

U C D : Status and perspectives

S. Derriere - derriere@astro.u-strasbg.fr

See <http://vizier.u-strasbg.fr/UCD/> for an introduction to UCDs and tools which use them.

This document summarizes recent discussions or suggestions concerning UCDs, their use, and possible evolution. I have quoted some relevant parts of discussions from various persons. A short list of people involved in those discussions is given below.

People who have worked on UCDs (not complete list):

- Thomas Boch (boch@astro.u-strasbg.fr)
- Sebastien Derriere (derriere@astro.u-strasbg.fr)
- Martin Hill (mchill@dial.pipex.com)
- Tony Linde (tol@star.le.ac.uk)
- Mireille Louys (louys@astro.u-strasbg.fr)
- Jonathan McDowell (jcm@head-cfa.harvard.edu)
- Ashish Mahabal (aam@astro.caltech.edu)
- Francois Ochsenbein (francois@astro.u-strasbg.fr)
- Patricio Ortiz (portiz@)
- Ray Plante (rplante@poplar.ncsa.uiuc.edu)
- Anita Richards (amsr@jb.man.ac.uk)
- Guy Rixon (gtr@ast.cam.ac.uk)
- Arnold Rots (arots@head-cfa.harvard.edu)
- Alex Szalay (szalay@jhu.edu)
- Doug Tody (tody@noao.edu)

1 UCD status - 02/2003

1.1 Where UCDs are currently used

- in VizieR catalogues
- in VOTable documents (ucd attribute in FIELD elements)
- in ConeSearch (POS_EQ_RA_MAIN, POS_EQ_DEC_MAIN, ...)
- in Aladin¹ (from v1.4, in filters, via a scripting language allowing operations on columns as identified by their UCD (e.g. display for each source a circle whose radius is proportional to PHOT_JHN_B)
- for Simple Image Access Protocol
- for IDHA Aladin server output (VOTable)

1.2 Where extensions to UCDs (in terms of new UCDs) have been suggested

- in Simple Image Access Protocol (VOX: namespace)
- for describing data models elements.

1.3 What has been suggested for more functionalities

- adding new UCDs / suppressing unuseful UCDs
- refactoring UCDs with atoms
- define standard prefixes/suffixes
- parametrize UCDs i.e. PHOT_FLUX(5cm) or PHOT_MAG(JHN,B)

1.4 Open discussions

There have been many discussions concerning:

1. What UCDs are?
2. What do we want to do with UCDs?

1.4.1 What are (currently) UCDs?

UCDs were designed for describing the contents of columns in tables, in order to ease the interoperability of astronomical catalogue tables. They are NOT

¹<http://aladin.u-strasbg.fr/>

”column names”: it’s possible to find the same UCD in a table several times.

UCDs are not:

- accurate definitions
- expressing physical dimensionality
- tied to units

UCDs and UNITS are different parameters... but are more or less coupled

- standard way of spelling units → see FITS WCS Paper II²
- values can be converted from one unit to another (java code available)

The hierarchical tree is not fundamental – it’s merely a way of classification (taxonomy).

1.4.2 What do we want to do with UCDs?

UCDs are already used for:

- describing contents of catalogues columns (especially VizieR)
- searches in catalogues or databases
- filtering (in Aladin, e.g. limit sample to galaxies brighter than 16 in blue)
- visualization (in Aladin, e.g. represent proper motion with an arrow, size proportional to value)

See <http://vizier.u-strasbg.fr/UCD/> for example tools.

Extending UCDs toward an ontology, or to describe datamodels, or as general standardized metadata is addressed below.

1.4.3 Use of UCDs for catalogues/databases outside VizieR

SDSS (A. Szalay)

- Trial to assign UCDs to the SDSS database structure (≈ 1000 columns)
- Creation of new UCDs (only a few ones needed !!!):

²http://fits.gsfc.nasa.gov/fits_wcs.html

STAT_STDEV
STAT_VARIANCE
STAT_COVARIANCE
FIT_PARAM_COVARIANCE
ID_VERSION
INST_SKY_SIGMA
PHOT_TRANSF_PARAM
POS_EQ_X
POS_EQ_Y
POS_EQ_Z

ESO: SAF/HST Databases (F. Pierfederici, M. Dolensky, B. Pirenne) — note that this is for image description! — suggestion of new UCDs to describe special columns in those databases mainly in the field of instrument/detector properties (conditions of observation).

2 UCDs and ontologies

Ontology is sort of a fuzzy abstraction for most astronomers. I won't get into the details of how information is stored in an ontology, how it is parsed or used by different kind of softwares: this is just a matter of implementation.

What is an ontology, and do we need one ?

Looking at tutorials concerning ontologies ³ it appears that:

An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of the basic concepts in the domain, and relations among them.

