Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

AN ENTITY MODEL FOR ONLINE RECIPE SEARCH

Usashi CHATTERJEE, Fausto GIUNCHIGLIA, Devika P. MADALLI, Vincenzo MALTESE

July 2016

Technical Report # DISI-16-002

Also sumbitted to the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016)

An entity model for online recipe search

Usashi CHATTERJEE[b], Fausto GIUNCHIGLIA[a], Devika P. MADALLI[b], Vincenzo MALTESE[a]
[a] *DISI - University of Trento, Italy*
[b] *DRTC - Indian Statistical Institute, Bangalore, India*

Abstract. In this paper we propose an entity model which allows us to represent and search for recipes in online environments. The entity model provides relevant entity types, attributes and relations between them. We advance the state of the art by covering aspects that are often not considered, or loosely considered, by existing models. The proposed model extends and formalizes Schema.org, the de-facto standard used by developers to structure Web content such that it can be better visualized and retrieved. The entity model is rooted in FRBR, the model used in Library Science to conceptualize creative works. We illustrate the methodology followed, the developed model, and the evaluation we conducted.

Keywords. Entity model, entity types, Schema.org, FRBR, recipes, search

1. INTRODUCTION

Searching for recipes on the Web is very popular as it gives the opportunity to people around the world to challenge themselves with the preparation of dishes from different regions and cultures. Several websites provide various thematic collections of recipes and offer dedicated facilities to search for them.

Recipe websites do not only differ in terms of content, but also in terms of language and terminology used, as well as in terms of properties designated to describe the recipes. For instance, some websites offer recipes in English, Italian, German or Hindi language; for the notion of *recipe instructions* used to describe the steps to be followed to prepare a dish, myrecipes.com uses the term *preparation,* chow.com uses the term *instruction*, and kraftrecipes.com uses *make it*; kraftrecipes.com explicitly provides *preparation time*, while myrecipes.com does not; vice versa, myrecipes.com explicitly provides *nutritional information* while kraftrecipes.com does not.

It is clear that in such situation the supported queries vary from website to website. Moreover, in absence of a reference standard, such diversity constitutes an obstacle towards a unified system aiming to search for recipes across websites.

Our main contribution to this problem is the definition of an entity model that provides a schema for recipes in terms of entity types and related properties. The model constitutes an advance w.r.t. the state of the art for at least the following reasons:

1. ***It covers aspects that are often not considered by existing models.*** To the best of our knowledge, there are surprisingly only a few attempts of defining a model tuned to online recipe search. Moreover, these models often do not consider aspects of crucial importance to adequately characterize recipes and

facilitate their retrieval. To ensure we cover all the important aspects, our methodology is based on the identification of competency questions [1].

2. ***The proposed model is compliant, extends and provides an ontological framework to Schema.org***. In fact, despite its wide adoption, its proposed schemas are informal, limited in the number of properties and often not adequate in their data types. Indeed, the fact that properties are often defined as plain text poses serious limits to the kind of queries it can support.
3. ***It is grounded on Library Science theories on creative works.*** Recipes are products of the human intellect. The way in which this is accounted for in Schema.org, i.e. by making sure that recipes inherit all the properties of creative works, does not allow to distinguish between recipes and dishes. None of the existing recipe models actually captures such distinction. Still, this distinction is fundamental to discriminate between the process (the recipe) and the product of the process (the dish). This is not a minor detail as they have different properties. For instance, while a recipe has an author and instructions to be followed, the dish has a cook that - in preparing it - gives her unique "touch". To adequately capture these aspects, we base our work on the FRBR model [12] that neatly distinguishes between a *work* and their possible physical *manifestations*. This distinction in the model allows us, for instance, to search for all the recipes that can be followed to prepare the same dish.
4. ***It formalizes the specific role played by the entity types in the model.*** Library Science principles allow us also to discriminate the role played by the entity types in the model w.r.t. the competency questions. In fact, we organize entity types into *common* (those of general applicability that can be re-used across domains [15]), *core* (those returned by the queries) and *auxiliary* (those that determine how to filter results).

In our work, we follow the approach described in [13][17]. The approach is centered on the notion of *entity* that is *any object so important in the real world to be given a name*. It consists of four consecutive steps aiming at defining the entity model, the formal language, the natural language, and corresponding instances. In this paper, due to space limitation, we concentrate on the first step.

The rest of the paper is organized as follows. In Section 2 we review the state of the art. In Section 3 we introduce the notion of entity model. In Section 4, we describe the steps in our methodology. In Section 5 we illustrate the process we followed to identify competency questions and to formalize them into general query patterns that the model will have to support. In Section 6 we describe the developed model. In Section 7 we illustrate the evaluation of the model; it was conducted against six recipe websites and shows that their information content comfortably fits into our model. Finally, Section 8 concludes the paper by summarizing the work done and by describing our next steps.

2. STATE OF THE ART

2.1. EXISTING WORK ON RECIPES

Table 1 provides a comparison of representative scholarly articles addressing recipes. They are compared in terms of general aim, the usage of competency questions to drive the development, the artifact(s) developed and the main aspects covered.

