International
Virtual
Observatory
Alliance

Units in the VO
Version 1.0

IVOA Working Draft December 16, 2011

This version:

http://www.ivoa.net/Documents/VOUnits/20111216/

Latest version:

http://www.ivoa.net/Documents/latest/VOUnits.html

Previous versions:

http://www.ivoa.net/internal/IVOA/UnitsDesc/WD-VOUnits-v0.3-20110508.pdf

Editor(s):

Sébastien Derrière

Authors:

Sébastien Derriere, François Bonnarel, Norman Gray, Mireille Louys, Jonathan Mc Dowell, François Ochsenbein, Pedro Osuna, Bruno Rino, Jesus Salgado

Contents

Abstract

This document describes common practices in manipulating units in astronomical metadata and will define a means of consistent representation within VO services.

The core of the document contains the rules for writing string representations of unit labels, called VOUnits.

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 rendered obsolete 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".

This document is a substantial update of the previous version 0.2 that was written within the Data Model IVOA Working Group. As decided in previous IVOA interoperability meetings, the Semantics working group is now in charge of the document. This document is intended to become a full IVOA recommendation, following agreement within the community and standard IVOA recommendation process.

The place for discussions related to this document is the Semantics IVOA mailing list `semantics@ivoa.net`.

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

1 Introduction

This document describes a proposal for the use of Units in the VO (hereafter simply "VOUnits"). The introduction gives the motivation for this proposal in the context of the VO architecture, from the legacy metadata available in the resource layer, to the requirements of the various VO protocols and standards, to the user layer, including VO applications.

After reviewing and comparing the current practices in the description and usage of units in section 2, section 3 details the proposal for VOUnits. Section 4 lists some use cases and reference implementations.

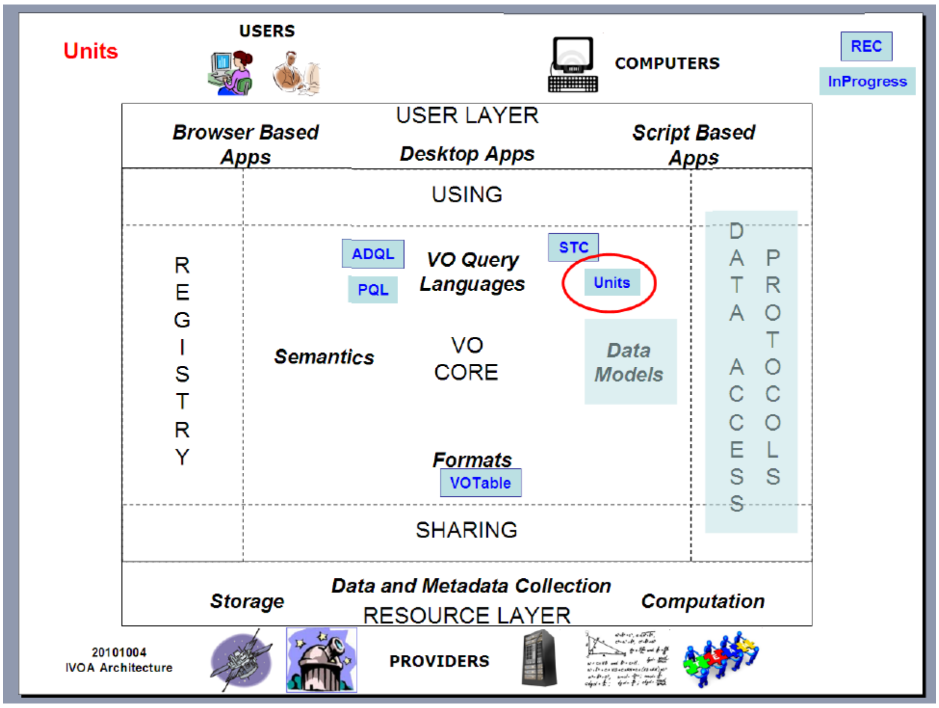

Figure 1: Units in the VO architecture

1.1 Units in the VO Architecture

Generally, every quantity provided in astronomy has a unit attached to its value or is unitless (e.g. a ratio, a numerical multiplier).

