# UTYPES

Jonathan McDowell

# Data Model fields

- A data model consists of a bunch of information (attributes) and a structure organizing them (classes)
- If we ignore the structure, we get a long checklist of all the information data providers need or might need to describe their data
- Each piece of information and each grouping (class) is called a data model field, and is given a name which we call a utype:
  - Example: Spectrum.Target.Name

# Utypes and VOTABLE

- VOTABLE is an XML document with a fixed header+table schema; to infer a data model structure we must impose it with extra clues
- The UTYPE attribute was introduced to do this
  - <PARAMETER name=”foo” utype=”Spectrum.Target.Name” value=”3C 273”/>
- UTYPE then became useful outside the VOTABLE context.

# UTYPE syntax

- Basic syntax is tokens separated by periods as **token.token.token**
- Leftmost is the highest level containing class, rightmost is the actual item in question
- Implies a “has-a” hierarchy: Spectrum.Target has a field whose 'simple utype' is Name and whose 'fully qualified utype' is Spectrum.Target.Name
- Case-insensitive, but recommend CamelCase style

# UTYPE uniqueness

- Utypes are unique within a **model** but not within an **instance**
- In VOTABLE can use GROUP for local scope

# Utypes and XML

- In an XML Schema based document instance, the information is represented in two ways:
  - The contents of elements or attributes
    - <answer>42</answer>
    - <distance unit=”furlongs”>100</distance>
  - The names of the elements themselves
    - <SpatialAxis>  versus <Axis type=”Spatial”>

# Utypes and XML -2

- Each data model field corresponds to an element in the schema (with a few exceptions)
- It makes sense to just use the field names (utypes) to build the XSD element names
- Limitations of XSD and XML, and implementation issues, result in some deviations of the XSD from the abstract model
  - Example: forcing automatic checking in STC

# UTYPES and FITS

- Introduce new TUTYPn keyword convention
  - TTYPE4 = 'WAVE_ERR'
  - TUTYP4 = 'Spectrum.Data.SpectralAxis.Accuracy.StatErr'
- Allows you to use arbitrary column names and map them to the data model
- No equivalent for header keywords, these must be hardcoded for now

# Namespaces and Scope

- If I see a utype 'Target.Name' what is Target?
  - Which Data Model defines Target?
  - What version of that data model?
- In XML the natural thing is to define a namespace
  - xmlns:foo=”http://ivoa.net/xml/SpectrumModel/spec1.01.xsd”

  and use a namespace prefix
  - utype=”foo:Target.Name”

# Namespaces contd.

- Another option is to have one IVOA namespace and declare certain public classes Spectrum, Char, Target etc.: all utypes must begin with one of this list. (A. Micol: utype searching should be simple with no parsing)

# Scope in VOTABLE

- In VOTABLE, I argued that it was natural to use GROUP to provide local scope and let utypes inherit context from the parent group:
  - <GROUP utype=”Target”>
  - <PARAMETER utype=”Name”>
  - </GROUP>
- Some opposition to this. Currently using fully qualified utypes in VOTABLE