To the best of our knowledge, limited work has been done so far in defining a model tuned to online recipe search. DeMiguel et al. [7] defined an ontology and developed a recipe search system based on ingredient similarity. It is quite limited in the aspects covered given that it only focuses on ingredients, meals, cuisine and dietary types. Sam et al. [4] is not a scientific work, but it is interesting as it provides the results of a classroom exercise in modeling recipes. It identifies common ontology patterns from different websites. It does not provide any concrete evaluation. Despite that, this work can be considered the closest to our aim and approach.

The other works do not specifically address online recipe search. They differ in aim, artifacts developed and aspects covered. In particular, they mainly focus on the definition of the ontology, i.e. the formal terminology (with no explicit model) required to support automated reasoning. Nanba et al. [8], given their focus on NLP, cover natural language terminology and lexical relations between terms. Some of the aspects that are important for recipe search (as we identified them from competency questions; see Section 5), such as cuisine and events, are rarely covered by the reviewed works.

Only some of these works follow a precise methodology or technique: Batista et al. [5] partially follow METHONOTOLOGY [10]; Nanba et al. [8] employ existing extraction techniques. Only some of them use competency questions in order to drive the development process.

Article	Aim	Competency Questions	Artifact(s) developed	Aspects Covered
DeMiguel et al. [7]	Recipe search based on ingredient similarity	NO (*)	Ontology & Search system	Ingredient, Meal, Cuisine, Dietary types
Sam e al. [4]	Recipe search	YES	Recipe Model	Recipes and related properties
Batista et al. [2]	To support the cooking process	YES	Ontology	Recipes, Food, Utensils, Units & Measures, Actions
Snae et al. [3]	Menu planning based on diet restrictions	YES	Ontology	Food, Dishes, Dietary restrictions, Nutrients, Cuisine, Utensils, Cooking methods
Krieg-Bruckner et al. [5]	Meal planning based on diet restrictions	NO (**)	Ontology	Recipes, Food, Meals, Dietary restrictions, Utensils
Nanba et al. [8]	NLP	NO	Ontology	Recipes, Ingredients, Utensils, Actions

Villarias [6]	Information Extraction	NO	Ontology	Recipes, Dishes, Ingredients, Nutrients, Courses, Actions
(*) questions are used in the evaluation; (**) A motivating scenario is used				

Table 1. Comparison of representative works addressing recipes

2.2. SCHEMA.ORG

Sponsored by the major search engines, Schema.org is a collection of schemas that is expressly designed and maintained such that the schemas can be used by webmasters to standardize the model used to structure Web content. This is achieved by incorporating invisible tags to the HTML of web pages. In this way, content can be better visualized and searched by search engines. Over 10 million websites already use Schema.org to markup their web pages.

Schema.org provides schemas in terms of generic *Types*, given in English, each of them associated with a set of *Properties*. Types are arranged in a hierarchy from more general to more specific Types. A second hierarchy provides primitive *Data Types* for Property values, such as Integer, Date or Text. The most general Type is *Thing*. Top level Types include Action, Creative work, Event, Intangible, Medical Entity, Organization, Person, Place and Product. Intangibles include Enumerations - used to standardize property values - as well as quantities, structured values, and some abstract entities such as Language. In Section 3, we propose a specification that, we believe, better discriminates ontologically among these various “things”.

Schema.org offers a schema to represent recipes[1]. However, the current version of this schema is quite limited as it only provides a small set of Properties. In addition, most of the Properties are loosely represented in that their corresponding Data Type is often an underspecified Text. This makes representation quite generic and search difficult. For instance, the fact that in Schema.org ingredients are represented as Text prevents users to search for recipes requiring a certain quantity of a given ingredient (e.g. less than 100gm). To overcome this, the model we propose, presented in Section 6, provides more specific Properties and Data Types such that such more elaborate queries can be supported.

3. ENTITY MODELS

The work presented in this paper follows the approach described in [13][17]. The approach takes into account diversity and exploits it to make explicit the local semantics, i.e. the meaning of words in a certain *domain* and *context*, such that information becomes unambiguous to humans as well as to machines. The approach is centered on the notion of *entity* that is *any object so important in the real world to be given a name*. We build a domain following four consecutive steps that are often not explicit in standard modelling approaches:

[1] https://schema.org/Recipe

1. **Defining the entity model**: the *entity model* specifies the schema, i.e. the entity types and the properties, in terms of attributes and relations between them, required to represent the key entities of the domain. For instance, it may specify that ingredient and preparation are properties of recipes.
2. **Defining the formal language**: the *formal language* specifies the ontology, i.e. the concepts and the semantic relations between them, thus formalizing the terms appearing in the entity model and those required to describe corresponding instances. It may indicate that chicken and rice are ingredients and that the concept of preparation denotes an activity;
3. **Defining the natural language**: the *natural language* specifies the vocabulary, i.e. the terminology used to lexicalize the concepts in multiple natural languages. The vocabulary may provide the information that the notion of preparation can be conveyed with the terms "preparation" and "instructions" in English and with the term "preparazione" in Italian;
4. **Defining the instances**: entity types are instantiated as entities in adherence with the schema and by using the formal and natural language provided. An example of instance of recipe is "chicken curry".

