

Use Case Implementation: The “Diet” Group

Jesse Weaver (Domain “Expert”)
Greg Williams (Software Engineer)
Ankesh Khandelwal (KR)

CSCI6962 Semantic eScience
Tetherless World Constellation
Rensselaer Polytechnic Institute

Use Case

NAME: Determine whether given foods are similar to known diabetic exchanges and provide additional nutritional information.

GOAL: Provide a mechanism by which a user can determine foods that are similar to known diabetic exchanges and also get related nutritional information.

Summary

For persons with diabetes, it's extremely important to monitor ones diet and to make sure that it complies with any recommendations from doctors or dieticians.

- Extreme hypoglycemia: diabetic coma
- Extreme (long-term) hyperglycemia: vision loss

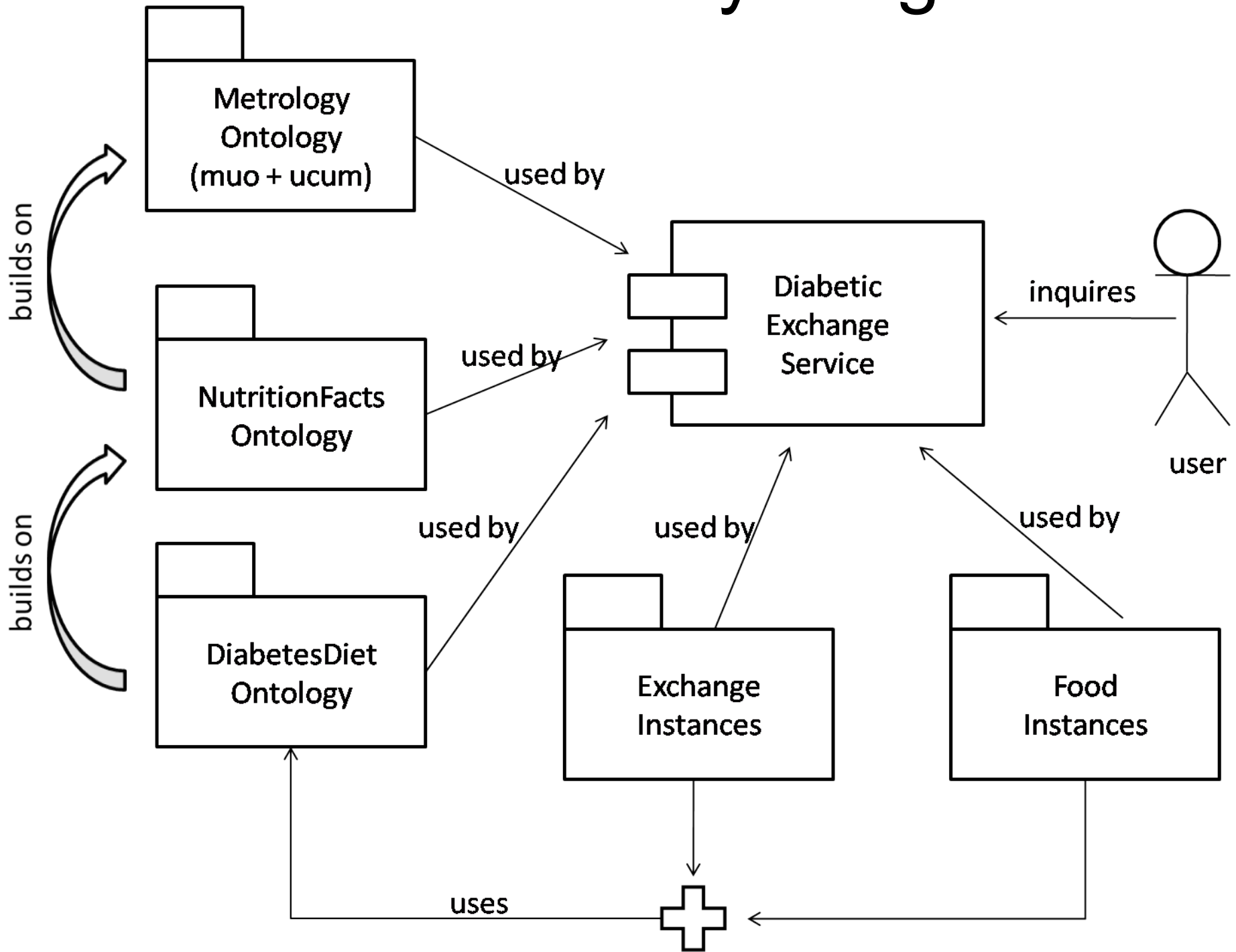
This use case aims to determine the number of diabetic exchanges for a given food.

