International

Virtual

Observatory

Alliance

Theory Semantics

Version 0.1

IVOA Technical Note 9/9/2006

This version:
0.1

Latest version:
0.1

Previous version(s):

Author(s):
Laurie Shaw
Herve Wozniak
Franck le Petit
Fabrice Roy
Miguel Cerviño

Supprimé : n

Abstract

Structured outline of UCD and Standard Vocabulary requirements for simulated data.

Status of This Document

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

Contents

1 Introduction

The purpose of this note is to outline the UCDs and various candidates for the standard vocabulary that will be required in order for a data or software provider to register a simulation data-set or snip of simulation code in the VO. Based on initial work on a Simulation data model, we have defined a set of categories, each of which represent an aspect of either a piece of simulation code or dataset. Some categories require just a single numerical or Boolean character, a string or URL specific to the context, to represent. Others might need more generic description taken from the IVOA standard vocabulary. For these we list the concepts, but do not attempt to define UCD type suggestions as the exact construction of Standard Vocabulary words is still to be determined. However, UCDs need to exist for the categories themselves – for these, we suggest appropriate existing UCDs or propose new ones.

2 List of Categories

Following the Theory Interest Group meetings at the interop in Victoria, a list of categories was constructed to describe the different components for simulation code/dataset metadata. Some of the categories below are relevant only for simulation data or codes. Indeed, if the simulation code used to generate a published dataset is itself published, then part of the metadata for the dataset may just point to that for the code. Categories marked with (c) are code specific, those with an (d), data specific. Those marked with both d & c are either relevant for both, or are required for datasets when the simulation code itself is not publicly available.

(d) Name of Dataset
(d,c) Name of the developer/team/contact
(d,c) Name of Code
(c) Version of Code
(d,c) Description of the code (text)
(d,c) Physical Objective
(d,c) Physical Process
(d,c) Subject(s)
(d,c) Algorithm
(d,c) Time evolution
(c) Protocol
(c,d) Result format
(d) Results Parameters

Below is a description of each of the above. Categories have been grouped according the aspect of the simulation data or code they describe. For each category a UCD label is suggested, and also sample entries – or concepts that must be covered – in the standard vocabulary are listed.

3 Basic Attributes

3.1 Name of Dataset, developer/team/contact

These are as for observational data.

3.2 Name of Code

Description: Name of code used to generate a particular dataset
UCD : **meta.id;comp.code.main**
List : no

Commentaire [MC1] : It allows to describe the results of codes that make y use of the results of other codes. Is is the case of evolutionary synthesis.

Secondary codes can be described at the same level by and UCD meta.id;comp.code.secondary

Examples : Gadget, Cloudy, Zeus

The content of this category can consist of either the name attributed to the code, especially (in the case of data) if the code is published in the VO. For unpublished codes it could consist of a reference to a paper or webpage where the code is described or available for download. In that case, the corresponding UDC words should be added (meta.ref.url, meta.ref.uri or meta.bib.bibcode) No additions to the standard vocabulary are required here.

For the case of code that make use of results of other codes (e.g. synthesis codes, or pipeline reduction of data) it would be included an additional attribute:

Description: Name of code or dataset used as input of the main code
UCD : comp.code.secondary
List : Yes

Formally, only the reference of these secondary codes are necessary since it is hopped that the secondary code will be registered some way in the VO.
[mcs: I am not sure if meta.id would be necessary in this case]

[DeYoung: Not all codes have names. e.g., MHD-TVD codes used here are not yet "named" because the authors think it is frivolous. How should these be included?]
[mcs: I have added some text in next item, it would solve this case?]

3.3 Version of Code

Description : Version number of the code.
UCD : **meta.version**

This is only required when the same piece of code is updated in a minor way. Hence, if the operation or purpose of the code changes significantly, i.e. Gadget to Gadget 2, then it should be registered as a new code separately.

However, the previous paragraph must be taken as a recommendation. It should be responsibility of the developers of the code to keep a correct versioning of it. It is not a task of the publishers of the code itself (or the code results) in the VO.

