University Of New Brunswick

CS6795 Semantic Web Techniques

Project Proposal

Title: - Specifying Units of Measure in POSL and RuleML 1.0: Targeting OO jDREW Execution

Instructor: Dr. Harold Boley.

Advisor: Dr. Tara Athan.

Team 7:

Chandan Bagai,
Sunil Kumar Ravikumar,
Udit Trivedi,

Introduction:

Specification of measurement units and conversion between various units is very essential in the field of Science, Engineering, Manufacturing, Commerce, Medicine and Environmental Regulation. [3]

Seven base dimensions:

SI base unit

Base quantity	Name	Symbol
length	meter	m
mass	kilogram	kg
Time	second	s
electric current	ampere	A
thermodynamic temperature	kelvin	K
amount of substance	mole	mol
luminous intensity	candela	cd

A quantity is a hypothetically measurable amount of something. We refer to those things whose amounts are described by physical-quantities as physical-dimensions. Time, length, mass, and energy are examples of physical-dimensions. Quantities are described in terms of reference quantities called units of measure. A meter is an example of a unit of measure for quantities of the length physical-dimension. [5]

Derived units are expressed algebraically in terms of base units or other derived units. The symbols for derived units are obtained by means of the mathematical operations of multiplication and division.

For example, the derived unit for the derived quantity ***Molar mass*** (mass divided by amount of substance) is the kilogram per mole, symbol **kg*/mol***. [5]

Our Project aims at building library for specifying units of measure in POSL and RuleML 1.0, by exploring Tom Gruber's work on ontoligua, this will be used to build the knowledge base in OO jDREW. Other part of the project is invertibility of transformation of units e.g. Fahrenheit to Celsius and vice versa.

Projectobjectives:

The primary goal of our project is to build library specifying units of measure in POSL and RuleML 1.0, by exploring Tom Gruber's work on ontolingua, this will be used to build the knowledge base in OO jDREW's Naf Hornlog after pre-processing. A Hierarchy of expressivity will be specified in First order logic (FOL) using Hornlog and datalog.

As per Tom Gruber's theory on units of measure with respect to ontologies, there is a hierarchy of concepts, commonly called taxonomy and there is a general class of dimensions; subclasses of this class are *base dimensions* and *derived dimensions*.[5]

Every physical quantities and their dimensions are defined as below in lisp.

```
 :def (defined (quantity.dimension ?x))

(define-instance METER (unit-of-measure)
  "SI length unit.  No conversion function is given
because this is a standard unit."

 :axiom-def (and (= (quantity.dimension meter) length-dimension)
        (SI-unit meter)))

(define-instance KILOMETER (unit-of-measure)
  ""
  := (* meter 1000)
  :axiom-def (= (quantity.dimension kilometer) length-dimension))

(define-instance INCH (unit-of-measure)
  "English length unit."
  := (/ meter 39.37)
  :axiom-def (= (quantity.dimension inch) length-dimension))

(define-instance FOOT (unit-of-measure)
  "English length unit of feet."
  := (* inch 12)
  :axiom-def (= (quantity.dimension foot) length-dimension))

(define-instance MILE (unit-of-measure)
  "English length unit."
  := (* foot 5280)
  :axiom-def (= (quantity.dimension mile) length-dimension))

(define-instance ANGSTROM (unit-of-measure)
  "ang.strom \'a<nj>-str<e>m also '<o.><nj>-\ n (1897)
 [Anders J. <A^o>ngstr<o:>m]
    :a unit of length equal to one ten-billionth of a meters -- used esp.
    for wavelengths of light."
  := (/ meter (expt 10 10))
```

```
:axiom-def (= (quantity.dimension angstrom) length-dimension))
```

The physical dimension of Area is defined as length squared.

```
:axiom-def (= Area
              (*LENGTH-DIMENSION  *LENGTH-DIMENSION))
```

The physical dimension of force is defined as mass times length over time squared.

```
:axiom-def (= FORCE-DIMENSION
              (* MASS-DIMENSION
                    (* LENGTH-DIMENSION (expt TIME-DIMENSION -2)))))
```

With respect to above mentioned axiom; mass, length and time are instance of base-dimension class and Force is an instance of derived-dimension class.

Final part of the project is to implement invertibility of transformation between units (e.g. Fahrenheit to Celsius and vice versa) using either multiple mode declaration or propagation constraint techniques.

Relationship between Fahrenheit and Celsius is $9C = 5(F - 32)$.