In this paper, due to space limitation, we concentrate on the first step. The interested reader can refer to [18][19][20] for a comprehensive example of the application of all the steps for the geographical domain.

We illustrate the main notions we use to define the entity model, and provide a corresponding graphical representation. In doing so, we extend and formalize Schema.org. The goal is to better discriminate ontologically among the various "things" it provides. Figure 1 shows an excerpt of the recipe model as initial example.

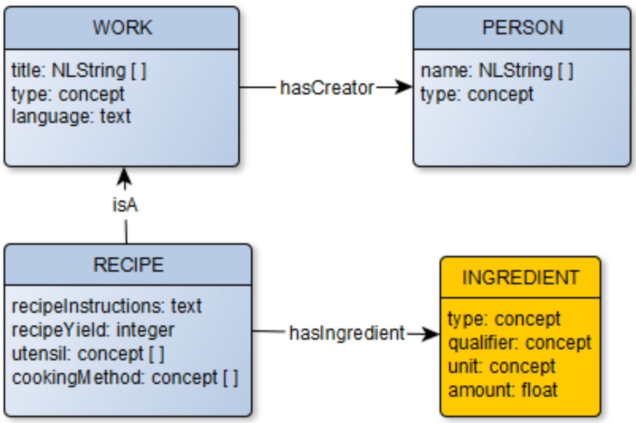

Figure 1. An example of entity model

Data Types. We support all primitive *Data Types* in Schema.org - such as *integer*, *float*, and *text* - and additional ones - such as *NLString* and *concept*. The first indicates a text in a language. It is particularly suited to support the representation of names. For instance, it can specify that "Italy" denotes the name of a country in English and that the same entity is called "Italia" in Italian. The second indicates that the values are taken from the formal language (i.e. they are concepts from an ontology).

Properties. We explicitly distinguish entity *Properties* into *Attributes* and *Relations*. Attributes describe entities. Therefore, the range of an Attribute is a *Data Type*. Relations relate entities between them. Therefore, the range of a Relation is an *Entity Type*. In Figure 1, Attributes (e.g. utensil) are represented within boxes, while Relations (e.g. hasCreator) are represented as arrows between boxes.

Structured attributes. Attributes and Relations can be grouped together to provide structured pieces of information by which entities are described. We call them *Structured Attributes*. In Figure 1, Structured Attributes are represented as yellow boxes. Ingredient is constituted by four Attributes, each of them having its own Data Type. Structured Attributes include some of the "things" classified as Intangibles in Schema.org such as Offer and Address.

Entity types. *Entity Types* describe concrete and abstract named entities. They include some of the "things" that Schema.org directly classifies under Thing - such as persons, locations, and events - as well as under Intangible - such as flights and reservations. In Figure 1 they are represented by blue boxes. Entity Types form a hierarchy whose root is *Entity*. Differently from Schema.org, we formalize inheritance of Properties via an explicit *isA* Relation between them. Each instance of Entity must have a *name* and a *type*. A name is any string in a language used to refer to an entity. Titles play the role of names in creative works [14]. The value of the type Attribute further specifies the kind of entity as more specific than the Entity Type, without the need to introduce a dedicated Entity Type for it. For example, Trento is an instance of the Entity Type Location with type set to City (City is more specific than Location in the ontology).

4. METHODOLOGY

In this section we briefly illustrate the methodology we employ to develop the entity model of the domain. It consists of three consecutive steps, described below.

Step 1: Identification of the query patterns. With this step, we collect informal competency questions, and generalize them into query patterns. The identification of a motivational scenario and the compilation of a corresponding list of competency questions are a well-established way to frame the scope and get insights about entities to be captured by the model to be developed [1]. We collect competency questions through interviews and from relevant state of the art works. The competency questions are then analyzed to identify common needs. The analysis leads to the identification of general query patterns where entities and properties are parametrized.

Step 2: Building the recipe model. This step leads to the definition of the entity model. The query patterns obtained from the previous step allow the identification of the entity types, their attributes and relations between them. They are those necessary to answer the queries. We define them and provide a corresponding graphical representation. The goal is to appropriately represent the domain in the model (representation) such that it can be effectively queried (reasoning).

