



*International
Virtual
Observatory
Alliance*

UCD (Unified Content Descriptor) - moving to UCD1+ Version 1.03

IVOA Working Draft 2004-04-26

This version:

<http://www.ivoa.net/Documents/UCD/WD-UCD-20040426.html>

Latest version:

<http://www.ivoa.net/Documents/latest/UCD.html>

Previous versions:

1.01 with additions from UCD 1.9.9*.

Editor(s):

S. Derriere, A. Preite Martinez

Authors:

Sébastien Derriere (derriere@astro.u-strasbg.fr)
Norman Gray (norman@astro.gla.ac.uk)
Robert Mann (rgm@roe.ac.uk)
Andrea Preite Martinez (andrea@rm.iasf.cnr.it)
Jonathan McDowell (jcm@cfa.harvard.edu)
Thomas Mc Glynn (Thomas.A.McGlynn@nasa.gov)
François Ochsenbein (francois@astro.u-strasbg.fr)
Pedro Osuna (Pedro.Osuna@esa.int)
Guy Rixon (gtr@ast.cam.ac.uk)
Roy Williams (roy@cacr.caltech.edu)

Abstract

This document describes the current understanding of the IVOA controlled vocabulary for describing astronomical data quantities, called Unified Content Descriptor (UCD).

It describes a new proposal (tentatively named UCD1+) for improving the first generation of UCD (hereafter UCD1). The basic idea is to adopt a new syntax, that will be compatible with forthcoming UCD2 and/or UCD3, while requiring little effort for people to adapt softwares already using UCD1.

We present how the proposed scheme has been successfully tested on Vizier, and what improvements it brings. We then describe how protocols and softwares using UCD1 could evolve to use these new terms, and what new functionalities could be made available.

As a practical example, we explain how matching functions can be built for the new scheme, and we conclude with simple scenarii of how UCD1+ can be used by astronomers/software.

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	Scope of UCD	3
1.1	A Controlled Vocabulary for Astronomy	3
1.2	Interoperability as a goal	3
2	UCD Syntax	4
2.1	Examples of Legal Syntax	4
2.2	Backus-Naur Form	4
3	Improving UCD1	5
3.1	From UCD1 to UCD1+	5
3.2	Defining simple words	5
3.3	Combining simple words	6
4	Consequences for VizieR	7
4.1	Building the list of simple words	7
4.2	Result	8
5	Consequences for services already using UCD1	8
6	Matching function	9
7	Use cases	10
7.1	Database Access and UCD: Translation Layer	10
7.2	UCDs in VO tools	11
7.3	UCDs in a registry	11

8	Software and Services	12
8.1	Services at CDS	12
8.1.1	Resolver	12
8.1.2	Listing and Browsing	12
8.1.3	Search Engine	12
9	UCD Steering Committee	13
9.1	Creation of a Board for New UCD Words	13
9.2	A procedure to request new UCD words	13
9.3	Creation of a Technical Board	13
9.4	Contact point for UCD issues	14
9.5	List of valid words and UCDS	14

1 Scope of UCD

1.1 A Controlled Vocabulary for Astronomy

The Unified Content Descriptor (UCD) is a formal vocabulary for astronomical data that is controlled by the International Virtual Observatory Alliance (IVOA). The vocabulary is *restricted* in order to avoid proliferation of terms and synonyms, and *controlled* in order to avoid ambiguities as far as possible. It is intended to be flexible, so that it is understandable to both humans and computers. UCD describe astronomical quantities, and they are built by combining words from the controlled vocabulary.

A UCD does not define the units or name of a quantity, but rather “*what sort of quantity is this?*”; for example **phys.temperature** represents a temperature, without implying a particular unit.

It would be possible to describe astronomical data quantities in a natural language such as English or Hungarian or Uzbek; however, it would be very difficult to expect a machine to ‘understand’ in any sense. At the opposite extreme, there is an attempt within the IVOA to describe astronomical data in terms of a hierarchical data model, so that there is a place for everything, and everything is in its place. The UCD vocabulary falls between these extremes, and is (we hope) understandable to both human and computer.

