Undefined 0 (0) 1
IOS Press

User Preferences in the Web of Data [1]

Editor(s): Fabian Abel, Delft University of Technology, The Netherlands; Laura Hollink, Delft University of Technology, The Netherlands; Geert-Jan Houben, Delft University of Technology, The Netherlands
Solicited review(s): Ricardo Kawase, Forschungszentrum L3S, Hannover, Germany; Knud Möller, DERI Galway, Ireland; Fabian Abel, Delft University of Technology, The Netherlands

Luis Polo [a], Iván Mínguez [a], Diego Berrueta [a], Carlos Ruiz [b] and José Manuel Gómez [b]
[a] *Fundación CTIC, C/ Ada Byron, 39, Edificio Centros Tecnológicos, 33203, Gijón (Spain)*
E-mail: {firstname}.{secondname}@fundacionctic.org
[b] *iSOCO S.A., Edificio Testa, Avenida del Partenón, 16-18, 28042, Madrid (Spain)*
E-mail: {cruiz,jmgomez}@isoco.com

Abstract. This article introduces a domain- and application-independent language for representing preferences as part of user profiles. It also describes the translation of statements of this language to RDF datasets using a new ontology named Framework for Ratings and Preferences (FRAP). The availability of this language and its RDF representation enable the effective exchange of user preferences across different applications in the web environment. The practical usage and limitations of this approach are also discussed in the article.

Keywords: Preferences, FRAP, FOAF, constraints, ontology, recommendation

1. Introduction

Preferences are an important part of user profiles for many applications. Despite a considerable number of proposed languages for representing user preferences, effective publication and reutilization of this information in the Web is far from being a reality nowadays. Users have to introduce their preferences repeatedly for each new application. Sometimes, systems are able to detect users' desires and preferences, but this information remains trapped in the applications. This article introduces an expressive and formal language with the aim of enabling open exchange of user preferences. To make it web-friendly, an RDF syntax of the language is also defined.

The proposal has a number of advantages with respect to the state of the art. Firstly, the preferences language is parametrized with specialized vocabularies and therefore the framework is independent of the domain (from retailing scenarios and multimedia content in mobile devices to social applications). It can be applied to any domain provided that a vocabulary or schema is available or can be built for that domain. Secondly, the language is also independent of the application that uses it. Preference interchange between applications becomes possible, given that they share the domain vocabulary. It is envisioned that this feature will be particularly useful in a web environment, for instance for portability of user profiles across social networks.

The RDF syntax of the language increases its reusability, as RDF is particularly suited to exchange descriptions in the Web. Moreover, it is possible to re-use existing RDFS vocabularies and OWL ontologies, as well as domain objects currently described as Linked Data (e.g. in LinkedMDB[1] or DBPedia[2]). Finally, by having preferences represented as RDF resources, their descriptions can be extended with rel-

[1]This work has been partially funded by Spanish CENIT Buscamedia project.

[1]`http://linkedmdb.org/`
[2]`http://dbpedia.org/`

evant metadata, such as contextual information (e.g., when or where a preference was first identified or applied).

This article is organized as follows. Firstly, the most relevant approaches for preferences representation are analyzed. Section 3 describes the main contribution of this article, namely the formal language for preferences and its translation to RDF datasets using the novel Framework for Ratings and Preferences (FRAP) vocabulary. Section 4 explains how preferences are connected to user profiles and how they can be embedded in FOAF profile documents. Afterwards, Section 5 discusses the use of the language in one of its target applications, recommendation systems. Finally, the last section concludes the article presenting some lines for future work.

2. Related Work

Preferences have been studied in many disciplines such as philosophy, economics and some fields of the AI related to decision-making processes. They play a major role in recommendation systems for e-business and social media sites. Preferences are statements of the form "Alice prefers A over B" or "Alice thinks A is better than B", captured by the logical relator ($\succ$) [9]. Basically they are user modal attributes which can be seen from different perspectives. On the one hand, preferences are ratings, i.e., quantitative measurements of the "appealingness" of a particular item to a user. Formally, a rating is defined [1] as a function μ that captures the satisfaction or appealingness of an item $i \in \mathcal{I}$ to user $u \in \mathcal{U}$ within a scale of numerical values, usually the real interval $[-1, 1]$, i.e.: $\mu : \mathcal{U} \times \mathcal{I} \rightarrow [-1, 1]$. In practice, other discrete scales are often used to measure users' opinions on items, like the five stars classification used by Amazon.

