VOQL session 1

ADQL

Unit

- Unit is exposed by "column" interface or "columns" table, so basically it is possible to do the unit conversion at the client side.
- Use same syntax as specified in VOTable WD.
  - http://vizier.u-strasbg.fr/doc/catstd-3.2.htx
- Is it required to implement unit conversion for all the units described in the above URL ?
  - CDS has software to do the conversion.
- Support at least the conversion among "deg", "arcmin", "arcsec", "radian"

Date/Time/Timestamp/TimeZone

- Many variety for ISO 8601 (STC) expressions
- standard SQL just covers only a part of ISO8601
- ADQL supports the yellow colored expression ?
- Recommend a client app to support all for user input

- 2006-05-16 (o)
- 2006-05 (x)
- 2006 (x)
- 2006-001 (x)
- 2004-W13-4 (x)

- 10:30:50.012 (o)
- 10:30:50 (o)
- 10:30 (x)
- 10:30.5 (x)
- 10 (x)
- 10.5 (x)

- +09:00 (o)
- Z (x)
- +09 (x)
- +0900 (x)

- 2006-05-16 10:30:50.012 (o)
- 2006-05-16T10:30:50.012 (x)

XMatch Function

- `XMATCH_CHI2(‘t1 t2 !t3’,3.5)`

-
  ```
  XMATCH_DISTANCE(t1.ra, t1.dec,
  t2.ra, t2.dec, 1.0 [arcsec])
  AND XMATCH_DISTANCE(t1.ra, t1.dec,
  t3.ra, t3.dec, 1.0 [arcsec])
  ```

Region Syntax

```
Region(‘<shape> [<frame>] <ra> <dec> <size>’)
<shape> ::= BOX | CIRCLE
<frame> ::= FK4, FK5, ICRS, Gala, what else ?
<ra> ::= <numeric literal>
<dec> ::= <numeric literal>
<size> ::= <numeric literal> [ <unit> ]
<unit> ::= deg | arcmin | arcsec
```

- “Sexagecimal” is not allowed in ADQL-x
- Recommend a client app to support “Sexagecimal” for user input (ADQL-s)
- Supported frames and units should be exposed by metadata interface

Region Syntax (XML)

- According to the comment from Arnold Rots (2006-05-09), XML representation of a region is:

```
<Condition xsi:type=“stc:STCRegion”>
  <stc:AstroCoordSystem xlink:type="simple"
      xlink:href=“ivo://STClib/CoordSys#UTC-FK5-TOPO”
      id="UTC-FK5-TOPO"/>
  <stc:Circle coord_system_id="UTC-FK5-TOPO">
    <stc:Center unit="deg">
      <stc:C1>333.817116</stc:C1>
      <stc:C2>55.832959</stc:C2>
    </stc:Center>
    <stc:Radius>2.5</stc:Radius>
  </stc:Circle>
</Condition>
```

Default Coordinate Frame and Unit

- When coordinate frame and/or unit are omitted, what default setting should be used ?
  - Service specific frame and unit
  - They should be exposed as metadata.
- It is natural to do a region search on the coordinate frame specific to the table.
  - Simulation data : if we define default frame and unit to e.g. “FK5” and “deg”, it will not be applicable to the simulation data.

Table Alias, Qualified Column

- Table alias name is mandatory
- All the column name must be qualified by table alias name not by table name.
- Recommend that client app allow a user to omit the table alias and column qualifier in a trivial case (single table query), and that the client app gives a default table alias name when submiting ADQL-x.

```
SELECT   ra, dec
FROM     qso
WHERE    Region('Circle 210 30 1.0')
```

Delimited Identifier

- According to the SQL standard, double quotations are used to specify the delimited identifier.
- Current ADQL uses "[" and "]" (dialect of SQLServer)
- Why not use the SQL standard
- This was discussed at the previous IVOA meeting, and there was no claim to use the standard SQL.

Select Into

- "Select into" is a dialect of SQLServer
- "Create table as (<select statement>)" is defined as a SQL standard (SQL99 ?)
- Why not use the SQL standard ?

