Exploiting Ontologies for better Recommendations

Abdulbaki Uzun
Service-centric Networking
Deutsche Telekom Laboratories, TU Berlin
abdulbaki.uzun@telekom.de

Christian Räck
Competence Center FAME
Fraunhofer Institute FOKUS
christian.raeck@fokus.fraunhofer.de

Abstract: Traditional recommender systems as they are mostly used in today's recommendation applications (e.g. the *SMART Recommendations Engine* of Fraunhofer FOKUS) primarily concentrate on recommending items to users. However, thinking of many modern (mobile) applications, contextual and semantic information may provide a significant preciseness to the recommendation process. That's why, Fraunhofer FOKUS' engine has been extended by two new extensions making the engine capable of incorporating contextual and semantic information when generating recommendations. This paper focuses on one of them, the *SMART Ontology Extension*.

1 Introduction

In a world of information overload, recommender systems filter relevant information and provide personalized content recommendations to users based on their interests and ratings. Numerous recommendation methods were designed over the years to enhance the preciseness of recommendations, such as content-based and collaborative filtering or hybrid approaches [AT05]. These traditional recommender systems primarily focus on recommending items to users. Existing ratings for items and content meta-data are the basis for effective recommendations. The *SMART Recommendations Engine* of Fraunhofer FOKUS [RS09], for example, belongs to this category of recommender systems.
However, thinking of many modern (mobile) applications, not only user and item, but also contextual and semantic information may provide a significant preciseness to the recommendation process. If for example, a user is vegan, eats only organic food, goes shopping nearby and tries to live economical, it would not make sense to recommend him stores far away or only discounters without taking his preference for vegan food into consideration. In order to make the *SMART Recommendations Engine* meet the demands of modern applications, Fraunhofer's engine has been extended by two recommender extensions. Inspired by Adomavicius et al. [ASST05], the *SMART Multidimensionality Extensions* enhance the two-dimensional matrix representation of recommender data (see Figure 1) by a multidimensional recommendation model enabling the incorporation of relevant contextual information into the recommendation process. The *SMART Ontology Extension*, on the other hand, exploits semantic ontology information in order to use implicit and semantic knowledge in the recommender. Since the *SMART Multidimensionality Extensions* are still in the conceptual phase, the scope of this paper is to present the functionality of the *SMART Ontology Extension*.

2 The SMART Recommendations Engine

The *SMART Recommendations Engine* developed by the Fraunhofer Institute FOKUS is a generic recommender system, which provides personalized recommendations for different applications. It can be licensed and used by various Internet businesses, rich media and entertainment services or SMEs. A flexible, general purpose algorithmic model is offered by the engine, which enables the formulation of application specific recommendation algorithms. These algorithms as well as the optimized entity-relationship-like data model are declared at configuration time by assembling the featured components. Through the provided API, custom components can be added as well extending the engine's capabilities to meet specific application demands. These components can be built using functional groups, such as basic mathematical operations, similarity and relevance computations, sorting and filtering, and data access. The recommender system also provides a custom query language called *Sugar Query Language (SuQL)*, which is used to request recommendations and related data at runtime.
In the *SMART Recommendations Engine*, data is represented in a data model consisting of entities and relationships between them. A domain represents a set of entities, whereas the relations between these entities are represented by matrices. A user domain, for example, can incorporate the set of all users, while an item domain can consist of all items in a certain application. The relation between the user and item domain can represent the ratings given by a user to an item stored as data values in the matrix table (see Figure 1).

Figure 1: Basic data model building block

A recommendation algorithm, which estimates predictions for each *User x Item* pair, is assembled at runtime configuration by defining a computation tree of matrix transformation components based on the requirements of the given application. Having some sort of data input (e.g. user profile, feedback) as a source, a number of transformations are applied in a hierarchical manner. The estimated utility function is provided by the top node of the tree. The engine also offers a variety of filters, which can be applied in a chain in order to alter the result set.

3 Related Work

In recent years, more and more researchers have recognized the importance of contextual and semantic information for recommendation processes and hence various approaches have been developed so far.