On the other hand, preferences are also viewed as qualitative descriptions of the desired attributes that items must ideally satisfy in order to be of interest for a user. In other words, preferences are conditions (implicitly or explicitly) expressed by the user about her interests and desires. This approach is adopted by the multi-attribute utility theory (MAUT [6]) for decision-making scenarios, and by some types of recommendation systems, such as knowledge-based ones [2,5]. Some languages have been proposed according to the latter paradigm. A language of preferences is defined in [13] for querying databases. This approach introduces an algebra and operators to represent the "whishes" of users. This formal, abstract language is then translated to extensions of SQL and XPath for relaxed queries. Preferences are interpreted as *soft constraints*. As it is not guaranteed that there will exist exact matches for all the conditions of a given query, preferences enable to look for the best possible matchmaking. In the same line, the authors of [18] define a preference-enabled query language as an extension to SPARQL. Soft constraints are captured by a new PREFERRING clause and a set of new operators for boolean preferences, AND, HIGHEST, LOWEST, and CASCADE. The results of the queries are arranged based on these preferences. Although there is a large overlap between expressiveness of this SPARQL-based language and the one defined in the next sections of this article, there is a significative difference between the approaches. The former is based on constraints, while the latter also considers ratings. Consequently, [18] defines an order relation among the results ($a \succ b$) and preferences, but cannot quantitatively express the relative "appealingness" (the ratio $\mu(a)/\mu(b)$).

In automatic configuration and planning tasks, preferences are also understood as soft constraints, where achieving a complete set of goals is not feasible and it is important to generate an optimal plan. In this case, preferences are desired goals that do not have to be necessarily satisfied. As well as the above scenarios, preferences apply to object (or component) attributes in order to evaluate and qualify the relative goodness of particular outcomes for a given problem [3]. An extension to the PDDL language has been proposed to express these soft constraints [8].

Domain-independent vocabularies have been defined to express ratings on RDF resources. The *Review* vocabulary[3] is a lightweight OWL ontology intended to capture ratings and reviews in RDF [11]. Similarly, the Weighted Interests vocabulary [4] applies weights to topics in order to represent user's preferences. A highlight of the latter is the fact that it captures the context by means of time and space elements regarding the applicability of preferences. However, these vocabularies cannot make complex preferences that combine different aspects of one resource. For instance, these vocabularies lack the expressiveness to represent preferences such as "I like Woody Allen movies shot in European cities".

[3] `http://vocab.org/review/terms.html`
[4] `http://smiy.sourceforge.net/wi/spec/weightedinterests.html`

Some efforts have been made to represent user preferences in particular domains and applications. The CC/PP vocabulary [16] is a W3C initiative for expressing device capabilities and user preferences to guide the adaptation of delivered content. CC/PP preferences are limited to desired attributes of device components to be considered in the client-server communication process. Further extensions have been proposed to CC/PP in the mobile domain, For instance, the profile module of the Mobile Ontology [20], a higher-level ontology for mobile communications, provides means for describing situation-based user preferences. The user can specify contextual conditions that facilitate service behavior customization.

In the multimedia domain, [19] proposes an OWL ontology to specify how to combine content filtering and browsing criteria with boolean operators. These filtering and search preferences (FASP) are designed to be applied to MPEG7 and MPEG21 specifications, which enable semantic multimedia content descriptions but lack powerful mechanisms to exploit them.

Other ontologies, focusing on ambient intelligence, directly introduce the concept of preference within the model. The SOUPA ontology [4] is used in the Context Broker Architecture (CoBRA) for pervasive context-aware systems. One of its modules provides some concepts to capture "mental states" of agents, such as preferences (`bdi:Desire`). Another OWL ontology for user preferences is STOUP [15], which allows expressing positive and negative preferences (likeness and dislikeness) of an agent regarding objects and some environmental conditions. It is worth mentioning that STOUP also permits to record a time stamp for each preference. Therefore, it is possible to analyze or detect modifications of people's thoughts.

A common hindrance of these domain- and application-specific ontologies is that they are hardly re-usable for other purposes. Moreover, it is up to the application to define the semantics of the preferences, as they are not sustained by a formal theory as the ones cited at the beginning of this section.

This article proposes a formal language that reconciles both notions of preferences, as ratings and as constraints, and introduces the user as a key element in its definition. To the best of our knowledge, this is the first proposed language that combines ratings and constraints. Moreover, contrary to most of the ontologies that have been enumerated above, the current proposal is domain-independent. Therefore, it can be mixed with any domain vocabulary and ontology to express user preferences over different resources: from items or products of a given marketplace, to activities or situations. The RDF syntax based on the FRAP ontology enables its usage in the web of data for preferences exchange across several applications and services. In a similar fashion, other formal languages also provide a RDF-based syntax, such as SPIN-SPARQL [14] and RIF in RDF [10].