The Units lie at the core of the VO architecture, as can be seen in Fig. 1. Most of the existing data and metadata collections accessible in the resource layer do have some legacy units, which are mandatory for any scientific use of the corresponding data. Units can be embedded in data (e.g. FITS headers) or be implied by convention and/or (preferably) specified in metadata.

Units also appear in the VOTable format [VOTable], through the use of a `unit` parameter that can be used in the `FIELD`, `PARAM` and `INFO` elements. Because of the widespread dependency of many other VO standards on VOTable, these standards inherit a dependency on Units.

The Units also appear in many Data Models, through the use of dedicated elements in the models and schemas. At present, each VO standard either refers to some external reference document, or provides explicit examples of the Units to be used in its scope, on a case-by-case basis.

The registry records can also contain units, for the description of table metadata. The definition of VO data protocols uses units by specifying in which units the input parameters have to be expressed, or by restricting the possible units in which some output must be returned.

And last but not least, the tools can interpret units, for example to display heterogeneous data in a single diagram by applying conversions to a reference unit on each axis.

1.2 Adopted terms and notations

Discussions about units often suffer from misunderstandings arising from cultural differences or ambiguities in the adopted vocabulary. For the sake of clarity, in this document, the following concepts are used :

- a **quantity** is the combination of a (numerical) *value*, measured for a *concept* and expressed in a given *unit*. In the VO context, the nature of the concept can be expressed with a UCD or a utype. This document does not address the full issue of representing quantities, but focusses on the *unit* part.
- a **unit** can be expressed in various forms: in natural language (e.g. *meters per second squared*), with a combination of symbols with typographic conventions (e.g. $\mathsf{m\ s^{-2}}$), or by a simplified text label (e.g. m.s-2). VOUnit deals with the label form, which is easier to standardize, parse and exchange. A VOUnit corresponds in the most general case to a combination of several (possibly prefixed) symbols with mathematical operations expressed with the appropriate symbols.
- a **base unit** is represented by a **base symbol**, with unambiguous meaning.
- a **prefix** or **scale factor** is prepended to a **base symbol** to scale it by factors of ten.
- a **symbol** or sym is either a base symbol or a prefixed base symbol.

Remark : some complex questions, more related to data modeling than to units, such as how a quantity is associated to its measurement error, or how groups of coordinates are described, are not addressed in this document. They can always be broken down, with appropriate modeling, into smaller bits to which VOUnits can be applied.

1.3 Purpose of this document

The purpose of this document is to provide a reference specification on how to write VOUnits, in order to maximize interoperability within the VO.

VOUnits will not try to reinvent the wheel, and be as compliant as possible with legacy metadata in major archives, and astronomers' habits.

By explicitly stating what must (and must not) be used, what may be used, and giving clear recommendations as to the preferred practices, it should avoid some common misinterpretations and confusions that are often done when dealing with units.

Data providers are encouraged to follow the VOUnits specifications for expressing their metadata. And application developers can rely on these specifications in order to know what VOUnits they should expect to face.

1.4 What this document will not do

This document is **not** prescribing what units data providers employ, nor enforcing that a given quantity be expressed in a unique way (e.g. all distances in m). So long as data are labelled in a recognised system, a translation layer can be provided. Data providers can customise the translation tools if required. Depending on preference and the operations required, the user may have a choice of units for their query and for the result.

VOUnits do not specify how transformation of quantities is done. But VOUnits describe the units of each quantity in a standard way, allowing to do the conversion if the underlying transformation model is known. This is the case for simple operations such as converting light wavelengths into frequencies, but also for more complex ones such as coordinate conversions or converting magnitudes into fluxes.

2 Current use of units

Many other projects have already produced lists of preferred representations of units. Those most commonly used in astronomy are described in this section.

The four first schemes described below are used as references for the comparison tables presented later in this document.

