Metamodeling Approach to Preference Management in the Semantic Web

Dilek Tapucu[1], Ozgu Can[2], Okan Bursa[2], Murat Osman Unalir[2]

[1] Department of Computer Engineering, Izmir Institute of Technology, 35430 Gulbahce, Urla, Izmir, Turkey
[1] LISI / ENSMA - Université de Poitiers, 86961 Futuroscope Cedex, France
dilektapucu@iyte.edu.tr
[2] Department of Computer Engineering, Ege University, 35100 Bornova, Izmir, Turkey
{ozgu.can,okan.bursa,murat.osman.unalir}@ege.edu.tr

Abstract

Preference is a superiority state to determine the preferable or the superior of one entity, property or constraint to another from a specified selection set. Preference issue is heavily studied in Semantic Web research area. The existing preference management approaches only consider the importance of concepts for capturing users' interests. This paper presents a metamodeling approach to preference management. Preference meta model consists of concepts and semantic relations to represent users' interests. Users may have the same type preferences in different domains. Thus, metamodeling must be used to define similar preferences for interoperability in different domains. In this paper, preference meta model defines a general storage structure to manage different types of preferences for personalized applications.

1. Introduction

Preference issue is heavily studied in Semantic Web research area. There are many preference definitions in Computer Science. In Philosophical definition, preferences are used to reason about values, desires, and duties [Hansson, 2001]; for Databases concept, preferences help in reducing the amount of information returned in response to user queries [Lacroix & Lavency, 1987]. In Mathematical Decision Theory, preferences are used to model people's economic behavior. For Artificial Intelligence, preference relations serve to establish an intervention goal of an agent [Gadomski, 1993]. Although, all these definitions are separated from each other, they have also common values like privity and effecting decision making process.

In our perspective, preference is a superiority state to determine the preferable or the superior of one entity, property or constraint to another from a specified selection set. Determining the preferable effects the decision making process by personalizing users' interests.

Personalization is also very important for Semantic Web. As preferences are naturally personal constraints, personalization provides support to satisfy the users' preferences.

Preferences are data on the domain model. Preferences can be one individual, a range of one property, a class or a selection of all. Therefore, defining preferences inside the domain model is hard to explain. Preferences defined for management systems must be modeled in the abstract level.

Users may have same type preferences in different domains. Thus, metamodeling must be used to define similar preferences for interoperability in different domains.

Our preference metamodeling architecture is developed by considering all preference descriptions mentioned above. It is built on four layered architecture of the Object Management Group (OMG) and supports different types of preferences which exist in the literature. The main goals of this paper are:

- Representing different preference types in preference meta model and storing users' preferences in their FOAF(Friend of a Friend)[1] documents.
- Developing the related ontologies for dynamic preference management which is achieved by managing the change in domain ontologies.
- Realizing preference management to observe user data usage over the domain model and storing observed data as preferences.

The paper is organized as follows. Section 2 presents the related research work for preferences and preference types. Our preference metamodel approach is discussed in

[1] http://www.foaf-project.org/

Section 3. Finally, in Section 4 our work is summarized and future research is given.

2. Related Work

Nowadays, a lot of research effort has been made for the representation of personalization. Personalization provides support to satisfy the users' preferences, but users' needs and preferences dynamically change over time. The major limitations of personalization for user perspective are proposed in [Omero et al., 2007]. Therefore, Computer Science has joined the set of fields with significant interest in personalization with preference handling and preference querying. Artificial Intelligence provides powerful techniques that can help people to address the personalization problem [Java et al., 2006]. Search engines can be very effective in locating items if users provide the correct queries [Koutrika and Ioannidis, 2004]. Agents, try to map their preferences to a query for finding the item that most closely matches their requirements [Nambir et al., 2007].

There have been several approaches towards capturing the user's interests such as [Claypool et al. 2001], [Gauch et al., 2003], [Teevan et al., 2005] and also modeling the user's preferences in a hierarchical structure [Koutrika & Ioannidis 2004], [Pretschner & Gauch, 1999]. Preference elicitation, was recently proposed to address wasted effort problems [Conen & Sandholm, 2001], and several papers have studied different types of elicitors [Hudson & Sandholm, 2004], [Smith, 2002]. There are various frameworks for preference models [Chomicki, 2003], [Kießling et al., 2004] [Holland et al., 2003].