Note that in the landscape of ontologies there exists also the Recommendations ontology[5] (RECO). However, the Recommendation ontology and FRAP completely diverge in purpose and aim. The former is devoted to represent ranked list of items given by a recommendation system, while FRAP captures user preferences.

3. Language for Preferences

The main contributions of this article are the formalization of a user-oriented preference language, as well as a lightweight OWL vocabulary that permits capturing this language expressions as RDF graphs. An injective mapping function (π) from language expressions to RDF graphs is also defined.

3.1. Definition of the Preferences Language

Given a domain vocabulary $\mathcal{V} = \langle C, R, I\rangle$ consisting of the sets of concepts C, binary roles R and constants I, as well as a set of infinite variables $Var = \{x_1, x_2, \dots\}$ disjoint of C, R and I, the language of preferences $\mathcal{L}$ is defined over a domain vocabulary $\mathcal{V}$, denoted as $\mathcal{L}(\mathcal{V})$. An expression of $\mathcal{L}(\mathcal{V})$ is a set of preferences over constraints, as defined below.

Definition 1 (Constraint)**.** *Constraints are conditions about desired or preferred attributes of the resources. A constraint ranges over a set of individuals represented by means of a variable x, which is called the **main variable**. A constraint $\sigma^x \in \mathcal{S}$, being $x \in Var$ the main variable, is either:*

1. *$c(x)$, where $c \in C$.*
2. *$r(x, y)$, where $r \in R$ and $y \in (I \cup Var)$.*
3. *$r(x, \oplus v)$, where $v \in I$ and $\oplus \in \{=, \neq, <, \leq, >, \geq, substring\}$ and is a boolean operator.*
4. *A conjunction of constraints $(\sigma_1^x \wedge \sigma_2^x)$, where σ_1 and σ_2 share the same main variable x.*

[5] `http://smiy.sourceforge.net/rec/spec/recommendationontology.html`

5. *A disjunction of constraints* $(\sigma_1^x \vee \sigma_2^x)$ *where* σ_1 *and* σ_2 *share the same main variable* x.
6. *A composition of constraints* $(r(x,y) \circ \sigma'^y)$, *where the composed constraint* σ' *has as main variable the secondary variable of the role* r*, that is* y.

The semantics of constraints is defined using a first order semantics. An *interpretation* of $\mathcal{L}(\mathcal{V})$ is a tuple $\mathcal{I} = \langle U, \cdot^{\mathcal{I}} \rangle$, where U is a non-empty set (called the *domain* of $\mathcal{I}$) and $\cdot^{\mathcal{I}}$ is a mapping function, which assigns to every $c \in C$ a subset $c^{\mathcal{I}} \subseteq U$, to every $r \in R$ a relation $r^{\mathcal{I}} \subseteq U \times U$, and to every constant $a \in I$, an element $a \in U$. A variable assignment B is a mapping that assigns an element $x^B \in U$ to every variable symbol $x \in Var$.

An interpretation $\mathcal{I}$ *satisfies* a constraint σ^x, given a variable assignment B, denoted by $(\mathcal{I}, B) \models \sigma^x$, if:

1. $(\mathcal{I}, B) \models c(x)$ iff $x^B \in c^{\mathcal{I}}$;
2. $(\mathcal{I}, B) \models r(x,y)$ iff $(x^B, y^B) \in r^{\mathcal{I}}$;
3. $(\mathcal{I}, B) \models r(x, \oplus v)$ iff $\exists z \in Var \mid (x^B, z^B) \in r^{\mathcal{I}}$ and $(z^B \oplus v^{\mathcal{I}})$;
4. $(\mathcal{I}, B) \models (\sigma_1 \wedge \sigma_2)$ iff $(\mathcal{I}, B) \models \sigma_1$ and $(\mathcal{I}, B) \models \sigma_2$;
5. $(\mathcal{I}, B) \models (\sigma_1 \vee \sigma_2)$ iff $(\mathcal{I}, B) \models \sigma_1$ or $(\mathcal{I}, B) \models \sigma_2$;
6. $(\mathcal{I}, B) \models (r(x,y) \circ \sigma'^y)$ iff $(\mathcal{I}, B) \models r(x,y)$ and $(\mathcal{I}, B) \models \sigma'^y$.

Definition 2 (Preference)**.** *A preference* $\rho \in \mathcal{P}$ *is an ordered pair* $(u, \sigma) \in \mathcal{U} \times \mathcal{S}$*, where* $\mathcal{U}$ *is the set of users and* $\mathcal{S}$ *is the set of constraints that can be generated by* $\mathcal{L}(\mathcal{V})$.

