A Profile-Policy Integration for A Personalized Lifestyle

Ozgu Can, Okan Bursa, Murat Osman Unalir
Department of Computer
Engineering, Ege University
Izmir, Turkey 35100
Email: {ozgu.can, okan.bursa, murat.osman.unalir}@ege.edu.tr

***Abstract*—As the amount of information increases and access to this information gets easier, the need for personalized systems is inevitable. A personalized system gives users the efficiency to meet their specific preferences by increasing usability and decreasing the unwanted content. The core component of creating a qualified personalized system is defining semantically rich user profiles. We propose to integrate user profiles with policy management concept to provide a rule-based personalization. The main contributions of this work are: developing a profiling methodology to define semantically rich user profiles and generating a profile-based policy management in order to satisfy the demands of a personalized system. We demonstrated our empirical approach for the health care domain to build a personalized lifestyle model. This user-adaptive system will also give the user a significant time reduction when searching specific items for a special user profile type by restricting the options based on the same profile when compared to a non-adaptive system.**

Keywords-Profile Management; Personalization; Policy Enforcement; Healthcare Systems

I. INTRODUCTION

The promising advantages of online networks create impressive occasions for users. The success of these occasions should be improved by adapting web services to each user's characteristics and behaviors. Personalized systems are the key component to achieve this improvement. As the amount of information increases, making decisions about information becomes difficult. Thus, a personalized system gives users the efficiency to meet their preferences.

Personalization is the process of giving decisions among the given choices according to the user's behavior, needs, preferences, interests and demographics. Hence, users can reach personalized contents such as customized web pages, advertisements, music albums and restaurants that match their profiles. Profiles can be used to describe a wide variety of knowledge about people [1] and this knowledge can have many levels according to the depth of the user information.

User-profile based personalization is the process of making decisions based upon stored user profile information. We use Friend-Of-A-Friend (FOAF) [2] ontologies to store static user profiles. Gruber [3] defines ontology as an explicit specification of a conceptualization. Ontologies are used to represent information in a machine-readable fashion and to model specific domain information by defining objects, concepts and relationships. A FOAF profile is a machine-readable page, which is describing a person, her activities and her relations to other people and objects.

FOAF is also a consistent and common vocabulary to describe the demographical information. Most of the ontological information on the web generally use personal FOAF files. There are more than 13,120,000 people using FOAF to describe their personal profiles [4]. We propose a user profiling methodology with multi-metamodeling by using FOAF profiles. This methodology gives us the opportunity to create a complex and personal profile, which is a demand for an effective personalized system.

Policies are used to control access to resources. Policy management in Semantic Web is used to define declarative rules for accessing a resource and to allow users to interpret and comply with these rules. Integrating profiles into policies is improving personalization under the influence of policy management.

In order to qualify personalization successfully, we are integrating user-profile based personalization with policy management. In this paper, we propose a personalized system to help users choose an item from a large set of items of the same type by filtering this large set using policies according to their defined user profiles. We demonstrate our empirical approach for food domain to meet the requirements of health care domain.

Today, many people care about their health. Therefore, they pay extra attention to what they eat, what ingredients do their meals include and how many calories do their meals have. In order to satisfy this demand, we focus on the food domain ontology to perform the profile-based policy concepts in a personalized system. A personalized system that we propose in our case study can serve several objectives:

– nutrition information to preserve health,
– caution for people who have specific conditions, such as allergies or diabetes,
– ingredient information of meal courses,
– calorie control mechanism to restrict a person's daily calorie intake.

Health care is an information-rich domain and needs to be handled in care. User profiling in such a delicate topic requires more abstraction and variation than a regular FOAF file. This variation in profiles gives more efficiency in building policies to achieve rule-based personalization.

 ISBN: 978-1-61208-293-6

Our profile methodology has the capability to describe the health domain profiles. We can describe several profiles using these profiles, such as diabetic profile, diet profile, individualized ingredient profile and personal profiles where personalization needs a complex domain knowledge, such as health. Profiles are the key ingredients to tailor a profile-based policy management to restrict personalized rules.

This paper is organized as follows: Section 2 describes the user personalization and explains our profiling methodology. Section 3 expresses policy representation and policy ontology concepts. Also, it clarifies the connection between profile and policy ontologies. In Section 4, a case study is presented. Additionally, the food domain ontology concepts, profile and policy examples are demonstrated in this section. Related Work is given in Section 5. Finally, Section 6 concludes and gives the future direction of our work.

II. User Personalization

The profile of a person is an abstract description of the person's demographic, social and behavioral condition. A profile is a representation of a person's daily or permanent properties. In their life time, people change their minds and their situation also changes, due to different conditions. Thus, a static profile, which consists of these properties, has to adapt itself. Demographic properties like a person's occupation, age or school can not change rapidly due to their static nature. However, on a daily basis, a person can have different moods, different roles and different social choices. For example, a person, who is a doctor, can have many daily roles, such as being a mother, a parent or a child. She may want to use different preferences and different identifications for each of these roles. But, as she is a person, she also has demographical properties. So, for all these situations, we have developed a profiling methodology to represent a person's daily profiles by using demographic, social and behavioral properties. This methodology consists of a domain ontology, profile ontology and a metaprofile ontology to represent profile attributes and general descriptions.