# Combining DMs

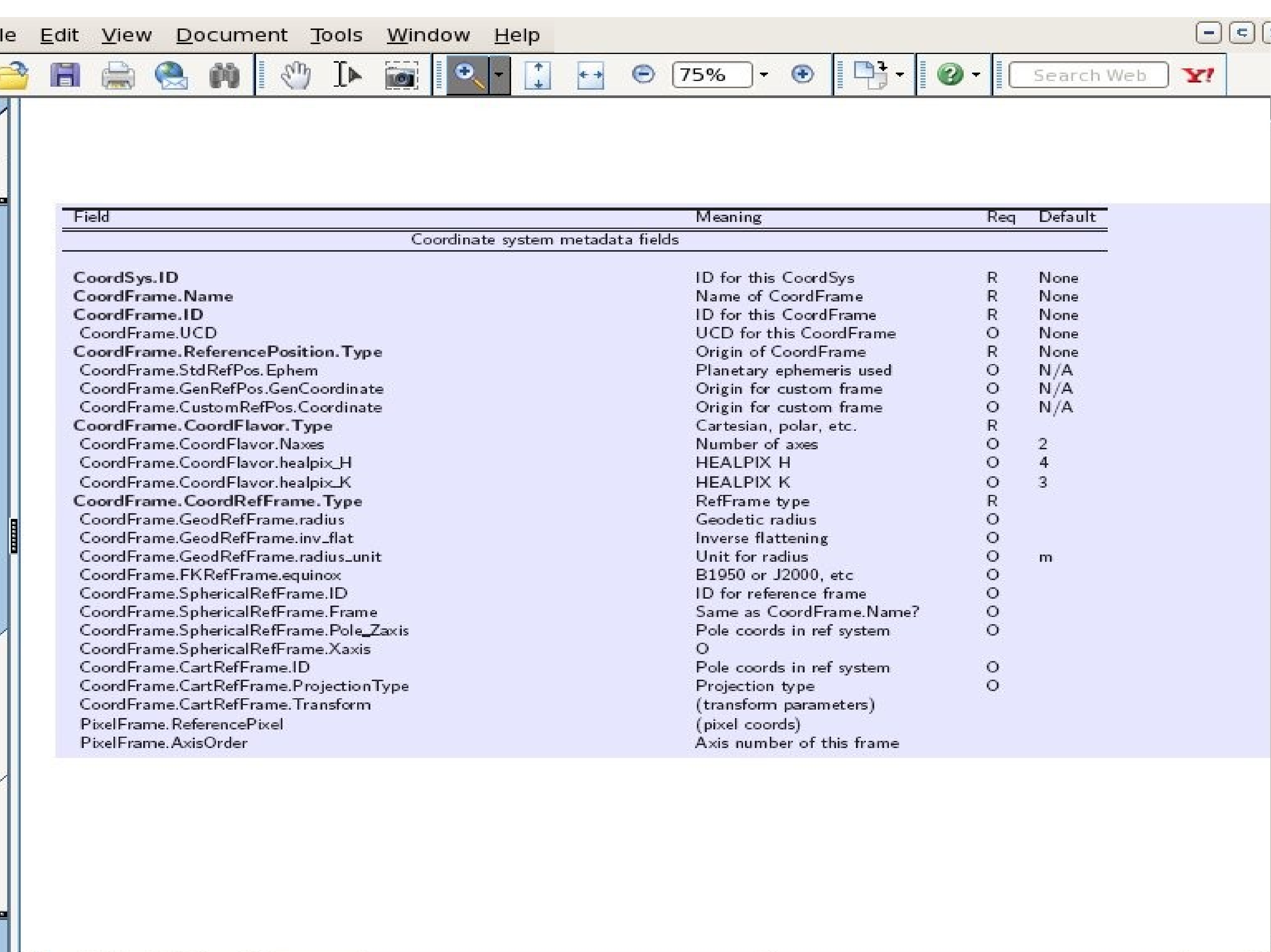
- In defining one DM we may want to refer to another. Example in Char:
  - Char.CharacterizationAxis.Coverage.Location
- Is the utype of an object of type STCCoordinate. The unit of the coord within this might have a utype
  - Coordinate.CValue.Unit
- How do we express the fact that this particular coord is in the Char DM with a specific meaning?