Preferences can be used to focus on search queries and to order the search results. For example personalized query composition [Koutrika & Ioannidis 2004] has to assemble the query using the various user preferences and personalized data. Query results presentation is adapted with the user's preferences [Kießling et al., 2004]. A set of preferences can be combined by using a generic combine operator which is instantiated with a value function [Agrawal & Wimmers, 2000]. Aggregating the preferences of humans, human-computer interaction issues must be considered because preferences are described as a complementary to rules and constraints [Junker & Mailharro, 2003].

Also preferences are expressed by the user for an entity which is described by a set of attributes; each attribute can take on values from a certain type. According to [Kießling, 2002] a preference is formulated on a set of attribute names with an associated domain of values.

In this paper, our approach is independent from the underlying preference model. Our primary work is putting together all preference types then collecting them under preference meta ontology. Related work for preferences and preference types are listed below:

- [Agrawal & Wimmers, 2000] presented preference function with "value dimension" and explained *numerical value based preferences*.
- [Kießling, 2002] presented formal semantics of preferences with base preference constructors that are "*pos*", "*neg*", "*pos/pos*" "*pos/neg*" and *lowest preference*. Kießling also described constructors for *complex preference, pareto preference*, *prioritized preference* and *numerical preference*.
- [Holland and Kießling, 2004] explained *situated preferences* with "long-term preference", "singular preference" and "non-singular preference".
- [Koutrika and Ioannidis, 2004] showed *atomic preferences* and explained *positive, negative, or indifferent* preferences, also presented *exact or elastic* preferences depending on whether the domain is categorical or numeric.
- [Oztürk & Tsoukiàs, 2005] showed different preference structures: *strict preferences* and *weak preferences*. They also explained the structure of the *numerical representation of the interval preferences*.
- [Dimopoulos et al., 2006] expressed three different preference layers which are *generic, contextual*, and *structural preferences*.
- [Siberski et al., 2006] specified *boolean preferences* with boolean condition and *scoring preferences* with value expression.
- [Fischer et al., 2006] explained *non-numerical base preferences*, *numerical base preferences*, and also *complex preferences.*
- [Dubois, 2004] expressed preferences in *qualitative* way with relativity approach for conditional preferences and specified *bipolar preferences* for possibility distributions.
- [Xu, 2007] explained *uncertain linguistic preference relations.*
- [Toninelli et al., 2008] defined two types of preference: *value preferences* and *priority preferences*. They showed a preference might be either object or data type depending on the property value. Thus, they explained how object preferences are modeled by means of the OWL ObjectProperty construct, and how data type preferences are modeled by means of the OWL DatatypeProperty construct. They also described requirements of meta-preferences.

3. Metamodeling Approach to Preference Management

In this section, we briefly describe our domain model, preference view ontology model and preference meta ontology model.

In our approach, we define a metamodel structure consists of FOAF, Domain Ontology, Preference View Domain (PVD) Ontology and Preference Meta Ontology (PMO).

- *FOAF* ontology has the personal data about the user.
- *Domain Ontology* defines the data about the domain knowledge.
- *Preference View Domain Ontology* decomposes the domain ontology and stores ontology resources in a hierarchy based on Ontology Definition Metamodel.
- *Preference Meta Ontology* defines different preference types.

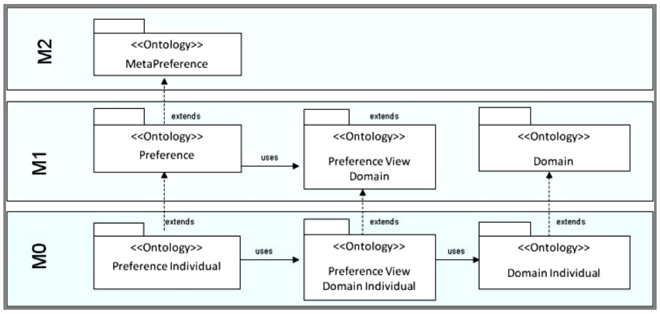

Figure 1: Generic View for Metamodel Architecture.