1.2 Interoperability as a goal

The UCD committee has tried to resist the temptation to allow the UCD syntax to be overly expressive. Every measurement in science has the possibility of essentially infinite description - the people, the instruments, the error analysis, the reasons, the funders, and so on. We have tried to find a way of organizing specifiers (words) so that it is easy to write simple software for machine use, but also possible to write better, more sophisticated software. We hope to build more sophisticated “intelligent” systems in the future, a project that has come to be called “UCD3”.

The major goal of UCD is to ensure interoperability between heterogeneous datasets. The use of a controlled vocabulary will hopefully allow an homogeneous, non-ambiguous description of concepts that will be shared between people and computers in the IVO.

We hope in the future to put more semantic expressiveness into the UCD framework, but always keeping a pragmatic eye on those who would create and use the software that is to “understand” UCD.

2 UCD Syntax

A UCD is a string which contains textual tokens that we shall call **words**, which are separated by semicolons (;). A word may be composed of several **atoms**, separated by period (.) characters. The order of these atoms induces a hierarchy. Standard UCD, which are validated by the IVOA, can start with the **ivoa:** namespace, but this namespace is optional. The use of namespaces, indicated by the presence of a colon in the word is possible, but should be avoided as far as possible. They should be used only temporarily, for words that are not yet included into the vocabulary validated by the IVOA, and they should be replaced by the standard word as soon as it is created. Section 9 describes a procedure for incorporation of new UCDs into the IVOA-approved list.

The character set that may be used in a UCD is the upper and lower-case alphabet, digits, hyphen and underscore. The colon, semicolon, and period are special characters as discussed above.

- There should be no space character within a UCD. Space characters or tabulations must be removed for a UCD to be well written.
- The UCD syntax is case-insensitive — all uppercase characters should be converted to lowercase before parsing.

2.1 Examples of Legal Syntax

The following examples have legal UCD syntax:

- **pos.eq.ra;meta.main**
- **meta.id;src**
- **phot.flux;em.radio;arith.ratio**
- **PHot.Flux;EM.Radio;ivoa:arith.Ratio**

Notice that the last two UCDs are identical because of the case insensitivity and because the default namespace is optional.

2.2 Backus-Naur Form

```
<Alpha> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
          |A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<char>  ::= <alpha>|<digit>|-|_
<semicolon> ::= ;
<period> ::= .
<colon> ::= :
<word-component> ::= <alpha>|<digit>|<word-component><char>
<namespace-ref> ::= <word-component>
<word> ::= <word-component>|<word><period><word-component>
<nword> ::= <namespace-ref><colon><word>|<word>
<UCD> ::= <nword>|<UCD><semicolon><nword>
```

Note: A UCD is always case-insensitive.

3 Improving UCD1

3.1 From UCD1 to UCD1+

What makes UCD1 easy to use is that they are simple strings: they can be considered as a single word. But the immediate drawback, as it has been discussed many times, is that this implies creating many new UCD1 for only slightly different things.

Consider the following list of 4 distinct UCD1 words:

- **POS_EQ_RA_MAIN**
- **POS_EQ_RA**
- **POS_EQ_DEC_MAIN**
- **POS_EQ_DEC**

They reduce in fact to only 3 elements (**POS_EQ_RA**, **POS_EQ_DEC**, and **MAIN**), that could be combined to build the 4 fully-qualified terms.

The idea of building UCDs by combining simple words makes the vocabulary less complex and more flexible (cf. “atomic UCDs” proposal by G. Rixon). The two questions that immediately arise are:

1. How do we define the simple words ?
2. How do we combine them to build fully-qualified UCDs ?

3.2 Defining simple words

There is no definitive answer to the first question, because selecting some terms for inclusion in the vocabulary and rejecting other terms is necessarily somehow subjective. The only possible validation of the selected vocabulary is to check its ability to describe properly a wide range of real data.