The multidimensional recommendation model proposed by Adomavicius et al. [ASST05] enhances the two-dimensional paradigm to a multidimensional matrix consisting of several context dimensions that can be related to each other. By doing so, it allows calculating different recommendations for different situations by taking different, but important aspects into consideration, such as user preferences, context or group information. A. Chen [Che05] presents in her paper a context-aware collaborative filtering system that generates item recommendations for a user based on different context situations.
The Semantic Web alleviates the search for information, enhances the visibility of knowledge in the web, and helps to gain implicit knowledge about a certain concept domain. Recommender systems can use these advantages to increase the preciseness of recommendations by exploiting semantic information, such as implicit knowledge and using them in the recommendation calculation process. One example for a semantic recommender system is described in the paper of Farsani and Nematbakhsh [FN06]. They suggest a methodology, which recommends semantic products to customers in the context of E-Commerce based on product and customer classification via OWL. Kim and Kwon [KK07], on the other hand, developed an ontology model with a multiple-level concept hierarchy for a grocery store scenario with four different ontologies.
Previous research activities are either focused on context or semantic information integration. However, incorporating both - context and semantic information - would increase the preciseness of recommendations decisively. The food scenario, for example, shows that the integration of both information types is necessary to satisfyingly answer a grocery recommendation request. That's why, Fraunhofer's engine was extended using both types of data.

4 The SMART Ontology Extension

The *SMART Ontology Extension* provides semantic ontology capabilities to the *SMART Recommendations Engine*. The first part of the extension is the *Ontology Mapping*. Here, the ontology structure of given semantic ontologies is mapped onto data matrices of the recommender. The second part makes use of the implicit knowledge present in the ontologies and generates semantic recommendations using the *Ontology Filter* on the previously created data matrices.
Mainly, OWL ontologies consist of individuals, classes, a class hierarchy, object properties, datatype properties and restrictions. These constructs are mapped onto data matrices of the recommender, so that the recommendation engine becomes capable of handling ontology information. Figure 2 shows an example for a property representation in the recommender of the ontology datatype property *eatingHabit*.

Once ontology data is stored in the recommender, the *Ontology Filter* can process the ontology information in the recommender. This filter is capable of performing two different operations on the data matrices, the *Concept Lookup* and the *Matrix Lookup* operations. The *Concept Lookup* is used to look up ontology concepts in the recommender. For the operation of the *Concept Lookup*, at least two different matrices are needed, whereas the column domain of the first matrix has to be the row domain of the second matrix. Applied

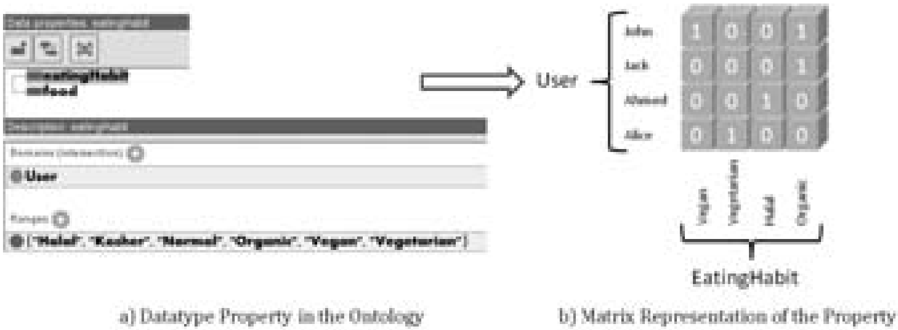

Figure 2: Ontology Mapping

on the first matrix, the *Concept Lookup* filters certain column elements for one single row element based on given filter constraints. The *Matrix Lookup* filters information in a matrix based on a given column domain result set of another matrix. Therefore, it also requires the use of two different matrices, whereas the column domain of the first matrix remains the column domain of the second matrix. Rows of the second matrix will be filtered based on the given column domain result set and a predefined set operation (*existential quantification* or *universal quantification*). The result is one set of filtered row elements.
Complex recommendation queries require combining both lookups to single a *Concept and Matrix Lookup* operation. An example can be seen in section 5.

5 Demonstration

In order to present the functionality of the *SMART Ontology Extension*, three ontologies were designed for the food scenario mentioned above. All data, such as food categories and products, ingredients, eating preferences or location information were manually included into these ontologies. After mapping all these data using the *Ontology Mapping* tool, the recommender can generate various recommendations based on different *SuQL* queries.
Assume that John is vegan, prefers only organic food and wants to get recommendations for snacks, bread and dairy products. And also assume that he already bought the brown bread product *Naturkind_SonnenblumenVollkornbrot_Geschnitten* and therefore rated this product implicitly. The *SuQL* query is built in that way that at first several semantic filterings are performed using the lookup operations several times in order to identify all desired products that fit John's eating preferences and his location. Afterwards, these elements are sorted by their relevance depending on the relevance predictions calculated by the recommendation algorithm.
In order to be able to answer John's query, the *Ontology Filter* first performs a *Concept Lookup* in the *User x EatingHabit* matrix that looks up John's eating preferences. In the second matrix (*EatingHabit x Ingredient*), his eating preferences (vegan and organic) are

mapped to the ingredients. The *Matrix Lookup* then looks up all groceries in the *Food x Ingredient* matrix for vegans and organic eating people individually. Both result sets are then unified to one single result set and inversed by the *Not* set operation. The result is a set of groceries, which can be eaten by John (see Figure 3). These groceries are also filtered by their categories, so that only snacks, bread and dairy products remain.