A profile is the representation of demographic properties of a person. Let us state a profile as p, a user as u and a FOAF profile of a person as f. As we can call a FOAF profile as a base, we can define many profiles inside the base by using the $hasProfile$ property, $F(u) = hasProfile(P)$. These profiles are meaningful when P has properties, which are included by F. So, we can add data type, D, and object type properties, O, to this definition. $f \in F$, $p \in P$, $d_n \in D$, $o_n \in O$;

$$f(u) \quad = \left\{ \begin{array}{c} hasProfile(p_1), .., hasProfile(p_n), \\ d_1, d_2, d_3, ..., d_n, \\ o_1, o_2, ..., o_n \end{array} \right\} \quad (1)$$

In our ontology metamodel, as seen in Figure 1, we propose a new metamodel based on OMG's Meta-Object-Facility(MOF)[5]. In our metamodel, FOAF documents are our individuals. Inside FOAF documents, we use definitions

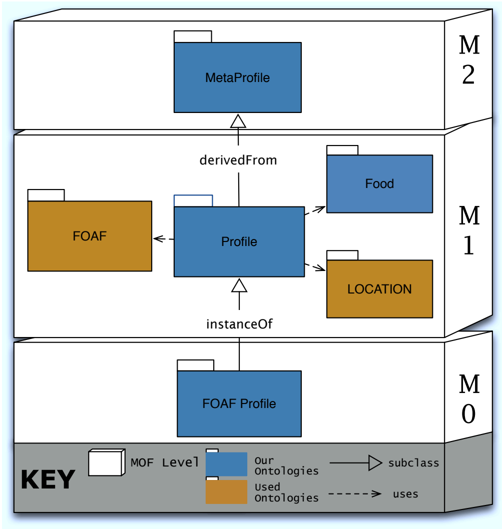

Fig. 1: Profile Methodology

and structures that are defined inside M1 level: Profile, FOAF definition, Location and Food ontologies. Profile ontology uses MetaProfile's ontological definitions. Metaprofile is independent from the domain. So, it consists of the basic properties to represent a social profile and its ancestors, a behavioral profile and its properties, and demographic properties of a person. These representations need to be designed inside a person's profile. Thus, we aim to define indicators. A profile indicator, pi , is the key property that defines a profile. $p \in P$, $pi \in PI$, $d_n \in D$, $o_n \in O$;

$$p = \{pi, d_1, ...d_n, o_1, ..., o_n\} \quad (2)$$

As an example, a diabetic profile is meaningful when a person has diabetes or regulations including diabetes inside the profile. Another example is a diet profile, which needs to include the definition of diet or maximum amount of calorie that a person should consume during the day. We develop three types of indicators. The first one is a point-based profile indicator, PB, which helps to define a basic profile property that has a singular value or individual.

$$PB = (d_1) \vee (o_1) \quad (3)$$

The second one is a range-based profile indicator, RB, which helps to define a range literal value with minimum and maximum values.

$$RB = (d_{1_{min}}, d_{1_{max}}) \quad (4)$$

The third one is a set-based profile indicator, SB, which includes a set of individuals. Profiles with set-based profile indicator could have individuals only described in this set-based profile indicator.

$$SB = \{o_1, o_2, o_3, o_4\} \quad (5)$$

Set-based profile indicator can be homogeneous or heterogeneous.

$$\begin{gathered} SB_{ho} = \{o_1, o_2, o_3\} : \\ \forall o \in SB_{ho} \mid o \in O_1 \\ SB_{he} = \{o_1, o_2, o_3\} : \\ \exists o_1 \in O_1 \land o_2, o_3 \in O_2 \land O_1 \neq O_2 \end{gathered} \quad (6)$$

This methodology gives us the ability to construct general profiles that can be explicitly defined in people's attributes. Thus, we can categorize people based on their profiles and represent these group profiles. Group profiles, G, are a generalization of a community of people based on their profile attributes. A group profile, $g \in G$, needs at least a profile identifier to describe itself. Also, later, this property will be the key to add user profiles into this group profile.

$$\begin{gathered} g = \{a_1, a_2, a_3, ..., a_n\} : \\ \exists a \in G \mid a \in PI \land PI \subset SB \cup RB \cup PB \end{gathered} \quad (7)$$

Group profiles can have these three profile identifier types: set-based, range-based and social. These identifiers are based on the key attribute(s) that they are constructed by. Moreover, a group profile may need two or more profile identifiers to describe itself. In this case, we define a set of profile identifiers.

$$g = a \land b : a, b \in PI \land PI \subset SB \cup RB \cup PB \quad (8)$$

Group profiles enable us to describe a policy for communities and persons based on their group or personal profiles.

III. Policy Representation

A policy is a declarative rule set that is based on constraints to control the behavior of entities. Policy rules define a declarative information on what an entity can do or cannot do. A policy consists of an entity, a constraint and a deontic object. An entity is the subject of the policy and a constraint defines the condition on a policy rule. A deontic object defines the concepts of permission, prohibition, obligation and dispensation. Permission is what an entity can do, prohibition is what an entity can not do, obligation is what an entity should do, and finally dispensation is what an entity need no longer do.

There are some general requirements that any policy representation should satisfy regardless of its field of applicability: expressiveness, simplicity, enforceability, scalability and analyzable [6]. In this work, by taking these requirements and ease of use criteria into consideration, we used Rei [7] policy language to represent policies. Rei policy language is composed of seven ontologies: ReiPolicy, ReiMetaPolicy, ReiEntity, ReiDeontic, ReiConstraint, ReiAnalysis, and ReiAction.