ADQL schema is split into two schemas

- ADQL-Core and ADQL-Full
- ADQL-Core schema conforms to the ADQL core specification
- ADQL-Core schema is aimed to be used for interoperability, update cycle will be longer than ADQL-Full (>10 years ?).
- ADQL-Full schema is aimed to be used for implementing advanced query functionality. Update cycle should be as long as possible (>5 year).

ADQL schema update

- Verbose ComplexTypes are replaced by one ComplexType → simplified
- Added SQL syntax (natural join, join using, subquery, exists, any, all) → higher functionality
- 56 complexType, 9 simpleType (ADQL 1.0)
- 53 complexType, 3 simpleType (ADQL 1.041)
- 35 complexType, 0 simpleType (ADQL Core)
- 1.0 is translatable to 1.041 without loss of information. Core is translatable to 1.041 w/o loss of information

ADQL schema update

- removed type definitions

  binaryOperatorType, unaryOperatorType, atomType, stringType, trigonometricFunctionType, trigonometricFunctionNameType, mathFunctionType, mathFunctionNameType, aggregateFunctionNameType, comparisonType, archiveTableType, xMatchTableAliasType, includeTableType, dropTableType, xMatchType, notLikePredType, exclusiveSearchType, notBetweenPredType, inverseSearchType, userDefinedFunctionType, ArrayOfFrmoTableType

ADQL schema update (cont.)

- Added complex type:

  xpathReferenceType, nonNumericType, subqueryTableType, joinConditionType crossJoin, onJoin, naturalJoin, usingJoin, booleanValueFunctionType, existsPredType, anyPredType, allPredType

ADQL schema update (cont.)

- + selectionLimitType: offset attribute is added
- + fromType: maxOccurs of Table element is changed from "unbounded" to "1"
- + searchType: “not” attribute is added
- + columnReferenceType: CaseSensitive attribute is added, xpathName attribute is removed as xpathReference is introduced.
- + functionType: abstract="true" is removed, Allow element is removed, number of appearance of an Args element changed to “unlimited”, Name attribute is added.
- + aggregateFunctionType: changed to extend scalarExpressionType, Name attribute is added, Allow and Arg elements is added.
- + numberType: unit attribute is added.
- + integerType: type of value attribute is changed from xs:long to xs:integer.
- + tableType: attributes “ShortName”, “Identifier” and “CaseSensitive” are added, “xpathName” is removed
- + joinTableType: “LeftTable” and “RightTable” are added, Qualifier, Tables elements are removed
- + joinTableQualifierType: "_OUTER" suffix is removed, "CROSS" is removed.
- + likePredType: type of Pattern element is changed to nonNumericType.
- + regionSearchType: ???

ADQL schema update (cont.)

- binaryOperatorType:
  - enumeration of strings "+", "-", "*", "/"
  - Removed to allow for service specific operators.
  - The operators that should be supported are described in another document (ADQL WD or note ?)
  - unaryOperatorType ("+", "-") and comparisonType ("=", "<", ">" …) are also removed for the same reason.
- atomType:
  - just a wrapper of literalType, unit is defined here.
  - Removed for verbosity
  - Unit is defined at NumericType
  - stringType is renamed as nunNumericType to be used for non-numeric type such as timestamp, boolean, spaceCoords, spaceRegion, and a service specific type.

ADQL schema update (cont.)

