Business Rules Taxonomy and a Standardized Syntax
(Listing Maintenance focus)

April 2016

Matt Cohen
Clareity Consulting
clareity.com
# The 2010 Business Case Effort Revived Late 2015 by R&D WG

## RESO R & D Business Case Worksheet

<table>
<thead>
<tr>
<th><strong>Title:</strong></th>
<th>MLS Business Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Submitted By:</strong></td>
<td>Matt Cohen - <a href="mailto:matt.cohen@callclareity.com">matt.cohen@callclareity.com</a></td>
</tr>
<tr>
<td><strong>Date:</strong></td>
<td>11/15/2010</td>
</tr>
<tr>
<td><strong>Summary:</strong></td>
<td>Enhance the RETS validation functions to encompass all MLS business rules. Outreach to MLSs to encourage business rule documentation. Vendors may need to improve capabilities to facilitate this.</td>
</tr>
<tr>
<td><strong>Stakeholders:</strong></td>
<td>MLS Vendors, MLS Executives, MLS Staff</td>
</tr>
<tr>
<td><strong>Value Proposition:</strong></td>
<td>MLSs with well documented business rules can move to a new MLS system, add additional front ends, or integrate other software that requires use of business rules - without manual work and inaccurate results. This will result in smoother conversions, more software choice, and enhanced competition and innovation.</td>
</tr>
<tr>
<td><strong>Detailed Synopsis:</strong></td>
<td>To facilitate the update transaction, field level</td>
</tr>
</tbody>
</table>
It’s six years later and:

- MLSs are still manually describing BR to new vendors.
- Some MLS vendors (and individual MLSs) have developed nice BR engines. Some can export BR - but typically the export is incomplete even for listing maintenance – and very incomplete in the grand scheme of all MLS BR!

None of the existing MLS BR Engines – let alone BR consuming platforms - “talk the same language” and can exchange BR data. Hence the need for RESO standards!
MLS Business Rule Breakdown:

1. Back-end
   a. Listing Maintenance (validation)
   b. Transformation in MLS / Repository

2. Front-end display and uses

The focus has been on #1 for now. The process:

- **FULLY UNDERSTAND** entire scope and complexities, via creation of “taxonomy” – a business rule breakdown.
- Created initial taxonomy documentation (12-2015)
- Collected feedback via reso.org (1-2/2016)
- Interviews with some MLS vendors (2-3/2016)
- Obtain many MLSs’ BR for analysis (2-3/2016)
- Refinement of initial taxonomy (3/2016)
- **CONSIDER** Existing Standards (2-3/2016)
- **Syntax Drafting** - Collab w/ stakeholders (3-4/2016)
- Engage R&D sub-group (when formed)
Existing BR Standards

- No need to entirely re-invent the wheel
- MANY BR standards were looked at for consideration and inspiration, including **but not limited to**:
  - Avro
  - Schematron
  - Easyb
  - Xpath
  - BRML (deprecated)
  - RuleML
  - Business Rule Language Definition Framework (BRLDF)
  - Business Action Language (BAL)
  - XPDL
  - Node-rules
  - JSON Rules Language
  - **RuleSpeak**
  - Semantics of Business Vocabulary & Business Rules (SBVR)
• **Selection Criteria:**
  • Understandable by non-technical stakeholders
  • Software language and platform agnostic
  • Easy to learn and use
  • Detailed enough to be unambiguous – can be translated to an easily interpretable form … and software code.

• **Results:**
  • The current proposal uses [SBVR](http://omg.org) compliant RuleSpeak® ([http://www.rulespeak.com](http://www.rulespeak.com)) to create Structured English expressions. RuleSpeak is longstanding, well-adopted set of guidelines for expressing business rules in concise, business-friendly fashion. It’s a set of best practices. **The proposal focus is creating example syntaxes** for MLS BRs.
  • A set of shorthand syntax expressions of the Structured English is also proposed.
Think Clearly About BR

It’s *important* to think clearly about business rules in order to document them correctly!

The following tips are from “Basic RuleSpeak Guidelines” by Business Rule Solutions, LLC:

**Business Rules must not be inscrutable!**

- *Sample from the restaurant menu:* “These items may or may not be available at all times, and sometimes not at all and other times all the time.”
Rules should be “declarative”, not procedural language

NO: "If the news is bad, then shoot the messenger."
YES: “The messenger of bad news must be shot.”
Still, not great, as business rules go!
Does the messenger need to be shot dead?
Does the messenger actually need to be shot, or is just killing him sufficient?