# Combining DMs-2

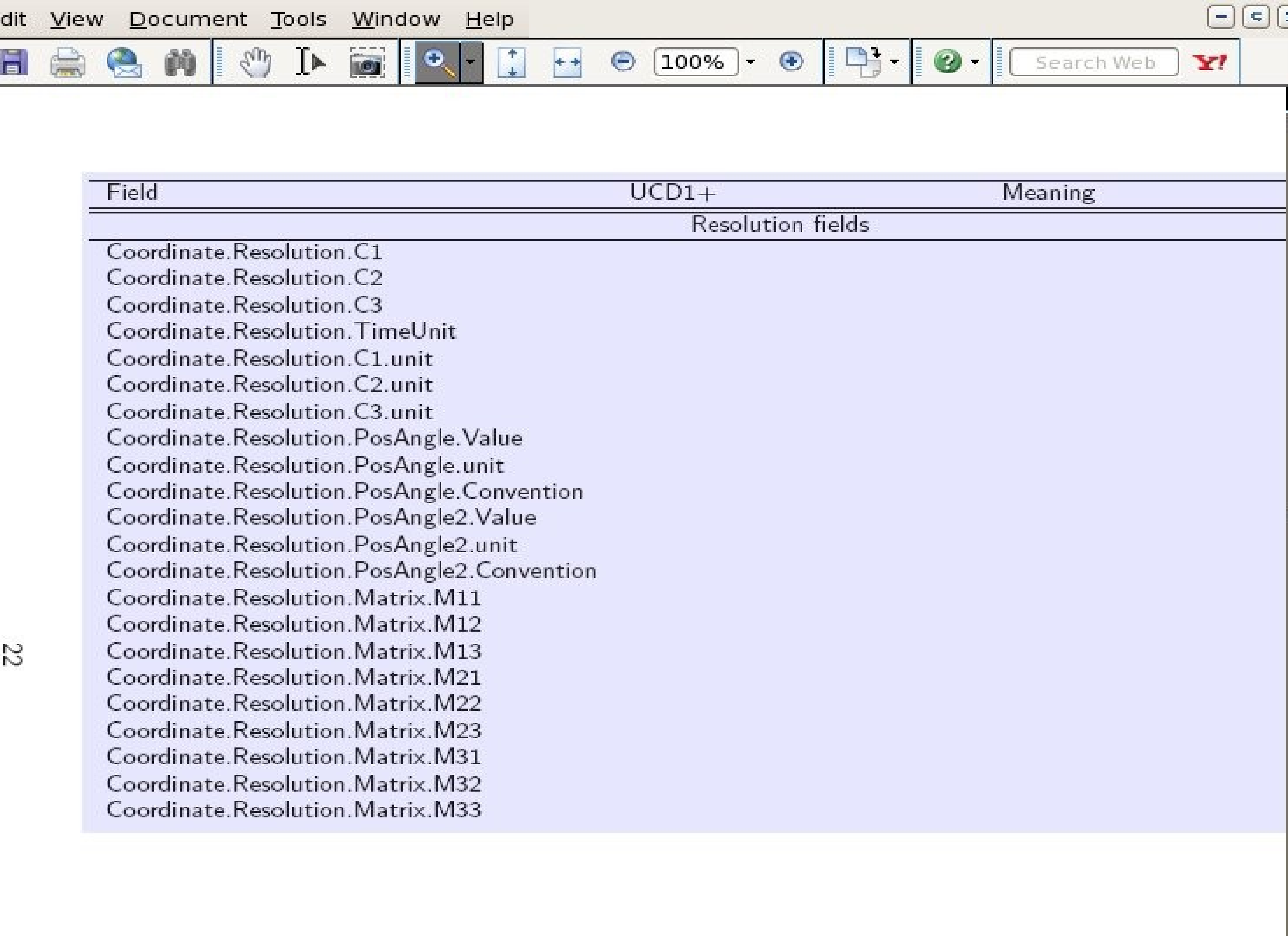
- We could say that since Location is of type Coordinate, it inherits Coordinate's utype substructure:
  - Char.CharacterizationAxis.Coverage.Location.CValue.Unit
- We could take the UCD approach and treat the higher level utype as an adjective:
  - Coordinate.CValue.Unit;Char.CharacterizationAxis.Coverage.Location
- Any other suggestions?