2.1 IAU 1989

In the section 5.1 of its Style Manual [IAU 1989], the IAU gives a set of recommendations for representing units in publications. This document therefore provides useful reference guidelines, but is not directly applicable to VOUnits because the recommendations are more intended for correct typesetting in journals than for standardized metadata exchange. The IAU style will be summarized in the second column of the comparison tables.

2.2 OGIP 1993

NASA has defined a list of character strings specifying the basic physical units used within OGIP FITS files [OGIP 1993]. Rules and guidelines on the construction of compound units are also outlined.

HEASARC datasets follow these conventions, presented in the third column of the comparison tables.

2.3 Standards for astronomical catalogues

The conventions adopted at CDS are summarized in the Standards for Astronomical Catalogues, Version 2.0 [StdCat 2000]. They are presented in the fourth column of the comparison tables.

2.4 FITS 2010

Section 4.3 of the reference FITS paper [FITS] describes how unit strings are to be expressed in FITS files. The recommendations are presented in the fifth column of the comparison tables.

2.5 Other usages

- `http://arxiv.org/pdf/astro-ph/0511616`
  Dimensional Analysis applied to spectrum handling in VO context (Osuna & Salgado 2008) offers a mathematical framework to guess and recompute SI units for any quantity in astronomy.
- `http://www.mel.nist.gov/msid/sima/07_ndml.htm`
  NIST (National Institute of Standards & Technology) project UnitsXML builds up an XML representation of units at the granularity level of a simple symbol string
- `https://jsr-275.dev.java.net/`
  JAVA JSR-275 specifies Java packages for the programmatic handling of physical quantities and their expression as numbers of units.
- `aips++` `http://aips2.nrao.edu/docs/aips++.html` and
  `casacore` `http://code.google.com/p/casacore/`
  contain modules handling units and quantities with high precision. The packages are mainly in use for radio astronomy but are designed to be modular and adaptable. (NB contrary to the statement on the casacore link, aips++ is still very much in use as the toolkit behind the CASA package.)

- IAU SOFA http://www.iau-sofa.rl.ac.uk/ and
  USNO NOVAS http://aa.usno.navy.mil/software/novas/novas_info.php
  implement the IAU 2000 recommendations.

3 Proposal for VOUnits

By systematically comparing four reference schemes, the rules for VOUnits are defined in this section. Various aspects are addressed :

- how the labels are encoded;
- what base symbols are allowed and how they are spelled;
- what prefixes are allowed and how they are used;
- how symbols are combined.

3.1 String representation and encoding

	IAU	OGIP	StdCats	FITS	VOUnits
Units are strings of chars		YES		YES	YES
Case sensitive	YES	YES	YES	YES	YES
Character set			No spaces	ASCII text	ASCII printable

Table 1: Comparison of string representation and encoding.

VOUnits must be unit labels suitable for IVOA use (or other electronic manipulation), and therefore need to be expressed unambiguously in encodings recognised by XML parsers, Java compilers, etc.

Following the current usage in unit labels (see Table 1), *VOUnits must be case-sensitive strings of chars consisting of printable ASCII characters* (20 to 7E in hexadecimal).

It is therefore not allowed to use special characters such as Å or µ in VOUnits.

3.2 Base units

There is a good agreement for the base symbols across the different schemes (see Tab. 2). VOUnits follow the same notations.

For masses, **kg** is preferred, but prefixes should be used with **g**.

	IAU	OGIP	StdCats	FITS	VOUnits
The 6+1 base SI units	m, s, A, K, mol, cd				idem
... mass base unit	kg, but use prefixes with g	kg	g	kg, note that g is allowed	g or kg ?
Dimensionless planar and solid angle	rad, sr				idem
	Tab. 2 Use s, NOT sec	Tab. 1		Tab 3,4. deg preferred for decimal angles	
Derived units with symbols	Hz, N, Pa, J, W, C, V, S, F, Wb, T, H, lm, lx				idem
	Ω Tab. 3	ohm	Ohm	Ohm	?

