLISHER>
    <DATEPUB>2002</DATEPUB>
</BOOK>
```

Quel est l'intérêt du XML?

- L'Interoperabilité nécessite l'utilisation de *structures de données* standardisées et une *description standardisée du contenu* (metadata)
  - Problème: FITS est bien une structure de données standardisée, mais les métadonnées sont hétérogènes
- XML built-ins:
  - tous les logiciels de vérification, validation, editions, ...
  - transformations et visualisation des documents XML (XSLT)
  - possibilités de requêtes (XPath, XQuery), XML-DB...

Structure d'une VOTable

RESOURCE
=
{TABLEs}



TABLE

METADATA
=
DESCRIPTION
+INFOs
+PARAMs
+FIELDs

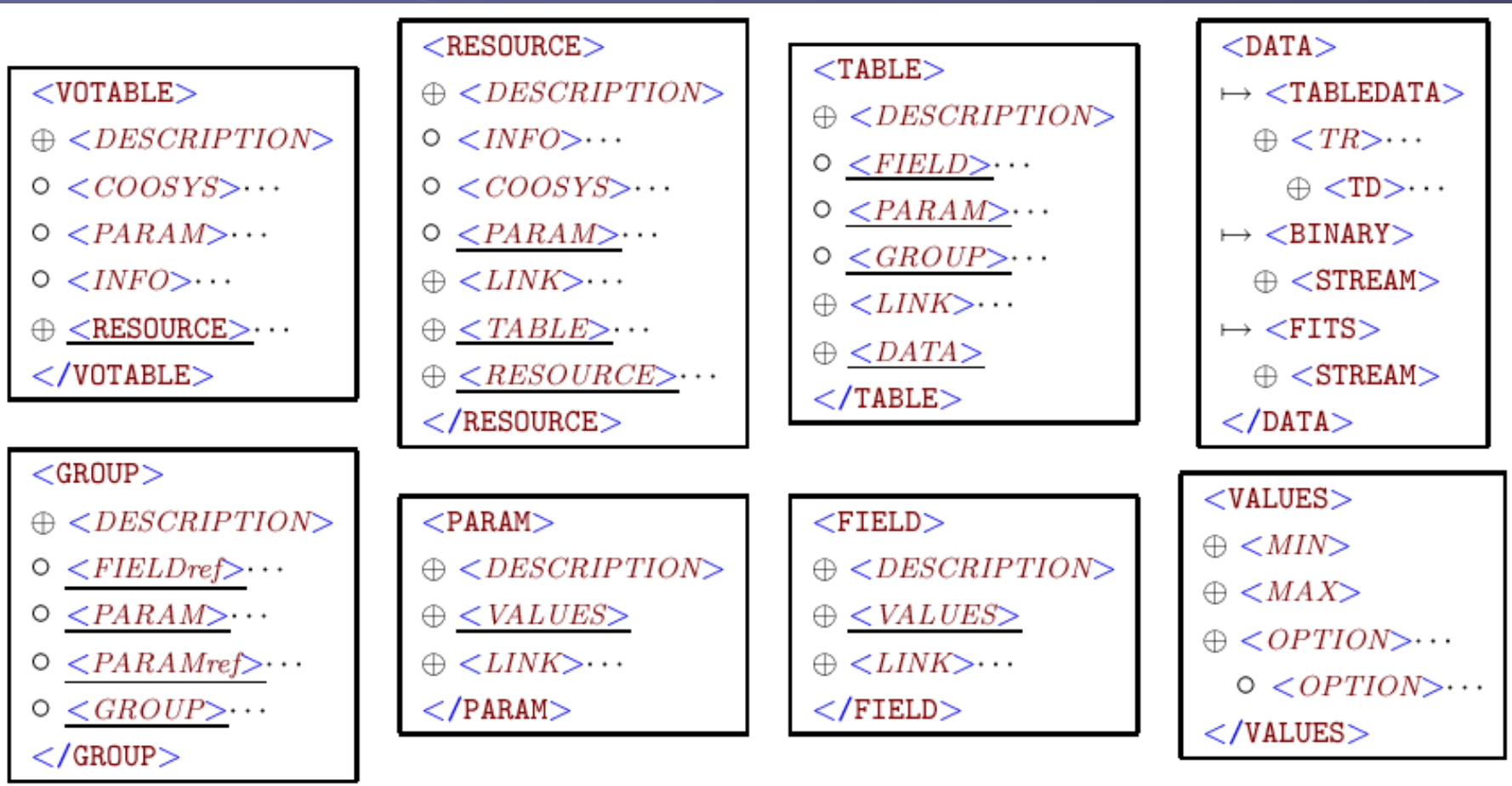
DATA

1	16.676	0.176	...