There are two caveats for the definition of the list of words:

- The words must not be too simple/ambiguous (or we have the same inconvenients as natural language);
- The words must not be too specific (or we come back to the UCD1 drawbacks).

In order to avoid ambiguities, **each word of the vocabulary will have an associated definition in plain text** (and possibly related keywords).

Words like **source** or **type** should only be used in the vocabulary with a very clear definition, and restrict only to one meaning in the case of homonyms (**source** can a priori mean an object in the sky, a program code, a bibliographic reference, ...).

For this reason, and also in order to group similar words, **words are composed of atoms**. The first atoms in a word generally help specifying the context, and help understanding the word without reading its definition (e.g. **pos.gal.lat** is the latitude in galactic coordinates while **pos.earth.lat** is the latitude on Earth).

3.3 Combining simple words

Our guideline for UCD1+ is that, while it is possible to build a UCD as a combination of several simple words, *the primary word carries most of the meaning as to “what the quantity is”*.

People or software who don’t want to manage composed UCD can use only the first word of the composed UCD (called *primary* word). This word must give a first order description of the quantity that is being described. It can be used as UCD1, with the only change that the underscore (`_`) is to be replaced by a period (`.`) in the parsing (cf. section 2 for syntax of UCD1+).

The choice of the primary word (when a complex element is to be described) should be guided by the answer to the question: “in *one* word, what is this element?”.

The units can give a hint to find the most appropriate primary word.

One UCD describes one element, and if several elements (e.g. columns of a table) are present, the possible relationships between the elements are not used for attributing UCDs.

Example: Consider a table containing 3 columns:

- A magnitude;
- A flag on this magnitude.
- An error on this magnitude;

We suggest that the primary word for the first column should be **phot.mag**: the contents of this column is a number, and the semantic meaning of this column is well described by the word **phot.mag** (whose definition is *photometric magnitude*).

The contents of the second column is a flag (often it is a symbol, like a, b or * that indicates, e.g. bad weather, unreliable values, ...). Therefore, the primary word should be **meta.code** (which means *code or flag*), because what is really described here is indeed a flag. The complete UCD could be written **meta.code;phot.mag**, to indicate that this flag applies a magnitude. A simple parser could keep only the primary word of this UCD, and still have a reliable description of what it is. It could also ignore the order of all secondary words.

The contents of the third column is an uncertainty, a measurement error. It can be expressed in magnitudes, but it is not a magnitude, so it is not correct to use **phot.mag** as primary word. One should use instead **stat.error** as the primary word, because the definition of this word corresponds precisely to the contents of the column. The complete UCD could be written **stat.error;phot.mag**, to indicate that this error applies a magnitude.

One could argue that these three columns are in fact related. This is correct, but it does not imply that the exact relation can be inferred from the UCD themselves. There are other expressive means to describe relationships between elements (e.g. use of `<GROUP>` tags in VOTable).

We decide to keep UCD1+ simple: they are just simple combinations of words that describe elements. The idea is that ultimately, a UCD3 system, using RDF and/or ontologies will allow a precise description of the relationships between elements. But this will lead to much more complex UCDs, that will most likely be no more human-readable (or writeable). **We hope**

however that most of the simple words that are defined in the UCD1+ vocabulary will be reusable in future evolution of UCDs.

We will see in section 7 how UCD1+ can be used in practice now, despite (or taking profit from?) their simplicity.

The order in which words are arranged after the primary word matters if the comparison of two UCD takes this order into account (see section 6 below). In the proposed scheme, UCD are built by adding words from left to right, with each new word specifying/qualifying the combination to its left.