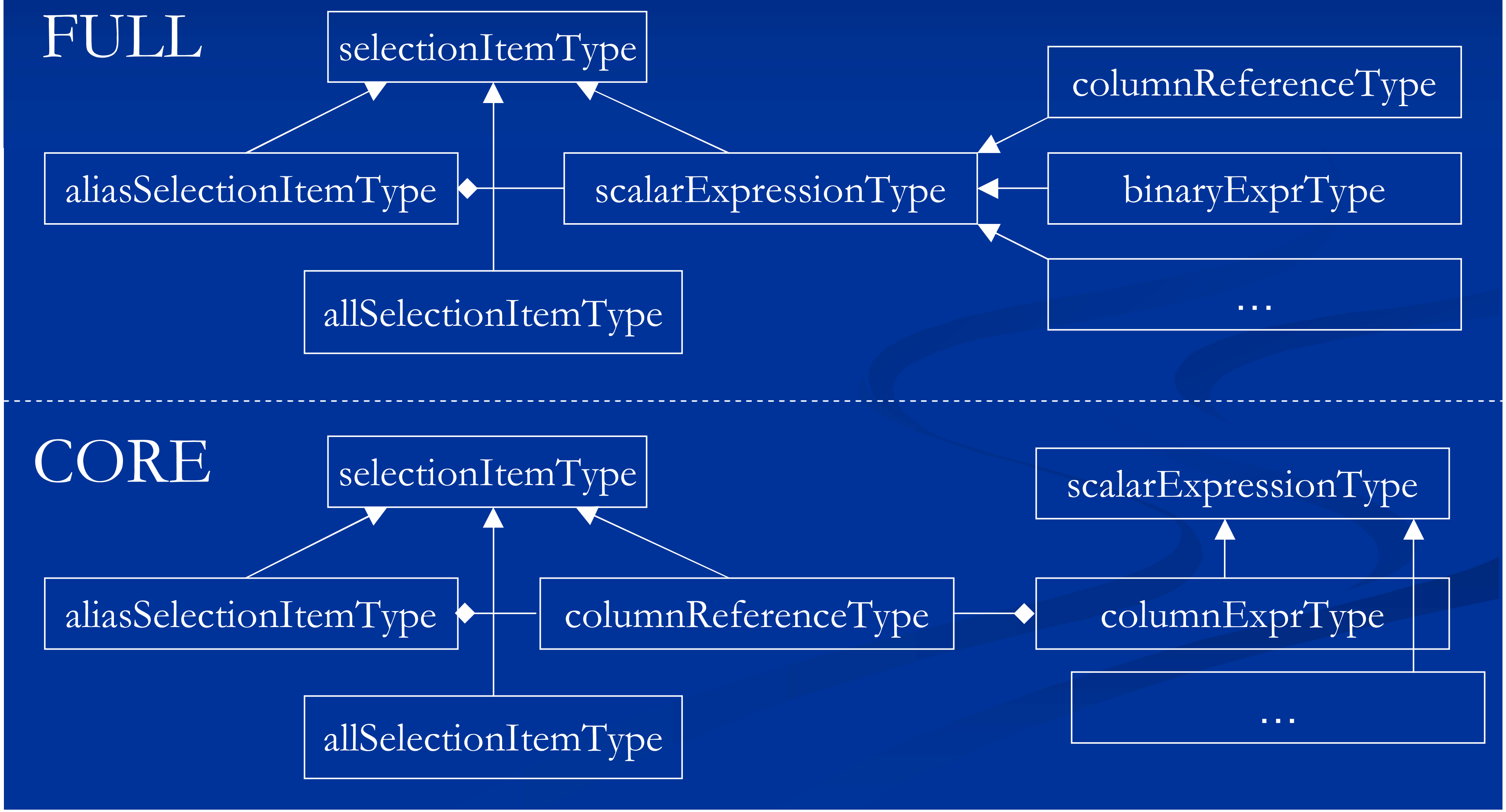
- FunctionType familiy
  - trigonometricFunctionType, mathFunctionType, userDefinedFunctionType are unified to a single FunctionType.
- ArchiveTableType
  - Identifier attribute is added TableType, so this is obsoleted.
- XMatchType family
  - xMatchType, xMatchTableAliasType and so on are removed
  - Xmatch is expressed by a FunctionType wrpped by booleanValueFunctionType

ADQL schema update (cont.)

