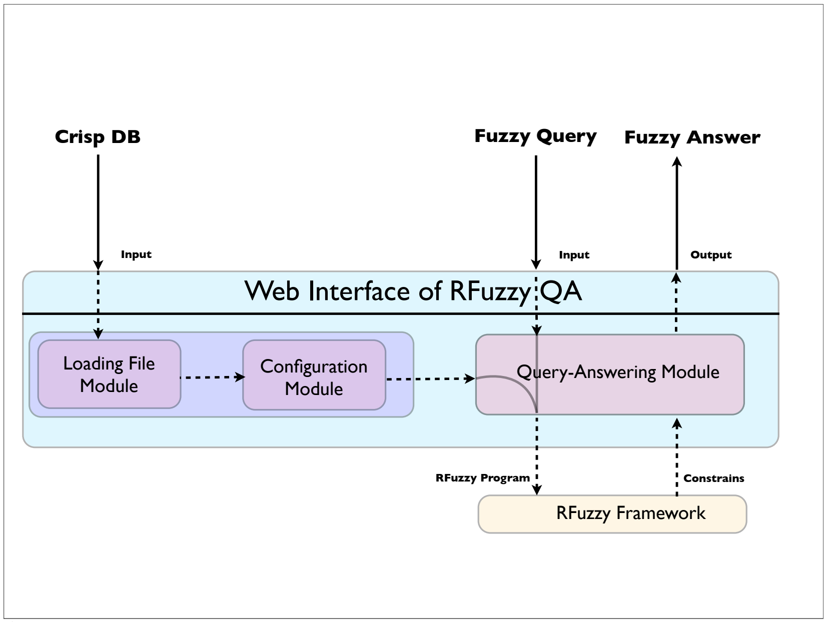
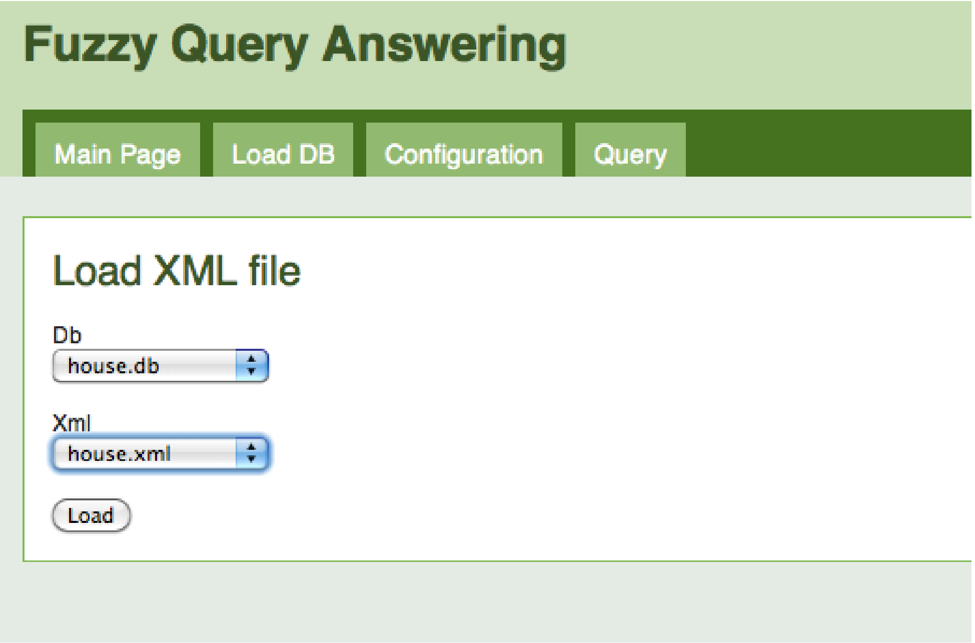
1 Dive into Fuzzy Query-Answering Web Interface



Sunday, February 20, 2011

1.1 Loading File Module

Loading File Module offers users to choose a crisp database with its schema from local file system, on which users intent to create fuzzy concepts and make fuzzy queries. In FQAS, the crisp database is in a prolog file with extension ".pl", and its schema is in a XML file with extension ".xml".



