Requirements for Theory in IVOA
Version 0.1

IVOA Theory IG Internal Draft 2006 May 01

Working Group:
http://www.ivoa.net/twiki/bin/view/IVOA/IvoaTheory
Authors (alphabetically):
Laurie D. Shaw
Nicholas Walton
Miguel Cerviño
theory interest group

Abstract

In this working draft we propose some requirements for including theoretical data to managed by the VO. It included not only the design of a Data Model but also the requirements needed to allow that third parties fell useful and easy to publish their theoretical results on a VOTable format and get involved in the VO community.

The general ideas can be found in *Theory in the VO* by G. Lemson and J. Colberg. The present working draft is based on the document by L.D. Shaw, N.A. Walton & J. Ostriker: *Towards the Incorporation of Cosmological Simulation Data into the Virtual Observatory* (that is almost completely included with some modifications in Section 2) and incorporates some extensions referred to the synthesis models (including their inputs and by-products). The present document is only an **incomplete** working draft and its only aim is to put on html/Twiki format the current documents for on-line discussion

Most of the document is focused in the inclusion of theoretical models in a VOTable format and their implications for other Working Groups in IVOA, in particular, the UCD, VOTable, Data Model and Resouce Metadata working Groups.

Status of this Document

This is an IVOA Working Draft for review by IVOA members and other interested parties.
It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/Documents/.

Acknowledgments

This document is based on the W3C documentation standards, but has been adapted for the IVOA.

Contents

1. Introduction

One of the most fundamental differences between theoretical work and observational data in astronomy is that observational data can be almost unambiguously defined by the coordinates of the object and related quantities like the type of instrument used etc... On the other hand, a theoretical result is defined, at least, by a list of parameters used by a given model, the code (including its version) that produce the result and the used inputs.

A second difference is the intrinsic pipe-line characteristic of theoretical models. One example can be taken from the theoretical modeling of stellar populations: a code needs as inputs the results of another theoretical code that produces isochrones, which are the output of another code that uses evolutionary tracks and so on; additionally, the stellar population code would also need a library of stellar atmospheres, and the resulting output can be extremely varied, ranging from from a synthetic color-magnitude diagram of a cluster to the spectro-chemical evolution of galaxy. A second example is Nbody dark matter simulations on a cosmological scale: where users must to be able to extract data from simulation archives, run their own analysis tools, compare simulated data directly to observed using tools that the VO offers and eventually even perform simulated observations of simulated data.

> In both cases, the intrinsic pipe-line characteristic of the process is relevant. This is in some way similar to a reduction of astronomical data from raw to "usable" quantities, but in the theoretical case, each of the by-products by itself would be a product that can be stored and used by the observational community.

A third difference is the amount of data that theoretical model can generate and how this data are used. To illustrate this point, if an evolutionary synthesis code makes use of theoretical libraries with a large resolution power, is output size requires some Gigabytes of disk space. Observational data are usually stored in FITS and compressed files; theoretical data are usually stored as simple ASCII tables ready for posterior use in other applications by direct reading of the complete data or streaming of the it.

The objective of this document is to discuss in the IVOA Theory community the best conditions so that the results of different theoretical models can be input into other utilities (either another theoretical model or a theory/observational interface).

Note: In 2004-01-20 there was a discussion in the "Interop IVOA" mail-list about how to establish a data workflow and if it is necessary or not to define a standard for workflow. Some of the ideas quoted there are relevant for this draft.

2. Simulation Data Model

A fundamental requirement for the implementation of theory in the VO would be to take advantage of the studies performed by other groups in the IVOA community: In the general case, any model should be compared with real observations, so it is quite useful that any data structure proposed in the theoretical domain agree with a similar structure proposed in the observational domain. As example that will be further developed, a model which final result is an spectrum, should mach the structured used for spectral tools of observational data (like the ones developed by the VOspec proposal). This general rule does not apply for the <COOSYS> keyword, and some additional keywords that describe the model must be incorporated.