Examples of UCD1+ and how they are built:

- *The maximum temperature of an instrument.* This is a *temperature*, so the primary word will be: **phys.temp**. This temperature is that of an *instrument*, so we specify it next: **phys.temp;instr**. And finally, we add a third word to indicate that this is the *maximum* value of a **phys.temp;instr**, giving the final UCD: **phys.temp;instr;stat.max**
- *The error on a magnitude measured in the V band.* The quantity is an *uncertainty*, so the primary word will be **stat.error**. This uncertainty applies to a *magnitude*, so we write **stat.error;phot.mag**. Then, we can specify the photometric band with another word, giving **stat.error;phot.mag;em.opt.V**.

In most cases, as we will show in section 4, one or two words are sufficient to form a UCD.

We can note that some of the words present in the vocabulary can not be used as primary words (e.g. most of the words starting with **em**. that only describe a part of the electromagnetic spectrum). Such words that can not be used as primary will be flagged in the list of standard words, so that people or tools trying to assign UCD1+ can avoid errors.

4 Consequences for VizieR

4.1 Building the list of simple words

This new scheme has been successfully applied to VizieR. Andrea Preite Martinez has been working a lot on the transformation of UCD1 into an improved version, concentrating on a very bottom-up approach, trying to build an homogeneous list of combinations of new words, describing all of the existing UCD1 terms.

The work of finding UCD1+ corresponding to UCD1 consists in finding combinations of simple words that will be used in practice (because they already are with UCD1 in VizieR), and thus an important step in settling UCD1+ on some solid ground.

In this process, the list of simple words forming the vocabulary of UCD1+ is built progressively: UCD1 are translated into word combinations, with new words created when necessary. Care was exercised in the choice of the words, so that those words are:

- unambiguous (avoid homonyms, and provide a clear definition for each word)
- reusable (because the UCD vocabulary should be flexible)
- consistently used (the same word is used to describe the same idea in different places)

Of course, the result will still certainly need a few iterations before some consensus is reached on the vocabulary.

4.2 Result

The first result of the translation of UCD1 into UCD1+ is a considerable simplification of the list of terms. The 1394 different UCD1 used in VizieR transform into 602 different UCD1+ combinations. These combinations use 416 different words in total, and only 347 different primary words are used (lists are available online, see 9.5).

The transition to UCD1+ brings some improvements:

- the new system is more flexible, because of the possibility to combine words: commonly used concepts (**meta.id**, or **phot.mag**) are described by simple words (that can possibly be used in combinations with other words to add some meaning) and allow simplifications of the vocabulary;
- the new system is more accurate than UCD1: one can be more expressive. For example, the UCD1 **ERROR** corresponds to the new word **stat.error**. But in practice, with UCD1+ we can assign many combinations starting with **stat.error** to the unique UCD1 **ERROR**;
- people who want to use one-word UCD can still interoperate with people/software that use composed UCD: the matching function should work because the primary word is the most important;
- the translation to the new system has allowed us to clean some problems in the current definition or usage of UCD1 within VizieR. These problems were encountered mainly with very specific UCD1 used in only one catalogue. There were also a few ambiguous definitions of UCD1, or problems not well assigned UCD in specific catalogues that were fixed.

For the moment we keep UCD1, but we would like to evolve to UCD1+ once agreement has been reached on the vocabulary.

5 Consequences for services already using UCD1

Services or protocols that already use UCD1 could evolve to use the new scheme with little extra work. This is because, in most cases, they use standard elements that can be easily expressed with simple combinations of words.

The flexibility of UCD1+ could also be exploited. For example, the Cone Search currently expects the use of the UCD1 **POS_EQ_DEC_MAIN**. This element would now be written **pos.eq.dec;meta.main**. The **main** word is in fact only useful when there are several values of declination in the same dataset. If there is only one value of a declination, it could be described by **pos.eq.dec**, and a flexible matching function could indicate that this UCD is compatible with the required **pos.eq.dec;meta.main** (cf section 6).