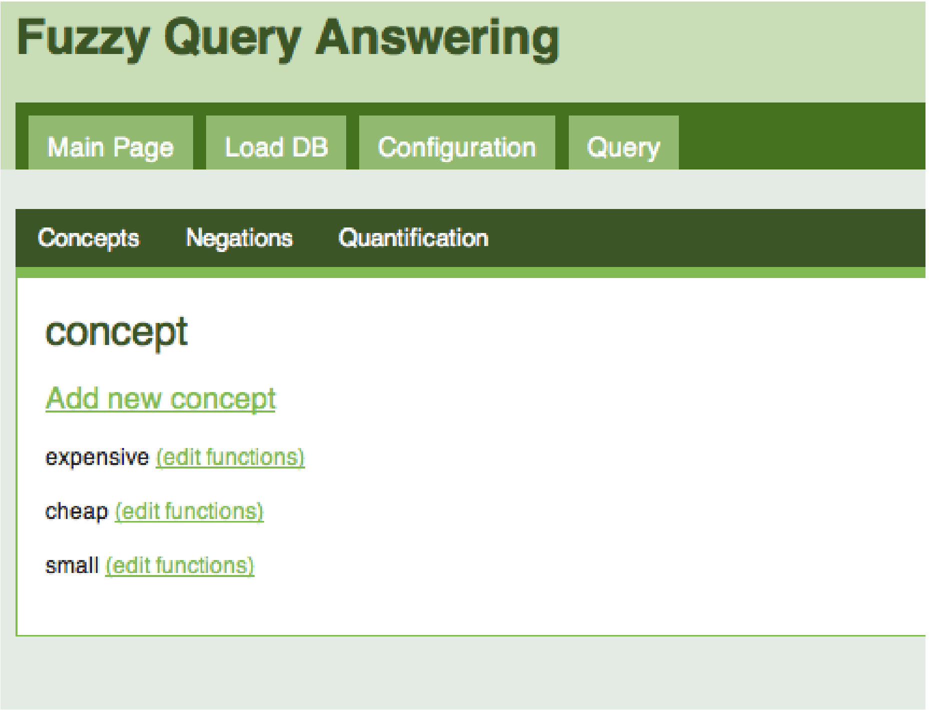
Db
house.db

Xml
house.xml

1.2 Configuration Module

Configuration Module functions the configuration of fuzzy concepts over crisp database, negation and quantification of natural language. The crisp database is the one load it in the Loading File Module. Users are entitled to create and edit fuzzy concepts such as “beautiful”, “old” by defining fuzzy functions over attributes in crisp database. Negations and quantifications can be created by defining the their functions over unit interval $[0, 1]$.

Configuration Page



Fuzzy Query Answering

concept

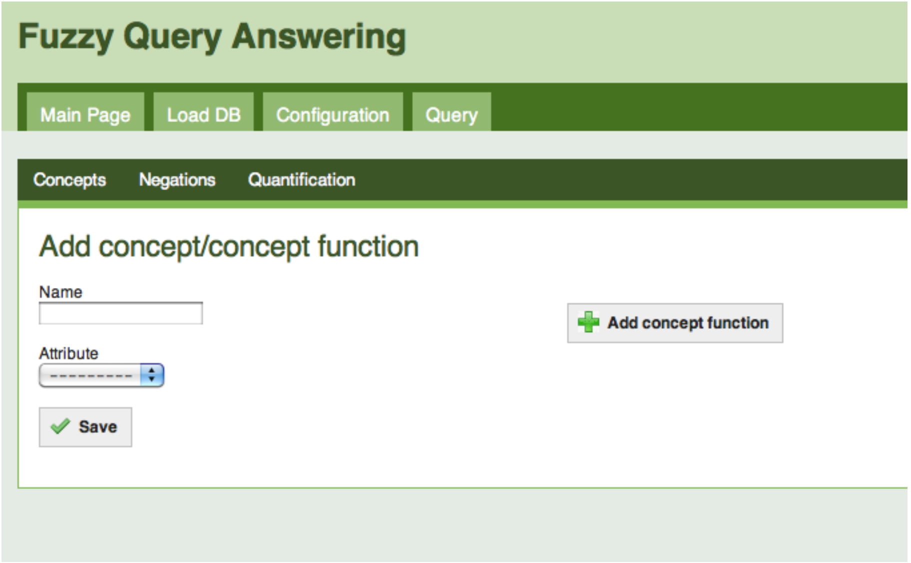
Add new concept

expensive (edit functions)

cheap (edit functions)

small (edit functions)