The definition of the utility function μ given in Section 2 is extended in order to add constraints to its domain. Therefore, $\mu : \mathcal{U} \times (I \cup \mathcal{S}) \to [-1, 1]$. This extension makes it possible to combine constraints and ratings for expressing quantitative measurements of the qualitative descriptions of the desired attributes of resources. To put it simply: constraints can be rated. The utility function introduces (as usual in utility theory) a total order in $\mathcal{P}$ by means of relators "$\succ$" and "$\sim$". Given $\rho_i, \rho_j \in \mathcal{P}$, a user u prefers ρ_i over ρ_j (denoted by $\rho_i \succ \rho_j$) iff $\mu(\rho_i) > \mu(\rho_j)$; and the user u has equal ratings for ρ_i and ρ_j ($\rho_i \sim \rho_j$) iff $\mu(\rho_i) = \mu(\rho_j)$.

The definitions above conciliate the alternative interpretations of preferences as ratings and as constraints:

- On the one hand, the language is able to capture sentences like "Alice has a strong preference for *Pink Floyd*", represented by the formula: $\mu(\text{Alice}, \text{Pink Floyd}) = 0.9$.
- On the other hand, it is also possible to capture that "Alice has a preference for *Pink Floyd's* early albums" as the pair: $\rho = (\text{Alice}, \sigma_1^x)$, where $\sigma_1^x = (\text{Album}(x) \wedge \text{Author}(x, \text{Pink Floyd}) \wedge \text{Released}(x, < 1980))$.
- Finally, the language permits combinations of both approaches. For instance, it is possible to express that the latter preference is weak: $\mu(\text{Alice}, \sigma^x) = 0.2$.

Notice that there is a semantic difference between rating a preference ($\mu(\text{Alice}, \sigma_1^x) = 0.2$) and rating an object ($\mu(\text{Alice}, \text{Pink Floyd}) = 0.9$). The latter is more intuitive and better matches the notion of "rating", while the former is useful to capture the mindset of a user. Continuing the example, Alice may assign different priorities or relative importance to her preferences. She does not attribute much value to the preference σ^x (*Pink Floyd* early albums), but she is much more interested in live albums, regardless of the year, i.e., $\mu(\text{Alice}, \sigma_2^x) = 0.7$, where σ_2^x is the latter preference.

One of the limitations of this language is that it is not possible to capture "exclusive" preferences, such as in "Alice only likes *Pink Floyd* music [and nothing else]". Even when the utility value is maximum $(+1)$, exclusivity is not assured due to the open world assumption. Neither is the problem solved by assigning a negative utility to the complement. In practice, this can be solved by adopting a categorization of preferences to express preference exclusivity as filters that are orthogonal to the utility function, as discussed in Section 5. In short, this kind of statements are transformed into hard constraints by some recommenders.

3.2. *RDF translation and the FRAP ontology*

The FRAP ontology[6] defines the vocabulary for representing formulas in the proposed language as RDF graphs. Notice that RDF is used as a convenient interchange format, and not as a translation of the formal language to semantically-equivalent RDF graphs. FRAP is a lightweight OWL vocabulary that provides domain-independent means to describe user profiles in a coherent and context-aware way. The FRAP namespace is `http://purl.org/frap#`, although for

[6] `http://purl.org/frap`

σ^x	$\pi(\sigma^x)$
$c(x)$	$\{l(x)$ `rdf:type` $l(c)\}$
$r(x,y)$	$\{l(x)\ l(r)\ l(y)\}$
$r(x, \oplus v)$	$\{l(x)$ `filter [ rdf:type Filter;` `operator` $l(\oplus)$; $l(r)\ l(v)]\}$
$\sigma_1^x \wedge \sigma_2^x$	$\pi(\sigma_1^x) \cup \pi(\sigma_2^x)$
$\sigma_1^x \vee \sigma_2^x$	$\{l(x)$ `union` $l(z)\} \cup \pi({}_{x\to z}\sigma_1^x) \cup$ $\cup\ \{l(x)$ `union` $l(w)\} \cup \pi({}_{x\to w}\sigma_2^x)$ where $z, w \in Var$ are fresh variables
$r(x,y) \circ \sigma'^y$	$\{l(x)\ l(r)\ l(y)\} \cup \pi(\sigma'^y)$

Table 1

Transformation function (π) for constraint expressions.