Preferences which are stored in FOAF ontology take the metamodel from PMO and data from PVD. Our metamodel shows that personal preferences for more than one domain can be stored and personalized in a metamodel architecture.

In this chapter, we explain the preference metamodel architecture.

3.1 Domain Model

Domain model defines objects and individuals for a specific domain. In a real world application, there may be several domains working together. These different domains can have similar objects or semantics. We model similar objects and their semantics from different domains by defining domain model and its metadata.

The domain model consists of two layers. In the domain metamodel layer (M1), domain resources and their relationships are defined. In the domain instance layer (M0), domain data is stored as individuals which are extended from the domain metamodel.

We illustrated the domain model by using DERI's e-tourism ontology (*http://e-tourism.deri.at/ont/e-tourism.owl*). Figure 2 shows an example of a Hotel Accommodation class and a Hotel individual.

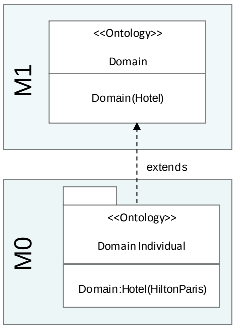

Figure 2: Model - Instance Relation for Domain Ontology.

3.2 Preference View Domain Ontology (PVD)

Preference View Domain decomposes the domain ontology into classes, properties (datatype, object) and individuals according to Ontology Definition Model (ODM) specification. ODM specification provides compatibility with OWL-DL, contains definition of concepts, individuals, relationships among them, and constraints. [Brockmans et. al., 2006]

In our approach, PVD works like ontology splitter. It captures resources from the domain ontology for preference management. Specifically, each concept and relation in the domain ontology is a choice for the user. So, all classes, properties and individuals in the domain ontology must be adapted into PVD.

PVD ontology is modeled as in Figure 3.

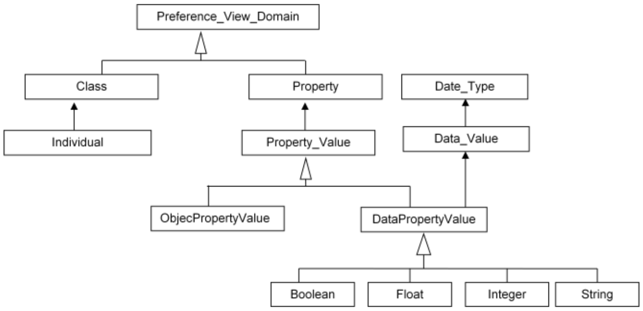

Figure 3: Preference View Domain (PVD) Ontology

Two layers of the domain model are mapped separately in PVD. In this mapping, classes, properties and constraints of the domain model are adapted according to the model shown in Figure 3. Each class, property and constraint can be a preference for the user, thus, they are set into classes defined in PVD.

Individuals, which are in M0 layer of the domain model, can be also preferences for the user. Therefore, individuals are also mapped into PVD and they are held in PVD individual class. The problem in this mapping is the spoilage in the semantics of the model. Constraint class of REI policy language [Kagal et al., 2003] is used to overcome this problem.

PVD model together with the domain model is shown in Figure 1. Table 1 shows how data attributes in the domain model are adapted to PVD.

Table 1: An example for Domain and PVD mapping

Domain Model	**PVD**
Class Accommodation	*Class* Accommodation
Individual:Accommodation HiltonParis	*Class:Individual* HiltonParis
Properties	*Class:Properties*
• *Object*	• *Class:Object*
hasRoom	hasRoom
• *Datatype*	• *Class:Datatype*
Boolean	Class:Boolean
petsAllowed	Individual:petsAllowed
Integer	Class:Integer
hasStarRating	Individual:hasStarRating
String	Class:String
hasType	Individual:hasType

3.3 Preference Meta Ontology (PMO)

After separating the domain model and entities inside the domain model, entities are modeled as preferences respect to preference types.

Preference Meta Ontology includes the required entities to define users' preferences. PMO accepts domain ontologies as a selection set and defines different preference types to illustrate users' personal preferences. Thus, preferences are kept in a type-based model inside the FOAF documents.

