**Recipe (Food) Ontology Domain**

Our domain includes recipes and the things used to make them. Our food is broken up into food groups that are ingredients to the recipe. We have many descriptors such as a difficulty level, the amount of time it takes to make the recipe, a food allergy, temperature, the size of the meal, the tools that are used, the type of cuisine, as well as how healthy the recipe is. The previous list contained all of the classes that we used in conjunction with our main recipe class to build a recipe. Our properties link the classes together with each other to make the recipe whole.

**Term Glossary**

Classes

* Recipe
* Food
* FoodGroups
* Fruits
* Vegetables
* Grains
* ProteinFoods
* Dairy
* Oils
* MiscFoods
* Cuisine
* Temperature
* MealSize
* HealthStatus
* FoodAllergy
* Difficulty
* KitchenTools

Properties

* hasIngredients
* dividedInto
* isClassifiedAs
* hasTemp
* hasSize
* isHealthy
* hasAllergy
* hasDifficulty
* takes
* uses
* name
* otherName
* occasion
* totalCalories
* includes
* origin
* minTempValue
* maxTempValue
* requiredAppliances
* minServings
* maxServings
* description
* severity
* difficultyLevel

**DL Representation**

**Interesting Decisions**

We changed the time it takes to complete a recipe to a property instead of a class. The change was because we wanted a more specific time instead of making 3 general classes to cover all of the recipes. The main classes were too general. When we removed the Time class, we actually forgot to remove the three classes that were created to classify the amount of time it took. The property and the Time class ended up both being used, but we caught the mistake and changed it to just implementing the property.

**Assumptions**

* We assume that the properties that are entered in are accurate (in the future)

**Scope Limitations**

* Drinks aren’t implemented but can be used, they just may not be classified correctly
* Quantities of the ingredients are not implemented

**Future Considerations**

* Drinks may be added
* Quantities can be implemented