A. Policy Ontology

In a policy ontology, a policy is shown with a triple as *(S, O, A)*, in which *S* is subject, *O* is object and *A* is action. The subject indicates the entity that wants to access a resource, the object indicates the resource, which is going to be accessed, and the action indicates an operation, which the entity wants to achieve on a resource. The set of subjects, objects and actions is represented as $S = \{s_1, s_2, ..., s_i\}$, $O = \{o_1, o_2, ..., o_j\}$ and $A = \{a_1, a_2, ..., a_k\}$, respectively. The set of deontic objects, which are used to form policy rules is represented as $DO = \{Permission, Prohibition, Obligation, Dispensation\}$

B. Connecting Profile Ontology with Policy Ontology

In order to integrate profiling methodology into policy management, we substitute the set of subjects with the set of profiles, $P = \{p_1, p_2, ..., p_n\}$. When creating policies using Rei policy language, the subject of the policy is related with `entity:Variable` class. `entity:Variable` is a class of `ReiEntity` ontology. While creating a profile-based policy ontology, instances of an action's actors are now profile instances of the profile ontology. Thus, profile instances are used instead of the instances of `entity:Variable` class as the subject of the policy. As a result, policy subjects are comprised of semantically rich profile ontology.

The OWL representation of a `Vegetarian` profile defined in `entity:Variable` class is as follows:

```
<owl:Thing rdf:about="#Vegetarian">
  <rdf:type rdf:resource="&ReiEntity;Variable"/>
</owl:Thing>
```

The OWL representation of a `Vegetarian` profile defined in profile ontology is as follows:

```
<owl:Thing rdf:about="#Vegetarian">
  <rdf:type rdf:resource="&Profile.owl;Vegetarian"/>
</owl:Thing>
```

In a profile-based policy management [8], policy rules are assigned to profiles. Subjects are assigned with profiles and access rights to objects are given to profiles. Profile-based policy determines the ideal behaviors of the user using the user profile information. Figure 2 shows the policy components of the model.

A subject is represented by a profile and a profile is comprised of the profile ontology, which uses metaprofile ontology. An action and an object are based on domain ontology. Profile, action and object triple is used to form policy objects. Policy objects are used to create policy ontology, which is also based on the metapolicy ontology.

IV. Case Study

In this section, we present a case study for personalization by using policy management based on profiling methodology. The following conditions are some examples for personalization:

1. A diabetic person who is looking for a restaurant, she can be permitted or prohibited for her meal course preferences according to her health condition.

2. A professor who has an obligation for beverages, like not

ISBN: 978-1-61208-293-6

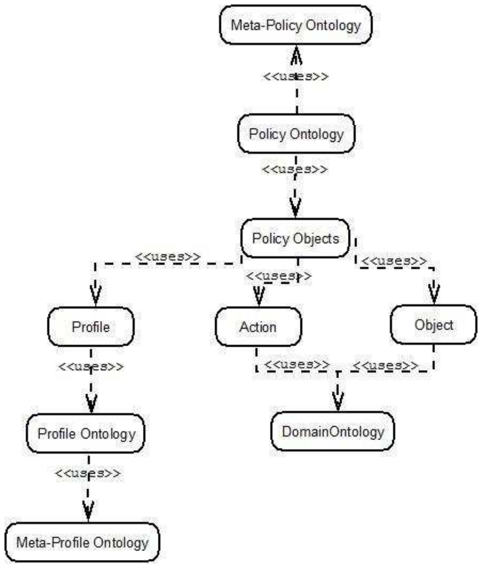

Fig. 2: Policy components of the model

drinking alcoholic beverages, when she is in a foreign country at a conference.

3. A person who is on a low calorie diet for a particular day may demand to be prohibited from choosing meal courses that have a high calorie content.

4. A peanut allergic person may demand to be prohibited from meal courses that include peanut.

5. A vegetarian person would like to know the meal courses that have vegetarian ingredients.

According to these examples given above, we build a food domain ontology. We use different sources to gather location and profile information. Unfortunately, we could not find any food ontology that combines all these ontologies together. So, we developed our own food ontology to overcome this problem. The next section explains in detail our domain ontology.

A. Domain Knowledge

As our case study needs a domain ontology to express the examples that are mentioned above, we build a food domain ontology. Figure 3 shows the class hierarchy of the food domain ontology.

Each item in a restaurant menu can be an individual of the food domain ontology. Each individual of appetizer, meal course, drink and dessert has an ingredient information, which has tied to `Ingredient` class with `hasIngredient` object property. Additionally, each individual has nutrition summary information defined with data properties. The nutrition summary values are taken from `fatsecret` [9] web site. Figure 4 shows object and data type properties of the food ontology, respectively.

Figure 5 shows an example of `Lasagna` individual of `MealCourse` class.

Besides the food domain ontology, a location ontology needs to be developed in order to provide a semantic