2	15.715	0.077	...
3	16.445	0.143	...
4	16.703	0.138	...
5	16.475	0.179	...
6	16.366	0.141	...
7	15.360	0.068	...
8	13.776	0.027	...

VOTable Data Model

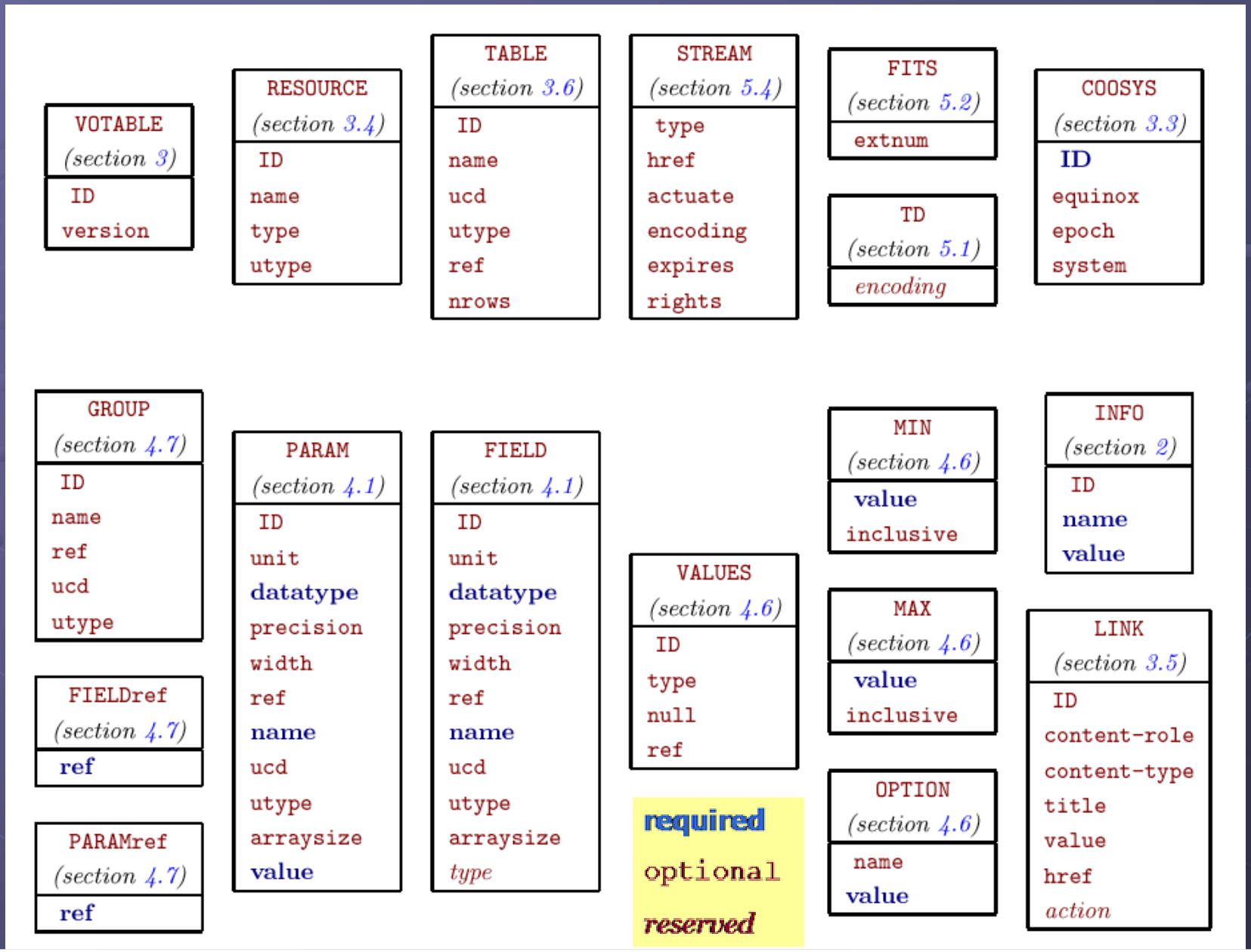
- **VOTable** = hiérarchie de **Resource**
- **Resource** = ensemble de **Tables**
- **Table** = **Metadata** + **Data**
  - **Metadata** = **Infos** + **Descriptions** + **Parameters** + **Links** + **Fields**
- **Data** = suite de **Rows** ou **STREAM** (remote)
- **Row** = liste de **Cells**
- **Cell** = une **Primitive**
  - ou liste (longueur fixe ou variable) de **Primitives**
  - ou ensemble multidimensionnel de **Primitives**
- **Primitive** = integer, character, float, floatComplex ,...