Table 2: Comparison of base units.

3.3 Scale factors

Following the usage in Tab. 3, there are 20 scale factors that can be used as prefixes to any base symbol. One must not use compound prefixes. Prefixes are concatenated to the base symbol without space, and can not be used without a base symbol.

	IAU Tab. 5	OGIP Tab. 3	StdCats 3.2.3	FITS Tab. 5	VOUnits
Scale factors, (multiple) prefixes	d, c, m, n, p, f, a				idem
	da, h, k, M, G, T, P, E				
	μ	u			u
		z, y, Z, Y			z, y, Z, Y

Prefix–symbol concatenation	no space, regarded as single symbol	no space, regarded as a single unit string	no space	no space (implicit)	no space
Prefix-able symbols	Not **kg** : use **g**	All units above, and **eV, pc, Jy, Crab** Only **mCrab** allowed	all	all	all
Use compound prefixes	should not	should never	must not	must not	must not

Table 3: Comparison of scale factors.

Remark: the letter **u** is used instead of the µ symbol to represent a factor of 10^{-6}, following the character set defined in section 3.1.

3.4 Astronomy symbols

The Table 4 lists symbols used in astronomy to describe times, angles, distances and a few additional quantities.

	IAU	OGIP	StdCats	FITS	VOUnits
minute	**min**, $^{\mathrm{m}}$	**min**	**min**	**min**	**min**
hour	**h**, $^{\mathrm{h}}$	**h**	**h**	**h**	**h**
day	**d**, $^{\mathrm{d}}$	**d**	**d**	**d**	**d**
year	**a**	**yr**	**a, yr**	**a, yr** Pa (peta a) forbidden	like FITS
arcsecond	″	**arcsec**	**arcsec**	**arcsec**	**arcsec**
arcminute	′	**arcmin**	**arcmin**	**arcmin**	**arcmin**
degree (of angle)	°	**deg**	**deg**	**deg**	**deg**
milliarcsecond	**mas** (use **nrad** !)		**mas**	**mas**	**mas**

microarcsec			uarcsec		?
cycle	c, ^{c}				not used
astronomical unit	au	AU	AU	AU	AU
parsec	pc				pc
atomic mass	u			u	u
electron Volt	eV				eV
jansky	Jy				Jy
Celsius degree	°C for meteorology, other use K				not used
century	ha, cy should not be used				no symbol

Table 4: Comparison of astronomy-related units.

Minutes, hours, and days of time must be represented in VOUnits by the symbols min, h and d. The year can be expressed by a (recommended by IAU) or yr (common practice), but peta-year must only be written Pyr because Pa is the base symbol for Pascal.

The astronomical unit must be expressed in upper-case, AU, following the legacy practice, and in order to avoid the (unlikely) confusion with atto-atomic mass.

There are no VOUnit symbols for Celcius degrees or century. Temperatures are expressed in Kelvin (K), and a century corresponds to ha or hyr.

3.5 Other symbols

Table 5 corresponds to Table 7 in the IAU document, and the IAU strongly recommends to no longer use these units.

	IAU	OGIP	StdCats	FITS	VOUnits
ångström	Å	angstrom	0.1nm	Angstrom	angstrom, Angstrom
micron	µ				
fermi	no symbol				
barn	b	barn	barn	barn	barn

cubic centimetre	cc				
dyne	dyn				
erg	erg	erg	No symbol. mW/m2 used for erg.cm-2.s-1	erg	erg
calorie	cal				
bar	bar				
atmosphere	atm				
gal	Gal				
eotvos	E				
gauss	G	G		G	G
gamma	γ				
oersted	Oe				
Imperial, non-metric	should not be used				not used

Table 5: Comparison of symbols deprecated by IAU.

In order to be compatible with legacy metadata, VOUnit parsers should be able to interpret symbols for Ångström, barn, erg and Gauss, as written in the sixth column of table 5. However, data producers are strongly advised to prefer the equivalent notation using symbols and prefixes listed in tables 2, 3 and 4.