Step 3: Evaluation of the model. With this step we evaluate the developed model. According to Gómez-Pérez [11], the goal of the evaluation process is to determine what the developed ontology defines correctly, does not define, or even defines incorrectly. This should be done in two steps: verification and validation. The purpose of verification is to check the syntactic correctness. The purpose of validation is to check its consistency, completeness and conciseness. Given that the presented work is focused to (the sub-task of) defining the entity model, we concentrate on *completeness* and *conciseness*. The model is *complete* if it fully captures what it is supposed to represent of the real world. The model is *concise* if it does not contain redundancies. We ensure by construction that the developed model is complete and concise w.r.t. the competency questions. In fact, we develop the model such that it represents exactly what is needed to support the identified query patterns (no more, no less), in terms of entity types, attributes and relations. In addition, we verify its *empirical adequacy* [16] by checking that information extracted from a random set of websites in the domain (in this case on recipes) can be comfortably represented by the entity model.

5. IDENTIFICATION OF THE QUERY PATTERNS

Informal competency questions were collected by conducting interview sessions and by complementing them with questions identified in relevant state of the art works. The collected questions were subsequently analyzed to identify the general needs they express and the requirements they suggest for the entity model. This led to the identification of a general pattern.

The participants involved in the interview sessions include Computer Science and Information Science students, research scholars and post-doctoral fellows, from different parts of the world (India, Italy, China, Georgia, Spain, Ukraine, Brazil and Belarus) working at the University of Trento in Italy and at the Indian Statistical Institute in Bangalore, India. They include 9 males and 4 females with age in the range 21 to 55 years. Each participant was interviewed individually and was asked to provide examples of recipe related queries typically used by them to search on the web. Each interview lasted for around 20-30 minutes. This lead to the identification of 30 informal questions that we complemented with 16 questions collected from relevant state of the art works, and in particular from [4][7] that are the closest to our aim.

The 46 informal questions we collected cover different usage scenarios and showcase different types of queries that an individual typically issue on a recipe website. By analyzing them, we identified similar needs expressed across different questions. We generalized the questions in 16 distinct queries clustered in 7 different scenarios. Details about the identified scenarios, the corresponding queries and the requirements they suggest - in terms of entity types and properties needed - are provided in appendix. We discovered that the 16 queries can be further generalized into a single pattern as follows:

Give me all *recipes* R which belong to *cuisine* C to prepare *dish* D appropriate for *event E* with specific property constraints.

This pattern clearly evidences the importance of four entity types: recipe, cuisine, dish and event. Property constraints are given in form of property-operator-value triples and include ingredient, authorship, rating, dietary, nutrition and preparation constraints.

6. THE RECIPE MODEL

The requirements collected and the general pattern identified with the previous step give clear indications about the entity types and the properties to be defined in the entity model. According to the function they play in the general pattern, we distinguish among common, core and auxiliary entity types:

- ***Common entity types*** are those of general applicability that can be (re-)used across domains. This notion resembles the idea of *common isolates* in Library Science as proposed by Ranganathan [15] that indicates those aspects of human knowledge playing a role in multiple domains. Within a domain, common entity types can become core or auxiliary. The common entity types playing a role in recipes are Person, Location, (Creative) Work and Event.
- ***Core entity types*** are those returned by the queries. As it can be noted from the general pattern, Recipe is the core entity type of the model. Notice that it extends the common entity type Work.
- ***Auxiliary entity types*** are those that in the general pattern determine how to filter results. They include Cuisine, Dish and Event. Notice that the common entity type Event plays the role of auxiliary for recipes.

Figure 2 provides a graphical representation of the recipe model we developed. It uses the notions presented in Section **Error! Reference source not found.**. The recipe model includes six entity types and three structured attributes. Common entity types are represented as white boxes with dashed boundary lines. Core entity types are represented as blue boxes with solid boundary lines. Auxiliary entity types are represented as blue boxes with dashed boundary lines. We give additional details about the developed entity types and their properties in the following.