# Querying Utypes

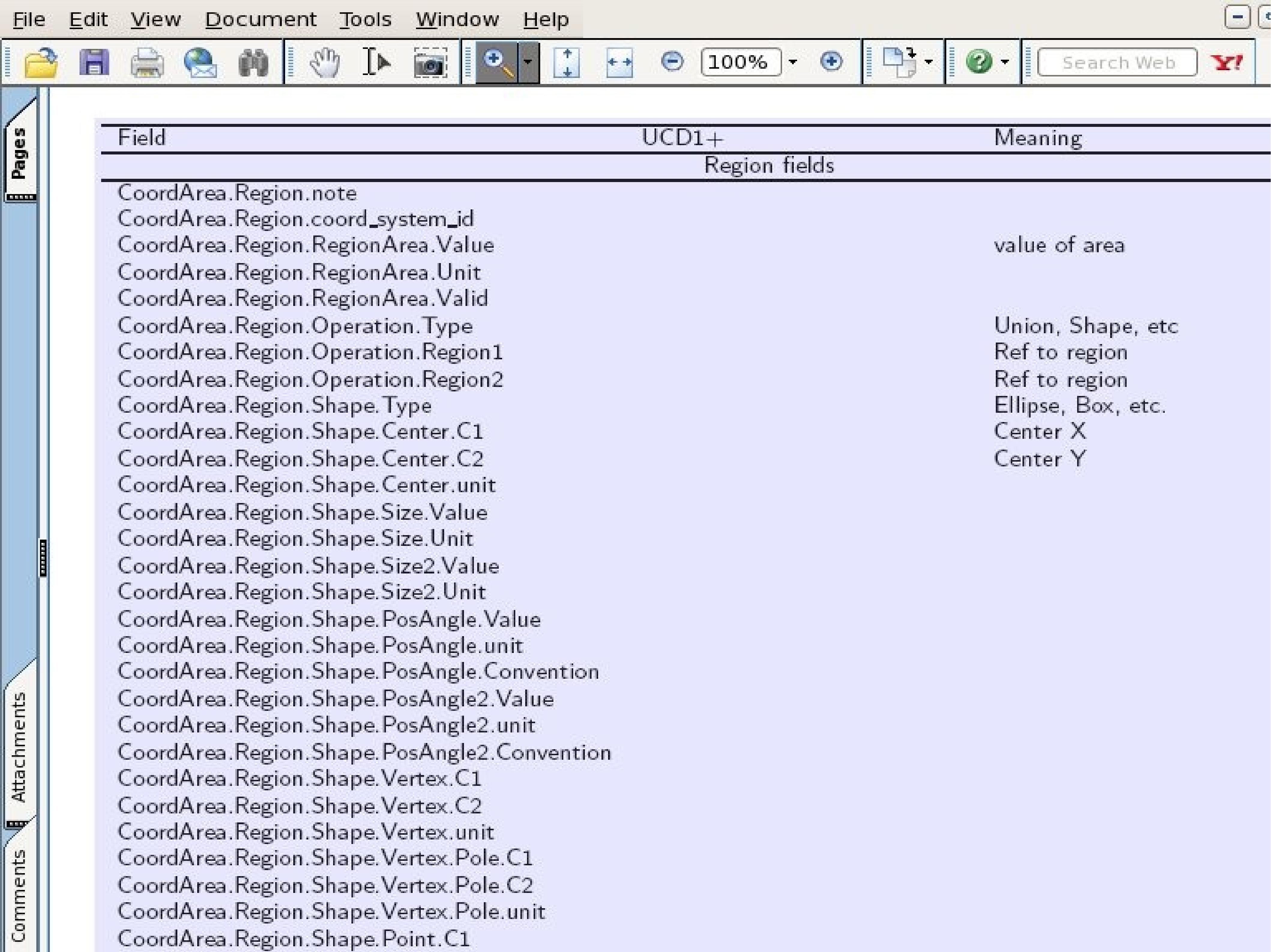
- Francois Bonnarel has argued for a query language form in which (similar to XPATH?) one could distinguish multiple instances of the same utype within a document by specifying the values of attributes of the relevant element or containing elements, e.g.
  - utype=token1.token2[attribute1=val1].token3.token4



| Field | Meaning | Req | Default |
|---|---|---|---|
| Coordinate system metadata fields | | | |
| **CoordSys.ID** | ID for this CoordSys | R | None |
| **CoordFrame.Name** | Name of CoordFrame | R | None |
| **CoordFrame.ID** | ID for this CoordFrame | R | None |
| CoordFrame.UCD | UCD for this CoordFrame | O | None |
| **CoordFrame.ReferencePosition.Type** | Origin of CoordFrame | R | None |
| CoordFrame.StdRefPos.Ephem | Planetary ephemeris used | O | N/A |
| CoordFrame.GenRefPos.GenCoordinate | Origin for custom frame | O | N/A |
| CoordFrame.CustomRefPos.Coordinate | Origin for custom frame | O | N/A |
| **CoordFrame.CoordFlavor.Type** | Cartesian, polar, etc. | R | |
| CoordFrame.CoordFlavor.Naxes | Number of axes | O | 2 |
| CoordFrame.CoordFlavor.healpix_H | HEALPIX H | O | 4 |
| CoordFrame.CoordFlavor.healpix_K | HEALPIX K | O | 3 |
| **CoordFrame.CoordRefFrame.Type** | RefFrame type | R | |
| CoordFrame.GeodRefFrame.radius | Geodetic radius | O | |
| CoordFrame.GeodRefFrame.inv_flat | Inverse flattening | O | |
| CoordFrame.GeodRefFrame.radius_unit | Unit for radius | O | m |
| CoordFrame.FKRefFrame.equinox | B1950 or J2000, etc | O | |
| CoordFrame.SphericalRefFrame.ID | ID for reference frame | O | |
| CoordFrame.SphericalRefFrame.Frame | Same as CoordFrame.Name? | O | |
| CoordFrame.SphericalRefFrame.Pole_Zaxis | Pole coords in ref system | O | |
| CoordFrame.SphericalRefFrame.Xaxis | O | | |
| CoordFrame.CartRefFrame.ID | Pole coords in ref system | O | |
| CoordFrame.CartRefFrame.ProjectionType | Projection type | O | |
| CoordFrame.CartRefFrame.Transform | (transform parameters) | | |
| PixelFrame.ReferencePixel | (pixel coords) | | |
| PixelFrame.AxisOrder | Axis number of this frame | | |