3.3.1 Preference Types. Preference Meta Ontology includes different types of preferences as mentioned in Section 2. These preference types are listed below:

- **Boolean preferences** [Siberski et al., 2006] are modeled by means of OWL Boolean Data Type Property Construct.

  Example: Hotel Hilton Paris has sauna.

- **Constraint preferences** are modeled by means of the OWL Object Property construct. Constraint preferences show the related domain class.

  Example: Tennis Court has opening hours from 10:00 am to 22:00 pm.

- **Individual preferences** take value from domain class.

  Example: User prefers hotel Hilton Paris.

- **Interval preferences** [Agrawal & Wimmers, 2000] have numerical, textual and date values either discrete or continuous.
  - **Numeric preferences** [Oztürk & Tsoukiàs, 2005]
    Example: User prefers hotel prices between 150-250 Euros.
  - **Textual preferences** [Fischer et al., 2006]
    Example: User prefers the beer named "Aloha".
  - **Date preferences**
    Example: User prefers to swim at weekend days.

Preferences defined in PMO are shown in Figure 4.

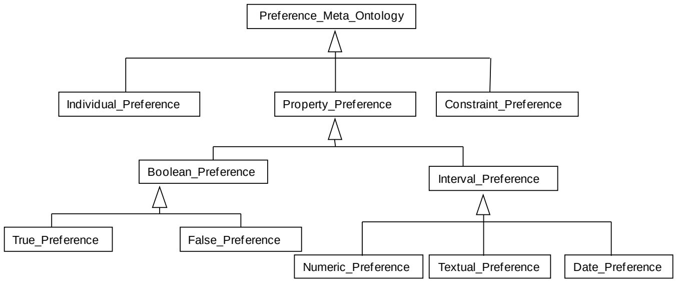

Figure 4: Preference Meta Ontology (PMO)

3.4 Preference Management

In this section, we describe how PVD ontology and PMO work together and how we store different types of preferences in FOAF ontology.

Domain knowledge is open and dynamic [Toninelli et al., 2008]. As we accept domain knowledge is a selection set of user, user preferences are also dynamic and open. For this reason, preference management must be defined in an open-world-assumption model and must be handled in a dynamic way. In this paper, preferences are defined in ontologies and stored in FOAF documents for dynamic processing.

Preference management is required for personalization and effective querying in Semantic Web. Personalization is achieved by storing personal preferences. Thus, to reduce the query time, personal preferences can be reused for similar queries in different domains.

In our preference meta model;

- preference definition comes from PMO,
- preference values come from PVD,
- preferred values are stored in FOAF.

In the following, we gave a scenario to explain preference management in preference metamodel architecture.

Example: Anna is a participant in a conference and wants to make a reservation for the conference hotel. She entered the hotel's web site and loaded her FOAF document. Preference management system recognized her hotel preferences for accommodation from her FOAF document. The system asked Anna her room preferences. She made her room preferences by selecting the hotel room's attributes and completed her reservation.

After the reservation, following preferences are identified by preference management system:

Room Preference-1: Prefers ocean view room.

$$\begin{aligned} &hasView \equiv FunctionalProperty(class, string)\\ &RoomPreference(roomPreference1)\\ &\qquad \equiv EndUser(Anna) \wedge hasView(room, "Ocean") \end{aligned}$$

Room Preference-2: Prefers not smokingAllowed room.

$$\begin{aligned} &smokingAllowed\\ &\equiv DataTypeProperty(\{GuestRoom, Room\}, boolean) \end{aligned}$$

$$\begin{aligned} &RoomPreference(roomPreference2)\\ &\qquad \equiv EndUser(Anna)\\ &\qquad \wedge smokingAllowed(room, false) \end{aligned}$$

Room Preference-3: Prefers wirelessConnection room.

$$wirelessConnection \ \equiv DatatypeProperty(class, boolean)$$

$$\begin{aligned} &RoomPreference(roomPreference3)\\ &\qquad \equiv EndUser(Anna)\\ &\qquad \wedge wirelessConnection(room, true) \end{aligned}$$

Hotel Preference-1: Prefers hasStarRating=5.

$$hasStarRating \equiv DatatypeProperty(Accomodation, \{2,3,4,5,6,7\})$$

$$\begin{aligned} &HotelPreference(hotelPreference1)\\ &\qquad \equiv EndUser(Anna)\\ &\qquad \wedge hasStarRating(AnnasAcc, 5) \end{aligned}$$