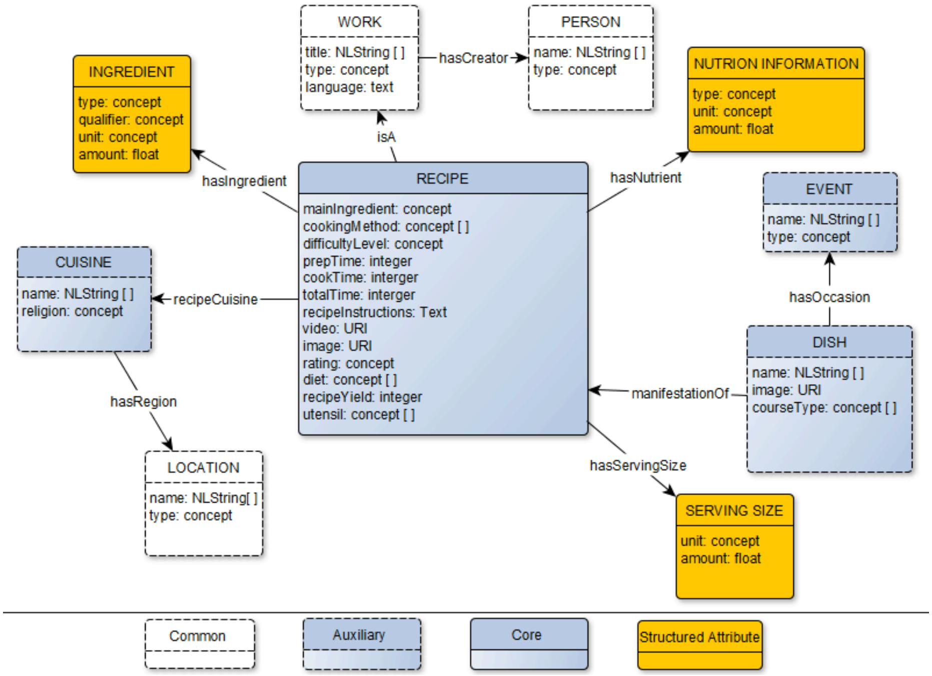

Figure 2. The recipe model

6.1. COMMON ENTITY TYPES

We assume that the common entity types Person, Location, Work and Event can be mapped to Person, Place, Creative Work and Event in Schema.org, respectively. In Figure 2, we exemplify them by providing their core properties as inherited from the root of the hierarchy Entity (not represented in the picture) and as strictly required to support the competency questions we identified.

6.2. CORE ENTITY TYPES

The core entity type in the domain is Recipe. A recipe describes the steps that need to be followed in the process of preparing a particular food item. Such description is clearly a product of the human intellect. For this reason, in our model Recipe extends Work and leads to the preparation of Dishes.

This is in line with what postulated by the eFRBR model [14] that we developed by formalizing FRBR [12] to the purpose of conceptualizing creative works. What is relevant of eFRBR for this work is that each Work has one or more creators (captured via the *hasCreator* relation) and that every entity type more specific than Work can have its own manifestations. In general, a manifestation is a possible way by which a work takes a physical form that can be experienced by people. We represent manifestations via an explicit *manifestationOf* relation between Dish and Recipe.

The properties we defined for Recipe include and extend those in Schema.org. We keep the attribute names it proposes, but we differ in the data types. In fact, in many cases it defines them generically as Text. We often substitute it with one that is more appropriate to support the queries. In particular, in our model *cookingMethod* assumes values from an enumeration of concepts taken from the formal language (e.g. frying, baking); *recipeYield* is represented as integer and indicates the number of people served and it is distinguished from *Serving Size* that is represented as structured attribute and indicates the amount produced for a single portion served to a person (e.g. 200gm); *recipeCategory* is dropped and captured via the attribute *courseType* in Dish (e.g. dessert, appetizer) or via *Events* (e.g. Christmas, Thanksgiving Day); *recipeCuisine* is a relation and *recipeIngredient* is a structured attribute.

recipeCuisine is a relation, whose range is Cuisine, such that we can query by its name (e.g. Mediterranean cuisine) or by providing regional (e.g. Italy) and religious (e.g. Muslim) constraints.

The attribute *recipeIngredient* is structured, whose range is Ingredient, such that we can query by the type (e.g. chicken), any qualifier (e.g. boneless), any amount (e.g. 1) or unit of measure (e.g. pound). Multiple values of *recipeIngredient* can be specified for a single recipe. Notice that *mainIngredient* differs from *recipeIngredient* as the former indicates the ingredient that characterizes the dish prepared.

Nutritional facts are associated to a recipe via the *hasNutrient* attribute that is represented in a way similar to Schema.org. The structured attribute (see Nutrition Information) provides necessary information about the type (e.g. carbohydrates, fat, proteins, sugar) and amount of nutrients brought by the prepared dish. However, while in Schema.org a fixed number of possible nutrients is given, we can extend them at any moment via an enumeration of concepts denoting types of nutrients. In addition we can vary the unit of measure that in Schema.org is fixed.

Recipe instructions can be given in textual (*recipeInstructions*) or other media (*video* and *image*) format.

Notice that the order in which properties are given in Recipe is not accidental. In fact, they are ordered according to the Actand(A)-Action(B)-Actor(C)-Tool(D) Principle given by Ranganathan [15]. The set A includes the attribute *mainIngredient* and the relation *hasIngredient*; B includes attributes from *cookingMethod* to *totalTime*; C includes attributes from *recipeInstructions* to *recipeYield* as well as the relations *manifestation-of* and *hasServingSize*; D includes *utensil*. This order can be used to order recipes, e.g. on the shelf of a library or to order query results.

6.3. AUXILIARY ENTITY TYPES

We identified three auxiliary entity types: Cuisine, Dish and Event. *Cuisine* allows organizing recipes into collections by cultural, regional or religious similarity. In fact, the Encyclopedia of global studies defines cuisine as an important expression of the impact of globalization on cultural practices having an impact on the ingredients used and the cooking methods employed [9]. *Dish* represents the final product of a recipe. Distinguishing dishes from recipes allows for instance the identification of all those recipes that lead to the preparation of the same dish. *Event* is the only auxiliary entity type having a corresponding type in Schema.org. Dishes are prepared for specific occasions. In our model, this is captured via the *hasOccasion* relation. We use Event to capture holidays (e.g. roasted turkey is usually prepared for the Thanksgiving Day), seasons (e.g. spring, summer) and meals (e.g. pasta in Italy is usually served at lunch). In fact, a meal can be seen as an occasion occurring every day. This will be codified in the formal language by specifying that holiday, season and meal are more specific than event and that, for instance, Thanksgiving Day is more specific than holiday, spring and summer are more specific than season, and lunch is more specific than meal. All these concepts will be possible values of Event.type.

