Summary of ADQL & SkyNode specification review meeting

Victoria, May 16th, 2006

Attendees:

Iñaki Ortiz, Aurelien Stebe, Pedro Osuna, Yuji Shirasaki, Paul Harris, Guy Rixon, Francois Ochsenbein, Peter Quinn, Andy Lawrence, Roy Williams, Christophe Arviset, Francoise Genova, Kevin Benson, Jonathan McDowell, Pat Dowler, Gerard Lenmson, Arnold Rots, Maria Nieto-Santisteban.

(Apologies if I missed any names or misspelled them ☺)

Meeting goal:

Understand what (ADQL Core, ADQL Extended) and (Basic SkyNode/ Full SkyNode) are in order to decide what ADQL Core should include and what is the minimum that VO services (data providers) need to implement.

Discussed:

1. **Should REGION (BOX and CIRCLE) be part of ADQL Core?**

- The REGION constraint is meaningless for services without spatial information.
- Not having REGION allows queries to be very close to pure SQL.
- REGION is simple enough that is easy to implement.
- REGION is important enough that should be included in the Core.
- It is up to the client to formulate sensible queries. If a table doesn't contain spatial information then makes no sense to add a REGION constraint.

2. **Should columns be added to the REGION constraint expression?**

- Catalogs may include more than two columns with positional information considering for example: myCatalog(ra1, dec1, ... ra2, dec2)

```
SELECT t.* FROM myCatalog t
WHERE POSITION(t.ra1, t.dec1) IN REGION('CIRCLE J2000 180 0 1')

SELECT t.* FROM myCatalog t
WHERE POSITION(t.ra2, t.dec2) IN REGION('CIRCLE J2000 180 0 1')
```

Although the idea was considered interesting, it was decided not to include it for now. The data provider chooses what set of columns is used in a REGION query. If data providers want to allow REGION queries on different set of columns, they can create views as:

```
myCatalog1(ra1, dec1, .... )
myCatalog2(ra2, dec2, .... )
```

```
SELECT t.* FROM myCatalog1 t
WHERE REGION('CIRCLE J2000 180 0 1')

SELECT t.* FROM myCatalog2 t
WHERE REGION('CIRCLE J2000 180 0 1')
```

3. **Should be XMATCH renamed?**

- XMATCH has a rich astronomical meaning. Astronomers think of XMATCH not as a positional match only but rather as the process of identifying that two or more objects correspond to one.

4. **Should be the XMATCH syntax part of ADQL? Is XMATCH a user defined function?**

- Big debate about this. I'm not sure we reached any clear final conclusion.
- Maria's opinion:
  - If XMATCH is part of the grammar, then the protocol defines the function signature and behavior. (Function signature = definition of input and output)
  - If XMATCH is a user defined function, then the service (user) defines the function signature and behavior.
  - In order to perform XMATCH between more than two nodes, all nodes need to agree in what XMATCH means in terms of signature and behavior. At that point, it really doesn't matter whether the function definition is native to the language or a user defined function.

5. **Why do we make the distinction between Basic/Full SkyNode?**

- Pedro is in favor of not making distinctions.
- Maria's opinion:
  - Classifying SkyNodes is the way clients can easily know from the registry whether or not the node is capable of handling execution plans and performing XMATCH. There is no other difference.

<u>Concluded:</u>

1. **REGION stays in ADQL Core as it is.**
2. **Using the word XMATCH to refer to positional match only, may not be the right word to use.**
3. **People feel uneasy about Chi2XMatch.**
4. **People left the room having a better understanding of what ADQL Core/Extended and Basic/Full SkyNodes are.**