Motivations to develop an ontology include :

- share common understanding of the structure of information (among people and software agents);
- make domain assumptions explicit.

³e.g. http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

In an ontology, there are:

- classes (=concepts)
- slots (=roles, =properties)
- facets (=role restrictions)

Classes describe concepts of the domain. There can be sub-classes (they are also classes, sharing same properties as the parent class). Slots describe features and attributes of classes. Facets are constraints on slots.

This gives the formal description of a domain, the ontology. Concrete realisations are instances of classes, and together with the ontology they constitute a knowledge base.

If we consider the current UCDs, we might find similarities with an ontology. The hierarchical structure indicates some class/subclass division. The length of UCDs often makes some implicit information explicit (like declination DEC, attached to POS_EQ to explicitly show that this is a quantity related to positions in equatorial coordinates, which is obvious for an astronomer but not necessarily for a software agent). But there is no clear distinction between classes, properties and instances, each of them can be described by a UCD, so the full set of UCDs is more a knowledge base than a ready-to-use ontology. We are also missing cross-links between different parts of the hierarchical UCD structure: this kind of missing link could be introduced by facets in an ontology.

Building an ontology in a domain is an iterative process that requires a lot of interaction to end up with a commonly agreed structure. If we want to use UCDs as standard elements for an Ontology of Astronomy, then we need to find a common agreement on its contents. This requires interaction, and *iterative building* of the UCDs.

Note that we probably don't need to conform strictly to existing ontology-related tools (DAML/Oil). Maybe one could build one real schema for an ontology, but most applications should be able to run simple implementations of the standard vocabulary. I really think that it is important, whatever metadata structure is finally adopted, that the concepts (or properties) can still be human-readable (e.g. in the form of formatted strings like current UCDs). This would allow to handle them simply in VOTable documents, as an attribute.

A problem with ontologies is that they are very well suited for formal explicit relations between concepts, but I haven't found yet examples of a formal description for *mathematical relations* between elements. e.g. how

do we describe that [photon-wavelength] is equal to [speed-of-light]/[photon-frequency] in the ontology?

3 UCDs and data models

Data models:

- Space-Time (A. Rots)
- Image Data Model (IDHA – M. Louys; CfA, CXC DM initiative – J. McDowell)

UCDs were designed for describing the contents of columns in tables, in order to ease the interoperability of astronomical catalogue tables. They have been deliberately oriented towards description of quantities that result from a physical measure in an experiment (e.g. magnitudes, fluxes, positions). UCDs were not built to cover every concept existing in astronomy: in particular, concepts which are "global metadata", that is to say concepts which are common to a whole dataset, are probably not covered by UCDs. For example, the name of an observatory is a concept which is global to all observations made from that observatory. But, if it is never explicitly used in at least one column of a table of data obtained from that observatory, no UCD will be created for this concept.

Because of the wide diversity of information in VizieR (3000 tables, 10^5 columns), it happens that some columns contain a quantity which will be global meta-information for other tables (it is the case for the observatory name: OBSTY_ID exists because some tables list data from various observatories and there is a column containing the name of the observatory).

But in general, UCDs do not cover all metadata concepts (they were not meant to do so).

The idea of attaching UCDs to elements of a data model is to have an homogeneous description of the concepts with a standard vocabulary. Because some of the concepts can already be described by some existing UCDs, they have been considered as a possible solution to make descriptions converge towards a standard vocabulary.

But as we have shown, some concepts can not be described with the current UCDs. In the case of the Image data models, this is often related to instrument configuration or reduction methods. This kind of information (size of the mirrors, version of pipeline software used for data reduction, ...)

are often only cited somewhere in the original papers describing the datasets, and are not part of the dataset (table). That's why some elements of the data models need some extra UCDs to be created, and in some cases, for part of the structure to be re-organized.

Suggested actions:

- define new concepts (i.e. not already described by UCDs) used in the data models
- create corresponding UCDs and place them in the metadata structure

Links:

Space-time metadata:

<http://bill.cacr.caltech.edu/cfdocs/usvo-pubs/files/SpaceTime.htm>

IDHA data model:

<http://alinda.u-strasbg.fr/IDHA/lastmodel/>

NVO data model:

<http://hea-www.harvard.edu/~jcm/vo/vo3.ps>

4 Suggestions for refactoring UCDs

There have been various suggestions on how UCDs could be improved, in order to be more coherent or be used in wider applications:

<http://wiki.astrogrid.org/bin/view/Astrogrid/RefactoringUCDs>

<http://wiki.astrogrid.org/bin/view/Astrogrid/UCDAtomsPreliminaryList>

<http://wiki.astrogrid.org/bin/view/Astrogrid/WhatAreUCDsFor>

<http://wiki.astrogrid.org/bin/view/Astrogrid/AreUCDsMetadata>

Main criticisms to the current structure are:

