



*International  
Virtual  
Observatory  
Alliance*

# Data Model serialisation in VOTable Version 1.00

**IVOA Note 2004-04-08**

**This version:**

<http://www.ivoa.net/Documents>

**Latest version:**

**Previous versions:**

**Editor(s):**

F.Bonnarel - [bonnarel@astro.u-strasbg.fr](mailto:bonnarel@astro.u-strasbg.fr)

**Authors:**

M.Louys - [louys@astro.u-strasbg.fr](mailto:louys@astro.u-strasbg.fr)

F.Bonnarel - [bonnarel@astro.u-strasbg.fr](mailto:bonnarel@astro.u-strasbg.fr)

## Contents

1	Starting point, UML description	2
2	VOTable solution	4
2.1	Why that? . . . . .	4
2.2	Classes, attributes and links . . . . .	5

## Introduction

We assume we have an UML description of a given data model, as a class diagram. A class expresses not only a structured data type but also a concept

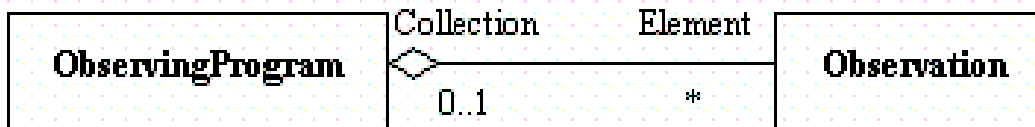


Figure 1: An aggregation link between a set, a collection and its elements.

either in astronomy or in astronomical data acquisition and processing. A class diagram shows the relationships between classes and may be rather complex. In this note for the sake of efficiency, serialising is meant to focus on some aspects of the model, showing the content of some classes and their interactions. This kind of serialisation offers only a view of the whole data model.

## 1 Starting point, UML description

We start with a description of classes, by their definition, interfaces and their properties. We express the logical relationship between classes using an UML diagram and some documentation text. In other words we have classes containing attributes (and methods). Each of these has a name, a data type and a text description. We have various relations between these classes:

- Aggregation Ex: an ObservingProgram object gathers a collection of Observation objects.
- Composition The Coverage class encompasses 3 aspects coded as Location, Support, Sensitivity Objects . It works like a union of classes.
- Association of any kind: This is the case when two classes are logically linked : For example, an Observation has a Provenance  
Association supports plenty of other cardinalities: 1 to n links, n to m links, etc
- Inheritance  
Ex: ObsData is a kind of Quantity

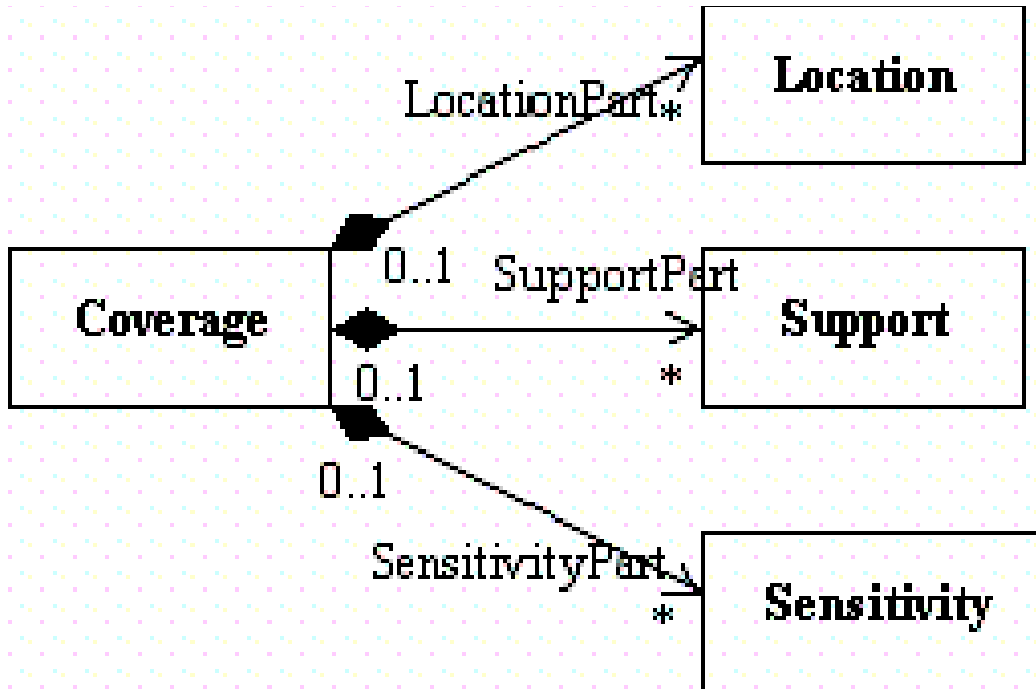


Figure 2: Composition example



Figure 3: An association link with cardinalities of 1 to 1, which means one instance of class Observation is connected to one instance of the Provenance class.