Hotel Preference-2: Prefers Hotel Central.

$$\begin{aligned} &HotelPreference(hotelPreference2)\\ &\qquad \equiv EndUser(Anna)\\ &\qquad \wedge Accomodation(HotelCentral) \end{aligned}$$

Hotel Preference-3: Prefers room price between 150-250 Euros.

$$hasPrice \equiv DatatypeProperty(room, float)$$

$$\begin{aligned} &HotelPreference(hotelPreference3)\\ &\qquad \equiv hasminPrice(AnnasAcc, 150)\\ &\qquad \wedge EndUser(Anna)\\ &\qquad \wedge hasmaxPrice(AnnasAcc, 250) \end{aligned}$$

PVD ontology takes these preferences and stores them as ontology classes as shown in Figure 3.

$$\begin{aligned} &String(hasView) \equiv textValue(\text{Ocean})\\ &\qquad \wedge constraint(hasView, rdf{:}type, hasView) \end{aligned}$$

$$\begin{aligned} &Boolean(notSmokingAllowed)\\ &\equiv False\\ &\wedge constraint(notSmokingAllowed, rdf{:}type, smokingAllowed) \end{aligned}$$

$$\begin{aligned} &Boolean(wirelessConnection)\\ &\equiv True\\ &\wedge constraint(wirelessConnection, rdf{:}type, wirelessConnection) \end{aligned}$$

$$\begin{aligned} &Integer(hasStarRating)\\ &\equiv ExactValue(5)\\ &\wedge constraint(hasStarRating, rdf{:}type, hasStarRating) \end{aligned}$$

$$\begin{aligned} &Individual(HotelCentral)\\ &\equiv constraint(HotelCentral, rdf{:}type, Accomodation) \end{aligned}$$

$$\begin{aligned} &Float(hasPrice) \equiv Float\_Value(150)\\ &\qquad \wedge constraint(hasPrice, rdf{:}type, hasPrice) \end{aligned}$$

PMO takes these values due to their class definitions and enclose them with preference type definitions.

$$Textual\_Preference(hasView) \equiv String(hasView)$$

$$\begin{aligned} &FalseBooleanPreference(notSmokingAllowed)\\ &\qquad \equiv Boolean(notSmokingAllowed) \end{aligned}$$

$$\begin{aligned} &TrueBooleanPreference(wirelessConnection)\\ &\qquad \equiv Boolean(wirelessConnection) \end{aligned}$$

$$\begin{aligned} &NumericIntegerPreferences(hasStarRating)\\ &\qquad \equiv Integer(hasStarRating)\\ &\qquad \wedge FoafMaxValue(5) \end{aligned}$$

$$\begin{aligned} &IndividualPreferences(HotelCentral)\\ &\qquad \equiv Individual(HotelCentral) \end{aligned}$$

$$\begin{aligned} &NumericFloatPreferences(hasPrice)\\ &\qquad \equiv Float(hasPrice) \wedge FoafMaxValue(150)\\ &\qquad \wedge FoafMinValue(250) \end{aligned}$$

Anna's room preferences in FOAF document are stored as:

$$\begin{aligned} &Preferences(Anna'sRoomPreference)\\ &\equiv Textual\_Preference(hasView)\\ &\wedge FalseBooleanPreference(notSmokingAllowed)\\ &\wedge TrueBooleanPreference(wirelessConnection) \end{aligned}$$

Other hotel preferences are also defined in the same way. Finally, Anna’s FOAF profile is linked to her preferences.

$$Person(Anna) \equiv Preferences(Anna'sRoomPreference) \\ \wedge Preferences(Anna'sHotelPreference)$$