The definition of a new list of words is also the occasion to describe in an homogeneous way elements that do not exclusively come from VizieR:

- a description of FITS keywords coming from different image headers adds useful words to the vocabulary;
- the VOX words that are used for the SIAP protocol can be replaced by combination of UCD1+ words. Only a few words not used in the vocabulary defined from VizieR are required;
- the atomic and molecular data description is also underway, to provide an homogeneous description of large reference databases in this domain (BASECOL, ...).

6 Matching function

The goal of a matching function μ is to compare two UCDs and return a result indicating the similarity of the two UCD. In general $\mu(u_1, u_2)$ returns 1 if the two UCDs are strictly identical, and 0 if they are completely different.

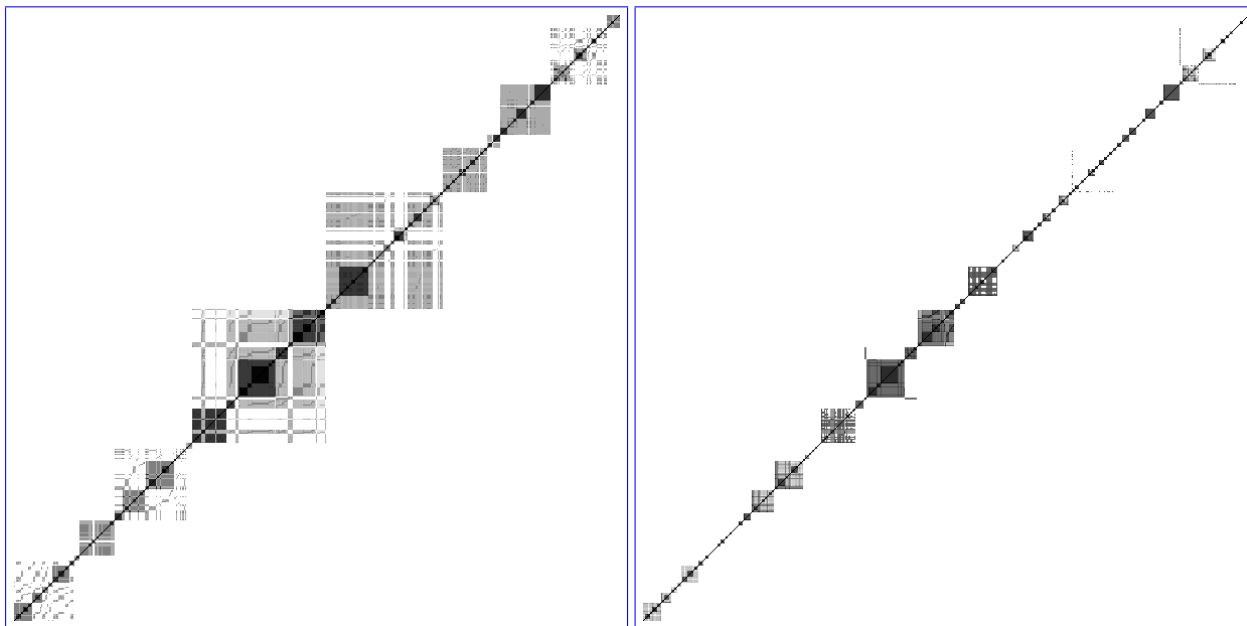


Figure 1: Illustration of two different matching functions (eq. 1 left panel, eq. 2 right panel). Images are made of 602x602 pixels. In each column, the greyscale encodes the value of the matching function ($\mu = 1$ is black, $\mu = 0$ is white) of one UCD compared to all other 602 UCD used in Vizier. The UCDs are sorted by alphabetic order, so that similar atoms for primary word are grouped together, giving the block-diagonal aspect. The diagonal corresponds to self-match with $\mu = 1$. One sees that matching functions can be made more or less restrictive.

In the simplest case, for UCD1, a simple string comparison can be used: if the two UCD1 are identical, $\mu = 1$, and if there is a difference, $\mu = 0$.