Regarding the values of the name of the code and its version it is suggested to use the most common values used in the literature.

3.4 Description of Code

Description : ASCII text describing code / simulation results

List : no
UCD : **meta.note**

(sort plain text description of code)

Commentaire [MC2] : Added "sort", otherwise VO applications will not manage it

Additionally, it should be possible to add url references for more detailed description of the code. For this last case, the corresponding UDCs has been yet included: meta.ref.url including the content.role="doc" in the LINK element (see appendix A in VOTable-20040811 recommendation).

4 Physics of Code

This section refers to the physical theory contained within the code.

[Franck & Fabrice:We think that in the three cases: Physical Objective, Physical Processes and Subject, le list of possible values have to be perfectly defined or a search in the registries will be unable to succeed because of synonyms. MNRAS keywords and some words used in VO-events should cover most of the possibilities.
If you agree we should explicitely add this in the text as it is done for Physical Objective.]

4.1 Physical Objective

Description: general keywords to give a short description of the phenomenon being simulated
UCD : **phys.process;comp.sim;meta.main** [might need a better suggestion]

[mcs:Maybe Physical objective sould include also the subject?: i.e. add the UDC **object?** In this case there is no collision with primary Physical Processes, althougth it sounds to be redundant… see note bellow…]

List : Yes
Examples : Galaxy/Planet/Galaxy Formation, Stellar evolution

Purpose is to give a general indication of overall phenomenon that is being simulated, or the physical objective. What are we trying to simulate? Aim is to provide an umbrella term for the collection of physical *processes* (see below) that are modeled or simulated.

Standard Vocab concepts here

- formation of astrophysical objects (star/planet/galaxy/universe/…)
- evolution of astrophysical objects (stellar/galactic/…)

Supprimé : evolution

- stellar population synthesis
- large scale structure
- accretion/mergers (binary stars, galaxies, discs, etc)
- etc (many more. See http://www.blackwellpublishing.com/pdf/mnraskey.pdf)

Supprimé : galaxy formation/

Supprimé : /evolution

4.2 Primary Physical Processes

Description: list of the main physical processes that are accounted for in the simulation
UCD : **phys.process;comp.sim** [laurie: not sure about this either]

[mcs: I think that it is a good UDC for this category]
List : Yes
Examples : GR, gravitational dynamics, radiative transfer

This category is used to describe the level of detail in which the phenomenon in the Physical Context category is investigated. Allows a user to determine the level of physical detail the simulation achieves. Does it account for this/that/etc? Consists of a list of words chosen from the standard vocabulary.

Supprimé : phenemonen

Standard Vocab words required:

- radiative transfer
- gravitational dynamics (collisional/collisionless [HW: I don't understand this refinement. 'Gravitational collisional dynamics' is hydro- or fluid- dynamics in presence of a gravitational fluid. so it can be labeled 'gravitational dynamics; hydrodynamics; etc.'. don't it ?)
- GR
- hydrodynamics
- fluid dynamics
- plasma physics
- magneto-hydrodynamics
- photoionization
- photodissociation

[DeYoung:The vocabulary here is very incomplete, as was noted by Laurie]
[mcs: There is any reference for a more complete standard vocab.??, I think that it would be better a link to this type of document instead to wirte here a complete list]

4.3 Subject

Description: General keywords describing the main type of object being simulated
UCD : **object;meta.main**
List : Yes
Examples : star, neutron star, dark matter halo

Words from the standard vocabulary to describe all the (astrophysical) objects in the simulation (or these could also be UCDs in a new ‘object’ branch):

- `object.`
- `object.star`
- `object.galaxy`
- `object.disk`
- `object.planet`
- `object.halo`
- `object.quasar`
- `object.molecularcloud`

Standard Vocab words required

- planet
- star
- stars
- stellar clusters
- galaxy
- halo
- volume of space
- accretion disk
- jet
- interstellar medium
- molecular cloud
- atmosphere (of planet, exo-planet, star...)