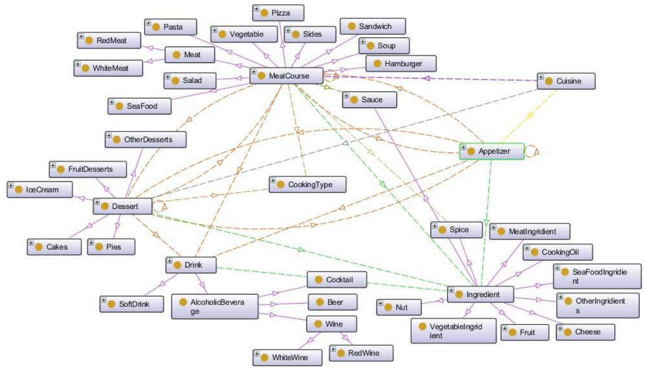

Fig. 3: Class hierarchy of the food domain ontology

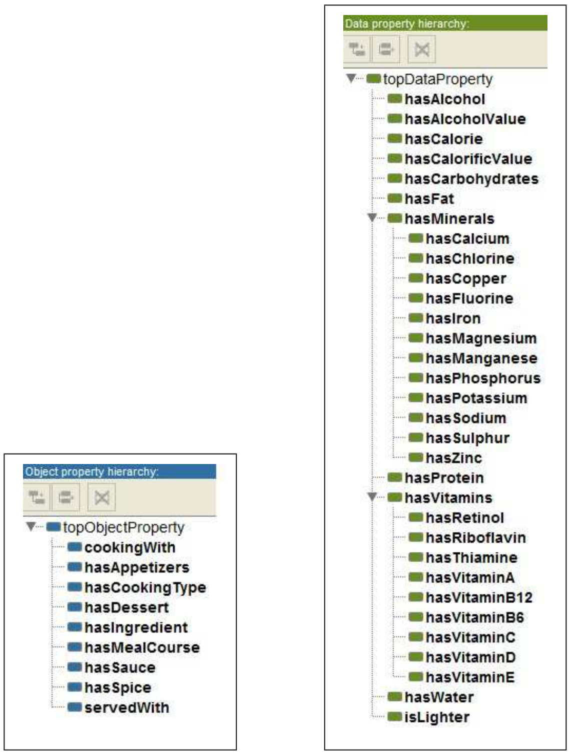

Fig. 4: Object and Data type properties of the food ontology

connection between a place and this place's food menu. For this purpose, we selected the schema.org's [10] ontology and adapted this ontology to our case study. Schema.org's ontology has a property to describe a menu item, but it is a general definition, which ranges to a `string` or the `Thing` class. Furthermore, a connection between the menu and the food domain ontology is a necessity. The relationship between the location ontology and the food domain ontology can be seen in Figure 6. This connection gives us the opportunity to build a profile-based policy description to handle the problems in our case study examples.

B. Profile Examples

The following examples define the profiles mentioned in the case study. These profiles are based on the profile methodology

ISBN: 978-1-61208-293-6

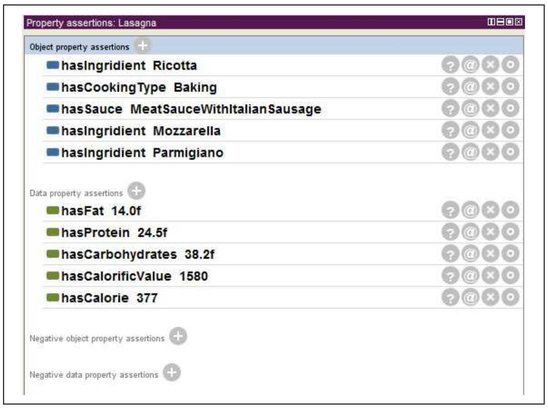

Fig. 5: `Lasagna` individual of `MealCourse` class

```
<!-- http://www.ifi.uzh.ch/ddis/people/bernstein/me -->
- <owl:NamedIndividual rdf:about="http://www.ifi.uzh.ch/ddis/people/bernstein/me">
    <rdf:type rdf:resource="http://efe.ege.edu.tr/~odo/Ontology/foaf_proje.owl/Person"/>
    <foaf:givenname>Abraham</foaf:givenname>
    <foaf:name>Abraham Bernstein</foaf:name>
    <foaf:title>Prof.</foaf:title>
    <foaf:mbox_sha1sum>8704ad77580618cb845036d3a15626d30fd828c3</foaf:mbox_sha1sum>
    <foaf:family_name>Bernstein</foaf:family_name>
    <foaf:nick>Avi</foaf:nick>
    <foaf:schoolHomepage rdf:resource="http://www.mit.edu"/>
    <Profile:hasOccupation rdf:resource="http://efe.ege.edu.tr/~odo/Ontology/Profile.owl#AcademicianOccupation"/>
    <foaf:hasProfiles rdf:resource="http://efe.ege.edu.tr/~odo/Ontology/Profile.owl#DietProfile"/>
    <foaf:hasProfiles rdf:resource="http://efe.ege.edu.tr/~odo/Ontology/Profile.owl#ProfessorTouristProfile"/>
    <foaf:workplaceHomepage rdf:resource="http://www.ifi.uzh.ch/"/>
    <foaf:workInfoHomepage rdf:resource="http://www.ifi.uzh.ch/ddis/"/>
    <foaf:homepage rdf:resource="http://www.ifi.uzh.ch/ddis/bernstein.html"/>
    <foaf:depiction rdf:resource="http://www.ifi.uzh.ch/ddis/uploads/pics/abernste_01.gif"/>
    <foaf:phone rdf:resource="tel:+41-44-635-4579"/>
  - <foaf:knows>
    - <rdf:Description>
        <rdf:type rdf:resource="http://efe.ege.edu.tr/~odo/Ontology/foaf_proje.owl/Person"/>
        <rdfs:seeAlso rdf:resource="http://seal.ifi.uzh.ch/fileadmin/User_Filemount/reif/rdf/reif_foaf.rdf"/>
      </rdf:Description>
    </foaf:knows>
```

Fig. 7: FOAF profile of Prof. Bernstein

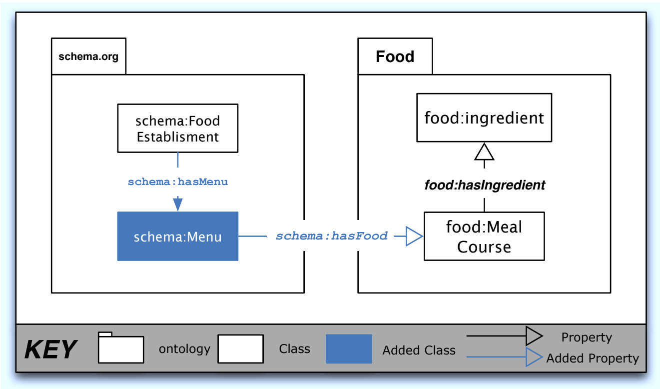

Fig. 6: Location and Food Ontology Relations

that we described in Section 2.