- Pick a clear subject, then use a rule key word (must, only) as fast as possible. Put the heavy lifting (“if” condition) toward the end of the sentence. IT people / programmers usually hate this but it makes things much more readable to business people.
Finally: enforcement and evaluation are separate concerns

A Business Rule should express only the ‘what’, not anything about:

- **How** it is to be evaluated or enforced.
- **Where** it is to be evaluated or enforced.
- **Who** is responsible for evaluating or enforcing it.
- **When** (what events cause) it to be enforced.
- **Why** it needs to be evaluated or enforced.
40 rule types in the taxonomy converted to 12 (with variations):

1. **ALLOW_EDIT** – Whether a user can alter data field(s)
   - Syntax 1: Single Field OR Single Listing,
   - Syntax 2: Group of fields OR Group of Listings

2. **REQUIRE_VALUE** - Whether a field must have a value
   - Syntax 1: Single field
   - Syntax 2: Group of fields

3. **REQUIRE_EMPTY** - Whether a field is required to have no value
   - Syntax 1: Single field
   - Syntax 2: Group of fields

4. **DERIVE_VALUE** - How a data field value is derived
   - auto-population from another source
   - computed [i.e. Days On Market], or
   - system generated [i.e. database key]
5. **CONSTRAIN_VALUE** - A field constraint that must be met by …
   - 1: User Enters Data - Specific value, Value Offset
   - 2: User Enters Data – Value Range
   - 3: User Enters Data – Inclusion/Exclusion List or Pattern
   - 4: User Enters Data - Prohibited values
   - 5: *User Chooses from List* - Inclusion/Exclusion List or Pattern

6. **CONSTRAIN_DATE_TIME** - A field constraint that must be met by a specific date/time constraint, with or without offset.
   - Syntax 1: Specific date, or Date Offset, Syntax 2: Date Range

**MOST MLS BUSINESS RULES ARE ADDRESSED BY THE ABOVE SIX RULE TYPES AND THEIR VARIATIONS, THEN...**
The Six Additional Rule Types

7. MEDIA
- Ensures media can be accepted – file types and size, pixels for images.

8. ALLOW_CROSSLIST – Cross-listing rules
- Cross Listing - Multiple listing types for same property

9. CLONE_LISTING
- Clone field values and media from existing listing into new listing

10. DELETE_LISTING
- Manual listing deletion

11. PREVENT_DUPLICATE_LISTING
- Prevent duplicate listing

Finally, a rule about what happens in a repository; not input:

12. AUTO_UPDATE
- Time triggered listing status change
ENFORCEMENT

\[\text{ENFORCEMENT} = \{\text{STRICT} \mid \text{WARNING} \mid \text{OVERRIDE}\}\]

// Strict: “This rule is strictly enforced”
// Warning: “Rule may be overridden with a warning”
// Override: “Rule can be overridden with no warning”

[\text{ALWAYS} \mid \text{IF} \ <\text{condition}> \ \text{ENDIF}]\]

.. 

// The enforcement statement may be repeated.

For example:

– **ENFORCEMENT** = WARNING IF UserRole INLIST (“MLS STAFF”, “SUPERUSER”) ENDIF
– **ENFORCEMENT** = STRICT IF UserRole INLIST (“AGENT”, “BROKER”) ENDIF
RULE SCOPE

“Applies to listings of transaction type X and Property Type Y”

- **Listing Transaction Type:** ALL, Sale, Lease, etc.
- **Listing Property Type:** ALL, Residential, Commercial, etc.
- **Listing Property SubType:** ALL, SFR, Mobile, Condo, etc.

SCOPE (ListingTxnType=ALL,
ListingPropertyType=INLIST (Residential, Lease),
ListingPropertySubType=ALL)

TRIGGERING

“This rule is triggered when the listing is [saved].”

- **TRIGGER** (OnSave | OnInput)
Structured English:
Field “Must Accept User Input”
Listing “Must Accept User Input”

Interpretable Form:
ALLOW_EDIT
{ FIELD <aFieldName> | LISTING <aListing> }  
[ YES | NO  ]
[  ALWAYS | IF <condition> ENDIF ]

PublicRemarks must accept user input.
ALLOW_EDIT PublicRemarks

ListingId must not accept user input.
ALLOW_EDIT ListingId NO

CloseDate must accept user input only if MlsStatus is Sold.
ALLOW_EDIT CloseDate YES IF MlsStatus="Sold"
Field Group level example:

Each of the following fields must accept user input if CommissionYN is “Yes”: BuyerAgencyCompensation, SubAgencyCompensation, and TransactionBrokerCompensation.