Mise en forme : Puces et numéros

[DeYoung: There seems to be a lot of overlap or redundancy here with section 4.1, which is supposed to give a description of the phenomenon being simulated] [mcs: I agree with Dave. However I am not completely sure if there will be rendundancy always… but I can not iamgine the case]

Mis en forme : Surlignage

5 Code Operation

5.1 Algorithm

Description: this is purely to describe the numerical procedure being used to evaluate the physical processes being simulated.

UCD : **comp.alg**
List : yes
Examples : Nbody, sph, amr, tree

This needs to be kept extremely general, otherwise the number of possibilities increases uncontrollably. There are many many minor variations of the same algorithm or approach and it will be almost impossible to maintain a complete list in a standard vocabulary. Can not be described to an acceptable degree using a few keywords – need reference to a paper, etc. Include at most the main variations of mesh (adaptive mesh, etc) and Nbody (tree, particle-mesh). Note, tree-particle-mesh could be described by listing both the "tree" and "particle-mesh" SV words.

Standard Vocabulary Entries:

- nbody
- mesh
- collisionless
- eulerian grid

[HW: what difference with mesh ? maybe do you mean Eulerian solver, e.g. Godunov, Rieman, piecewise parabolic solvers etc for Eulerian hydrodynamical codes ? Need to classify codes like RAMSES which is not a SPH code…]

- lagrangian grid

[HW: what is a lagrangian grid. In Lagrangian formalism, we follow the flow, so at my knowledge, there is no mesh/grid; the difference between Eulerian and Lagrangian formalisms is very important for hydro codes]

- tree
- adaptive refinement mesh
- adaptive refinement tree
- sph
- particle-mesh
- particle-particle
- fokker-planck codes
- vlasov solvers
- orbit solver
- Friends-of-Friends
- Denmax

[HW: real specific algorithm ? I don't know…]

5.2 Time Evolution

Description: flag indicating whether the code is time dependant (i.e. evolves a system with time), or is stationary
UCD : **comp.sim.timeEvolution**
List : yes – two possiblities (yes/no)

5.3 Protocol

Description: if a code is parallel, gives the protocol used

[HW: at my knowledge codes can be serial, vectorized or parallel; for parallel code we can further refined with the model (simd, mimd, etc…), the way memory is used (shared, distributed) but is the library used (openmp, mpi, pvm, globus, etc.) very useful ?]

[Franck & Fabrice: This section is indeed usefull in the case of the search for a service to be used on grids. In this case we should provide not only informations about parallelisation but all the technical requirements to run the code. For exemple compilers, libraries (Blas, Lapack, NAG, ...).
This makes me think to another problem. In the case of a code launched on a grid via a VO-service such as Astrogrid, do we plan at any level to describe the requirements in hardware: memory, processor... I think that may be useful at a point but maybe it is not required at this level of our work.]

[DeYoung: I agree with others that the specification of the library is useful]

UCD : **comp.protocol**
List : yes
Examples : OpenMP, MPI,

Decribes type of parallelisation – useful for grid and webservices group?

6 Results Metadata

6.1 Type of Result

Description: give the type of result produced by simulation
UCD : **meta.id**
List : yes
Examples : spectra, object catalogue, statistics, fit parameters, raw particle/grid data

Gives an indication of the type of output produced by a simulation – is it raw unprocessed particle data, or mock images/spectra or a halo catalogue from FoF etc.