Add New Concept Page



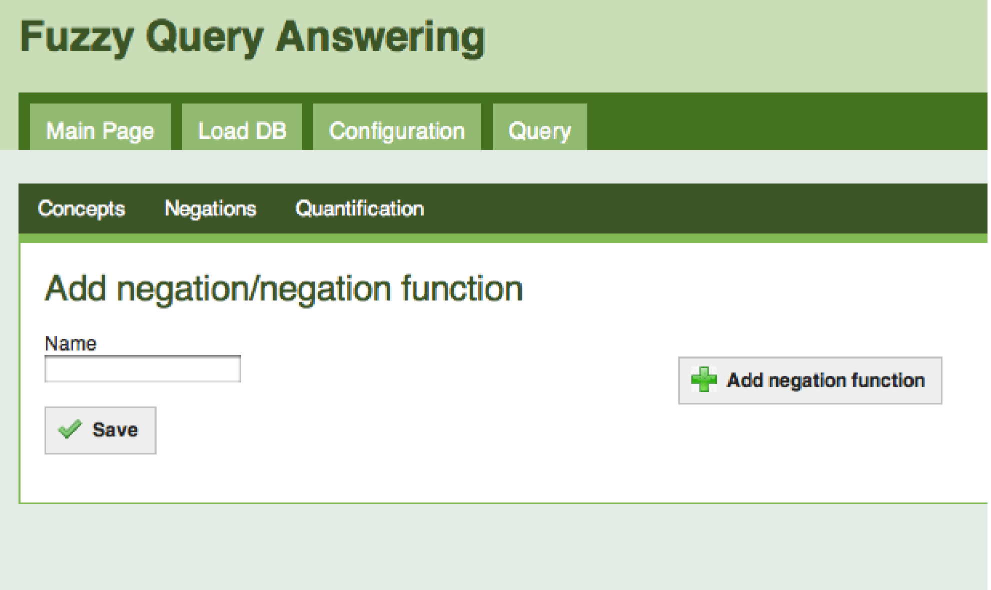
Fuzzy Query Answering

Add concept/concept function

Name

Attribute

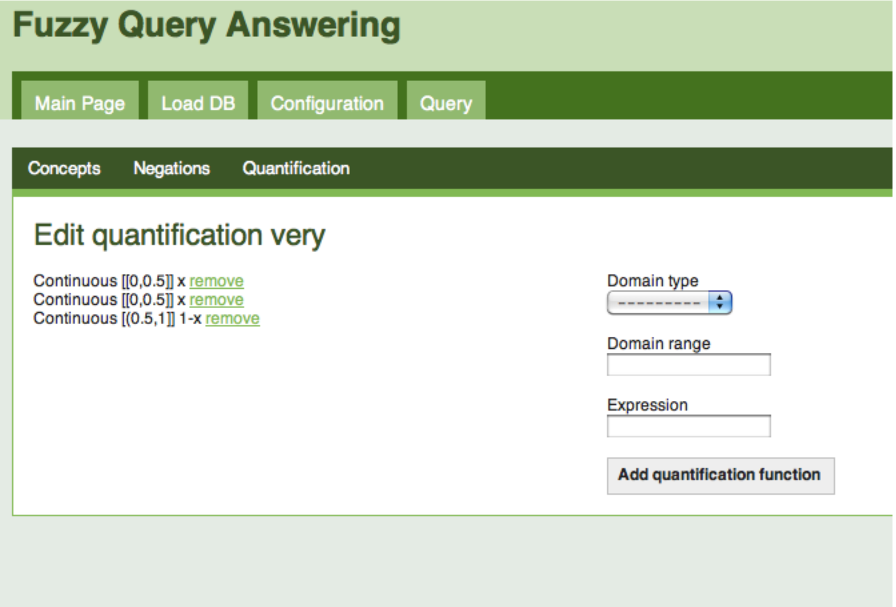
Add New Negation Page



Add negation/negation function

Name

Edit Quantification



Edit quantification very

Continuous [[0,0.5]] x remove
Continuous [[0,0.5]] x remove
Continuous [(0.5,1]] 1-x remove

Domain type: ---------

Domain range

Expression

1.3 Query-Answering Module

Query-Answering Module offers users to create simple fuzzy queries or complex fuzzy queries by choosing negations, quantifications and fuzzy concepts which are defined in Configuration Modules.