We suggest that this simple comparison with a binary result can still be used with UCD1+, with a comparison of the primary words of u_1 and u_2 respectively.

But it is possible to use more flexible matching functions, returning intermediate results between 0 and 1. The general idea is to compute a distance d between u_1 and u_2 . This distance can be computed by comparing the primary word w_{11} of u_1 with the primary word w_{21} of u_2 , and then the 2nd word w_{12} of u_1 with the 2nd word w_{22} of u_2 , etc... This distance can be a value between 0 and 1.

Because the primary word carries most of the meaning, it can have a more important weight. And subsequent words can have decreasing weights, like higher order terms in a series development.

For example, matching functions could use distances:

$$d = \left[d_1(w_{11}, w_{21}) + \frac{1}{2!} (d_2(w_{12}, w_{22}))^2 + \frac{1}{3!} (d_3(w_{13}, w_{23}))^3 \dots \right] \quad (1)$$

or

$$d = \left[d_1(w_{11}, w_{21}) + \frac{1}{2}d_2(w_{12}, w_{22}) + \frac{1}{3}d_3(w_{13}, w_{23})\dots \right] \quad (2)$$

An define $\mu(u_1, u_2) = \max(0, 1 - d)$ to ensure a result between 0 and 1. The individual distances between words can also be expressed as a series of terms built upon binary atom-to-atom comparison:

$$d_1(w_{11}, w_{21}) = \left[c(a_{111}, a_{211}) + \frac{1}{2}c(a_{112}, a_{212}) + \dots \right]$$

where $c(a_x, a_y)$ is an atom comparison function returning 0 if $a_x = a_y$ and 1 else.

With all distances truncated to 1, the above rules give interesting results. For two UCDs with completely different primary word (different first atom), the match is 0. The match comes closer to 1 when there are more identical atoms, and more similar words. And $\mu = 1$ when there is absolutely no difference.

Figure 1 illustrates the behaviour of two different matching functions (based on eq. 1 and 2, independently written by S.D and A.P.M., respectively) for the 602 different combinations of words used in VizieR.

These examples are of course not mandatory, and it is possible to imagine many different forms of matching functions for different purposes. What is interesting here is the flexibility offered by UCD1+ to compare slightly different elements: this allows for fuzzy searches of “quite similar” UCD.

Example: The matching function 2 will give the following result when evaluating a match of **phot.mag;em.opt.R**, with the 602 UCD1+ combinations in VizieR:

μ	UCD1+
1.00	phot.mag;em.opt.R
0.97	phot.mag;em.opt.R.Halpha
0.94	phot.mag;em.opt
0.89	phot.mag.sb;em.opt.R
0.83	phot.mag;em.opt.B
0.83	phot.mag;em.opt.I
0.83	phot.mag;em.opt.U
0.83	phot.mag;em.opt.V

7 Use cases

7.1 Database Access and UCD: Translation Layer

UCD will be used in practice for **exchanging** information using a controlled vocabulary. They are used in the VOTable standard to attach a standard description to table column names, for example. The data providers do not need to change the internal descriptions of their existing databases. Nor is it required that people building from scratch a new VO-compliant service use UCD in the core of their system.

What is needed for interoperation with other systems is a “**translation layer**” that is able to associate UCD to the parameters that are used internally, so that the output of the service contains a standard description that can be interpreted by other VO services.

