

# COBOL & Topic Maps?

- COBOL had support for data definitions and manipulation in its core
- Object-oriented languages help to model “things” but they are not optimized to model knowledge about “things”
  - *time sensitive values, multiple sources of information, provenance, security*
- Other issues: persistence, remote editing, notifications, offline support, dynamic/schema based GUI
- Lisp, Prolog, Python, Ruby etc. allow building domain specific languages

# Ruby example

```
class Knows < Association
  acts_as_historical
  acts_as_symmetrical

  role :person, :player_type => :person,
        :as_property => :knows,
        :negation => :does_not_know,
        :card_min => 2,
        :card_max => 2
end

class Person < Topic
  name :first_name, :scope => {:historical => :now, :multilingual => "en"}
  name :last_name, :scope => {:historical => :now, :multilingual => "en"}
  occurrence :date_of_birth
  property :knows, :card_min => 0, :card_max => :unbound
end

a=Person.create(:psi=>"JohnSmith")
a.first_name="John"
a.last_name="Smith"

b=Person.create(:psi=>"BobStone", :first_name => "Bob", :last_name => "Stone")

a.knows! b, :at_date => "2007-10-01"
a.save
b.save
```

# New subject-centric programming language

- TMQL, TMCL and CTM provide a good basis for a new subject-centric programming language
  - *data, time, multiple sources, provenance, security in its core*