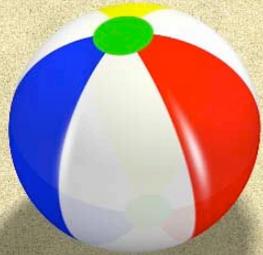# RSL: A Meta-Case Study in Theory Formulation and Use

Jim Cordy
Queen's University
CSER, Toronto, Oct 2006

Conjecture

* Theory == DSL
  (Domain Specific Language)

* Case study:
  * Propose theory of table recognition
  * Design DSL to encode theory
 * Encode methods to compare in DSL
  * Validate, apply, compare

Problem:

* Understand & compare table recognition methods
  * Only documentation is source code for tools themselves
  * Imperative code in various languages (mostly C)
  * Unreadable
* Only comparison by results on examples, no benchmarks

Theory I: Data

* Directed graph with attributes represent both input and recognition results *(interpretations)*
  * Nodes represent physical *regions*
  * Edges represent *relations* on regions
* Recognition steps locally transform graph
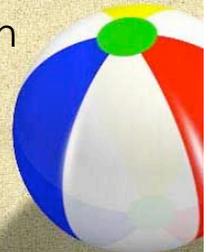
Theory II: Operations

* Region *creation*, *classification*
* *Segment, merge, re-segment* regions
* *Relate* regions (e.g. horizontal adjacency)
* *Reject* classifications, relation elements
* *Accept* and *reject* interpretations
* Guards for conditional application

DSL for Theory: RSL

```
strategy main
  adapt aResolution using
      getScanResolution()
    observing
        {Image} regions

  classify {Word} regions as {Cell}

  relate {Cell} regions with {adjacent_right} using
      defineRightAdjacency(sMaxRowSeparation,
      aResolution)

  segment {Cell} regions into {Row} using
      mergeRowsFromCells()
    observing
        {adjacent_right} relations

  accept interpretations
end strategy
```
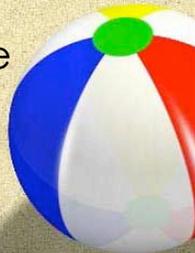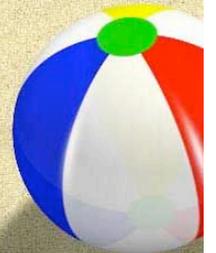
Applying RSL

* Table structure recognition methods dominated by Handley, Hu
* No one knows why they work well, or how they relate
* Encode each in RSL to expose method according to theory
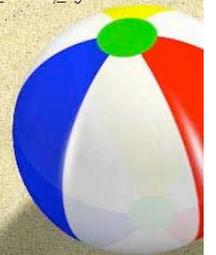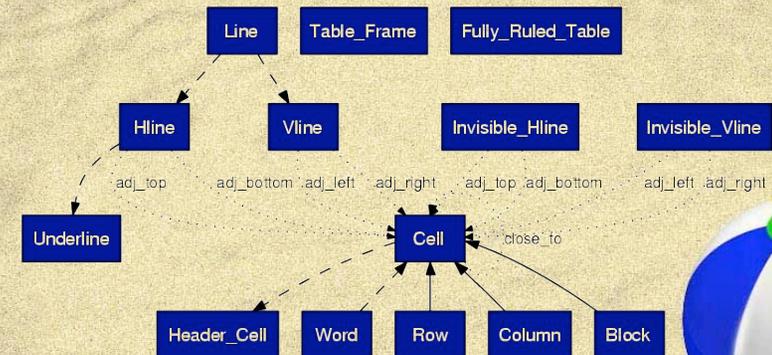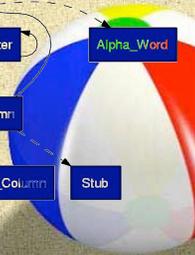
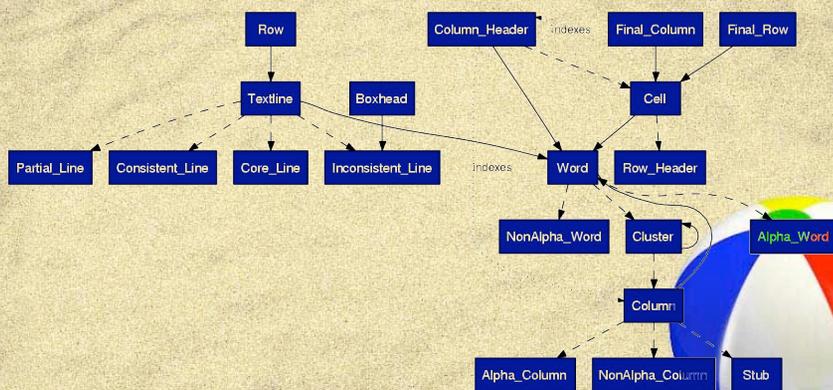(Insert two years of huge work here)

Applying RSL

* Validate by running RSL encodings vs original programs
* 9 pages (Handley), 3 pages (Hu)

* So what do we do with it?
  * Map back to theory
  * Compare methods in detail using RSL models

RSL to Theory (Handley)

RSL to Theory (Hu)

Comparison: Handley vs Hu

* Run RSL interpretations of both over a large set of same example inputs
* Store history of decision tree (model according to theory)
* Match and compare historical decision trees to understand process similarities and differences

Moral

* DSL's (which we understand) can encode theories (which Steve says we don't)
* Provides a bridge to a theory, exposing method models
* Enables detailed empirical comparison of incomparable methods

Downside

* It's a lot of work
  - (doesn't surprise scientists)
  - (but scares software engineers)
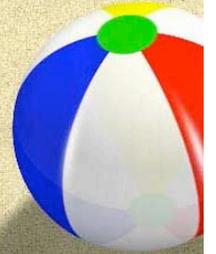
Acknowledgments

* This is PhD work of Richard Zanibbi co-supervised with Dorothea Blostein
* References:

R. Zanibbi, A Language for Specifying and Comparing Table Recognition Strategies. *PhD thesis, Queen's University*, 2004.

R. Zanibbi, D. Blostein and J.R. Cordy, "The Recognition Strategy Language", *Proc. ICDAR 2005 - IAPR 8th International Conference on Document Analysis and Recognition,* Seoul, Korea, August 2005, pp. 565-569.

R. Zanibbi, D. Blostein and J.R. Cordy, "Historical Recall and Precision: Summarizing Generated Hypotheses", *Proc. ICDAR 2005 - IAPR 8th International Conference on Document Analysis and Recognition,* Seoul, Korea, August 2005, pp. 202-206.