Basic Flow

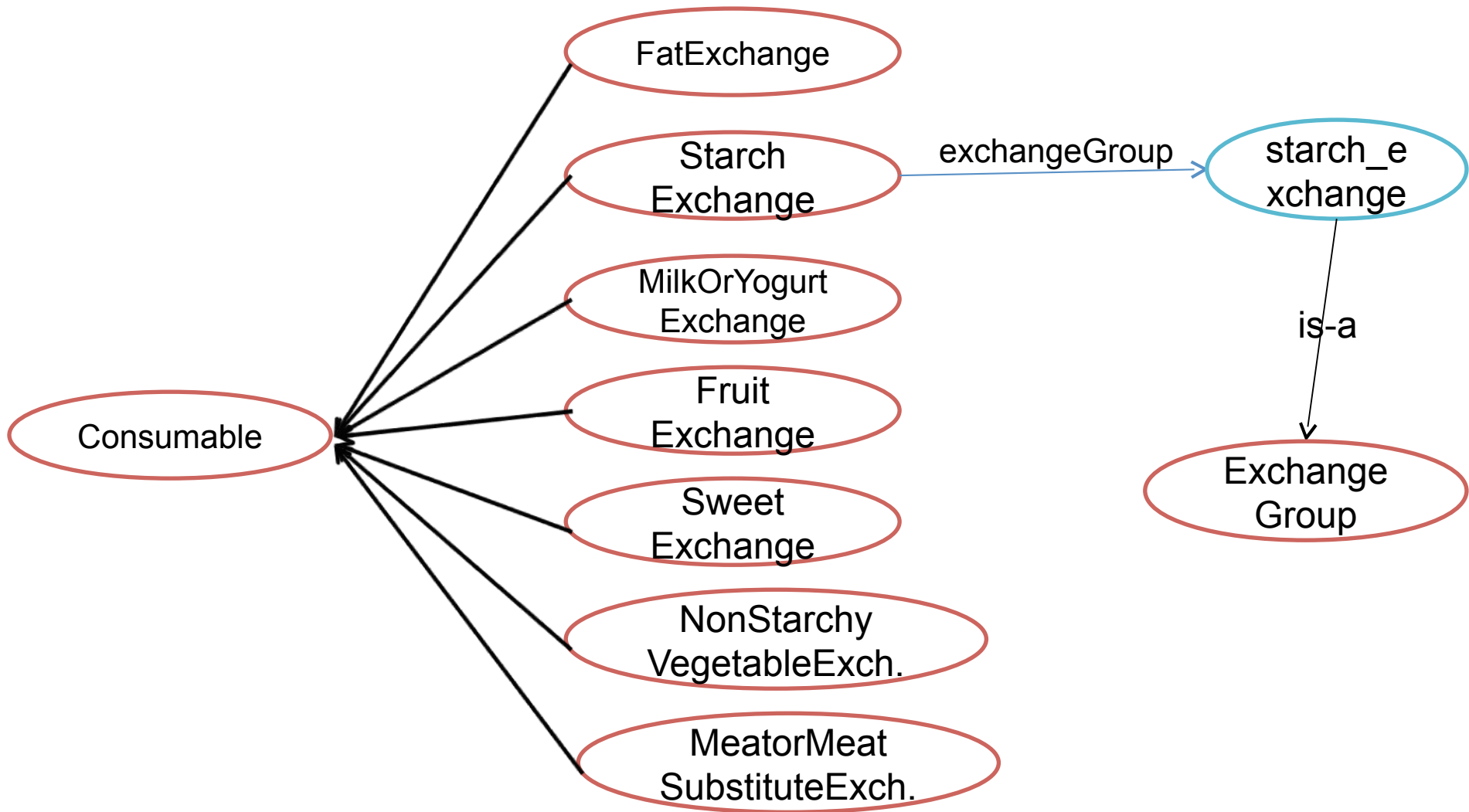
- Person starts application.
- Person selects an exchange group and possibly further constrains interest with a keyword (e.g., “apple”).
- Person submits search.
- System lists foods similar to foods in the selected exchange group, and if a keyword was provided, the descriptions of the foods contain the keyword.
- Person selects the appropriate (desired) food.
- System returns exchange information and nutritional information for the food.