Table 6 compares other various symbols. The VOUnits symbol for magnitudes is mag.

	IAU	OGIP	StdCats	FITS	VOUnits
magnitude	mag				mag
rydberg			Ry	Ry	
solar mass	$M_\odot$		solMass	solMass	
solar luminosity			solLum	solLum	
solar radius			solRad	solRad	
light year		lyr		lyr	
count		count	ct	ct, count	

photon		photon		ph, photon	
rayleigh				R	
pixel		pixel	pix	pix, pixel	
debye			D	D	
relative to Sun			Sun	Sun	
channel		chan		chan	
bin		bin		bin	
voxel		voxel		voxel	
bit			bit	bit	
byte		byte	byte	byte	
adu				adu	
beam				beam	
		Crab avoid use			
No unit, dimensionless		blank string	-		empty string
Unitless in percent			%		
unknown		UNKNOWN			

Table 6: Comparison of other symbols.

It can be noted that some of the units listed in Tab 6 are questionable. They arise in fact from a need to describe quantities, when the only piece of metadata available is the unit label. Count, photon, pixel, bin, voxel, bit, byte are concepts, just as apple or banana. The associated quantities could be fully described with a UCD, a value and an void unit label.

It is possible to count a number of bananas, or to express a distance measured in bananas, but does this make banana a reference unit?

The FITS document provides the most general description of all the compared schemes, and VOUnits adopts similar definitions, in order to acknowledge at best legacy metadata. Note that all symbols like count, photon, pixel are always used in lower case and singular form.

The recommended way to indicate that there is no unit (for example a quantity that is a character string) is to use an empty string.

3.6 Combination of symbols and mathematical expressions

Table 7 summarizes how mathematical operations can be applied on unit symbols for exponentiation, multiplication, division, and other computations.

	IAU	OGIP	StdCats	FITS
Multiplication	space, except if previous unit ends with superscript; dot (.) may be used	one or more spaces OR one asterix (*) with optional spaces on either side	dot (.), no space	single space OR asterix (*, no spaces) OR dot (., no spaces)
Division	per. Use negative index or solidus (/)	slash (/) with optional spaces on either side, space not recommended after / OR negative index	/ with no spaces	/ with no spaces
Use of multiple /	MUST never two	allowed	allowed	may be used, discouraged, math precedence rule
sym raised to the power y	superscript	sym**(y) parenthesis optional if $y > 0$	nothing: sym$y$ use +/- sign for 10+21	sym$y$ OR sym**(y) OR sym^(y), no space

Exponential of sym		exp(sym)		exp(sym)
Natural log of sym		ln(sym)		ln(sym)
Decimal log of sym		log(sym)	[sym]	log(sym) dimensionless argument
Square root of sym		sqrt(sym)		sqrt(sym)
Other math		sin(sym), cos(sym), tan(sym), asin(sym), acos(sym), atan(sym), sinh(sym), cosh(sym), tanh(sym)		
( )		any allowed	allowed	optional around powers
powers	superscripts	decimal and integer fractions allowed	integers only	integer (sign and () optional), OR decimal or ratio between ()

Table 7: Comparison of mathematical expressions and symbols combinations.

The combination rules is the point where the largest discrepancies between the different schemes appear. The FITS document provides a nice summary trying to accomodate at best the existing schemes, quoting section 4.3.1 of the paper :

> Two or more base units strings (called str1 and str2 (...)) may be combined using the restricted set of (explicit or implicit) operators that provide for multiplication, division, exponentiation, raising arguments to powers, or taking the logarithm or square-root of an argument. Note that functions such as log actually require dimensionless arguments, so that log(Hz), for example,