In this example, when she selected her room preferences, each preference in domain ontology is mapped to preference view domain ontology. System overlaps preferences in PVD ontology to classes in PMO. Her preferences are stored as an instance of the preference class in PMO. Anna’s preferences in FOAF document is shown in Figure 5 .

```
<foaf:Person rdf:ID="Anna">
  <foaf:preferences
  rdf:resource="#Anna_s_Room_Preferences"/>
  <foaf:preferences
  rdf:resource="#Anna_s_Accommodation_Preferences"/>
</foaf:Person>
<foaf:Preferences
rdf:ID="Anna_s_Accommodation_Preferences">
  <pmo:Individual_Preference
  rdf:resource="&pvd;HotelCentral"/>
  <pmo:Numeric_integer_Preference
  rdf:resource="&pvd;hasStarRating"/>
</foaf:Preferences>
<foaf:Preferences rdf:ID="Anna_s_Room_Preferences">
  <pmo:True_boolean_Preferences
  rdf:resource="&pvd;wirelessConnection"/>
  <pmo:True_boolean_Preferences
  rdf:resource="&pvd;notSmokingAllowed"/>
 <pmo:Textual_Preference rdf:resource="&pvd;hasView"/>
</foaf:Preferences>
```

Figure 5: Anna’s preferences in FOAF.

4. Discussion and Future Work

Preference management in information systems is an ongoing research for personalization. Our work differs from other studies in the following ways: the management of user preferences and setting user preferences according to domain ontologies stored in the knowledge base. There is no research in the literature for setting user preferences based on domain ontologies.

As future work, we plan to extend preference meta ontology to manage complex preferences such as preferring an entity to another and fuzzy preferences using weighted preferences. Thus, preferences can be managed by ensuring priority by adding a list of preferences and fuzzy linguistic terms will be achieved by adding fuzzy preferences.

Preference conflicts which may occur in querying preferences will be solved by using policy management. For example, in our scenario, if there is a rule like “*Conference participants can only make reservation for mountain view rooms.*” is defined then, Anna’s room preference will conflict with this conference rule. Policy management will be used for the preference conflict resolution.

Social networks can be created by using FOAF documents. We can group people who have the same preferences by using preferences which are stored in FOAF documents. This method leads us to the profiling issue in the Semantic Web research area. Thus, we will manage preferences with using profiles in further issues of our approach.

Finally, we will develop preference, profile and policy management by using Semantic Web Portal applications.

5. References

Agrawal R. and Wimmers E., 2000. *A framework for expressing and Combining Preferences.* In Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 297–306.

Claypool M., Le P., Waseda M., and Brown D., 2001. *Implicit Interest Indicators*. In Proceedings of ACM Intelligent User Interfaces Conference (IUI), pages 33–40, Santa Fe, New Mexico, USA.

Chomicki J., 2003. *Preference Formulas in Relational Queries.* ACM Transactions on Database Systems, vol. 28 (4), pp. 1-39.

Conen W. and Sandholm T., 2001. *Preference elicitation in combinatorial auctions: Extended abstract.* In Proceedings of the ACM Conference on Electronic Commerce (ACM-EC), pages 256–259, Tampa, FL.

Dimopoulos, Y., Moraitis, P. Tsoukias, A., 2006. *Qualitative Preference Modelling in Constraint Satisfaction.* Decision Theory and Multi-Agent Planning, G. Della Riccia, D. Dubois, R. Kruse and H.J. Lenz (Eds.), pp. 15-31, Springer Verlag, NewYork.

Dubois D., Kaci S., Prade H., 2004. *Representing preferences in the possibilistic setting.* Preferences 2004.

Fischer S., Kießling W. and Preisinger T., 2006. *Preference based Quality Assessment and Presentation of Query Results.* Studies in Fuzziness and Soft Computing 978-3-540-33288-6. Springer pp. 91-121.

Gadomski A.M., 1993. *AI technologies in ENEA: Architecture and decision-making of intelligent agents*. The seminar materials of System Research Institute, Polish Academy of Sciences, Warsaw.

Gauch S., Chaffee J., and Pretschner A., 2003. *Ontology based Personalized Search and Browsing*. Web Intelligent and Agent Sys., 1(3-4):219–234.

Hansson S. O., 2001. *Preference Logic. In Handbook of Philosophical Logic*, D. Gabbay, Ed. Vol.8.

Hessling A., Kleemann T., Sinner. A., 2004. *Semantic User Profiles and their Applications in a Mobile Environment*. In the Proc. of Artificial Intelligence in Mobile Systems 2004 (AIMS’04) In conjunction with UbiComp2004.

