Semantic Data Aggregation through Semiotics
Facilitating querying and inferencing

Sebastián Samaruga
http://xama.dev.java.net
xama@dev.java.net

Abstract

Trying to fill the gap between real business (intelligence) domain applications and semantics through extensive data aggregation and a functional approach to knowledge representation through semiotics.

Introduction

Given Semiotics, and Semantics, which is a branch of Semiotics, regarding Peirce, along with Syntax / Grammar, and Pragmatics, the relationship arises that given three entities regarded as: Sign – Concept – Object, considering (Sowa[1]) "A sign has three aspects: it is (1) an *entity* that represents (2) another *entity* to (3) an *agent"* that our underlying model can be composed of three classes, namely:

- Type
- Value
- Name

Given this basic 'units' of knowledge, we should model our data according to some rules so we can make useful things given this arrangement. The first step is to find a common 'meta – meta – model" for the model stated before so we can 'import' data from disparate sources into it. The data is ultimately aggregated into this three structures given meaningful parsing of it (and configuration files).

Meta Meta Model

The underlying common model for entities coming from diverse data sources should allow to covert from and to the 'model' easyly. Let's begin considering what a data structure could become after decomposing it a little. We should consider, for example, rows, or statements (from RDF), predicates or columns from a relational database and tables or types (rdf:Type) for example, from these two kind of data sources (RDBMs and RDF).

Lets arrange them into objects of different classes. The name in the left is the name of the class, and the three value tuple named 'statements' is the arrangement of statements about other entities the object has:

Mapping:
Statements: <Context, Entity, Role>
Entity:
extends Mapping. Statements: <Context, Mapping, Role>
Context:
extends Entity, Statements: <Entity, Mapping, Role>
Role:
extends Context, Statements: <Context, Mapping, Entity>

So, the Statements part is the references the object has to other objects in the data space, in the form of 3-tuples. The inheritance relationship is for allowing reification and further composition. The correspondences between these objects and a data source are roughly this:

A Mapping represents a row (in a database table) or an RDF statement.
An entity represents a value in a table cell or an RDF object.
A context represents a table in a database or rdf:Type value of statement in RDF.
A role represents a database column or a RDF predicate.

The population of the model should allow for triadic relationships to be stated over the model, and to be accessible for querying in a meaningful way. (And the use of configuration mapping files for population of upper models)

For example, in a Value x, let's say (200Km), we could 'operate' semiotically on it and 'ask' it for a reference to its related Type object, given a Name, let's say ('Distance'), and once we have the Type we arrived from the value, regarding it as that name, ask the Type object for a Value named ('Speed') and get (100Km/h). If we query using the same mechanism for a Value named ('Time') we should get (2h).

Architecture

The idea is building level over level based on mapping configurations files in XML that describe how entities in a lower level populates entities in an upper level. This should give us the layers of metametamodel (for data load), model (for inference, semantics) and later a business and agent layer to ease to provide user interface, reporting and interaction layers.

The whole system should provide, through the use of accessory packages, such as a framework for FCA (FCA[2]) and information gathering and retrieval (Watson[3]) for the build up of a kind of Business Intelligence (2.0) application framework, with dimensional and aggregated views of semantically integrated data.

The project page of the ongoing development effort for this framework is online and available at:
http://xama.dev.java.net

References

1. JF Sowa,
“Ontology, Metadata and Semiotics”
http://users.bestweb.net/~sowa/peirce/ontometa.htm

2. Formal concept analysis:
http://en.wikipedia.org/wiki/Formal_concept_analysis

3. Mark Watson ,
“Practical Artificial Intelligence Programming With Java , Third Edition”