

UCD discussion from SSIG

B. Cecconi, S. Erard, P. Le Sidaner
Observatoire de Paris

Previous proposal (1)

Plasma environment modeling

- magnetic potential vector.
`phys.magfield;phys.potential` OR `phys.magfield.potentialvector`
- electric current density (total current density of all charged particles: electrons, protons, ions...)
`phys.flux;phys.atmol.ionstage` is not satisfactory.

Illumination conditions

- **Note:** only `pos.phaseAngle` available.
- **New identified needs**
 - Incidence angle. Same as “solar zenithal angle”
`pos.incidenceAng`
 - Emergence angle:
`pos.emergenceAng`
 - Azimuth angle:
`pos.azimuthAng`

Previous proposal (2)

Coordinates and ephemeris

- **Coordinates**

- planetary magnetospheric coordinates use colatitude and not latitude (spherical coordinates).

- `pos.bodyrc.colat` (or `pos.az.zd?`)

- **Orbital Parameter**

- perifocal distance:

- `pos.distance;src.orbital.perifocal`

- **Generic coordinate systems**

- current coordinate systems in “pos.” UCDs are: AZ, BodyRC, Cartesian, Earth, Ecliptic, EQ, Galactic.

- Adding generic cylindrical system would be useful:

- `pos.cylindrical.r` / `pos.cylindrical.th` / `pos.cylindrical.z`

- or `pos.distance` (`pos.radius?`) / `pos.az.azi` / `pos.distance` (`pos.height?`)

- **Vector or matrix components**

- Adding a way to say “this a component of a vector or a matrix, and not the full set of information”:

- `phys.component`

- **Rotation parameter description**

- necessary for describing attitude and orientation parameters

- `pos.rotation.eulerAng`

- `pos.rotation.quaternion`

- `pos.rotation.matrix`

- `pos.rotation.axis`

- **More info here** : <https://voparis-confluence.obspm.fr/display/VES/VESPA+Contribution+to+NASA-JPL+WebGeoCalc+tool>

Previous proposal (3)

EPN TAP keywords

- **Spatial Resolution**

- We need spatial resolution (spatial sampling: in situ or projected on target) and angular resolution

- `pos.resolution`

- `pos.angResolution`

- **Heliospheric coordinates**

- There is a heliocentric related UCD, but it is a generic reference frame qualifier. Adding heliocentric longitude coordinates would be useful.

- `pos.heliocentric.lon`

- while there, let's add also heliocentric latitude.

- `pos.heliocentric.lat`

Metadata

- **New identified needs**

- checksums: MD5 hash

- `meta.cryptic;meta.file (?)`

- `meta.checksum;meta.file`

- modification date

- `time.processing;meta.file`

- `time.update;meta.file`

to be compared with creation date

- `time.creation;meta.file`

and release date

- `time.release;meta.file`

Updated proposal

Spectroscopic and photometric measurements

- **Note**

In Spectrum DM, spectral dependencies are given in UCD. For instance:

- Flux Density per unit wave: `phys.flux.density;em.wl`

- Surface Brightness per unit frequency: `phys.flux.density.sb;em.freq`

Is this something that we want to keep for the future?

Tools are using UCDs to detect spectral quantities. This works now because it is simple. More complex description needed => tools broken ?

- **New identified needs**

Detailed description:

<https://voparis-confluence.obspm.fr/display/VES/Spectral+quantities+in+use+for+Planetary+Science>

Many spectral data related to observations of the Solar System are not calibrated / distributed in flux, but as different quantities adapted to resolved sources:

- radiance : power from the source, per unit area (intrinsic quantity)
- reflectance: generalization of “albedo”, i.e., measured radiance relative to incoming power. Several variations exist depending on context.
- The radiance of a source can be derived from a measured flux but has to be calibrated (depends if the field of view is filled, and on observer-source distance), therefore no simple conversion can be performed on the fly.

Data.FluxAxis.Value / name	Data.FluxAxis.ucd	Data.FluxAxis.unit	Possible equivalent UCD
Radiance (=I)	phot.radiance (exists in UCD 1.3, but see below)	$W * m^{(-2)} * sr^{(-1)} * \mu^{(-1)}$ (and variations)	phys.luminosity;phys.angArea;em.wl (heavy...)
“Reflectance“ = Radiance factor (RADF) = πr = “I/F ratio” (I/F may include thermally emitted light)	phys.reflectance? (would not cover emitted light) phys.I_over_F? (would be more general)	dimensionless	phys.albedo;em.wl (but ambiguous, and only when emission is negligible). Equal to “normal albedo” at $i=e=0^\circ$ (and phase = 0°) only arith.ratio;phot.flux.density - not even sure because this is the ratio of a radiance to a flux (in sr^{-1})
Derived quantities			
Bidirectional reflectance = $r = I/\pi F$	phys.bidir?	$sr^{(-1)}$	
bidirectional reflectance distribution function (BRDF) = r / μ_0	phys.brdf?	$sr^{(-1)}$	(mostly used in terrestrial remote sensing)
Reflectance factor (REFF) = Radiance coefficient $\pi r/\mu_0$	phys.reff?	dimensionless	(used with laboratory samples)
Normalized reflectance	arith.ratio;phys.reflectance ? does not seem adapted M4ast currently uses phys.reflectance;em.wl;arith.ratio	dimensionless	Normalized to 1 at some wavelength
+ thermal / emission quantities	Same as I/F for spectral measurements		I/F also apply to observations in the thermal range. What about lab measurements?
+ Hemispherical quantities			TBC

Table 1: Quantities in use (μ_0 stands for cosine of incidence angle; acronyms in parenthesis are from Hapke 1981 and are commonly used)

New proposals

- **Spectral Matrix (or Jones Matrix)**: matrix of auto- and cross-correlations between colocated antenna with different polarizations (similar to “auto”-visibilities = visibility with null base-line)
Could be either :
 - a child of `phys.polarization`, as it is a raw measure of the polarization
 - a child of `instr` as it tells what type of instrument/mode set up is used.This UCD would be used to advertise clients that a data product contains this specific type of data (used in `obscore:o_ucd` or `epn_core:measurement_type`) in order to select the tools to send the product to.
- UCD for **gravitational field** "Power Spectrum of Spherical Harmonic Coefficients of Lunar Gravity Model":
 - `phys.gravitation`
- **Shape model (full 3D shape) or Terrain Model (or Elevation model)** with respect to reference geoid or ellipsoid:
 - `phys.shape`
 - `phys.shape.elevation`
- **Spatial Resolution**: The `pos.resolution` UCD has been deprecated at some point and replaced by `pos.AngResolution`. We need it back for, e.g., cartesian axes.

Status and update of UCD

- What is the status of the UCD update discussed last year?
- What is the result of the tests done with provided examples?
- We tried a system based on RT (Request Tracker) for managing new UCD proposals, but it did not work. Other options ?