Proposed Database Design for *OmniLingua*

Prof. Dennis Shasha
Dept of Computer Science
Courant Institute, NYU

Design Goals

- Want to include
  language properties
  examples
- Want system to allow the addition of new properties, but with some discipline (avoid the reinvention of terms).

Database Design 101

- Most database schemas (table designs) are fairly rigid, e.g. employees have names, departments, managers, and salaries.
  Emp(name, dept, manager, salary)
- Changes are changes within the framework of the schema, e.g. add a new employee record or change a salary.

When Flexibility Is Needed

- There already exists an application area where properties must be added in a flexible way: ecommerce.
- Imagine a website that sells stuff. For pants you have an inseam. Now if you sell wristwatches, you have to add wrist circumference. If you sell wetsuits, you need to add thickness.

Ecommerce Approach “attributes as data”

- Generic(object, attribute, value)
  e.g. generic(‘pants’, ‘inseamlength’, 32’’)
  generic(‘watch’, ‘wristcircumference’, 5’’)
- That is, attributes become data. If we did this with the employee example, we might have:
  generic(empid_53, name, ‘joe’)
  generic(empid_53, salary, 60,000)

OmniLingua Schema
first cut

- language(languageid, property, value)
  example(sentenceid, languageid, property, value)
  property(propertyname, description)
- Here are some records:
  language(‘english’, ‘order’, ‘verb-object’)
  example(3, ‘nigeria1’, ‘text’, ‘me kp0 Kofi’)
  example(3, ‘nigeria1’, ‘gloss’, ‘I see.past Kofi’)

OmniLingua Schema extensions

- Generic fields: e.g. author, date, annotations, display instructions, time
- Special features for hierarchies.
  X is_a_type_of Y
  X is_a_part_of Y

OmniLingua Query Examples

- select sentenceid, languageid from example where property = ‘voice’ and value = ‘passive’
- Find language pairs that both have passive voice examples. (SQL is a little more complex but not too bad.)

Why a Relational Database?

- Gary Simons has pointed out that RDF could be an alternative implementation platform as it also offers an attribute-as-data approach.
- I agree but with a relational database, we get: recovery from failure, concurrency control, and a query language.

Example Design

- My suggestion is to agree on a core set of example sentences for purposes of discovering minimal pairs.
- This could be based on a questionnaire to explore many features of the languages.

Property Freedom

- With this model, it is possible to define new properties.
- How do we prevent chaos?
  Tools – if a researcher defines a new property, we would ask for a description and see whether we could suggest ‘similar’ properties. Researcher would be asked to distinguish new from old.
  Training – Examples + properties

How Much Effort

- Creating the database schema and a core set of queries is not a lot of work (e.g. a month once the community has suggested the queries of interest).
- Creating a user interface is likely to be much longer, but is mostly a policy issue of how you want to interact with the system.

How Much Effort II

- First implementation is three months using a current tool like ruby on rails on initial data.
- Get feedback from users to see how to improve, e.g. new queries, new sets of examples.
- Hosting on single machines around the world serving as mirror sites.
- All eminently doable.