$expr$	$\pi(expr)$
$\rho \in \mathcal{P}$	$\{l(\rho)$ `rdf:type Preference`$\}$
$u \in \mathcal{U}$	$\{l(u)$ `rdf:type User`$\}$
$\sigma^x \in \mathcal{S}$	$\{l(x)$ `rdf:type Pattern`$\}$
$\rho = (u, \sigma^x) \in \mathcal{P}$	$\{l(u)$ `holds` $l(\rho)$. $l(\rho)$ `rdf:type Preference` ; `about` $l(x)\}$
$\mu(u, \sigma^x) = r$, such that $\rho = (u, \sigma^x)$	$\{[$ `rdf:type Rating` ; `rates` $l(\rho)$; `utility` $l(r)$; `assignedBy` $l(u)]\}$
$\mu(u, i) = r$, where $i \in \mathcal{I}$	$\{[$ `rdf:type Rating` ; `rates` $l(i)$; `utility` $l(r)$; `assignedBy` $l(u)]\}$

Table 2

Transformation function π for preferences and ratings.

concision, in the following it is assumed to be the default namespace.

RDF distinguishes three sets of disjoint syntactic entities. Let U denote the set of *URI references* and Bl the set of *blank nodes*, i.e., variables. Let L be the set of *literals*, i.e., data values such as floats or strings. An RDF graph G is a set of *triples*, where the tuple $\{s\ p\ o\} \in (U \cup Bl) \times U \times (U \cup Bl \cup L)$ is called an RDF triple. In the tuple, s is the subject, p is the predicate and o is the object.

The π function transforms expressions in the preferences language to sets of RDF triples. Tables 1 and 2 define the function for constraint expressions and user preferences, respectively. Notice that Table 2 contains an additional entry for constraints factorizing the typing of constraint expressions in Table 1. N3 syntax for RDF is used in the right column.

An auxiliary labeling function $l : (Var \cup C \cup R \cup I \cup \oplus \cup \mathcal{P} \cup \mathcal{U}) \to (U \cup Bl \cup L)$ produces RDF entities for each symbol of the preferences language and the domain vocabulary. For operators ($\oplus$), the labeling function produces URIs from the XPath specification in order to ensure interoperability[7].

Given $x, y \in Var$, the substitution operator ${}_{x\to z}\sigma$ replaces all occurrences of x by z in the constraint σ.

FRAP introduces the concept `Preference` which reifies the relation between a user profile and a constraint. This relation is realized in the graph by means of the property `holds`. On the other hand, an auxiliary concept for transformations, called `Pattern`, is introduced in the ontology in order to capture the constraints σ^x of the preferences language.

Regarding the utility function, which output is a ternary relation $\langle u, i, r\rangle$, the ontology introduces another new concept `Rating` which reifies this tuple. Three specific properties capture the relationships between the rating and the user u (`assignedBy`), the item i (`rates`) and the utility value r (`utiliy`). Notice that the property `rates` has an open range, therefore ratings can be applied to anything, including preferences themselves.

```
ex:Alice a :User ;
    :holds _:p1 .

_:p1 a :Preference ;
    :about [ a :Pattern ;
       a db-owl:Album ;
       db-owl:artist db:Pink_Floyd ;
       :filter [ a :Filter ;
         :operator op:date-less-than ;
         db-owl:releaseDate
            "1980-01-01"^^xsd:date ] ] .

[  a :Rating ;
   :assignedBy ex:Alice ;
   :rates  _:p1 ;
   :utility  "0.2"^^xsd:float ] .
```

Fig. 1. RDF graph in N3 syntax representing "Alice has a weak preference for *Pink Floyd* early albums".

Figure 1 shows π translation of a preference into a FRAP representation. It is worth reminding that the vocabulary mixes with domain-specific properties and concepts from existing ontologies. In this example, DBpedia vocabularies are combined with the `Pattern` class in order to describe desired and preferred attributes of musical resources. More precisely, the band *Pink Floyd* is identified by its DBpedia URI (`db:Pink_Floyd`) and an album release date is identified by the `db-owl:releaseDate` property.

[7]The prefix `op` is used in the examples to refer to the XPath namespace: `http://www.w3.org/TR/xpath-functions/`

This mechanism allows to compositionally build complex constraints in RDF, where each `Pattern` captures the main variable x of a constraint σ^x. Notice that all conjoined constraints, which share the same variable (the "album" in the example), are reduced to just one `Pattern` instance in the RDF graph. Although this example requires only one `Pattern`, more complex constraints using disjunctions and compositions would require multiple `Pattern` instances.

4. Preferences in the Web of Data

FRAP is a user-oriented vocabulary to represent preferences in the Web, where users are instances of the `User` class. In practice, there are previously existent and widely adopted vocabularies to represent user profiles, such as FOAF. In particular, FRAP users can be interpreted as instances of `foaf:Person` and `foaf:Group`, which define individuals and collectives respectively. The combination of both vocabularies allows then expressing sentences like "Alice loves wines from Rioja country in Spain" and "the friends of Alice do not like dark beer too much".