7. EVALUATION

The methodology we employed ensures that the developed model is by-construction complete and concise w.r.t. the competency questions. In addition, we evaluated the *empirical adequacy* [16] of the recipe model by checking that information extracted from six popular recipe websites can be comfortably represented using the proposed entity types and properties. The six websites are: eatingwell.com, foodnetwork.com, allrecipes.com, sanjeevkapoor.com, cooking.nytimes.com and food.ndtv.com. This has been done by selecting random recipes from them and by manually mapping the schema used in each website with our recipe model. For each recipe we came up with a graphical representation of the model and a comparison table of the properties used.

Due to space limitation, we provide the graphical representation (see Figure 3) and the comparison table (see Table 2) for a single recipe[2] taken from the eatingwell.com website. As it can be seen from the comparison table, the mapping is typically 1-to-n. This indicates that properties in our model have a more fine-grained representation.

Table 2. The comparison table of the terminology used by a recipe from eatingwell.com mapped to the terminology used in our recipe model

Website Recipe Terminology	Proposed Recipe Terminology
Ingredients	INGREDIENT type qualifier unit amount
Preparation	recipeInstructions
Nutrition	NUTRITION INFORMATION type

[2] eatingwell.com/recipes/grilled_chicken_salad_with_a_fresh_strawberry_dressing.html

	unit amount
Makes	SERVING SIZE unit amount
Active Time	prepTime
Total Time	totalTime
Health & Diet Considerations	Diet
Meal/Course	EVENT type
Type of Dish	DISH courseType
Preparation/Technique	cookingMethod
Season	EVENT type
Ease of Preparation	difficultyLevel
Ethnic / Regional	CUISINE name
Servings	recipeYield
Main Ingredient	mainIngredient
Rating	rating

