NRC·CNRC

DALI: Next

Patrick Dowler
Canadian Astronomy Data Centre

Data Access Layer - ~~Wed May 15~~ Thurs May 16

DALI-1.2

- **scope: define new xtype(s)** for
  - identifying values in services (e.g. columns of a TAP service)
  - serialising values (e.g. in VOTable)
- use cases:
  - varying coordinate frame
  - polymorphic shapes
  - disjoint shapes
  - all-sky
  - registry resource coverage
- needed by:
  - ObsCore, ADQL, RegTAP, (retro: SIA, SODA)

- details: https://wiki.ivoa.net/twiki/bin/view/IVOA/DALI-1_1-Next

DALI-1.2: MOC in the Registry

- registry use cases:
  - describe coverage of data collections, surveys, etc

- proposed: **xtype=”moc” datatype=”char” arraysize=”*”**

  - future-proof: smoc? tmoc, stmoc?

  - ascii serialisation format specified in MOC-1.1
  - of course: usage not limited to RegTAP

  - TBD: ADQL-2.x MOC({string}) function ?
  - TBD: ADQL INTERSECTS({moc}, {moc}) ?
  - TBD: other ADQL functions ?

DALI-1.2: Varying Coordinate Frame

- EPNtap use case:
  - frame varies from row to row
  - non-normative STC-S in TAP-1.0 allowed frame value within geometry constructs
  - mixing metadata with values considered harmful: subsequent geometry usage rejected that (DALI, SIA, SODA)
  - ADQL deprecating coordsys argument to geometry
    - users advised to use empty string
    - TAP services advised to ignore it

DALI-1.2: Varying Coordinate Frame

- proposed solution: coordinates in a frame is a DM thing
  - use one column for values
  - if frame is constant: metadata about the column
  - if frame varying: second column
  - VO-DML mapping markup to connect the frame(s) + value(s)

DALI-1.2: Use Cases

- polymorphism
  - spatial coverage as polygon or point depending on available metadata (sometimes position only) [ESO]
  - spatial coverages that are circle or polygon due to instrument characteristics [CADC, ESAC, MAST, others]
  - SIA & SODA: POS param allows circle, range, polygon
- disjoint shapes
  - mosaic camera data may have significant gaps between sampled areas [CADC, ESO, others]
  - non-contiguously sampled regions that are built up over time [CADC]
- all-sky:
  - DALI polygon cannot specify an all-sky region

DALI-1.2: Proposed Solutions

- consider ~4-5 new constructs → 1-5 new xtype(s)

	simple	**complex: disjoint**
polymorphic	SIA & SODA parameters: **shape**	normative replacement for STC-S (finally) **region**
typed	DALI-1.1 xtypes: **point** **circle** **polygon**	CAOM **multi-polygon** **multi-interval**

DALI-1.2: Obvious Solutions

- polymorphism: **xtype=”shape” datatype=”char” arraysize=”*”**
  - describe parameter values (e.g. SIA, SODA)
  - stored (e.g. TAP) and serialised (VOTable) values
- serialisation: **<type label> <simple DALI value>**
  - label are not case sensitive
  - POINT {long} {lat}
  - circle {long} {lat} {radius}
  - rAnGe {long} {long} {lat} {lat}
  - PoLyGoN {long} {lat} {long} {lat} {long} {lat} …
  - consistent with SIA & SODA POS parameter syntax
  - just defining an xtype for the current values

DALI-1.2: More Questions than Solutions

- disjoint shapes: requires a “UNION” operator
  - minimum requirement: union of polygons
- holes in shapes: requires an “INTERSECT” operator
  - use cases exist -- not compelling enough at this time
  - probably should consider it when designing solution(s)

- operator style vs function style
  - **<value> UNION <value> …**
  - **UNION <value> <value> …**

  - operator style lets you append
  - function style identifies as complex up front
  - TBD: typed multi-polygon requires separator (operator style?)
  - TBD: do we really want/need to add ( )?
  - TBD: operators with different shapes?

DALI-1.2: Going forward...

- serialisation principles
  - should be trivial to up-convert from non-polymorphic to polymorphic values (by adding the type)
  - simple values should be valid complex values

- really need a whiteboard to sketch this out, but:
  - simple: typed → polymorphic (new xtype=”shape”)
  - simple → complex favours using operators to append values

Splinter meeting with whiteboard to work out details?

DALI-1.2: Debatable Solution

	simple	**complex: disjoint**
polymorphic	shape: point {point} | circle {circle} | range {range} | polygon {polygon}	region: polygon {polygon} [U {polygon} …] | circle {circle} [U {circle} …]? | point {point} [U {point} …]?? {shape} U {shape} X {region} {op} {region} XX
typed	interval: {lower} {upper} point: {long} {lat} circle: {long} {lat} {radius} polygon: {long} {lat} {long} {lat} {long} {lat} … range: {long} {long} {lat} {lat}?	{polygon} [U {polygon} …] {interval} [U {interval} …] {circle} [U {circle} …]? {point} [U {point} …]??