Demo!

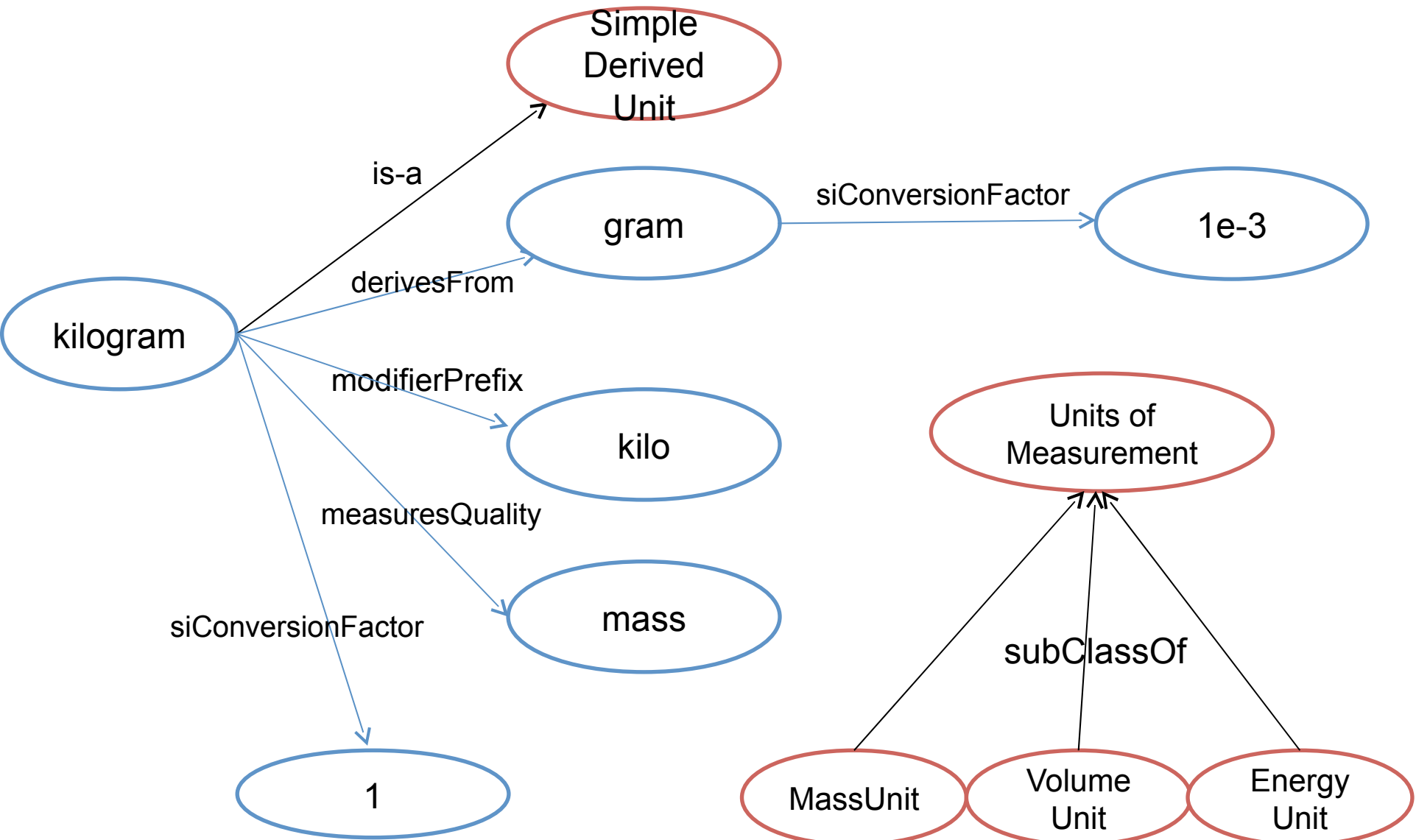
Activity Diagram



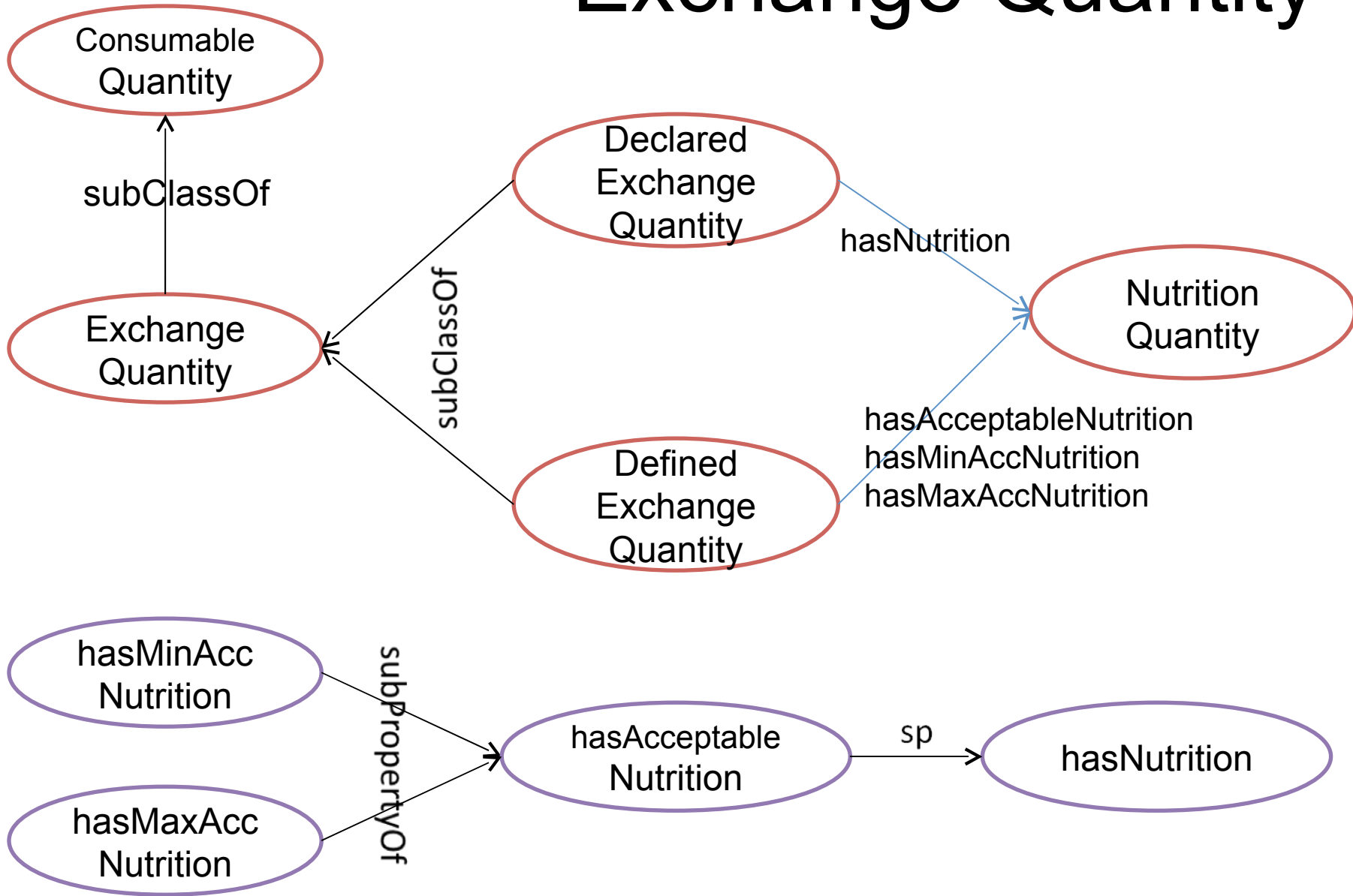
Exchange Groups



Metrology (muo & ucum)



Exchange Quantity



Architecture

Diabetic Exchange Code

Perl RDF Toolkit

SPARQL Protocol

Apache + mod_perl

Sesame SPARQL Endpoint

Modeling Example

```
:ideal_starch a :Consumable ;
  :exchangeGroup :starch_exchange ;
  :hasQuantity [
    a :DefinedExchangeQuantity ;
    :hasMinimumAcceptableNutrition [
      a :NutritionQuantity ;
      :quantityOf :protein ;
      muo:qualityValue [
        a muo:QualityValue ;
        muo:numericalValue 0.0 ;
        muo:measuredIn :gram ;
      ] ;
    ] ;
    :hasMaximumAcceptableNutrition [
      a :NutritionQuantity ;
      :quantityOf :protein ;
      muo:qualityValue [