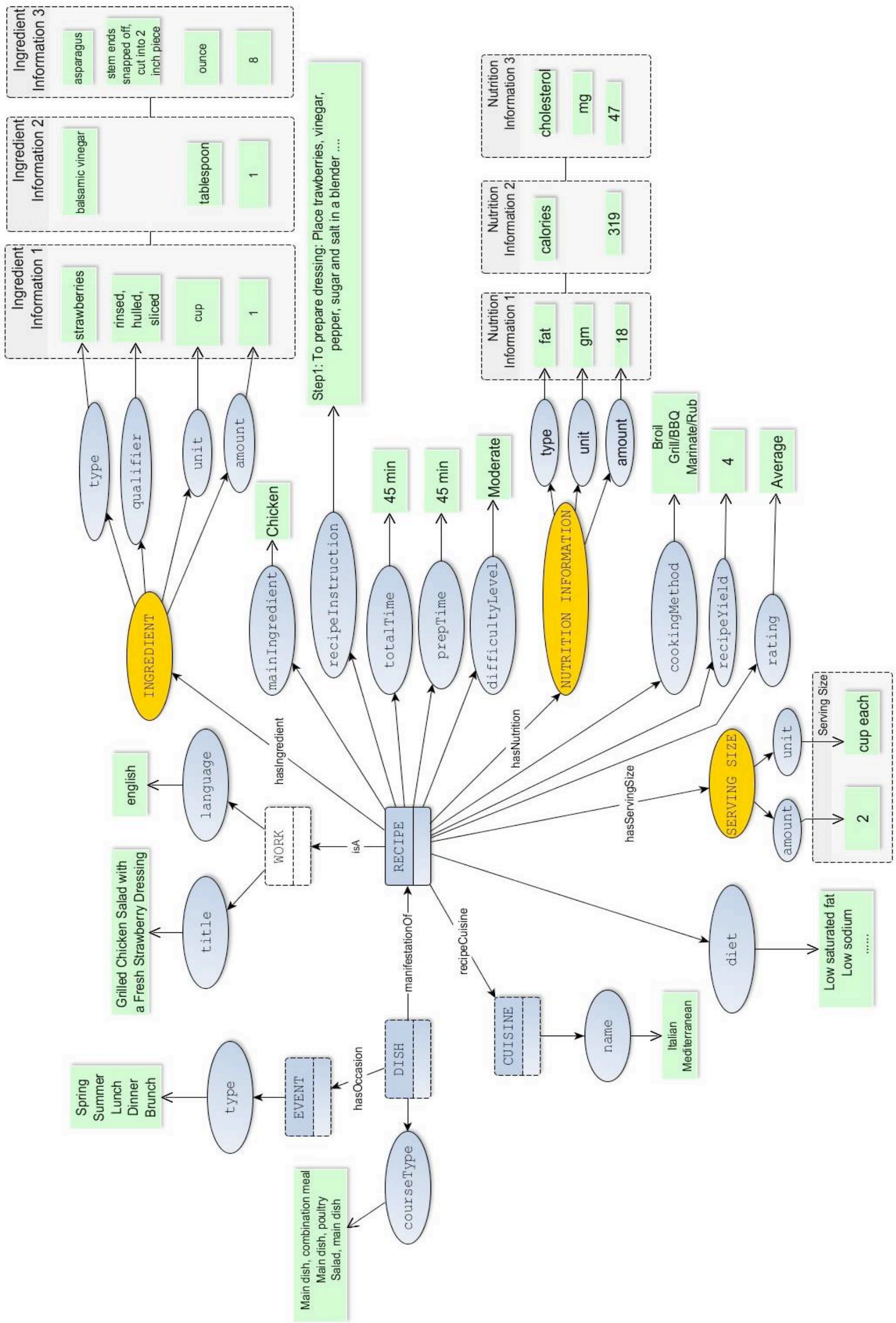

Figure 3. The graphical representation of a recipe from eatingwell.com mapped to our recipe model

8. CONCLUSIONS

In this paper we proposed an entity model tuned to support online search for recipes. It constitutes an advance w.r.t. the state of the art in that it covers aspects which are often not considered, or loosely considered, by existing models. In particular, it extends and gives an ontological valence to Schema.org. In addition, it is based on the principles of the FRBR model used in Library Science to conceptualize creative works. In particular, FRBR allows distinguishing recipes from dishes. Among other things, this distinction allows us to associate the right properties to them and supports the identification of all those recipes that lead to the preparation of the same dish.

The methodology employed ensures that the developed model is complete and concise w.r.t. an initial set of competency questions as it takes into account all the necessary entity types and properties. The analysis and generalization of the competency questions led to the identification of entity types playing different roles in answering the queries. The evaluation confirms the adequacy of the model as it is able to capture information randomly selected from six recipe websites.

This paper described only the first step of a broader methodology, based on the approach presented in [13][17], aiming at the development of domain knowledge in terms of entity model, formal language, natural language and corresponding instances. We plan to write a journal paper describing the application of all the steps on recipes.

ACKNOWLEDGMENTS

This work has received funding from the ITPAR India-Italy Cooperation Programme for Advanced Research.

REFERENCES

[1] Grüninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies. In *IJCAI '95Workshop on Basic Ontological Issues in Knowledge Sharing*, *IJCAI.*

[2] Batista, F., Pardal, J. P., Mamede, P.V.N., & Ribeiro, R. (2006). Ontology construction: cooking domain. *Artificial Intelligence: Methodology, Systems, and Applications*, *41*, 1-30.

[3] Snae, C., & Brückner, M. (2008). FOODS: a food-oriented ontology-driven system. In *Digital Ecosystems and Technologies, DEST 2008, 2nd IEEE International Conference on IEEE*, 168-176.

[4] Sam, M., Krisnadhi, A. A., Wang, C., Gallagher, J., & Hitzler. (2014). An Ontology Design Pattern for Cooking Recipes–Classroom Created. In: Victor de Boer, Aldo Gangemi, Krzysztof Janowicz, Agnieszka Lawrynowicz (Eds.), *WOP , Workshop on Ontology and Semantic Web* Patterns.

[5] Krieg-Brückner, B., Autexier, S., Rink, M., & Nokam. S. G. (2015). Formal Modelling for Cooking Assistance. In *Software, Services, and Systems* (pp. 355-376). Springer International Publishing.

[6] Villarias, L. G. (2004). Ontology-based semantic querying of the web with respect to food recipes. Doctoral Dissertation, Technical University of Denmark, Denmark.

[7] DeMiguel, J.,Plaza, L., & Díaz-Agudo, B. (2008). ColibriCook: A CBR System for Ontology-Based Recipe Retrieval and Adaptation. In *ECCBR Workshops*,199-208.

[8] Nanba, H., Takezawa, T., Doi, Y., Sumiya, K., & Tsujita, M. (2014). Construction of a cooking ontology from cooking recipes and patents. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 507-516.

[9] Anheier, H. K., & Juergensmeyer, M. (Eds.). (2012). *Encyclopedia of global studies*. SAGE Publications.

[10] Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. *AAAI Spring Symposium Series*, AAAI Press, 33–40.

[11] Gómez-Pérez, A. (2001). Evaluation of ontologies*.* *International Journal of intelligent systems*, *16*(3), 391-409.