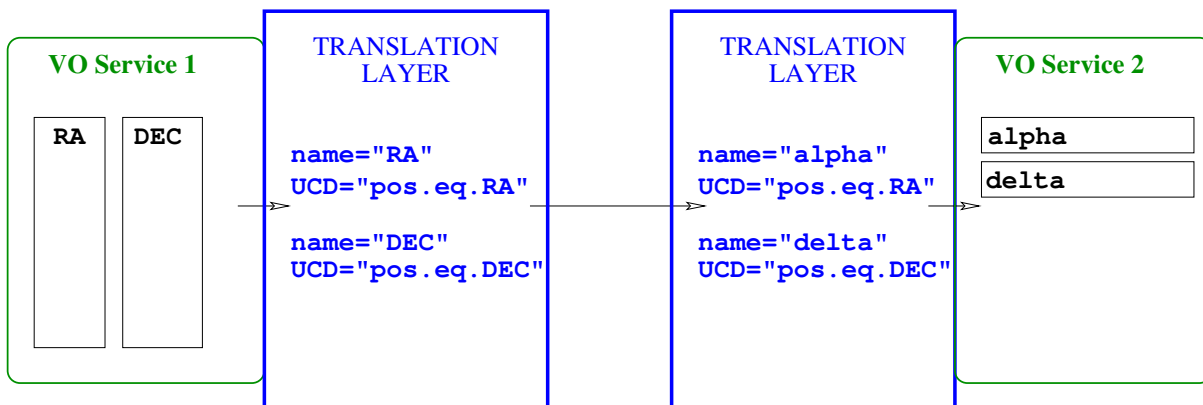


Figure 2: Services use UCD to exchange information. A translation layer is used to interpret the internal description in terms of UCD.

In Fig. 2, a first VO service describes internally the right ascension and declination with names RA and DEC. For sending data to another service expecting right ascension and declination as an input, it uses a translation layer to attach UCD to its parameters. The second service also has a translation layer that can interpret UCD into its own parameters.

The mapping done by the translation layer can be done using XML files. For the second service above, we can specify that quantities corresponding to UCD **pos.eq.ra** and **pos.eq.dec** are to be found in the database table Obs-Table, which has column names alpha and delta:

```
<?xml version='1.0'?>
<!DOCTYPE ucdToDb SYSTEM 'ucdToDb.dtd'>
<ucdToDb>
  <ucd name="pos.eq.RA" table="Obs-Table" col="alpha" />
  <ucd name="pos.eq.DEC" table="Obs-Table" col="delta" />
  <ucd name= ... />
</ucdToDb>
```

7.2 UCDs in VO tools

There are already applications that use UCD to manipulate or display some data to the user, or to find required fields (VOPlot, Filters in Aladin), ...

If they want not to change, they can use the primary word only.

With UCD1+, it is possible to be more flexible, and to find the “most appropriate” element in a dataset.

Consider a tool that expects to find a field with UCD **pos.eq.ra;meta.main**. Using a custom matching function to analyze the contents of a VOTable file, this tool could consider that **pos.eq.ra** matches in the absence of **pos.eq.ra;meta.main**, and pick that column as the expected one.

7.3 UCDs in a registry

Consider a registry containing descriptions of catalogues, with the associated UCD. The benefit of having acces to the contents in terms of UCD is that it is possible to explore the contents of a catalogue more extensively than with simple keywords.

E.g., a catalogue dedicated to very accurate measurement of proper motions and parallaxes will certainly put keywords for these, but it might also contain a column that measures a radial velocity. With UCDs assigned, this column could be identified and the catalogue selected for someone searching for radial velocities, even if this is not the primary goal of the catalogue.

It is however not necessary to describe every element of a dataset by UCDs... Only the most relevant column need have UCDs attached to them. Parameters used for internal processing by a service do not need to have UCDs attached.

Consider the catalogue above described with UCDs in a registry. A query by UCD allows to locate this catalogue and find that it contains radial velocities.

Once the resource is located, one can then send a query to this resource, either on its specific parameters or again using UCDs.

Because UCD1+ have a more flexible syntax, it is possible to make some kind of fuzzy search, with the help of matching functions in the case of the search in a registry.

The different possible levels of granularity in the description allow more interoperability.

8 Software and Services

What is the nature of the software and services that will work with UCD?