First profile has diabetes, so she possesses a `Diabetic` profile. Diabetic profile has a rule, which states that *a diabetic person can not drink an alcoholic beverage*.

$$\begin{gathered} diabetic = canDrink(n), \\ \forall n \in NonAlcoholicBeverage \cup \\ SetBasedProfileIndicator \cup Demographic \\ \rightarrow n \notin AlcoholicBeverage \end{gathered} \tag{9}$$

The second profile is that of a special professor profile who does not want to drink any alcoholic beverage when she is attending a conference in a foreign country.

$$\begin{gathered} professorAbroad = canDrink(n) \wedge visits(c), \\ \forall n : n \in NonAlcoholicBeverage \cup \\ SetBasedProfileIndicator \cup Demographic \wedge \\ \in Country \rightarrow n \notin AlcoholicBeverage \wedge \\ c \neq homeCountry(professorAbroad) \cup \\ SetBasedProfileIndicator \cup GeoDemographic \end{gathered} \tag{10}$$

The third profile is a behavioral profile, which describes the diet of a person. This profile has a range-based profile indicator to describe a `lowCalorie` profile, which has a minimum and maximum range in calorie calculation.

$$\begin{gathered} lowCalorieProfile = hasMood(lowCalorie) \\ lowCalorie = hasMaximum(maximumCalorie) \wedge \\ hasMinimum(minimumCalorie), \\ lowCalorie \in RangeBasedProfileIndicator \cup Mood \end{gathered} \tag{11}$$

The fourth profile is a `peanutAllergic` profile who has an allergic reaction to peanuts. This profile needs to be defined based on `hasIngredient` object property that defines ingredients of a meal course.

$$\begin{gathered} peanutAllergic = hasAllergic(p) \\ p = hasIngredient("peanut"), \\ \forall p : p \in PeanutAllergicFood \cup \\ SetBasedProfileIndicator \cup Demographic \end{gathered} \tag{12}$$

The last profile is a `Vegetarian` profile who only eats vegetarian food.

$$\begin{gathered} vegetarian = canEat(f) \\ f = hasIngredient(i), \forall i : i \in VegetarianFood \wedge \\ \forall f : f \in MealCourse \cup \\ SetBasedProfileIndicator \cup Demographic \end{gathered} \tag{13}$$

Figure 7 shows a professor who has a FOAF profile as mentioned in the second example of the case study. The Professor has many different profiles inside his FOAF profile. So, when he travels abroad for a conference and wants to have a light lunch according to his daily diet, there will be some restrictions on the lunch menu of the restaurant he choses. As seen from Figure 7, he has a `professorConference` profile and a `dietProfile`. These profiles have preference restrictions on alcoholic beverages and the total calorie limit for his lunch menu.

C. *Policy Examples*

This section demonstrates policy examples and their Semantic Web Rule Language (SWRL) [11] rules for the related case study examples. The following example shows a prohibition for a `Diabetic` profile. According to this rule, if a `Diabetic` profile chooses `ScillianScampi` from

`Appetizer`, she will be prohibited, because `cookingWith` property has `Chardonnay` individual, which has `true` value for its `hasAlcohol` property.

$$\begin{array}{c} Profile(?Diabetic) \wedge Appetizer(?x) \\ \wedge cookingWith(?x, ?y) \wedge hasAlcohol(?y, true) \implies \\ Prohibition\_ \\ orderScillianScampi(?Diabetic, ?x) \end{array} \tag{14}$$

In the second policy example, `professorAbroad` profile will be permitted when she chose `FruitPunch` that has `false` value for its boolean `hasAlcohol` data property.

$$\begin{array}{c} Profile(?professorAbroad) \wedge Drink(?x) \\ \wedge cookingWith(?x, ?y) \wedge hasAlcohol(?y, true) \\ \wedge hasAlcohol(?x, false) \implies \\ Permission\_ \\ orderFruitPunch(?professorAbroad, ?x) \end{array} \tag{15}$$

The following policy example gives an obligation to the `lowCalorie` profile according to the profile's daily calorie range for one meal course defined in the profile that has a range between minimum 400 and maximum 500. Thus, when she chose `Herb-GrilledSalmon`, if its `hasCalorie` property is less than the maximum calorie defined for `lowCalorie` profile, then she will be permitted to order `Herb-GrilledSalmon`, otherwise she will be prohibited.

$$\begin{array}{c} Profile(?lowCalorie) \\ \wedge hasCalorie(?Herb-GrilledSalmon, ?x) \\ \wedge hasMaximumCalorie(?lowCalorie, ?y) \\ \wedge isLessThan(?x, ?y) \implies \\ Obligation\_ \\ orderHerb-GrilledSalmon \\ (?lowCalorie, ?Herb-GrilledSalmon) \end{array} \tag{16}$$

A prohibition will be given to the `peanutAllergic` profile when she chose `PumpkinPie`, which has `Peanut` value for its `hasIngredient` property.

