Data Dictionary

MLSListings Experience

By Olga Ermolin

Background

- MLSListings - 16,000+ subscribers over 5 home counties in SF Bay Area
- In-house built RETS server 1.5, 1.7,1.7.2, 1.8
- We operate & support close to 200 RETS data feeds – brokers and vendors
- Ran parallel backward compatible servers for many years to transition.

Data Share

- 2008-2010 – Data Share and Reciprocal access project with 6 neighboring MLSs.

- Defined NEW common schema, added hundreds of fields and lookups (Data Dictionary would have been very helpful at this point !)

- Extended Areas, Schools, Districts, Cities – became resources

New Listing Management

- This year we are rolling out new Listing Management system compatible with aggregated data set

- Opportunity to change schema and input forms

- Adopted Data Dictionary “way” where we were doing things differently (Media Resource, Unit Fields for residential Income)

Data Dictionary 1.3 core fields

- Majority of fields just require Standard Names update

- Exceptions-few
  - – Garage YN vs. lookup value – will add DD field
  - – Resources vs. Lookups for geography lists

Data Dictionary 1.3 - Exceptions

- Exceptions - several patterns
  - One field to many
    - Kitchen vs. Appliances and Room Type DD
  - Many to one (can’t use duplicate standard name)
  - Different types
    - Boolean vs. lookup value
    - Coded ranges (building height 5-10) vs. integer
  - Income properties – Unit Descriptions – we adopted Data Dictionary

Data Dictionary 1.3 - Implementation

- Possible solutions and considerations
  - Change original schema to comply with DD
    - Full Compatibility with Data Dictionary
    - Very disruptive: impose constant change on clients as DD matures or wait until DD is fully defined. No backward compatibility.
  - Define missing or mismatching fields in the metadata, keep original fields for backward compatibility.
    - One metadata, one server, flexibility
    - Redundancy in Metadata, could be confusing
  - Define brand new Data Dictionary schema, ETL data, run two servers during transition
    - Full Compatibility. No redundancy, adopting such server should be easy. Backward compatibility
    - Load on us to maintain two sets of metadata, two servers, and ETL

Lessons Learned

- Change is painful, but necessary

- Backward compatibility is essential
  - Without it adoption might be impacted

- Participation in RESO workgroups – important : useful ideas + some of our exceptions were worked in into the standard + contribution