Ruby on Semantic Web

Vadim Eisenberg and Yaron Kanza
Department of Computer Science, Technion, Israel

Introduction

The Impedance Mismatch problem between RDF (Semantic Web) and OO, is similar to the Impedance Mismatch between the Relational Model and OO.

- **OO**: Objects
  - have identity
  - have methods and attributes
  - each one is a member of a single (most specific) class
  - cannot change their class

- **RDF**: Individuals
  - have (global) URIs
  - have properties (no methods)
  - each one can be a member of multiple classes
  - can change their classes dynamically

The Solution: Ruby on Semantic Web

- Extension of the Ruby Programming Language
- Hybrid model between OO and RDF:
  - RDF individuals and OO objects are first-class citizens of the language
  - Much more than RDF-to-OO mapping!
- Persistent Programming Language
- Domain Specific Language (DSL)

The Language Principles

- URIs and Access Transparency
- Persistence Independence
- Orthogonal Persistence
- Location, Migration and Data Model Transparency
- Language-Integrated Queries
- Logical Inference
- Procedural Attachment

URIs and Access Transparency

- Processing Data in multiple data sources
- Data associated by "DNS" - mapping from URIs to data sources
- Data sources are transparent to the programmer

SPARQL Queries and Reasoning Integrated in the Code

- Integrating SPARQL* queries with programming language constructs, variables, and arithmetic operations
- Using inferred facts from the ontology of the enterprise

Persistence Independence

- Attaching RDF properties to native objects
- Running SPARQL queries on native objects
- Attaching methods to RDF individuals

The Architecture

- All the data in an enterprise mapped to RDF via adapters
- An association between data sources and URIs is created ("DNS")
- Reasoners are plugged in to provide Logical Inference
- Ruby on Semantic Web is built on top of Active RDF: [http://activerdf.org/](http://activerdf.org/)

*Using q::var instead of ?var syntax of SPARQL