$$\begin{array}{c} Profile(?peanutAllergic) \\ \wedge hasIngredient(?x, ?Peanut) \\ \implies Prohibition\_ \\ orderPumpkinPie(?peanutAllergic, ?x) \end{array} \tag{17}$$

The last policy example prohibits the `Vegetarian` profile when she chose `VegetableLasagna`, because the course's `hasSauce` property's value is `MeatSauce`, which also has `ItalianSausage` value for its `hasIngredient` property. Figure 8 shows the OWL representation of this policy.

$$\begin{array}{c} Profile(?Vegetarian) \\ \wedge hasSauce(?x, ?y) \\ \wedge hasIngredient(?y, ?ItalianSausage) \\ \implies Prohibition\_ \\ orderVegetableLasagna(?Vegetarian, ?x) \end{array} \tag{18}$$

All these profile definitions and their integration with policies are described manually by the domain experts.

```
<owl:Thing rdf:about="#Prohibition__orderVegetableLasagna ">
  <rdf:type rdf:resource="&ReiDeontic;Prohibition"/>
    <ReiPolicy:desc rdf:datatype="&xsd;string"> A vegetarian profile is
        prohibited to order a meal that includes non-vegetarian food.
    </ReiPolicy:desc>
    <ReiDeontic:action rdf:resource="#order_VegetableLasagna"/>
    <ReiDeontic:actor rdf:resource="&Profile;Vegetarian"/>
    <ReiDeontic:constraint rdf:resource="IsVegetarianFood">
      <rdf:type rdf:resource="&ReiConstraint;And"/>
       <ReiConstraint:first rdf:resource="IsVegetarianFoodHasSauce">
         <ReiConstraint:subject rdf:resource="&FoodMenu;VegetarianLasagna"/>
         <ReiConstraint:predicate rdf:resource="&FoodMenu;hasSauce"/>
         <ReiConstraint:object rdf:resource="&FoodMenu;MeatSauce"/>
       </ReiConstraint:first>
       <ReiConstraint:second rdf:resource="IsMeatSauceHasItalianSausage"/>
         <ReiConstraint:subject rdf:resource="&FoodMenu;MeatSauce"/>
         <ReiConstraint:predicate rdf:resource="&FoodMenu;hasIngredient"/>
         <ReiConstraint:object rdf:resource="&FoodMenu;ItalianSausage"/>
       </ReiConstraint:second>
    </ReiDeontic:constraint>
</owl:Thing>
```

Fig. 8: OWL representation for `Vegetarian` profile policy

D. Practical Application

In our scenario, we used Prof. Bernstein's FOAF profile [12] as our FOAF Person. Firstly, we changed the `FOAF URI` in order to access the metalevel profile and the hometown property. As an example in our scenario, the professor uses his FOAF profile to order meals through the system. When he attends a conference in a foreign country, he chooses his `AcademicianTourist` profile, which has an restriction on alcoholic beverages. Besides, he is also on a diet. Thus, his diet profile must be active. His FOAF profile can be seen in Figure 7.

As he is an academician, he has an `AcademicianProfile`. When he attends a conference, he has `ProfessorTouristProfile` and also `DietProfile`. During the conference, he wants to dine in a good restaurant with his colleagues and his colleagues offer to go to a place named with `Winter Garden`. But first, he wants to check the menu of `Winter Garden` and uses his mobile application. After he loads his FOAF profile and policy definitions, his mobile application checks the restrictions connected to his profiles. The process of using restrictions with profiles needs an ontology parser and rule engine. This overall architecture can be seen in Figure 9.

The mobile application can query the SPARQL [13] end-point to get the restricted or granted menu items from the Ontology DB. As we have not developed a mobile application yet, our mock-up for mobile application interface can be seen in Figure 10.

In this interface, granted menu items are green, and restricted menu items are red and not selectable. The mobile application queries the Ontology DB by using the SPARQL. An example query is given in Figure 11.

 ISBN: 978-1-61208-293-6

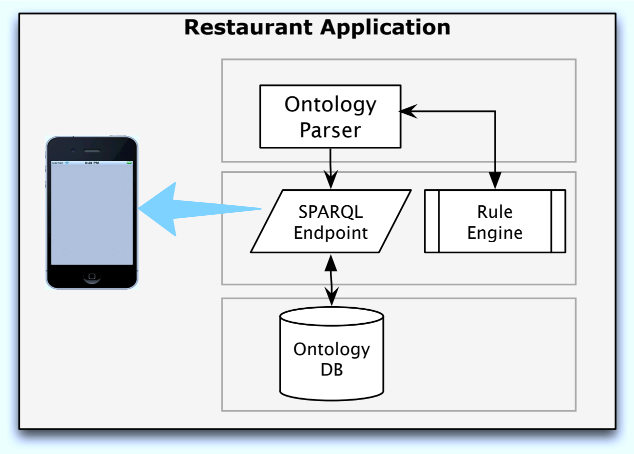

Fig. 9: General view of the architecture

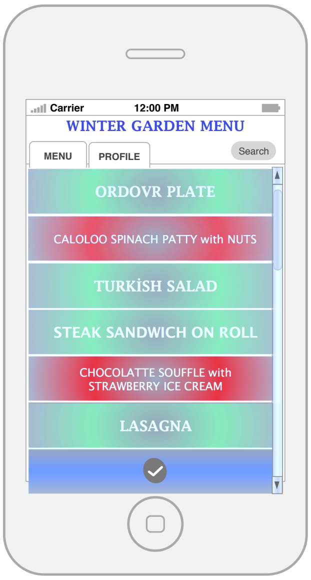

Fig. 10: Mobile application example

V. Related Work

Different domains need different profiling methodologies. The spectrum of profile description in the literature is wide. A profile is a storage that keeps the usable properties of a user and profiling is storing this user information. In [14], user profiles are used as a static storage document for basic information, calendar for daily meetings and so on. As the system becomes more complex, developing a profiling methodology also becomes a complex task. In order to provide services such as recommendation [15] and location based personalization [16], a profile can include different types of properties like online social network information, last visited web page and last clicked advertisement information.