8.1 Services at CDS

Several web services have been implemented at CDS Strasbourg to aid in the exploitation of UCD. Those below are available at <http://cdsweb.u-strasbg.fr/UCD/>.

These services were originally built for UCD1, and they will be upgraded to make use of the new UCD1+.

The following list covers some of these:

8.1.1 Resolver

The resolver service: given a UCD, the associations of the previous section will enable us to get a textual description of what it means.

8.1.2 Listing and Browsing

These services allow a dynamic view of the tree of UCD, either as a single text file, or as a Javascript-enabled tree-browser.

8.1.3 Search Engine

This service allows the input of natural language, and it searches for matches in the text description of the UCDs. A further extension connects to metadata about Vizier tables that use those UCDs. This tool can be used to find an appropriate UCD for labeling data. A batch-oriented version accepts a file of keywords, data types, and other information and tries to find suitable UCDs.

9 UCD Steering Committee

9.1 Creation of a Board for New UCD Words

We believe that the inclusion of new UCD words must be a flexible process, yet controlled. The best way to accomplish these two needs would be to create a proper scientific board that would study new UCD requirements and, within a given period of time, give an answer as to whether a new UCD must or must not be included in the UCD standards.

The use of “mission-specific” namespaces has been addressed in many occasions, and we believe that namespaces should be avoided as much as possible. There has been an exercise in revising the VOX words for the SIAP protocol and trying to assign existing UCDs to them, or proposing new UCD words for the non-existing ones.

The responsibility of the board would consist of studying the cases where a UCD word is proposed and to figure out whether the proposed word should be accepted or rejected, and in case of rejection recommending the closest existing word that should be used.

In case a new word is accepted into the main tree, an internal procedure should be established so that the new UCD becomes live after a proper internal new release in a short period of time. It should be agreed whether this board would study the proposed cases in an “on demand” basis or would collect requests and study them on a periodic basis.

A suggestion on the formation of this scientific committee would be that it might contain people from CDS (as they have the experience and the resources) but it should be offered to all relevant parties. It would also be very important to have a member from the data providers community, as the scientists view on some issues might not include other important views from data providers.

9.2 A procedure to request new UCD words

A procedural document should be created to make it easy to a user to ask for a new UCD and to understand the implications of doing so. This document would address:

- the contact point to ask for new UCD
- the life-cycle of the process of asking for a new UCD
- when and how a new UCD becomes live
- what to do if a UCD is rejected

This type of actions could (and should) be supported by tools like an automatic form that is filled in and sent to the scientific board, giving an answer back to the user acknowledging the request, and giving a time estimate for an answer. All these issues will be suggested in a separate point. Lessons should be learnt from other projects where similar boards exist. There should be a thorough investigation (maybe from the board mentioned above) of how other projects have worked in this direction (like the Planetary Data System (PDS), the FITS consortium, the W3C) and try to get the right things from them while avoiding the wrong ones.

9.3 Creation of a Technical Board

There should be tools available for the user to check for the existence of UCDs, etc. Some of these tools exist already in CDS, and they are good candidates to become the sort of “official” tools for the UCD standards. However, we feel it is necessary to have a proper technical board that could, eventually, decide on what tools are really necessary to make the UCD work feasible and as easy

as possible for the user. This board would be mainly in charge of writing proper requirements for the tools. The management of resources, etc., should be handled by the concepts wanting to work for the VO project, but the definitions of requirements, etc., should be centralized on this board.

9.4 Contact point for UCD issues

We feel the necessity to create a contact point to which all UCD related matters can be addressed. This contact point could be a web address devoted explicitly to that in the context of the VO, a properly organized web place, where all the tools would be available, as well as all documents and procedures for creation of new UCD words, etc., with practical examples and the like.

9.5 List of valid words and UCDs

The list of valid words is not included in this document, as it is subject to changes. The list of valid words is available online, together with previous versions and the history of changes that have been made: <http://vizier.u-strasbg.fr/UCD/lists/>