> actually means $\log(x/1\text{Hz})$. The final units string is the compound string, or a compound of compounds, preceded by an optional numeric multiplier of the form 10**k, 10^k, or 10±k where k is an integer, optionally surrounded by parentheses with the sign character required in the third form in the absence of parentheses. Creators of FITS files are encouraged to use the numeric multiplier only when the available standard scale factors (...) will not suffice. Parentheses are used for symbol grouping and are strongly recommended whenever the order of operations might be subject to misinterpretation. A space character implies multiplication which can also be conveyed explicitly with an asterisk or a period. Therefore, although spaces are allowed as symbol separators, their use is discouraged. Note that, per IAU convention, case is significant throughout. The IAU style manual forbids the use of more than one slash (/) character in a units string. However, since normal mathematical precedence rules apply in this context, more than one slash may be used but is discouraged.
>
> A unit raised to a power is indicated by the unit string followed, with no intervening spaces, by the optional symbols ** or ^ followed by the power given as a numeric expression (...). The power may be a simple integer, with or without sign, optionally surrounded by parentheses. It may also be a decimal number (e.g., 1.5, 0.5) or a ratio of two integers (e.g., 7/9), with or without sign, which must be surrounded by parentheses. Thus meters squared may be indicated by m**(2), m**+2, m+2, m2, m^2, m^(+2), etc. and per meter cubed may be indicated by m**-3, m-3, m^(-3), /m3, and so forth. Meters to the three-halves power may be indicated by m(1.5), m^(1.5), m**(1.5), m(3/2), m**(3/2), and m^(3/2), but not by m^3/2 or m1.5.

VOUnits follow the same rules as FITS, as summarized in Tab. 8.

3.7 Remarks and good practices

VOUnits, as described in this document, provide a string serialization of unit labels compatible with the two principal means of representation expected in the VO :

- as an attribute string in a VOTable document : `unit="..."`
- as a unit element in an XML serialization of some data model : `<unit>...</unit>`

str1 str2	Multiplication
str1*str2	Multiplication
str1.str2	Multiplication
str1/str2	Division
str1**expr	Raised to the power expr
str1^expr	Raised to the power expr
str1expr	Raised to the power expr
log(str1)	Common Logarithm (to base 10)
ln(str1)	Natural Logarithm
exp(str1)	Exponential (e^{str1})
sqrt(str1)	Square root

Table 8: Combination rules and mathematical expressions for VOUnits.

Avoiding the use of special characters (Å, ', ", ...) will reduce the complexity of using VOUnits in query languages or other programmation environments.

Even if enclosing VOUnits with simple or double quotes should be sufficient to facilitate parsing of a general query containing VOUnits, avoiding the use of white spaces for multiplication should make the parsing even easier, with the unit label being a single "word", and therefore the notation in the first line of Tab 8 is discouraged.

Complex data formats are not addressed by VOUnits. While possibly convenient for humans, sexagesimal coordinates or calendar dates expressed in ISO 8601 are quantities represented in a complex format, encoded as strings, and as such the corresponding VOUnit should be an empty string. Expressions such as "d:m:s" or "ISO 8601" are not valid VOUnits. This should not be a problem, as existing VO standards already recommend that coordinates be expressed in decimal degrees.

One can also remark that some quantities like the Modified Julian Date (MJD) are not recognized VOUnits. As described in section 1.2, the quantity MJD can be seen as a concept (described by the appropriate UCD or utype), and the corresponding value will most likely be expressed in days, so the VOUnit will be d. There is no need in overloading VOUnits to incorporate the description of concepts themselves.

The notion of unit conversion and quantity manipulation is discussed in section 4.3.

4 Use cases and applications

4.1 Unit parsing

The rules defined in section 3 allow to build VOUnit parsers. Several services can be built on top of a VOUnit parser :

1. Validation. A service checking that a VOUnit is well written. The output of such a service can have different levels: fully valid unit; valid syntax, but not the preferred one (e.g. use of deprecated symbols); parsing error.
2. Explanation. A service returning a plain-text explanation of the unit label.
3. Typesetting. A service returning an equivalent of the unit label suitable for inclusion in a LaTeX or HTML document.
4. Dimensional equation. As described in [DimEq], VOUnits can be translated into a dimensional equation, allowing to build up conversions methods from one string representation to another one (see also section 4.3).