Eléments principaux de VOTable



o unordered ⊕ ordered □□□□□□choice □

Détails des Attributs



Détail d'une VOTable

contient les éléments suivants:

- COOSYS (système de coordonnées, sera remplacé par une référence (utype) au modèle STC=Space-Time Coordinates)
- RESOURCE contient une DESCRIPTION et une liste de tables, éventuellement d'autres RESOURCEs
- TABLE contient:
  - une DESCRIPTION de la table
  - une liste de FIELD (colonnes) qui décrivent le schéma de la table + des PARAMETER pour précision éventuelle de *constantes*
  - éventuellement des GROUP faisant référence a d'autres colonnes
  - des liens LINK pour récupérer des détails, des données annexes...
  - la partie DATA qui peut être incluse dans le document, ou renvoyer à des fichiers FITS, du binaire, une URL...
- Exemple tiré de VizieR

<TABLE>

- Contient DESCRIPTION + une collection d'éléments FIELD + PARAMETER éventuels + LINK + GROUP = *métadonnées;* suivi par le bloc DATA
  - FIELD décrit une colonne de la table
  - PARAMETER peut être vu comme une *colonne constante*
  - GROUP permet de définir des *associations de colonnes*
- DATA démarre la partie données, qui contient les valeurs de chaque colonne dans le même ordre que leur déclaration, groupées en lignes (tuples)

<FIELD> en détail

- Possède les sous-éléments DESCRIPTION, LINK, et éventuellement VALUES pour préciser le domaine
- Rôle = définir aussi précisément que possible le contenu de la colonne qui se trouve dans la partie DATA
- L'identification d'un FIELD se fait par `name`, et par `ID`
  - `name` est destiné par ex. à l'affichage
  - `ID` est un identifiant XML (formé d'un jeu de caractères restreint, doit être unique dans tout le document XML)
- FIELD doit préciser son `datatype`
- Chaque colonne peut contenir un vecteur multidimensionnel du type précisé par `arraysize`, qui peut avoir la dernière dimension indéfinie (`64x64x*`)

Datatypes

datatype	Meaning	FITS	Bytes
"boolean"	Logical	"L"	1
"bit"	Bit	"X"	*
"unsignedByte"	Byte (0 to 255)	"B"	1
"short"	Short Integer	"I"	2
"int"	Integer	"J"	4
"long"	Long integer	"K"	8
"char"	ASCII Character	"A"	1
"unicodeChar"	Unicode Character		2
"float"	Floating point	"E"	4
"double"	Double	"D"	8
"floatComplex"	Float Complex	"C"	8
"doubleComplex"	Double Complex	"M"	16

Field cont..

- `unit` précise l'unité (vocabulaire contrôlé)
- `ucd` (Unified Content Descriptors)
  - donne un "type sémantique" de la donnée contenue dans la colonne
  - Pour l'instant VizieR donne par défaut des UCD1, évolution imminente...
- `utype` (v1.1) permet de référencer un modèle externe
- `precision, width` nécessaires l'édition d'1 valeur
- VALUES
  - précise les informations relatives au *domaine :* min, max, null, liste des valeurs permises.
- LINK
  - sert de pointeur vers d'autres documents ou serveurs par le biais d'UR[IL]
  - peut être typé ("doc" ...)