        a muo:QualityValue ;
        muo:numericalValue 3.0 ;
        muo:measuredIn :gram ;
      ] ;
    ] ;
  ] ;
  :hasAcceptableNutrition _:total_calories_80.0_cal ;
  :hasAcceptableNutrition _:total_carbohydrate_15.0_gram ;
  :hasMinimumAcceptableNutrition _:total_fat_0.0_gram ;
  :hasMaximumAcceptableNutrition _:total_fat_1.0_gram ;
] .
```

```
:lenders_bagel_whole_grain_plain
  a :Starch ;
  rdfs:label "Lender's Bagel Whole Grain, Plain" ;
  dc:identifier "lenders_bagel_whole_grain_plain" ;
  :hasQuantity [
    a :SampleQuantity ;
    muo:qualityValue [
      a muo:QualityValue ;
      muo:numericalValue 104.0 ;
      muo:measuredIn :gram ;
    ] ;
    :hasNutrition [
      a :NutritionQuantity ;
      :quantityOf :calories_from_fat ;
      muo:qualityValue [
        a muo:QualityValue ;
        muo:numericalValue 20.0 ;
        muo:measuredIn :nutrition-label-Calories ;
      ] ;
    ], [
      a :NutritionQuantity ;
      :quantityOf :cholesterol ;
      muo:qualityValue [
        a muo:QualityValue ;
        muo:numericalValue 0.0 ;
        muo:measuredIn :milligram ;
      ] ;
    ],
    :dietary_fiber_8_gram,
    :protein_12_gram,
    :saturated_fat_0_gram,
    :sodium_410_milligram,
    :sugars 9_gram,
```

Modeling Example

Benefits of Semantics

- Based upon existing quantities (and their associated units), we can determine the relevant units with which the user would probably like to specify a quantity.
- Declared and defined exchanges can be treated similarly depending on contexts.

Questions?