User profiles can be used as a static document but it is more convenient as a dynamic and social projection, which saves a person's daily activities, social roles and preferences.

```
SELECT DISTINCT ?Person ?Profile ?prohibition ?ingredient
WHERE{?person rdf:type foaf:Person.
?person foaf:hasProfiles ?profile.
?restaurant rdf:type location:Restaurant.
?restaurant  menu:hasMenu ?menu.
?menu menu:hasFood ?MealCourse.
?MealCourse ?ObjectProperty ?ingredient.
?prohibition rdf:type reideontic:prohibition.
?prohibition reideontic:actor ?profile.
?prohibition reideontic:reiconstraint ?foodconstraint.
?foodconstraint rdf:type reiconstraint:And.
?foodconstraint ?numberOfConstraint ?firstconstraint.
?foodconstrain reiconstraint:predicate ?ObjectProperty.
?firstconstraint reiconstraint:object ?ingredient.
}
```

Fig. 11: SPARQL Example

In [17], a profiling methodology has been developed to store user preferences. The study presents a User Profile Ontology based on user characterization. This ontology provides an extensible user profile model that focuses on the modeling of dynamic and static user aspects. On the contrary to [17], preference handling needs a complex methodology to extract the possible preferences from domain knowledge and cover these preferences inside appropriate preference types as proposed in [18].

User profiling is also an asset for Quality of Service. In [19], user profiling is a solution for a group of workers who need to be authorized based on different authorization grants. Authorization based on group and individual user profile is a good solution. Besides, it is a strict solution and very hard to change or adapt to different domains. These profile definitions are convenient to be used in small data environments.

However, when data gets bigger, profiling becomes a tough problem. Likewise in [20], profiling is designed inside social networks and a general profile is constructed. As social networks emerge in time exponentially, profiling data emerges elsewhere, so that, describing policies with such a big data becomes a problem. In our work, we are proposing an abstraction to profiling methodology by using metamodel levels [5]. Thus, handling such a huge data becomes less problematic.

User profiles can be integrated into policy management mechanisms. There are various developed policy languages. KAoS [21], Rei [7] and Ponder [22] are the most common policy languages. KAoS is a DAML/OWL policy language. It is a collection of policy and domain management services for web services. KAoS distinguishes between authorizations and obligations. Rei is a policy specification language based on OWL-Lite. It allows users to express and represent the concepts of rights, prohibitions, obligations, and dispensations. Ponder is a declarative, object-oriented policy language for several types of management policies for distributed systems and also provides techniques for policy administration. Ponder has four basic policy types: authorizations, obligations, refrains and delegations. Tonti [6] gives a comparison of these three policy languages.

A framework that offers tools to specify adaptation policies

 ISBN: 978-1-61208-293-6

in the form of rules on profile attributes is presented in [23]. However, this is not sufficient to achieve the development of semantically rich applications. In [8], profile-based policy management is studied in order to make use of semantically rich policies in terms of the personalization scope.

An ontology-based solution to personalized clinical management is presented in [24]. The proposed ontology provides a solution for the personalized care challenges in home-based telemonitoring scenarios, and aims to model the tasks specified within a patient profile. Unlike this work, we use FOAF to specify profiles and integrate them with policies for personalization. A health care domain ontology is developed in [25] and access control policies are created based on this domain to manage patient's health records.

Since there are numerous ontology developers, there are also several food ontologies developed. A food-oriented ontology was developed in [26]. Additionally, Cantais [27] proposes a health care domain designed as a part of PIPS (Personalized Information Platform for Health and Life Services) project. However, both of these works do not fulfill the semantics of our scope and the relationship that we need to establish between location and food ontology. Thus, we built a new food menu ontology to achieve the semantically rich data representation.

VI. Conclusion and Future Work

Personalization should lead users to reach person specific information and customization by preventing unwanted content. User profiles are used to construct the personalized content by determining the user's choices and behaviors. User profiles can also be used as subjects for policy management. If profiles are well defined, they can give the exact and direct information about user's behaviors. We proposed an empirical approach to user profiling and built a profile-based policy management model to demonstrate a qualified personalized lifestyle system. We developed a new food ontology to be able to calculate calorie measures. Calorie measures of a menu can be calculated with this information and this makes the policy enforcement possible with the help of individuals profile selections. Profile, as means of a user's daily life role, is used to personalize policies to be able to define different policy rules for different daily situations. We showed policy and profile definitions of our case study examples. We also explained the policy rules and how we enforced these rules by using SPARQL. Our profiling methodology gives a richer user information to policy framework to provide a rule-based personalization. This information is useful to simulate real world problems into policy management.

As part of our future work, we will add new features to the food domain ontology and build a visual tool that allows users to create their profiles and make their meal choices from the restricted menu list. We are currently working on completing our mock-up based mobile application. Therefore, we will be able to gather user experience feedbacks of the methodology. A comparison between an user-adaptive and a non-adaptive system in the measurement of time that is spent for searching a specific item for a specific profile type will also be experimented. Additionally, we will automatize our food ontology's calorie extraction by using FatSecret's Platform API [28].