In this situation, it is necessary to know the current data model that is performed by other groups in the IVOA comunity.

2.1 Review of the Observation Data Model

A comprehensive data model named *Observation* for observational data is currently being defined. This model attempts to identify the different aspects that fully describe either a single observation of the sky, or a dataset derived from a number of observations. It therefore represents a description of all the metadata that may be required by both data discovery and retrieval services and data analysis applications. An example of the typical categories that make up a complete description of an observation is dis- played in Figure 1 (taken from the current IVOA Data Modelling 'observations' draft Observation Data model).

Figure 1: The general model for Observation. See text for description

Figure 1 demonstrates that an observation can essentially be broken down into three main categories - Observation Data, Characterisation and Provenance:

- *Observation Data* describes the units and dimension of the data. It inherits from the Quantity data model (currently in development) which assigns the units and metadata to either single or arrays of values.
- *Characterisation* describes how the data can be used. It can be broken down into Coverage (within what limits the data is valid) and Reso- lution and Precision (different aspects of how accurately we are able to measure any single value).
- Provenance describes how the data was generated. This includes the telescope/instrument configurations, calibrations, the data reduction pipelines and the target itself.

2.2 Simulation Data Model

A first attempt to define a data model for simulation data (named *Simulation*) within the framework outlined by the Observation model is shown in Figure 2). The three main sub-categories - Simulation Data, Characterisation and Provenance are still applicable. However, *for simulation data it is the Provenance object, rather than the Characterisation, that contains the real descriptive content of the model*. We now describe below each of the three main parts of the *Simulation* model, noting the similarities or differences to their equivalents in *Observation*.

Figure 2: The general model for Simulation (described below)

2.2.1 Simulation Data

This object remains essentially the same as in the *Observation* model - a subclass of the Quantity object, used to contain the main data output of the simulation (see Quantity Data Model). However, for simulated data there is potentially a much wider range of quantities to be stored. In *Observation* at least one quantity in the data must be an observable; this is not the case in *Simulation*. The metadata structure - the set of Universal Content Descriptors (UCD's) - used to describe each quantity must be enlarged to incorporate data clearly labelled as being 'theoretically derived'. It must be flexible enough to be able to describe the many different quantities that can be measured from a simulation, yet accurate enough to allow their identification by a general query service. We are currently working on how this can be done.

2.2.2 Characterisation

Although simulation data is fundamentally different to observed data (we can know everything about a simulated object), the Characterisation outlined in *Observation* is in many ways still applicable to the equivalent in *Simulation*. Even though simulated data is normally not subject to enforced data gaps or exposure times, concepts such as Coverage, Resolution and even Sensitivity are still relevant. Rather than image resolution, Resolution in the *Simulation* context would refers, in the case of Cosmological Simulations, to mass resolution (the mass of the particles), temporal resolution (time step) spatial resolution (grid size, or the particle-particle interaction length... or to mass resolution in the case of isochrones, spectral resolution in the case of atmosphere models or synthesis models etc...

Others quantities would play a role depending on the model quantity in the *Simulation* object. As example, Coverage would play a minor role for Cosmological Models, but it play the a similar role than in the *Observation* object for synthesis models (e.g. the spectral coverage of the model will define over which data the model can be compared). And a similar situation happens for other quantities, like Sensitivity or Precision, that would show the theoretical dispersion of the results, etc...

Summing up, Characterisation in a *Simulation* object, would depend on the kind of theoretical object that are defined by this class. It poses a first conclusion:

> *Characterization in the **Simulation** object would include not only the same subclasses that the **Observation** object, but also a specification of which kind of simulation contains actually the simulation data*

2.2.3 Provenance

As stated above, the Provenance object contains most of the information describing the simulation. This is because, unlike during an observation, most of the effort in acquiring the data is not through measurement but through the execution of numerical routines, thus creating the data set. The Provenance object is defined as **the description of how the dataset was created** which for a simulation we are able to describe entirely.