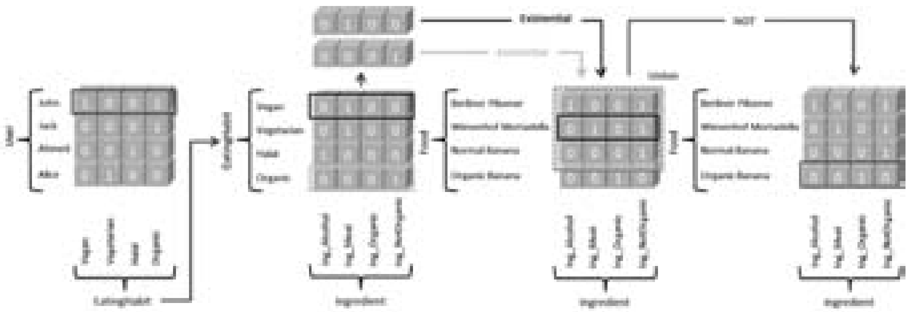

Figure 3: Ontology Filter - Concept and Matrix Lookup

Finally, the recommendation algorithm is used in the recommendation process. The content-based filtering approach calculates relevance predictions using the similarity of content keywords and user feedback. This algorithm can be extended to an *ontology-based filtering approach*, in which the ontology class structure data can represent content features. In the food scenario, for instance, similarities between groceries can be calculated based on the categories of the products (e.g. *brown bread* is more similar to *white bread* than to *snacks*).
For John's query, the recommender responds as seen in Figure 4. There is only one snack fitting his eating preferences with a low relevance value since John did not purchase any snacks yet. The recommender can present him a big choice of brown bread, but it has no vegan and organic dairy products to offer.

6 Conclusion

As seen above, the *SMART Ontology Extension* provided all necessary tools to generate semantic recommendations using the *SMART Recommendations Engine*. While the *Ontology Mapping* tool prepared the engine for utilizing ontology information, such as implicit knowledge or classification; recommendations were generated using the *Ontology Filter* with both lookups in an ontology-based filtering algorithm. Valuable information, such as a user's eating preferences as well as ontology classification (e.g. food categories) were integrated into the recommendation process providing much more precise recommendations than usual recommender systems. All in all, the *SMART Ontology Extension* affords an added value to the *SMART Recommendations Engine* by enabling the engine to provide

```xml
<Response>
  <User Name="John">
    <EatingHabit Name="Organic" />
    <EatingHabit Name="Vegan" />
    <Class Name="Snack">
      <Food Name="Naturkind_Grissini" Relevance="0.166666672">
        <GroceryStore Name="Kaisers" />
      </Food>
    </Class>
    <Class Name="Bread">
      <Food Name="Naturkind_SonnenblumenVollkornbrot_Geschnitten"
Relevance="0.5714286">
        <GroceryStore Name="Kaisers" />
      </Food>
      <Food Name="Stoehr_Bio-Mehrkornbrot" Relevance="0.5714286">
        <GroceryStore Name="OrganicShop" />
      </Food>
      <Food Name="Naturkind_Vollkornbrot_Geschnitten" Relevance="0.5714286">
        <GroceryStore Name="Kaisers" />
      </Food>
      <Food Name="Stoehr_Bio-Roggenvollkornbrot" Relevance="0.5714286">
        <GroceryStore Name="OrganicShop" />
      </Food>
    </Class>
    <Class Name="DairyProduct" />
  </User>
</Response>
```

Figure 4: SuQL Response of the Engine

accurate semantic and contextual recommendations.

References

[ASST05] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.

[Che05] Annie Chen. Context-aware collaborative filtering system: predicting the user's preferences in ubiquitous computing. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1110–1111, New York, NY, USA, 2005. ACM.

[FN06] H.K. Farsani and M. Nematbakhsh. A Semantic Recommendation Procedure for Electronic Product Catalog. *International Journal of Applied Mathematics and Computer Sciences*, 3:86–91, 2006.

[KK07] S. Kim and J. Kwon. Effective Context-aware Recommendation on the Semantic Web. *International Journal of Computer Science and Network Security*, 7:154–159, 2007.

[RS09] Christian Raeck and Fabian Steinert. Fraunhofer Institute FOKUS, CC FAME, myLab Research Laboratory, SMART Recommendations, 2009. http://mylab.fokus.fraunhofer.de/content/smartrecommendations/overview.