Holland S., Ester M., and Kießling W., 2003. *Preference Mining: A Novel Approach on Mining User Preferences for Personalized Application*s. Proc. 7th European Conference on Principles and

Practice of Knowledge Discovery in Databases (PKDD 2003), pp. 204-216.

Holland S., Kießling W., 2004. *Situated Preferences and Preference Repositories for Personalized Database Applications*, 23rd International Conference on Conceptual Modeling (ER 2004). S. 511-523, Shanghai, China.

Hudson B. and Sandholm T, 2004. *Effectiveness of Query Types and Policies for Preference Elicitation in Combinatorial Auctions*. In: Proc. 3rd Int. Joint. Conf. on Autonomous Agents and Multi Agent Systems, pp. 386-393.

Java A., Finin T., and Nirenburg S., 2006. *A Semantic News Framework*. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06), pages 1939-1940.

Junker U. and Mailharro D., 2003. *Preference programming: Advanced Problem Solving for Configuration*, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 17, pp. 13–29.

Kagal L. *et al.*, 2003. *A Policy Based Approach to Security for the Semantic Web,* In Proceedings of the 2nd International Semantic Web Conference (ISWC2003), pp. 402-418, Sanibel Island, FL, USA.

Kießling W., 2002. *Foundations of Preferences in Database Systems*. Proc. 28th Inter. Conference on Very Large Data Bases (VLDB 2002), pp. 311-322.

Kießling W., Fischer S., and Döring S., 2004. *COSIMA^{B2B}– Sales Automation for E Procurement*. 6th IEEE Conference on E-Commerce Technology (CEC 2004), San Diego, CA.

Koutrika G., Ioannidis Y., 2004. *Personalization of Queries in Database Systems*. Proc. 20th International Conference on Data Engineering, Boston, USA.

Lacroix M., Lavency P., 1987. *Preferences: Putting More Knowledge into Queries*, Proceedings of the 13th International Conference on Very Large Data Bases (VLDB), pp. 217-225.

Nambiar U., Gupta H., Mohania M., 2007. *CallAssist: Helping Call Center Agents in Preference Elicitation*. 33rd Very Large Data Bases Conference (VLDB '07), pp. 1338-1341.

Omero P., Polesello N., Tasso C., 2007. *Exploiting Personalized Information Access within an Online Digital Library for Medicine: the BIBLIOMED Project*, 10th DELOS Thematic Workshop on Personalized Access, Profile Management, and Context Awareness in Digital Libraries (PersDL2007), Corfu, Greece.

Oztürk M. and Tsoukiàs A., 2005. *Preference on intervals: a general framework.* In Proceedings of the IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling, Edinburg, Scotland.

Pretschner A. and Gauch S., 1999. *Ontology based Personalized Search*. Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence, pp. 391-398, Chicago.

Siberski W., Pan J., and Thaden U., 2006. *Querying the Semantic Web with Preferences*. International Semantic Web Conference, pp:612-624, Athens, GA, USA.

Smith T, Sandholm T., and Simmons R., 2002. *Constructing and Clearing Combinatorial Exchanges Using Preference Elicitation*. In AAAI-02 workshop on Preferences in AI and CP: Symbolic Approaches, pages 87–93.

Teevan J., Dumais S., and Horvitz E., 2005. *Personalizing search via automated analysis of interests and activities.* ACM SIGIR conference on Research and development in information retrieval, pp. 449 – 456.

Toninelli A., Corradi A., Montanari R., 2008. *Semantic-based discovery to support mobile context-aware service access.* Computer Communications archive Volume 31 , Issue 5., pp. 935-949.

Xu, Y., 2007, *A Method for Multiple Attribute Decision Making with Incomplete Weight Information under Uncertain Linguistic Preference Relations*. IEEE International Conference on Grey Systems and Intelligent Services, pp. 768-773.

Appendix

You can access the ontologies developed in this work from the following URL:

http://efe.ege.edu.tr/~odo/MPREF2008/PMO.owl

http://efe.ege.edu.tr/~odo/MPREF2008/PVD.owl

http://efe.ege.edu.tr/~odo/MPREF2008/I_Tour.owl

http://efe.ege.edu.tr/~odo/MPREF2008/foaf.owl