One of the applications of combining FOAF and FRAP is take advantage of preferences to classify individuals into groups defined by intension. For instance, it is possible to define a cluster such as "people that love hard rock music, but not Metallica". This clusterization can go beyond the borders of social networks silos, allowing to query the global web data space for people with certain preferences. Similarity metrics can be defined between pairs of users based on the set of preferences they share.

Together with the WebID proposal, FRAP permits the portability of preferences attached to the user web identity. From an application point of view, the preferences of a new user arriving to a web site can be retrieved and used to offer tailored contents, such as movies in the case of a multimedia on-line database, or products in an e-commerce portal. To this end, preferences must be added to the personal FOAF profile, as shown in Figure 2. Notice that the ability to openly exchange preferences must be counterbalanced with adequate access control policies, which actually restrict the visibility of subsets of the user profile. Unfortunately, at the present moment, the mechanisms to enforce these policies are not widespread.

The question still remains about the origin or generation of FRAP preferences, specially the complex ones. Although it is conceivable that a form may be

```
ex:Alice a foaf:Person ;
         foaf:name "Alice" ;
         :holds _:p1 .

[ a rsa:RSAPublicKey ;
         rsa:modulus "..." ;
         cert:identity ex:Alice ] .
```

Fig. 2. FOAF profile for Alice with preferences and WebID public information.

used to capture users' preferences through a UI (a simplistic example is shown in Figure 3), it is also possible that FRAP preferences can be automatically inferred from users' behavior, or translated from another preference language.

5. Preferences and Recommendation

Preferences can be exploited by a number of applications in multiple personalization scenarios. In combination with recommendation agents, preferences become a powerful tool, since they help users to discover items that are likely to be of their interest. For instance, preferences on restaurants that are typically used by rating portals (see Figure 3) are easily captured by statements using the FRAP vocabulary (Figure 4).

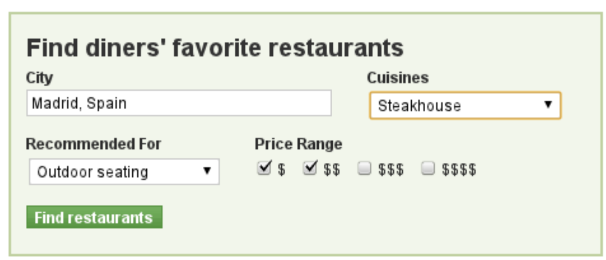

Fig. 3. Restaurant search form from a typical ratings portal.

Recommenders help users by suggesting the items that better fit their needs. The recommendation challenge deals with estimating ratings for items not previously rated by the user. Once ratings are estimated, a ranking of items is made up to present the user with the items which received the highest score. Recommendation algorithms try to choose, for each user $c \in \mathcal{U}$, such item $i \in I$ that maximizes the user's utility $\mu(c, i)$.

Preferences expressed in FRAP may be the input of several kind of recommendation systems. A running example illustrates how the following set of FRAP

```
ex:Alice a :User ;
    :holds _:p1 .

[  a :Rating ;
   :assignedBy ex:Alice ;
   :rates  _:p1 ;
   :utility  "0.8"^^xsd:float ] .

_:p1 a :Preference ;
    :about [ a :Pattern ;
       a ex:SteakHouse ;
       ex:feature ex:outdoor_seating
       :filter [ a :Filter ;
         :operator op:less-than ;
         ex:price
            "100"^^xsd:float ] ] .
```

Fig. 4. A preference on restaurants as an RDF graph in N3 syntax.

preferences can be translated in order to feed different recommenders.

$$\sigma_1^x = \text{Album}(x) \wedge \text{Author}(x, \text{Pink Floyd}) \wedge \wedge \text{Released}(x, < 1980) \quad (1a)$$

$$\sigma_2^x = \text{LiveAlbum}(x) \wedge \text{Author}(x, \text{Pink Floyd}) \quad (1b)$$

$$\mu(Alice, \text{Pink Floyd}) = 0.9 \quad (1c)$$

$$\mu(Alice, \sigma_1^x) = 0.2 \quad (1d)$$

$$\mu(Alice, \sigma_2^x) = 0.7 \quad (1e)$$

On the one hand, collaborative filtering systems are primarily based on the concept of "preference as a rating" [17]. These recommenders employ statistical techniques to find a set of similar users, known as neighbors, who have rated the same items, and to calculate the final utility of the items. Ratings are usually represented as a two-dimensional matrix (user $\times$ item). For instance, Table 3 has rated the artists based on preference $1c$. It is tempting to derive also the rating matrix in Table 4 for the extension of preference $1e$, in this case, the albums. However, this is not correct. The rating in $1e$ applies to the preference itself, and not to the items (albums) that satisfy the conditions of the formula. Otherwise, it would lead to inconsistencies, e.g., Pink Floyd's 1969 live album *Ummagumma* fulfills both constraints. The matrix cannot simultaneously capture two ratings for the same item.