[12] International Federation of Library Associations, & Institutions. Section on Cataloguing. Standing Committee (1997-2009). Functional Requirements for Bibliographic Records: final report. IFLA Study Group on the Functional Requirements for Bibliographic Records (Ed.).
[13] Giunchiglia, F., Maltese, V., & Dutta, B. (2012). Domains and context: first steps towards managing diversity in knowledge. *Journal of Web Semantics, 12-13*, 53-63.
[14] Maltese, V., Giunchiglia, F., Sarangi, A., & Margonar, S (2015). *eFRBR: An entity model for FRBR*. Paper presented at ISKO UK biennial conference, London.
[15] Ranganathan, S. R. (1967). *Prolegomena to library classification*. London: Asia Pub. House.
[16] Suppe, F. (2000). *Understanding scientific theories: An assessment of developments, 1969-1998*. Philosophy of Science, 67, 102-115.
[17] Giunchiglia, F., Dutta, B., Maltese, V. (2014). *From Knowledge Organization to Knowledge Representation*. Knowledge Organization Journal, 41(1), 44-56.
[18] Giunchiglia, F., Dutta, B., Maltese, V., Farazi, F. (2012). *A facet-based methodology for the construction of a large-scale geospatial ontology*. Journal on Data Semantics, 1 (1), 57-73
[19] Farazi, F., Maltese, V., Giunchiglia, F., Ivanyukovich, A. (2011). *A faceted ontology for a semantic geo-catalogue*. In Proc. of Extended Semantic Web Conference (ESWC), 169-182
[20] Maltese, V., Farazi, F. (2013). *A semantic schema for GeoNames*. INSPIRE conference.
[21] Giunchiglia F., Sebastiani R. (200). *Building Decision Procedures for Modal Logics from Propositional Decision Procedures: The Case Study of Modal K(m)*. Information and computation, Academic Press, 162 (1/2), 158-178.
[22] Giunchiglia, E., Giunchiglia, F., Sebastiani, R., Tacchella, A. (1998). *More evaluation of decision procedures for modal logics*. In A.G. Cohn, L. Schubert and S.C. Shapiro (editors), Proc. of KR'98, 6th International Conference on Principles of Knowledge Representation and Reasoning, 626-635.

APPENDIX: COMPETENCY QUESTIONS CLUSTERED BY SCENARIO

SCENARIO 1: looking for recipes given ingredients and quantity constraints

Give me pasta recipes for 4 people
Give me recipes using 200gm brown rice, 1 pound boneless chicken.

REQUIREMENTS: Often people search for recipes based on the availability of ingredients they have at home or on the basis of the main ingredient of the recipe. The model needs to capture different aspects of ingredients, such as type (pasta or chicken), qualifiers (boneless), amounts and units of measure (200 gm or 1 pound). The model needs to distinguish between common ingredients and the main ingredient, i.e. the one that characterizes the dish prepared.

SCENARIO 2: looking for recipes which are adequate for a certain diet or having certain nutrition constraints

Give me recipes for vegetarians
Give me recipes of sugarless dessert for diabetic person
Give me a recipe for a gluten free appetizer containing less than 100 calories

REQUIREMENTS: People's choice of food is often bound to the diet they follow. The diet followed by individuals is affected by various aspects which includes religious and ethical norms, clinical or medical reasons, or personal preference. Therefore, the model needs to capture diet and nutrition information. The latter should describe the type of nutrients (sugar, calories), their amount and units of measure. The queries also suggest that the model should include the course type (dessert, appetizer).

SCENARIO 3: looking for recipes belonging to a specific cuisine or typical of a specific region

Give me recipes for spicy Japanese noodles.
Give me Kerala recipes.

REQUIREMENTS: People are interested in experimenting with food prepared following recipes that belong to different cuisines (Japanese cuisine). Each cuisine has its own unique identity in terms of ingredients and cooking style - with a great variance from region to region

(Kerala) - that often reflects cultural and religious practices.
SCENARIO 4: looking for recipes with certain preparation constraints
Give me recipes with eggs that are easy to cook. Give me a recipe to cook Italian pasta prepared using microwave oven. Give me recipes for 4 people which can be prepared in less than 20 minutes. Give me recipes prepared with fried cauliflower. **REQUIREMENTS:** Queries often embed preparation related constraints such as level of difficulty (easy to cook), utensils/cookware used (microwave oven), preparation time (less than 20 minutes), and cooking method (frying).
SCENARIO 5: looking for popular recipes
Give me recipes of Gordon Ramsay which have high rating. **REQUIREMENTS:** People may look for popular recipes, i.e. those created by a famous person (Gordon Ramsay) or rated high by other users. Therefore, the model needs to capture the fact that recipes are creative works and that they can be rated.
SCENARIO 6: looking for recipes of a specific dish
Give me a recipe of chicken biryani. **REQUIREMENTS:** People search for recipes to prepare a particular dish. Recipes provide the instructions that need to be followed to prepare the corresponding dish. There might be different recipes, with different titles, to prepare the same dish.
SCENARIO 7: looking for recipes of dishes which are cooked in special occasions or for specific meal
Give me a recipe to prepare roasted turkey for Thanksgiving celebration Give me a recipe of halal food prepared for Iftar party. Give me breakfast recipes **REQUIREMENTS:** Religious and cultural events (e.g., Thanksgiving) are often characterized by specific dishes (roasted turkey) prepared to celebrate them. Also, meals (breakfast) consumed at different time of the day have different characteristics.