4.2 Libraries

There are several existing libraries able to interpret unit labels. In all cases, some software effort is required if they are to be used in translating between data provider unit labels, and those to be adopted by the IVOA for internal use.

One of the most widely-used specialised astronomical libraries is AST which includes a unit conversion facility attached to astronomical coordinate systems [AST].

Another library has been developed at CDS[1], and can be tested online[2]. This library covers all the symbols and notations defined in the standard for astronomical catalogues [StdCat 2000], as well as additional symbols and notations.

Recently, Norman Gray started developing Unity[3], a new library intended to parse VOUnits, and compatible with the OGIP, StdCats and FITS formats.

[1] http://cds.u-strasbg.fr/resources/doku.php?id=units
[2] http://cdsweb.u-strasbg.fr/cgi-bin/Unit
[3] https://bitbucket.org/nxg/unity

4.3 Unit conversion and quantity transformation

Unit conversion is the simple task of converting a quantity expressed in a given unit into a different unit, while the concept remains the same. For example, converting a distance in **pc** in to a distance in **AU** or **km**. Or converting a flux from **mJy** to **W.m-2.Hz-1**. This is rather easy with existing libraries, using dimensional analysis or SI units as a reference.

Quantity transformation consists in deriving a new quantity from one or several original quantities. It is more complex, because it requires to have a precise model (a simple equation in simple cases) for computing the transformation. The model involves quantities, each described with a UCD or utype, value and VOUnit. Some of the quantities involved might be physical constants (e.g. Boltzmann's constant k_{B}).

Examples of such transformations can be :

- linear unit conversion: a distance is measured in **pixel** in an image, and needs to be transformed in the corresponding angular separation in **arcsec**. This can be done if the quantity representing the pixel scale is given, with its value and a compatible unit like **deg/pixel**.
- converting a photon wavelength in the corresponding photon energy or frequency.
- deriving the flux for a given photon emission rate (in `W`) from Planck's constant (`6.63` 10^{-34} `J.s`), the radiation frequency (in `GHz`), and the number of photons emitted per second.
- transforming a magnitude into a flux, as needed for SED building.

VOUnits can help in quantity transformation if all quantities are qualified with proper VOUnits.

4.4 Query languages

Including VOUnits in queries is not an easy task. Some guidelines were defined in the reflexion on ADQL.

1. All data providers should be encouraged to supply units for each column of a table. Columns should also have associated UCDs, so that quantities can be properly identified.
2. The IVOA needs to provide a parser to relate the native units to the standard IVOA labels.
3. The default response to a query which does not specify units, will be in the native units of the table.

4. Where queries involve combining or otherwise operating on the content of columns to produce an output column with modified units, we can provide libraries and a parser to assist in assigning and checking a new unit, and attach this to the returned values via the SQL CAST operator. This is implemented already in database related applications such as Saada, for instance. If any column used in responding to a query lacks a necessary unit, the output involving that column will be unitless.
5. If the user wants to change the output units with respect to the table units, this could be done by specifying the units in the initial SELECT statement. There are several issues to consider:
   (a) Does the user also need to include the conversion expression, or does the unit parser take care of that?
   (b) Can the user use this to assign units (based on prior knowledge) to output from a column lacking a unit?

4.5 Broader use in the VO

Different VO entities require and consume metadata with units attached like registries, applications and interoperate via protocols. Figure 2 illustrates the places where the IVOA could intervene to ensure consistent use of units.

Acknowledgements

We thank all those participants in IVOA and EuroVO workshops who have contributed by exposing use cases and proving comments, especially Paddy Leahy, Jeff Lusted, Arnold Rots, Mark Taylor, Brian Thomas and recent contributors on the DM forum. This document is just an update of the previous draft discussed in Strasbourg Interop meeting in 2009. It will be completed with appropriate grammar and tools to check different levels of consistency for a unit string used in the VO context.