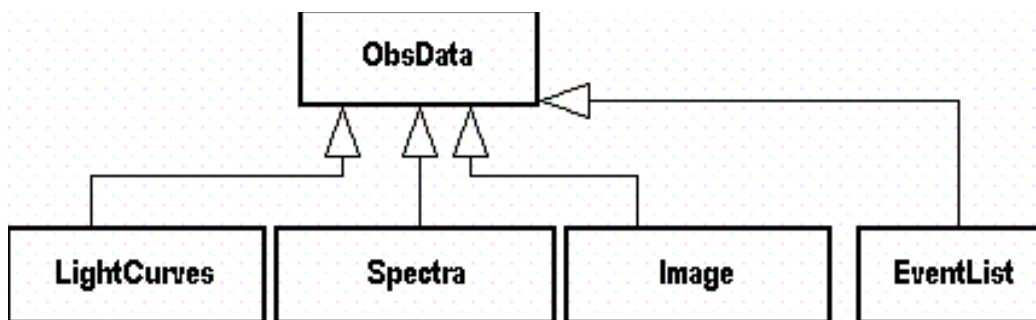


Figure 4: Inheritance : A Spectra is "a kind of" , a subclass of ObsData

## 2 VOTable solution

### 2.1 Why that?

What is our problem: creating a file where data objects belonging to this scheme can be instantiated, and the various logical links between classes expressed. XML is a widely admitted standard for description and exchange of data. But definition of XML documents for a specific purpose, here describing entities and metadata according to a data model will require the definition of a DTD or an XML schema. Such DTDs or Schema are available for data model themselves, but how do we serialize metadata for sets of real objects, like collections of images, spectra, ?

From the elements of discussion initiated by Ed Shaya concerning serialisation of pure XML description of data models, (cf. DM list), and the Quantity data model serialisation proposals by Brian Thomas we concluded that agreement about a pure XML serialisation protocol is probably far to be reached. We inferred that there is some room left for a more simple "quick and dirty" proposal.

We consider that the IVOA has defined an XML DTD (and schema) for description of tables of data such as astronomical catalogues: VOTable. If we take into account that metadata description and structuration are part of the goals for a datamodel, we must consider the fact that such kind of description is some special case of a structured catalogue, where the columns are metadata, the lines are the instances of objects.. The VOTable DTD is generally considered as rather poor by XML purists, and it is true that the semantic of the elements and their attributes is rather simple. **However here, we explicitly choose to accept that DTD provided by the VOTable group and to serialize our datamodel using it.** Actually the semantic and richness of content of an astronomical catalogue field definition is present in the actual values taken by the attributes of the FIELD

or PARAM elements. We will use this feature extensively to transfer the semantic of our datamodel in the XML document. **Classes and attributes names are rendered as VALUES of attributes not as attribute or elements NAMES.** Behind VOTable is lying the so called Relational model. So our problem is simply to map our object data model into a relational one.

## 2.2 Classes, attributes and links

The basic way to do it is to consider a class as a table and a class attribute as a table field. An instance of a class is then associated to a tuple of the corresponding table. Associations between classes are rendered by at least two ways:

- The use of a reference field in the VOTable description of the two classes . We can generally use a Name or Identifier field as the common field. Note that for most classes instances can be uniquely identified by a Name or Identifier attribute if not both. In practice, the link is defined by a field in one table and uses a reference to it in the second one. This simply implements a pointer from one table to the other.
- Due to the recursive nature of VOTable a RESOURCE can be defined by a composition of other RESOURCE or TABLE elements . Therefore here one can define one classe as a top level RESOURCE of the document and the second one as a nested RESOURCE. For example the relationship: Observation has a Coverage, can be represented by a Coverage RESOURCE nested under the Observation RESOURCE. The first feature is adapted to any kind of links but specifically well to 1 to 1 associations. The second one is more adapted to 1 to n associations and to aggregations and it materializes tree structures in the the document structure. It is mainly a way to render a point of vue on the data model issue. Several trees involving the same sequence of classes can be coded using several times the same sequence of RESOURCES.

It is also possible to refer to the parent table from the children one by using a reference field to the parent identifier or name field N to n (or n to 1) links can be seen as a set of 1 to n links and can be described in a similar way.

For a composition, classes "belonging" to a class can be either referenced in the container class by using a specific field with a VOTable reference to the ID of the Table associated to the included class, or as an inclusion of this included class attributes as fields in the container class/table. The first feature is well adapted for content rich classes (ie where the contained class has a lot of attributes). The second one is better for poorer classes. In the

second case we have to show that the included fields belongs to a special class. We can achieve that by using a utype description for each of these field referring to the actual data model name for this field. We could also use the grouping feature of VOTable to take this Into account . We can even use both.

For inheritance we can define generic tables somewhere, create a group (or groups) on its fields and use these groups as references in the inherited classes.

The IDHA format (<http://www.euro-vo.org/twiki/bin/view/Avo/MetadataTree/>) and the proposal for a SIAP evolution (<http://alinda.u-strasbg.fr/SiapEvolution.pdf>) are two examples of where these simple principles have been used.