References

[1] E. Rich, Users are Individuals: Individualizing User Models, International Journal of Man-Machine Studies, Vol.18, pp. 199-214, 1983.

[2] D. Brickley, Friend-Of-A-Friend (FOAF), http://www.foaf-project.org, Last Access: July 3, 2013.

[3] T. R. Gruber, A Translation Approach to Portable Ontology Specification, Knowledge Acquisition 5(2), 1993, pp. 199-220.

[4] J. Goldbeck, M. Rothstein, Linking Social Networks on the Web with FOAF, Proceeding AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence, Vol 2, 2008, pp. 1138-1143.

[5] OMG's Meta Object Facility (MOF) 2012, http://www.omg.org/mof/, Last Access: July 3, 2013.

[6] G. Tonti, J. M. Bradshaw, R. Jeffers1, R. Montanari, N. Suri, and A. Uszok, Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder, International Semantic Web Conference 2003, Vol. 2870, 2003, pp. 419-437.

[7] L. Kagal, T. Finin, and A. Joshi, A Policy Language for A Pervasive Computing Environment, In IEEE 4th International Workshop on Policies for Distributed Systems and Networks, 2003, pp. 63-74.

[8] O. Can, O. Bursa, and M. O. Unalir, Personalizable Ontology-Based Access Control, Gazi University Journal of Science, 23(4), 2010, pp.465-474.

[9] FatSecret, Calories Nutrition All Things about food and diet, http://www.fatsecret.com/calories-nutrition, Last Access: July 3, 2013.

[10] Schema.org Schema.org with its RDFS, http://schama.rdfs.org, Last Access: July 3, 2013.

[11] W3C, SWRL: A Semantic Web Rule Language Combining OWL and RuleML W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/, Last Access: July 3, 2013.

[12] FOAF rdf, http://www.ifi.uzh.ch/ddis/people/bernstein/foaf.rdf, Last Access: July 3, 2013.

[13] W3C, SPARQL:SPARQL Query Language for RDF W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/, Last Access: July 3, 2013.

[14] H. Chen, F. Perich, D. Chakraborty, T. Finin, and A. Joshi, Intelligent agents meet semantic web in a smart meeting room, In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), Volume 2, Washington DC, USA, 2004, pp. 854-861,

[15] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, Ontological User Profiling in Recommender Systems, ACM Transactions on Information Systems, Vol. 22, 2004, pp. 54-88.

[16] K. W. -T. Leung, D. L. Lee, and Wang-Chien Lee, Personalized Web search with location preferences, IEEE 26th International Conference on Data Engineering (ICDE), 2010, pp. 701 - 712.

[17] K.L.Skillen, L.Chen, C.D. Nugent, M.P. Donnelly, W. Burns, and I. Solheim, Ontological User Profile Modeling for Context-Aware Application Personalization, UCAmI'12 Proceedings of the 6th International Conference on Ubiquitous Computing and Ambient Intelligence,, 2012, pp. 261-268,

[18] D. Tapucu, O. Can, O. Bursa, and M. O. Unalir, Metamodeling Approach to Preference Management in the Semantic Web, 4th Multidisciplinary Workshop on Advances in Preference Handling (M-PREF 2008) (In conjunction with AAAI 2008), Chicago, Illinois, July 13-14, 2008.

[19] J. C. Royer, R. Willrich, and M. Diaz, User Profile-Based Authorization Policies for Network QoS Services, Seventh IEEE International Symposium on Network Computing and Applications, 2008, pp. 68-75.

[20] Z. Iqbal and J. Noll, Toward User-Centric Privacy-Aware User Profile Ontology for Future Services, Third International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Athens, TBD, Greece, 2010, pp. 249 - 254.

[21] A. Uszok, J. M. Bradshaw, and R. Jeffers, KAoS: A Policy and Domain Services Framework for Grid Computing and Semantic Web Services, In Trust Management (iTrust 2004), Volume 2995 of Lecture Notes in Computer Science, 2004, pp. 16-26.

 ISBN: 978-1-61208-293-6

[22] N. Dulay, E. Lupu, M. Sloman, and N. Damianou, A Policy Deployment Model for The Ponder Language, In Proceedings of IEEE/IFIP International Symposium on Integrated Network Management (IM 2001), 2001, pp. 14-18.
[23] A. Agostini, C. Bettini, N. Cesa-Bianchi, D. Maggiorini, and D. Riboni, Integrated Profile and Policy Management for Mobile-Oriented Internet Services, Technical Report Firb-Web-Minds N. TR-WEBMINDS-04, 2003.
[24] N. Lasierra, A. Alesanco, S. Guillen, J. Garcia, A three stage ontology-driven solution to provide personalized care to chronic patients at home, Journal of Biomedical Informatics, 46, 2013, pp. 516529.
[25] J. Calvillo, I. Romana, L.M. Roa, Empowering citizens with access control mechanisms to their personal health resources, International Journal of Medical Informatics, 82, 2013, pp. 58-72
[26] C. Snae, M. Brckner, FOODS: A Food-Oriented Ontology-Driven System, 2nd IEEE International Conference on Digital Ecosystems and Technologies (DEST 2008), 2008, pp. 168-176.
[27] J. Cantais, D. Dominguez, V. Gigante, L. Laera, and V. Tamma, An Example of Food Ontology for Diabetes Control, Proceedings of the International Semantic Web Conference 2005 Workshop on Ontology Patterns for the Semantic Web, 2005.
[28] FatSecret Platform API 2013, http://platform.fatsecret.com/api/, Last Access: July 3, 2013.