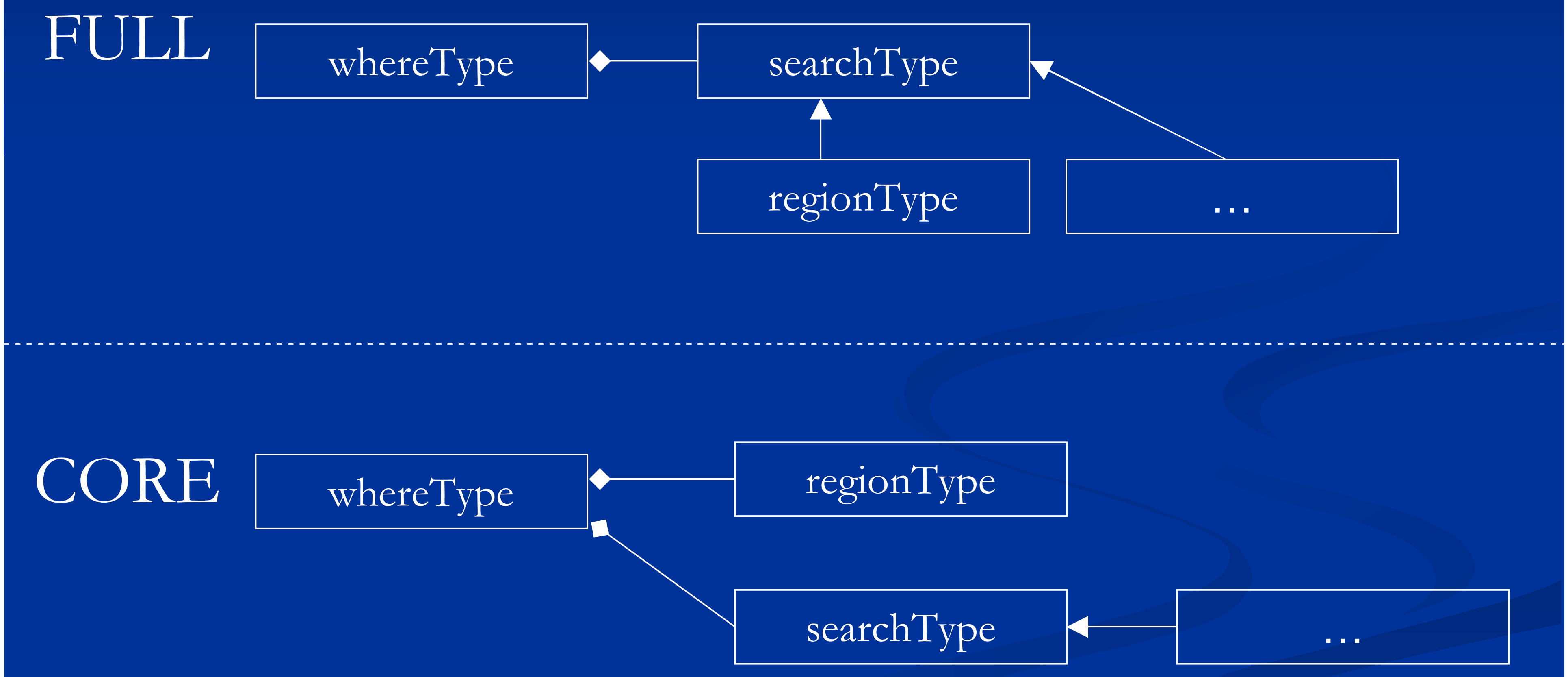
- NOT family
  - notLikePredType, exclusiveSearchType, notBetweenPredType, inverseSearchType are removed
  - Not attribute is added to the searchType
- EXISTS, ANY, ALL
  - EXISTS (subquery)
  - Column = ANY (subquery)
  - Column = ALL (subquery)

ADQL Core schema

- Structure is slightly different from ADQL full schema.



ADQL Core schema (cont.)



FULL

whereType

searchType

regionType

…

CORE

whereType

regionType

searchType

…

Only one region can be specified.

ADQL Core schema (cont.)

- TableJoin family is removed
- Only one TableType is specified.