So, for the *Simulation* class, Provenance turns to be the most important object. From this object it should able to reproduced the Data Model by third parties (i.e. similar to repeat the observation in the observational word).

Provenance can be broken down into the Theory, Computation and Parameters. Theory describes the underlying fundamental physics upon which the simulation is based. For example, in a dark matter n-body simulation it would include what gravity processes have been accounted for and which have been ignored, in the case of evolutionary tracks or isochrones it would contain details about overshooting or rotation, in the case of synthesis models details about the combination of atmosphere models libraries with isochrones, etc...

Computation describes the technique used to evaluate the physics described in The- ory through the execution of numeric routines. The main components of Computation are the organised sequence of algorithms that compute the various stages of the sim- ulation. The algorithms are often chosen to provide a balance between the time taken to complete the simulation, the numerical accuracy and resolution, the complexity and requirements of the physics and so on, based on the hardware and software resources available. Often the sequence of algorithms will involve the 'main' simulation fol- lowed by a number of analysis routines. For example,Figure 3 demonstrates the main steps towards the creation of a 'mock universe' - a catalogue of dark matter halos halos (identified in large scale dark matter nbody simulations) populated with galaxies taken from observational surveys. Stage 1 represents the bulk of the simulation, the nbody Tree Particle Mesh code (TPM, see Bode, P., & Ostriker, J.P. 2003, ApJS, 145, 1) that evolves the dark matter particles from their distribution in the early universe to the present day. The stages that immediately follow are all 'reduction' algorithms that extract the useful information from the raw output of Stage 1. For example, the purpose of Stages 3 & 4 are to identify structures (halos) in the particle data and then to fit a density profile to each of them. This is roughly anal- ogous to the data reduction performed on the raw data from observations. The details of the physics that goes on at each stage are unimportant here; the figure is just a good example of the sequence of algorithms that may be involved in such a simulation.

Figure 3: Flowchart outlining the main steps towards the creation of a mock universe, from the raw TPM simulation (see text) to assigning gas and galaxies to the halos and sub-halos.

The third component of the Provenance is Parameters. The input parameters not only define the physical context of the simulation, but also the resolution and detail. If the algorithms are analogous to a mathematical function, the parameters are the values of the input variables. They are identified here as being seperate to the Theory and Computation as they are normally the only elements that change between different runs of the simulation.

Parameters can be broken into Physical and Technical sub-classes. In any cos- mological simulation there are at least five physical parameters that must be defined (baryonic and total matter densities, intial density perturbations, etc). The technical parameters such as 'box size' (representing the volume of space being modelled) and the total number of particles will define the accuracy and resolution of the simulation and in some ways, the amount of processing power required. In the case of synthesis models the Physical parameters would refer to the set of evolutionary tracks/isochrones used, the set of stellar atmospheres used and the Initial Mass function and Star Formation History assumed. The technical parameters would refer to the age resolution of the output and the normalization scale (mass of the modelled cluster), as example.

In an alternative view, the physical parameters could be seen as an input to Theory and the Technical paramerters as an input to

Algorithm. However, it seems sensible to seperate the variable and static elements of a simulation. Parameters will be contained by the Quantity object. Although UCDs will probably already exist for the physcial parameters, a new cate- gory will need to be created for the technical parameters. Work is currently in progress attempting to define the requirments of this new catagory.

It is worth to note that no credits should be lost. Contrary to observational data, managed by institutions, most theoretical models are managed and developed by small groups, and sometimes by individual (examples would Kurucz atmosphere models or the photoionization code Cloudy, by G. Ferland, which are be the extensive used by the community). If the data model is not able to recover the corresponding credits, some groups could decide not join the theoretical VO iniciative, since their work would receive no credit.