Artists	**Pink Floyd**	**The Who**
Alice	0.9	...
Bob	...	...

Table 3

Rating matrix derived from FRAP preferences.

Albums	**The Wall**	**Pulse**
Alice	0.7	0.7
Bob	...	...

Table 4

Album rating matrix that does *not* follow from σ_2^x.

On the other hand, knowledge-based recommenders are based on the concept of "preference as a constraint" [7]. These techniques typically describe the attributes of the items to be recommended and compare these attributes with the preferences of the users. These recommenders rely on various inference mechanisms, such as Description Logics reasoning, rule engines, or SPARQL and SQL query answering. Moreover, these recommenders often distinguish between two kinds of statements:

1. Hard constraints, which are matchmaking conditions that items *must* fulfill in order to be considered utile for the user (e.g.: "Alice only likes *Pink Floyd* music") . These conditions act as boolean filters. Note that the "mandatory" aspect of these preferences is out of the expressiviness of the language defined in this article, although this is not a limitation but an exclusive characteristic of some recommenders.
2. Preferences (or soft constraints), which are matchmaking conditions that items *should* fulfill and impact in their final utility or ranking.

The proposed preference language can be translated to specific languages of different recommendation systems. For instance, [12] introduces soft constraints into SPARQL queries. The query in Figure 5 combines σ_1^x and σ_2^x as two independent Pareto preferences, each one with multiple cascading lexicographic preferences. In this language it is not possible to indicate that σ_2^x is more important for Alice than σ_1^x.

In addition, TeRRAS[8] is a recommendation system based on matchmaking. It is available as a web service, empowering third-parties to take advantage of it for developing personalized applications. TeRRAS provides two alternative implementations of the semantics of

[8] http://terras.sourceforge.net/

```
SELECT ?album
WHERE {
  ?album a db-owl:Album .
  ?album db-owl:artist ?artist .
  ?album db-owl:releaseDate ?date .
  ?album dct:subject ? kind
}
PREFERRING
  ?artist = db-owl:Pink_Floyd
  CASCADE
  ?date < 1980
AND
  ?artist = db-owl:Pink_Floyd
  CASCADE
  ?kind = cat:Live_albums
```

Fig. 5. Preferences expressed in PPSS language.

```
OBLIGATORY   // hard constraints
  rdf:type db-owl:Album .
OPTIONAL     // soft constraints
 db-owl:artist some {db:Pink_Floyd} AND
  db-owl:releaseDate some date(<1980),
  0.2.
 db-owl:artist some {db:Pink_Floyd} AND
  dct:subject some {cat:Live_albums},
  0.7.
```

Fig. 6. Preferences expressed in QIL language.

the language, namely the transformation from FRAP to OWL class expressions and SPARQL queries. Both share the same input, a specific language called QIL. Figure 6 illustrates the translation of rated preferences σ_1^x and σ_2^x to QIL. As in the previous example, the condition Album(x) has been converted into a hard constraint, due to an application-specific requirement (an assumption is made that these recommenders focus on music albums). In this case, QIL is able to deal with the fact that Alice is really interested in live albums rather than pre-1980 Pink Floyd works.

Finally, the integration in a single format of both preferences-as-constraints and preferences-as-ratings opens the door to hybrid recommendation systems. Combining the strengths of complementary recommendation techniques can help to overcome their individual limitations. The same input (FRAP) can be used for different algorithms to fit the particular requirements of each scenario.

6. Conclusions and Future Work

This paper presents a novel approach to represent and combine complementary views of preferences. It also suggests how they can be exchanged in the Web as RDF graphs. The authors plan to enhance the proposed formal language in ways that are already supported by the companion FRAP ontology. These aspects include the ability to group preferences for the representation of requests for knowledge-based recommendation systems (usually known as "demands" in matchmaking literature).

The semantics of the language can be extended in many directions. For instance, it would be desirable to capture preferences with order functions (such as "the lowest the price, the better") and fuzzy operators ("the temperature must be around 21 degrees").

In the introduction it was suggested that having preferences as RDF resources turns them into suitable candidates to be described with metadata properties such as `dct:created`, `dct:temporal` and `dct:spatial`. FRAP shares this feature with Weighted Interests Ontology, which opens the door to augment preferences description with contextual information and fine-grained profiles. Future work on context description will make it possible to capture statements such as "Alice does not like to drink wine at home" or "Alice prefers radio news when she drives". Furthermore, the dynamic nature of people's thoughts, including preferences, is also challenging. Consider for instance "Alice used to like Mickey Mouse cartoons when she was young, but nowadays she prefers action movies".