<PARAM>

- PARAM est très semblable à FIELD : peut contenir une DESCRIPTION, possède des attributs `unit`, `ucd`, `utype`, `name`,`ID`,... plus l'attribut obligatoire `value`
- peut être interprété comme une colonne *constante*
- est utile également dans un GROUP
- Utilisation typique:
  - frequence ou λ utilisé
  - erreur statistique moyenne (par ex. précision astrométrique ou photométrique)

<DATA>

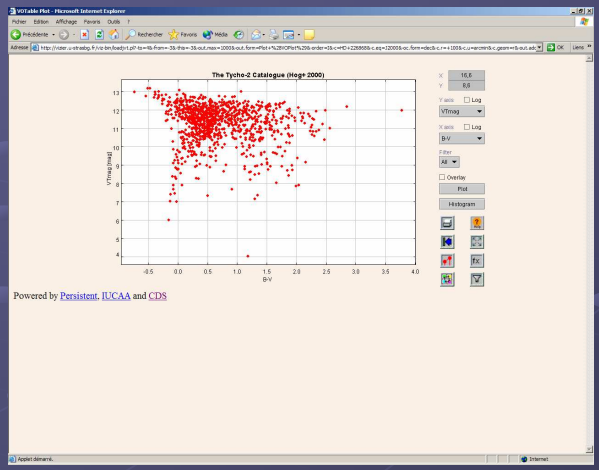
- Un seul bloc DATA par table.
- Plusieurs façons d'accéder aux données, dans un des 3 formats:
  - TABLEDATA lorsque les données sont en XML
  - FITS permet d'accéder à un fichier FITS externe
  - BINARY pour accéder à des données binaires (pour plus d'efficacité)
- STREAM pour accéder à des données non XML
  - Les données peuvent être dans le flux XML (éventuellement encodées), ou stockées séparément
  - Exemple

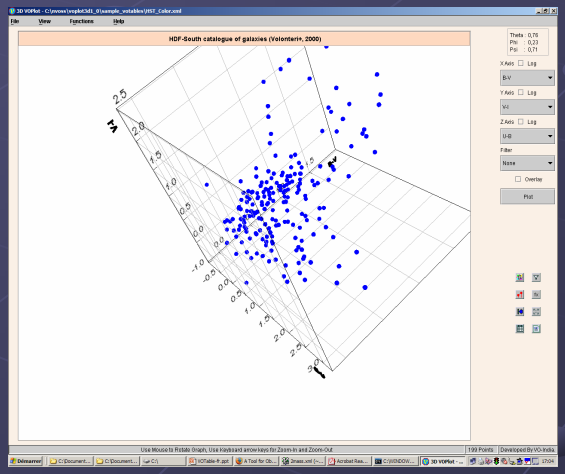
Quelques utilisations de VOTable

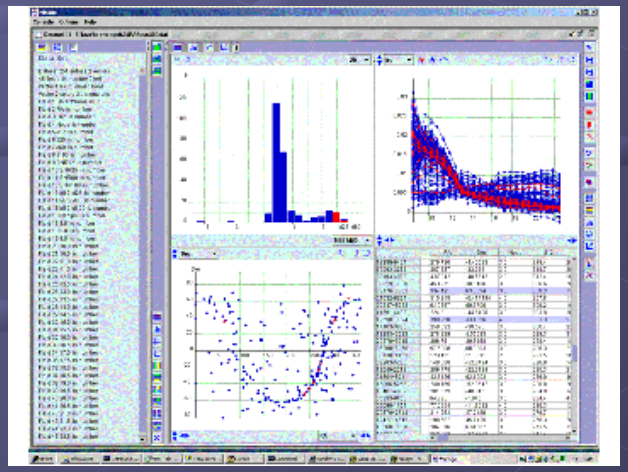
- SkyNode (plusieurs niveaux)
  - Cone Search (simple HTTP-get avec les arguments RA= DEC= SR=)
  - Retournent du VOTable
- SIA (interrogation serveur d'image) ou SSA (serveur de spectres) retournent du VOTable

Quelques outils

- En java, ingestion de VOTable (savot, javot, ...)
- Conversion FITS VOTable (toujours possible), topcat
- VOPlot (à partir de VizieR) + VOPlot3D (standalone)







- Mirage http://www.bell-labs.com/project/mirage/

VOTable vs VOList

Les "puristes" reprochent à VOTable que son schéma ne s'applique strictement qu'aux métadonnées, la partie "data", même sous forme de TABLEDATA, n'étant pas strictement "validable" et "interrogeable" par les outils XML standard (XPath, XQuery)

- l'utilisation de FITS ou BINARY devient impossible
- chaque table devrait avoir sa propre définition
- l'efficacité en pâtirait pour les très longues listes...

Exemple de pur XML: Sesame