References

[AST] AST Library, D. Berry at al. 2009, http://starlink.jach.hawaii.edu/starlink/AST

[DimEq] Osuna P. & Salgado, J., 2008 http://arxiv.org/pdf/astro-ph/0511616

[FITS] Pence et al., 2010, A&A 524, A42, Definition of the Flexible Image Transport System (FITS), version 3.0

[IAU 1989] IAU Style Manual by G.A. Wilkins, Comm. 5, in IAU Transactions XXB (1989)
http://www.iau.org/static/publications/stylemanual1989.pdf

[OGIP 1993] George I. M. and Angelini L., Specification of Physical Units within OGIP FITS files
http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/docs/general/ogip_93_001/

[SLALIB] http.www.starlink.rl.ac.uk/star/docs/sun67.htx/sun67.html

[StdCat 2000] Standards for Astronomical Catalogues, Version 2.0, 2000,
http://cds.u-strasbg.fr/doc/catstd-3.2.htx

[VOTable] Ochsenbein, F. et Williams, R., 2009, VOTable Format Definition 1.2 http://www.ivoa.net/Documents/VOTable/20091130/

A Updates of this document

- 0.3 to 1.0-20111216 Major rework of the document.

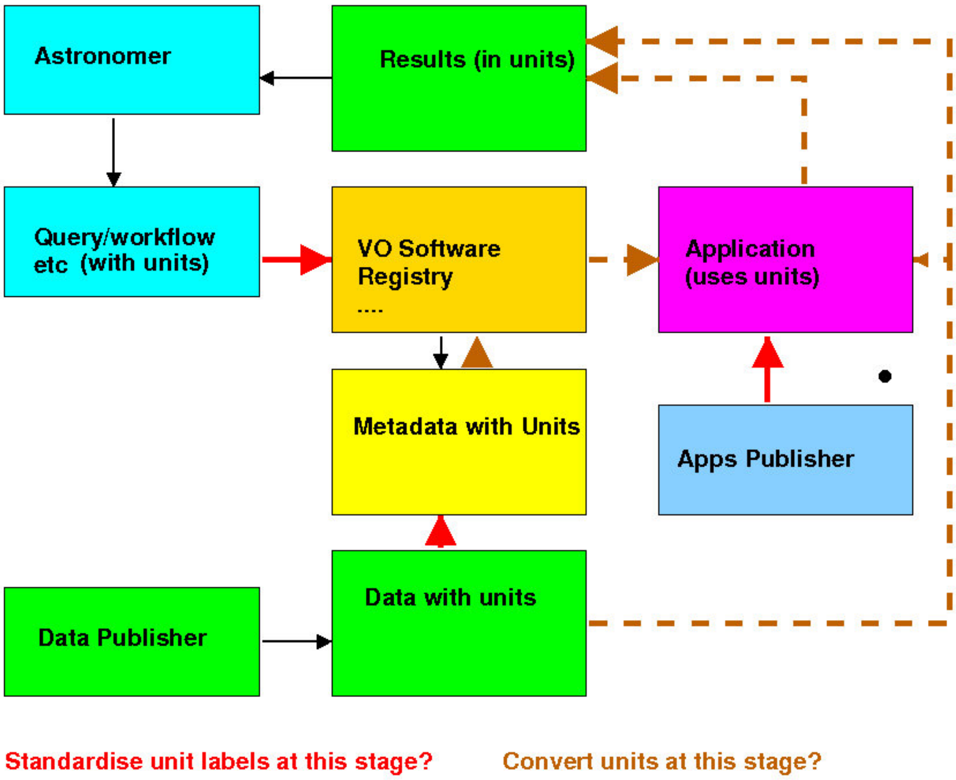

Figure 2: This shows the levels at which conversions might be done. Thick red arrows: At the point where an astronomer or data provider submits input to the VO, we should provide tools to ensure that units are labeled consistently according to VOUnits. This implies that a units parsing step is included prior to metadata ingestion into the VO. Dashed brown arrows: Conversions required to supply results to the user in specified or reasonable units e.g. `J.s-1` to `W`, are done where and when they are required.