From a metadata perspective it is anticipated that the Theory, Computation and Parameters objects should include references and/or links to relevant papers and (in the case of Com- putation) a reference to the code itself (this could be secondary function of the web services that provide the astronomy community access to the simulation tools). But additional text fieds (may be in a "latex-ready" format) that explain what has been used by the references would be required: In several cases a single reference is not enough to actually describe univocally the input data, , like papers with several sets of isochrones with different metallicities. In the same way, the reference field would be best if described by a latex-bibitem environment with labels similar (or equal) to ADS/CDS format.

3. Additional requirements for a theoretical model: Interoperativility

From the previous section it is clear that:

- *Observation* and *Simulation* objects would share most of their properties, with some important differences.
- The <COOSYS> keyword in the *Observation* object must be changed into something like a <SIMULATION_TYPE> keyword that describes what *kind* of data is included in the simulation data, or a specific keyword (with a suitable UCD) must be included in the Characterisation class.
- The most important class in the *Simulation* object should be the Provenance, that should be considered like a "log" with all the relevant information for the Simulation Data to be recovered.
- In particular, Provenance should include the references/links to the relevant sites (suggested: in latex-ready format) plus a description of what information of the references must be taken into account. Ideally, with the Provenance object alone it would be possible to recover the rest of the classes (Simulation Data and Characterisation)

In this context, it should be clear that for any pipe-line process performed with VO utilities, the Provenance object should store all the information. i.e. it should be an object where additional fields can be added but not deleted out. Note that this is different from the Characterisation class, where fields can change. As an example, a VO utility that obtains colors from simulated spectra would add some parameters (e.g. source of the filter system) to the Provenance class, and would change the Characterisation in the resolution field.

At this moment it is worth to ask if it would be interesting to store Simulation Data, or would be better to provide a service that can reproduce them directly only with the Provenance class. However, such simulation data would be the result of several hours (or even days) of CPU computing, so, even in a perfect case of reproductivility, to provide the result is useful. Additionally, although in a future if would be possible to work only with the Provenance class for some cases, the corresponding VO tools would be needed to perform this task. In any case, the present considerations suggest that future VO tools and applications should be designed to work with the Provenance class.

Takening into account the previous requirements, we describe now the implications that they have for a data model and its output in VOTable format:

3.1 Unique definition of the model

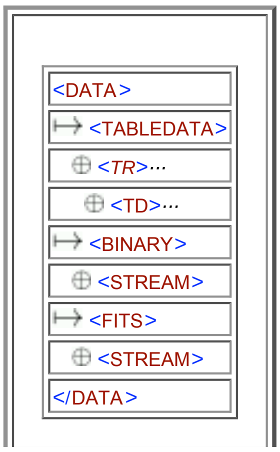
Would be needed a "particular" data model for each kind of theoretical results?

3.2 Resource and bibliographic management

....

4. Data Storage and streaming

Taken into account the pipe-line character of theoretical models in their input data, it would be desirable the use of catalogues that can be used to access the data. In the current VOTable version format definition the <DATA> are defined in the following way:



Where ↦ means a choice between alternatives, the dots ... indicate that the element can be repeated and ⊕ indicate that the order of the elements is mandatory.

This schema of the <DATA> element does not allow for including <STREAM> elements (as used in the SIAP protocol). This situation would be overcome in the next VOTable extension when it would be possible to use a <FIELD> as a data pointer:

Questions:

- **VOTables with pointers to others VO tables? How distinguish between *final data* (e.g. SED) and *intermediate data* (e.g. atmosphere library)**
- **Tehoretical VOTable format: plain ascii (i.e. XML, easier for streaming) as a primary option**
- **FITS? Are there theoretical models that produce fits directly?**
- **As the format implications for posterior developments? (e.g. GRID)**

5. Implications for UCD VOTables and Resource Metadata

5.1 Implications for UCD

Appendix A: ...

If you have an appendix, put content here

References

[1] Author(s), *Title*
http://