- structure is not flexible enough (missing UCDs)
- browsing the UCD tree to find what you need is not easy

Main suggestions are:

- rearrange the roots of the tree (first level nodes)
- define some common suffixes (or prefixes)
- break the hierarchical structure, and use atoms. These atoms serve as elements in a multi-dimensional parameter space.
- make UCDs parameterizable

Concerning reorganisation of the roots, e.g. J. McDowell first suggested⁴

⁴<http://hea-www.harvard.edu/~jcm/misc/ucd.ps>

splitting the problem domain into:

- observations
- sources
- physics

and then into 3 categories of fields:

- about the universe (PHYS, POP, SAMPLE, OBJ, MED, LOS)
- about observations of the universe (SRC, BKG, OBS, PHOT, SPEC, POS, TIME)
- about analysis of observations (DATA, REFER)

This root structure is still not complete (image parameters, data reduction are not included), and might be confusing (distinction between population and sample, or object and source is very subtle). It is clear that the definition of the *right* roots does not exist, and is influenced by the user's own needs. The current structure of UCDs can only be reorganised if there is a consensus on the new form to be adopted.

But it is also possible to imagine having different structures corresponding to different (not necessarily complete) views of a complex model (ontology or other).

The most requested change is to use ERROR as a suffix that can be added to any measure (instead of the single ERROR used in current UCD). This is somewhat similar to the convention used in VizieR ReadMe files, where errors column names are prefixed with "e_". There are a number of similar conventions (for notes, differences, etc...). This is also in the same spirit as the various dimensions suggested by G. Rixon, for atomic UCDs. The syntax of the main parameter from additional suffixes/dimensions (separation by / \ | ; or whatever symbol) is of second importance. First a set of mutually exclusive attributes must be defined (e.g. _ERROR, _NOTE, _CODE, _FLAG, _QUAL, _MAX, _MIN, _SCALE, _DATATYPE), in order to place them on orthogonal axes. A UCD could be seen as a point in the multi-dimensional space defined by those axes. This will provide a high degree of flexibility in the definition of new UCDs.

Splitting UCDs in atomic components that could allow building new UCDs, and parametrization of some UCDs gives full flexibility. This has pro and cons. For non-existing UCDs, it is a useful way to create new ones, from standard atoms or standard structures (PHOT_MAG[PHOT_SYSTEM,

PHOT_SYSTEM_BAND]). But if this creation of new combinations is completely free and unsupervised, it could lead to different dialects (2 different combinations of atoms for the same parameter in 2 different datacenters). This is what we do not want, because we would lose all the benefits of a common description for interoperability. This is why I find it important that we have in the VO a set of tools to query and modify a common repository of the existing UCDs. In practice, you don't create new UCDs just for fun. You use UCDs to describe fields in a dataset that will be shared with others. The questions are:

- how do I find the proper UCDs to describe my dataset ?
- what do I do if there is no UCD ?

I would suggest, for common definitions of UCDs in the IVOA framework:

- sharing a common dictionary (not necessarily centralized, can be duplicated in several places with daemons keeping different copies up-to-date)
- versioning - each time modifications are done, keep track of the changes/additions/suppressions, so that applications using UCDs can say: (e.g.) compatible with version 1.2 of UCDs
- a robust validation mechanism, to avoid creating synonyms of already existing UCDs

An assignment form (or Web Service) would take in input a basic description of the dataset to be described by UCDs, and suggest some UCDs (possibly together with a list of atoms or combinations of atoms, and also an indication of where this UCD is already used). If no UCD is relevant, the person must be able to send a suggestion of a new UCD in the common dictionary (with a short definition of what it means). This proposition can be studied by a group of experts in charge of maintaining the dictionary, and included in the dictionary if needed. This might look tedious, but it is needed if we want a *common* description of contents. At least, this is needed for datasets that are to be widely accessible, and conform to some standard description.

In other disciplines, there are some collaborative work to develop ontologies. This is the case in the Gene Ontology consortium⁵. In this project, they

⁵<http://www.geneontology.org/>

have opened a SourceForge.net account where it is possible to submit suggestions for new terms: <http://sourceforge.net/projects/geneontology>

This might be a way of updating a common dictionary for our purpose?

There has also been some work done by the Consultative Committee for Space Data Systems (CCSDS)⁶, on Data Entity Dictionaries. They provide on their website documents describing the abstract syntax and XML/DTD syntax for Data Entity Dictionary Specification Language (DEDSL).

Extension of the current UCDs to new fields has also been studied

- proposal of specific UCDs for radio data (A. Richards)
- proposal of specific UCDs for X-ray data (J. McDowell)

Here again, the level of granularity achieved in the description of the field must be reasonable. An infrared astronomer who wants to use radio data could be lost in over-detailed radio parameters.

⁶<http://www.ccsds.org/>