ALLOW_EDIT FIELDGROUP CommissionFieldGroup
IF CommissionYN = “Y” ENDF

DEFINITION
ENDDEFINITION
Structured English:
Field must have a value.

Example: Field GarageSpaces must have a value if GarageYN has a value of “Y”.

Interpretable Form:
REQUIRE_VALUE
FIELD <aFieldName>
[ ALWAYS | IF <condition> ENDIF ]
[ DEFAULT (<aValue>) ]

Example: REQUIRE_VALUE FIELD GarageSpaces
REQUIRED IF GarageYN = "Y" ENDIF
Structured English:
Field, out of this group of fields, must have a value.

Interpretable Form:
REQUIRE_VALUE
FIELDGROUP <aFieldGroup>
[ ALWAYS | IF <condition> ENDIF ]
[ MINCOUNT [ 1 | <aInteger> ] ]
[ MAXCOUNT [ MAX | <aInteger> ] ]

DEFINITION { <aFieldGroup>} ENDDEFINITION
Structured English:
Exactly 1 data field in the following list must have a value if CommissionYN has the value "Yes":
BuyerAgencyCompensation
SubAgencyCompensation
TransactionBrokerCompensation

Interpretable Form:
REQUIRE_VALUE
FIELDGROUP CommissionFieldGroup
IF CommissionYN = “Y” ENDIF
MINCOUNT 1 MAXCOUNT 1

DEFINITION CommissionFieldGroup =
INLIST (BuyerAgencyCompensation, SubAgencyCompensation, TransactionBrokerCompensation)
ENDDEFINITION
Other Rule Examples
ReserveListPrice must have no value if SpecialListingConditions is not equal to “Auction”

```
REQUIRE_EMPTY FIELD ReserveListPrice
IF SpecialListingConditions NOTEQUAL "Auction" ENDIF
```

Autopopulate the value for ParcelNumber. CoreLogic is the data source If State is “NY”. LPS is the data source if State is “NJ”. CRS is the data source if State is “PA”.

```
DERIVE_VALUE ParcelNumber
ALWAYS
AUTOPOP CoreLogic IF State="NY" ENDIF
AUTOPOP LPS IF State="NJ" ENDIF
AUTOPOP CRS IF State="PA" ENDIF
```
MlsStatus of an Active Listing of Residential Property Type may only change to one of the following: "Active", "Cancelled", "Extended", "Under Agreement", "Temporarily Withdrawn". MLS Staff or SuperUser may override this restriction.

**SCOPE** (ListingPropertyType="Residential")

**CONSTRRAIN_VALUE** MlsStatus SUMLIST INLIST("Active","Cancelled","Extended","Under Agreement","Temporarily Withdrawn") IF MLSStatus="Active" ENDIF

**ENFORCEMENT** = STRICT IF INLIST ("AGENT", "BROKER") ENDIF

**ENFORCEMENT** = OVERRIDE IF INLIST ("MLS STAFF", "SUPERUSER") ENDIF
A property may be cross-listed only in Property Types Residential and Farm.

ALLOW_CROSSTLIST
XLISTFIELDNAMES (PropertyType) YES IF
XLISTFIELDVALUES INLIST (RES, Farm) ENDIF

ContractStatusChangeDate must be a date on or after ListingContractDate.

CONSTRAIN_DATE_TIME ContractStatusChangeDate
ONORAFTER ListingContractDate
Listing Status must be set to Expired on or after Expiration Date if Current Listing Status is not in (Expired, Pending, Sold, Leased).

```
AUTO_UPDATE LISTINGSTATUS "Expired" WHEN Date GE ExpirationDate ENDWHEN IF ListingStatus NOTINLIST ("Expired", "Pending", "Sold", "Leased") ENDIF
```

Many of the preceding rules seem simple (only 12 simple forms!) but there is much advanced syntax possible and the rules combine to be very powerful and cover a wide array of MLS business rules!
Next Steps:

1. Form Business Rules sub-group in R&D workgroup.
2. Evaluate and iterate on proposed syntax.
3. Package BR as payloads, reference from transport.
4. As vendors adopt this for MLSs, it will enable increased innovation and competition – and make system implementations much better for MLSs.

Thank you!