| Field | UCD1+ | Meaning |
|---|---|---|
| | Resolution fields | |
| Coordinate.Resolution.C1 | | |
| Coordinate.Resolution.C2 | | |
| Coordinate.Resolution.C3 | | |
| Coordinate.Resolution.TimeUnit | | |
| Coordinate.Resolution.C1.unit | | |
| Coordinate.Resolution.C2.unit | | |
| Coordinate.Resolution.C3.unit | | |
| Coordinate.Resolution.PosAngle.Value | | |
| Coordinate.Resolution.PosAngle.unit | | |
| Coordinate.Resolution.PosAngle.Convention | | |
| Coordinate.Resolution.PosAngle2.Value | | |
| Coordinate.Resolution.PosAngle2.unit | | |
| Coordinate.Resolution.PosAngle2.Convention | | |
| Coordinate.Resolution.Matrix.M11 | | |
| Coordinate.Resolution.Matrix.M12 | | |
| Coordinate.Resolution.Matrix.M13 | | |
| Coordinate.Resolution.Matrix.M21 | | |
| Coordinate.Resolution.Matrix.M22 | | |
| Coordinate.Resolution.Matrix.M23 | | |
| Coordinate.Resolution.Matrix.M31 | | |
| Coordinate.Resolution.Matrix.M32 | | |
| Coordinate.Resolution.Matrix.M33 | | |



| Field | UCD1+ | Meaning |
|---|---|---|
| Region fields | | |
| CoordArea.Region.note | | |
| CoordArea.Region.coord_system_id | | |
| CoordArea.Region.RegionArea.Value | | value of area |
| CoordArea.Region.RegionArea.Unit | | |
| CoordArea.Region.RegionArea.Valid | | |
| CoordArea.Region.Operation.Type | | Union, Shape, etc |
| CoordArea.Region.Operation.Region1 | | Ref to region |
| CoordArea.Region.Operation.Region2 | | Ref to region |
| CoordArea.Region.Shape.Type | | Ellipse, Box, etc. |
| CoordArea.Region.Shape.Center.C1 | | Center X |
| CoordArea.Region.Shape.Center.C2 | | Center Y |
| CoordArea.Region.Shape.Center.unit | | |
| CoordArea.Region.Shape.Size.Value | | |
| CoordArea.Region.Shape.Size.Unit | | |
| CoordArea.Region.Shape.Size2.Value | | |
| CoordArea.Region.Shape.Size2.Unit | | |
| CoordArea.Region.Shape.PosAngle.Value | | |
| CoordArea.Region.Shape.PosAngle.unit | | |
| CoordArea.Region.Shape.PosAngle.Convention | | |
| CoordArea.Region.Shape.PosAngle2.Value | | |
| CoordArea.Region.Shape.PosAngle2.unit | | |
| CoordArea.Region.Shape.PosAngle2.Convention | | |
| CoordArea.Region.Shape.Vertex.C1 | | |
| CoordArea.Region.Shape.Vertex.C2 | | |
| CoordArea.Region.Shape.Vertex.unit | | |
| CoordArea.Region.Shape.Vertex.Pole.C1 | | |
| CoordArea.Region.Shape.Vertex.Pole.C2 | | |
| CoordArea.Region.Shape.Vertex.Pole.unit | | |
| CoordArea.Region.Shape.Point.C1 | | |