It will be implemented individually for Fahrenheit to Celsius conversion and vice versa by using flattening the variables in function. [6]

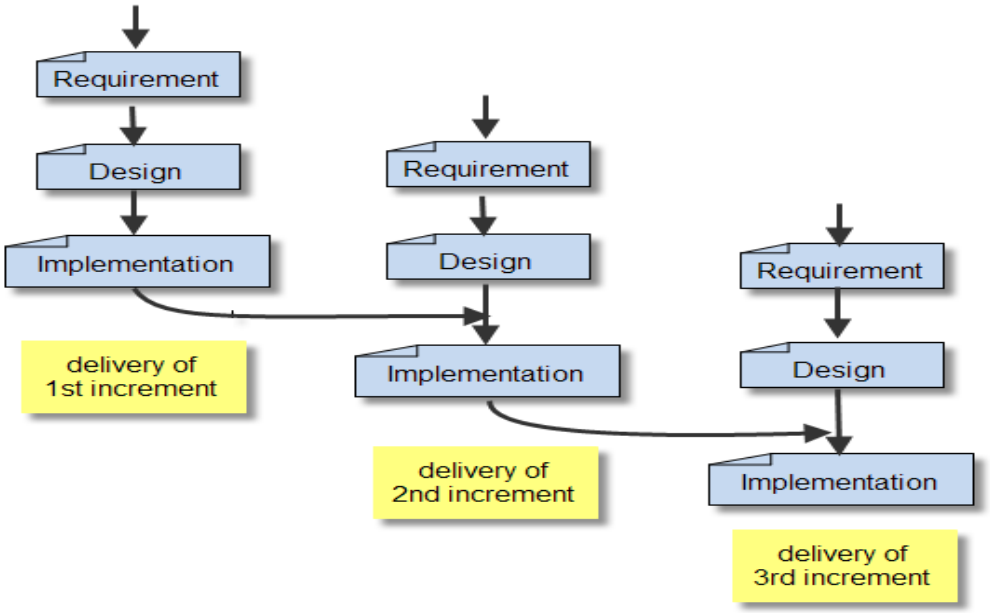
Methodology:

The Methodology used for this project will implement Incremental model. We will accomplish this goal by the following tasks:-

Increment 1: A library will be build specifying Units of Measure in POSL and RuleML of which most part will be executable.

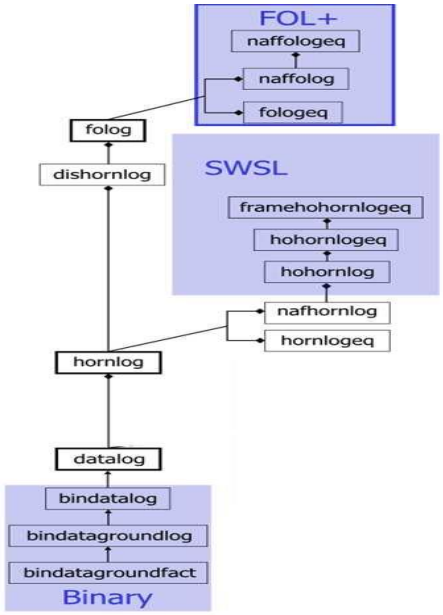
Increment 2: Certain considerations will be made and decision will be made depending upon which type of relationship it is. If it is a binary relationship then we can go for binary Datalog and if it is polyadic relationship than we can Hornlog.

Increment 3: Implementing invertibility of transformations between units via constraint propagation techniques. E.g. Conversion of Celsius to Fahrenheit and vice versa.



Incremental Development Model

Project's Tools and technologies:



For implementing this project, we will create knowledge base in OOjDREW using POSL and RuleML. As discussed in the introduction, there will be a subclass of the class dimensions, called derived dimensions. We will use the following logical expressions to represent expression based on the number of arguments associated with the relation.

Binary datalog
Binary datalog can be used only with two arguments (two basic dimensions or constants, e.g. Area: = length * length).

Datalog
For polyadic relations, binary datalog is not sufficient for expressivity, but we can use Datalog with a limitation of using functions (e.g. Acceleration which can be expressed in terms of basic dimension as Distance/ (Time*Time)).

Hornlog
If functions are necessary, then datalog will not be sufficient for expressivity. So, for that, we will use Hornlog (e.g. dimension Force (Force: = mass * acceleration, where acceleration is function which will be defined as mention in datalog above).

We will execute the knowledge base in the OO jDREW (Object Oriented Java Deductive Reasoning Engine for the Web), which is the reference implementation of the (Naf Hornlog) RuleML Web rule language. It is an Object Oriented extension to jDREW.

References:

1. POSL- An Integrated *Positional-Slotted Language* for Semantic Web Knowledge
   http://ruleml.org/submission/ruleml-shortation.html
2. Units of Measure
   http://physics.nist.gov/Pubs/SP811/sec04.html
3. Frank Olken: An Ontology of Measurement Units and Dimensions:
   http://ontolog.cim3.net/file/work/OntologySummit2009/OntologySummit2009_Symposium_20090406-07/units-ontology-talk-v01--FrankOlken_20090406.pdf
4. NIST UnitsML:
   http://unitsml.nist.gov/Presentations/UnitsML-SCC20.ppt
5. Ontolingua Quantities and Units:
   http://www-ksl.stanford.edu/htw/dme/thermal-kb-tour/physical-quantities.html
   http://www-ksl.stanford.edu/htw/dme/thermal-kb-tour/standard-units.html
6. Functional-Logic Programming
   http://www.cs.unb.ca/~boley/FLP/notes.html