SV entries:

- snapshot
- animation
- table
- FITS
- catalogue of objects
- statistics of objects

6.2 Results Formats

Description: output format of the results type
UCD : **meta.format**?
List : yes
Examples : ASCII, binary, hdf5, fits, jpeg, etc

So the user knows what tools to use to obtain/analyse the results

6.3 Algorithm Parameters

[Frank & Fabrice: I think we should not focus too much on this part yet since it is linked to the datamodel to describe results. This part is fine as it is presently.]
[mcs: I agree, it is fine for me]
[DeYoung: Here there is also a lot of overlap with section 5.1, Algortithm. Perhaps these two should be merged or at least placed in the same general section]

Description: simulation parameters used to generate results
UCD : **comp.sim.params;comp.alg**
List : yes
Examples : number of particles, box size

Difficult to know how to deal with this information, as it contains key information that determines how relevant/accurate data is. Will come from a standard vocabulary, however it might be nice to have UCDs for the more general ones:

- comp.sim.boxsize (Simulation box)
- comp.sim.gravsoft (gravitational softening)
- comp.sim.particles (simulation particles)

- comp.sim.snapshot (output of a simulation box)
- comp.sim.mesh (simulation mesh (for hydro simulations))

.

So that we could have

- `meta.num;comp.sim.particles`
- `meta.num;comp.sim.grid`
- `phys.mass;comp.sim.particles`
- `phys.size;comp.sim.gravsoft`
- `phys.size;comp.sim.boxside`

Alternate SV words required:

- number of particles (of different types, where relevant)
- particle mass
- mesh size
- gravitational softening
- box size
- timestep
- distance to the observer
- smoothing length

6.4 Physical Parameters

Description: physical parameters used to generate results
UCD : **comp.sim.params;phys**
List : yes
Examples : number of particles, box size

Same as above, but for input physical parameters to simulation. Should really be UCDs for all this, e.g.

```
phys.cosmology         (cosmology)
phys.cosmology.omega   (matter/energy density of universe)
phys.cosmology.hubble  (hubble constant)
phys.cosmology.sigma8  (Normalisation of matter power-spectrum)
```

and also (maybe from the standard vocab):

```
phys.matter.dark       (dark matter tag)
phys.matter.baryon     (baryonic matter tag)
phys.DarkEnergy        (dark energy tag)
```

So, Omega_Lambda, Omega_DM, Omega_baryon would be

```
phys.cosmology.omega;phys.DarkEnery,
```

phys.cosmology.omega;phys.matter.dark
phys.comsology.omega;phys.matter.baryonic

<u>List of Suggested UCD Candidates for Controlled Vocabulary</u>

‘comp’ branch

```
Q comp.                        [computational astrophysics branch]
P comp.protocol                [parallelization/grid protocols]
P comp.alg                     [algorithm]
Q comp.sim                     [simulation (to distuguish between
                               simulations and post-processing of obs]
P comp.sim.params              [simulation parameters, technical or
                               physical]
P comp.sim.timeEvolution       [time evolution tag for sims]
```

and possibly also

P comp.sim.boxsize	[Simulation box side length]
P comp.sim.gravsoft	[gravitational softening)
S comp.sim.particles	[simulation particles, for mass, number of, etc]
S comp.sim.mesh	[simulation grid (for mesh size, number of mesh points]

‘object’ branch – though more likely this is just in the standard vocab

```
Q object.                      [astrophysical object]
Q object.star
Q object.galaxy
Q object.disk
Q object.planet
Q object.halo
Q object.quasar
Q object.molecularcloud
Q etc
```

others

```
Q phys.process                 [a physical process UCD for simulations,
                               which allows a description of what is
                               being simulated]

P phys.cosmology                [cosmology]
P phys.cosmology.omega          [matter/energy density of universe]
P phys.cosmology.hubble         [hubble constant]
P phys.cosmology.sigma8         [Normalisation of matter power-spectrum]
```

and also (from the standard vocab):

```
S phys.matter.dark              [dark matter tag]
```

```
S phys.matter.baryon           [baryonic matter tag]
S phys.DarkEnergy              [dark energy tag[
```

So, Omega_Lambda, Omega_DM, Omega_baryon would be

```
phys.cosmology.omega;phys.DarkEnery,
phys.cosmology.omega;phys.matter.dark
phys.comsology.omega;phys.matter.baryonic
```

```
meta.format                   [enables output format of a simulation to
                              be labeled]
```