In RFuzzy Framework, we define the fuzzy query as a pair $< A, v >$, where $A \in TB_{\Pi,\Sigma,V}$ and v is either a "new" variable that represents the initially unknown truth value of A or it is a concrete value $v \in [0,1]$ that is asked to be the truth value of A. We extend this simple query into complex one.

Definition 1.1. ***(Complex fuzzy query).***

$$Answer(\vec{t}, v) \overset{c,F_c}{\longleftarrow} F(p_1(\vec{t_1}, v_1), ..., p_m(\vec{t_m}, v_m))$$

where $p_i s$ are predicates in RFuzzy program P, $Answer$ is a predicate which never appears in program P, v_i and v could be unknown truth value for their associated atoms or concrete value v assigned to their atoms.

By introducing quantifiers into RFuzzy framework, it could be used to enhance the expressivity of the query, which is defined as follow,

Definition 1.2. ***(Simple fuzzy query with negation and quantification).***

$$< A, v >$$

where A is an atom $q(x_1, \ldots, x_n)$. q is represented as a regular expression,

$$q = (Negation|Quantification)^* Predicate$$

Definition 1.3. ***(Complex fuzzy query with negation and quantification).***

$$Answer(\vec{t}, v) \overset{c,F_c}{\longleftarrow} F(q_1(\vec{t_1}, v_1), ..., q_m(\vec{t_m}, v_m))$$

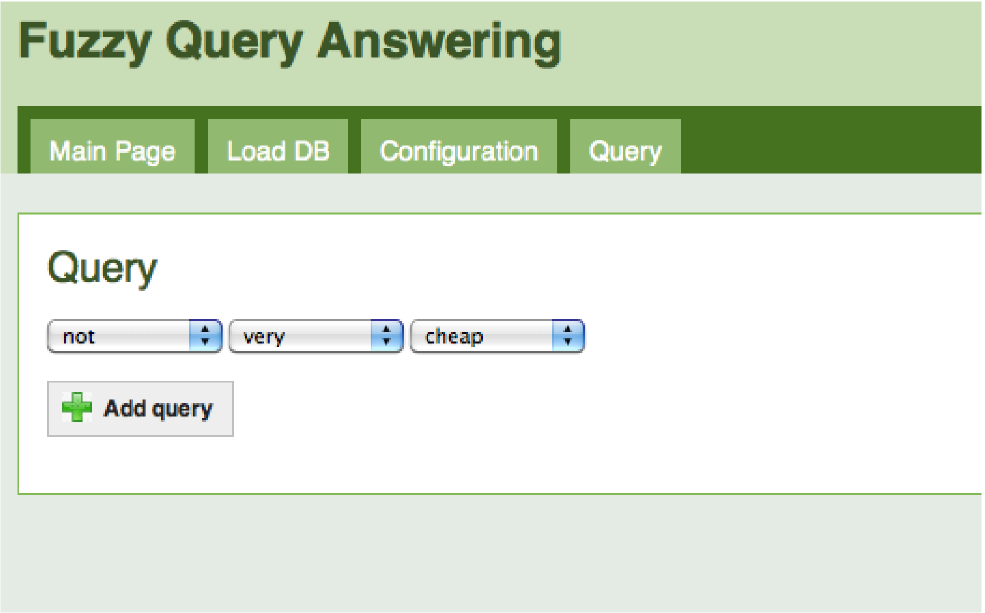
where $Answer$ is a predicate that never appears in program P, q_i is represented as a regular expression,

$$q_i = (Negation|Quantification)^* Predicate$$

Example 1.4. *A set of negations is $N = \{not, seldom\}$, a set of quantifiers is $Q = \{very, extremely\}$, and a set of fuzzy concept is $C = \{beautiful, clear\}$. The simple query with negation and quantification can be represented as " not very beautiful girls ?", "seldom extremely clear statements ?" and so on. A Complex query with negation and quantification is generated by joining those simple query together with fuzzy rules. The simplest complex query could be " not very beautiful and seldom extremely clear landscape ?", which is formalized in fuzzy logic as,*

$$Answer(Landscape, V) \longleftarrow \boldsymbol{min} \quad \begin{array}{l} not(very(beautiful(Landscape))), \\ seldom(extremely(clear(Landscape))). \end{array}$$

Query Page


Fuzzy Query Answering
Main Page
Load DB
Configuration
Query
Query
not
very
cheap
Add query