Moreover, the preferences associated to an individual (either directly or through a profile) can be contradictory or even inconsistent. The study of these situations, which are likely to happen when partial user profiles are merged, are also part of the authors' work roadmap.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] A. Borgida, T. Walsh, and H. Hirsh. Towards Measuring Similarity in Description Logics. In Ian Horrocks, Ulrike Sattler, and Frank Wolter, editors, *Proceedings of the 2005 International Workshop on Description Logics*, volume 147 of *CEUR Workshop Proceedings*, Edinburgh, Scotland, UK,

2005. URL http://www.ceur-ws.org/Vol-147/25-BorgidaEtAl.pdf.

[3] C. Boutilier, R. Brafman, C. Geib, and D. Poole. A Constraint-based Approach to Preference Elicitation and Decision Making. In *AAAI Spring Symposium on Qualitative Decision Theory*, pages 19–28, Stanford, CA, 1997.

[4] H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. MOBIQUITOUS'04, pages 258–267, Cambridge, MA, USA, 2004. IEEE Computer Society.

[5] T. Di Noia, E. Sciascio, and F. Donini. Semantic Matchmaking as Non-Monotonic Reasoning: A Description Logic Approach. *Journal of Artificial Intelligence Research*, 29(1):269–307, 2007.

[6] J.S. Dyer. MAUT – multiattribute utility theory. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 265–285. Springer Verlag, Boston, Dordrecht, London, 2005.

[7] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Developing constraint-based recommenders. In F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors, *Recommender Systems Handbook*, pages 187–215. Springer, New York, Dordrecht, London, Heildelberg, 2011.

[8] A. Gerevini and D. Long. Preferences and Soft Constraints in PDDL3. In A. Gerevini and D. Long, editors, *Proceedings of the ICAPS Workshop on Preferences and Soft Constraints in Planning*, pages 46–53, The English Lake District, Cumbria, UK, 2006.

[9] S. Hansson and T. Grüne-Yanoff. Preferences. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab, Stanford University, fall 2011 edition, 2011. URL http://plato.stanford.edu/archives/spr2009/entries/preferences/.

[10] S. Hawke and A. Polleres. RIF In RDF. Working Group Note, W3C, May, 2011. URL http://www.w3.org/TR/rif-in-rdf/.

[11] T. Heath and E. Motta. Revyu: Linking Reviews and ratings into the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):266–273, 2008.

[12] P. Kärger, D. Olmedilla, F. Abel, E. Herder, and W. Siberski. What Do You Prefer? Using Preferences to Enhance Learning Technology. *IEEE Transactions on Learning Technologies*, 1 (1):20–33, 2008.

[13] W. Kießling. Foundations of preferences in database systems. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 311–322, Hong Kong, China, 2002. VLDB Endowment.

[14] H. Knublauch. SPIN - SPARQL Syntax. W3C Member Submission, February, 2011. URL http://www.w3.org/Submission/spin-sparql/.

[15] K.A.P. Ngoc, Y.K. Lee, and S.Y. Lee. OWL-based User Preference and Behavior Routine Ontology for Ubiquitous System. In *Proceedings of the 2005 OTM Confederated international conference on On the Move to Meaningful Internet Systems: CoopIS, COA, and ODBASE - Volume Part II*, OTM'05, pages 1615–1622, Agia Napa, Cyprus, 2005. Springer-Verlag.

[16] H. Ohto, M. Butler, C. Woodrow, F. Reynolds, G. Klyne, L. Tran, and J. Hjelm. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation, W3C, January 2004. URL http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[17] B Sarwar, G Karypis, J Konstan, and J Reidl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, Hong Kong, China, 2001. ACM.

[18] W. Siberski, J. Pan, and U. Thaden. Querying the Semantic Web with Preferences. In *Proceedings of the 5th International Semantic Web Conference (ISWC2006)*, ISWC'06, pages 612–624, Athens, GA, 2006. Springer-Verlag.

[19] C. Tsinaraki and S. Christodoulakis. A Multimedia User Preference Model that Supports Semantics and its Application to MPEG 7/21. In *Proceedings of the Multimedia Modeling 2006 Conference*, MMM '06, pages 35–42, Beijing, China, 2006.

[20] C. Villalonga, M. Strohbach, N. Snoeck, M. Sutterer, M. Belaunde, E. Kovacs, A. Zhdanova, L. Goix, and O. Droegehorn. Mobile Ontology: Towards a Standardized Semantic Model for the Mobile Domain. In E. Nitto and M. Ripeanu, editors, *Proceedings of the 5th Internation Conference Service-Oriented Computing 2007 - ICSOC Workshops*, pages 248–257, Vienna, Austria, 2009. Springer-Verlang.