

Semantically Enhanced Enforcement of Mobile Consumer's Privacy Preferences

Mahmoud Youssef
Arab Academy for Science & Tech.
P.O. Box 1029 - Miami
Alexandria - Egypt
youssefm@aast.edu

Nabil R. Adam
Rutgers University
180 University Ave.
Newark, NJ 07102
adam@cimic.rutgers.edu

Vijayalakshmi Atluri
Rutgers University
180 University Ave.
Newark, NJ 07102
atluri@cimic.rutgers.edu

ABSTRACT

In such applications as location-based advertising, merchants use consumers' information to send them personalized advertisements. These applications provide convenience to consumers and competitive advantage to merchants. However, the improper use of consumers' information presents a serious threat to their privacy. It is also important to observe that among the motives for the consumers to accept advertisements is the *incentive* offered by the merchant. Therefore, such incentive should become a criterion upon which consumers decide to grant or deny access to their information. We propose modeling mobile consumer preferences including incentive-related preferences in an ontology using the Ontology Web Language (OWL) and enforcing these preferences using reasoning techniques. We present modeling of consumer preferences and merchant queries in that ontology and describe how to match them. Moreover, we present a prototype implementation and an evaluation study that shows that query size is more significant than the ontology size.

Categories and Subject Descriptors

K.4.1 [Computer Milieux]: Computers and Society – *Public Policy Issues, Privacy.*

General Terms

Security, Human Factors.

Keywords

Incentive, Privacy, Mobile Consumer, Semantic Enforcement.

1. INTRODUCTION

In such applications as location-based advertising, merchants use consumers' information to send them personalized advertisements. These applications provide convenience to consumers and competitive advantage to merchants. However, the improper use of consumers' information presents a serious threat to their privacy. In previous work [7], we discussed the requirements for protecting consumers' information. These

requirements include: 1) preventing illegal sharing of consumer information, 2) enabling consumers to express privacy preferences at different levels of granularity, 3) specifying spatio-temporal constraints in the privacy rules, and 4) providing efficient enforcement of the privacy policies. We also discussed the convenience that location-based advertising offers to mobile consumers by providing timely and location-based offers. Nevertheless, another motive for the consumers to accept advertisements is the *incentive* offered by the merchant, and hence, it should become a criterion upon which consumers decide whether to grant or deny access to their information.

Modeling incentives is not an easy task. In the proposed solution in [7], each component of the privacy policy (i.e., the subject, the object, and the spatio-temporal constraints) was represented as a hierarchy. Using encoding, we were able to evaluate these policies using syntactic matching. However, the structural properties of hierarchies are not expressive enough to capture the semantics of incentives. For instance, multiple representations of the same concept are difficult to achieve in typical hierarchies. Consider a consumer who is interested in discounted tickets for events that include fighting sports, e.g., boxing, at the same time, a merchant may specify boxing as a ring game. In this case, not only that syntactical matching will fail, but also hierarchical representation will fail too. In addition, hierarchical representations are usually not appropriate for representing exceptions. Consider a location hierarchy that includes country, region, state, county, and city. In such hierarchy, it is assumed that counties consist of disjoint cities, states consists of disjoint counties, etc. However, some cities violate that assumption. For example, New York City consists of five counties. For such knowledge-rich domains, we propose using Knowledge Representation (KR) techniques.

KR methodologies can generally be classified into logic-based approaches and non logic-based approaches. Recently, the Semantic Web initiative [13] led to intensifying efforts to merge logic-based KR approaches with database technology to provide machine understandable search and retrieval. These efforts led to the development of several web standards, e.g., the Ontology Web Language (OWL) [5], the Resource Description Framework (RDF) [12], and several development and reasoning tools (e.g., [11] [15]). Using OWL, a domain can be represented in an ontology; then, using a reasoner, several types of inference operations can be performed on that ontology. We propose modeling mobile consumer preferences including those related to incentives using OWL and enforcing them using Description Logic (DL) reasoning techniques.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'06, April, 23-27, 2006, Dijon, France.

Copyright 2006 ACM 1-59593-108-2/06/0004...\$5.00.

The rest of this paper is organized as follows. Section 2 describes the proposed solution, Section 3 presents a prototype implementation, Section 4 presents a performance evaluations study, Section 5 presents the related work, and Section 6 summarizes the work.

2. PROPOSED SOLUTION

The idea of our solution is to build an OWL ontology that captures the semantics of incentives and the other consumer preferences in concepts, properties, and restrictions, and then to answer merchant queries by reasoning about them against the ontology. The ontology includes six major taxonomies (terminologies): Incentive types, Incentive values, Time, Location, Merchants, and Products. We call the overall ontology, the *preferences* ontology. The ontology was designed so that both consumer preferences and merchant queries are *subsumed* by a class called Consumer Privacy Preferences (CPP). We assert consumer preferences as follows: each consumer has a set of preferences. Each of these preferences is a *description* e.g., a subclass of CPP with a set of properties and property restrictions. Merchants' queries are asserted upon receiving them where each of the queries is a *description* that includes the same main components as the consumer preferences but with different values.

The processing of a query involves finding the consumer preferences that match the merchant query. The result of the matching is one of three cases: *granted*, *denied*, and *non-determined*. Although it makes sense to represent preferences and queries as individuals, we represent them as concept to benefit from the performance gain of TBox subsumption over individual matching. The *grant* case is obtained when the query is equivalent to or subsumed by the consumer preferences and the *deny* case is obtained when the query and the consumer preferences are disjoint. The case when the query subsumes the consumer preferences produces the *non-determined* result. This type can be resolved, for instance, by always considering them granted or denied.

It is important to observe that consumer preferences do not include a grant/deny flag. Rather, we assume that the consumer's privacy policy is a closed policy where all grants are specified. However, the merchant's query has to semantically match the query in order to receive grant permission. Therefore, the consumer preferences in the ontology can be seen as sets of conditions under which the consumer is willing to accept advertisements where each set is represented as a class. Since the mismatch between a merchant query and a consumer preference does not suggest or even imply that the consumer does not want to receive that advertisement, there is no need to resolve conflicts. In fact, to exploit this property, the evaluation algorithm applies a short circuit technique in which it does not evaluate further preferences of a consumer who has already produced a grant result.

2.1 An Ontology of Mobile Consumer Preferences

As advised by the designer of OWL, each ontology should be designed to answer specific questions [5]. We follow this advice in designing the preferences ontology by focusing on the needs of the application at hand. For instance, our time hierarchy, although it could be useful for other purposes, it was mainly designed to

help consumers express temporal preferences towards advertisements. The following is the upper part of the preferences ontology:

CPP	⊆	T	// T is the universal concept
AllLocations	⊆	CPP	
AllTimes	⊆	CPP	
AllProduct_Services	⊆	CPP	
IncentiveType	⊆	CPP	
IncentiveValue	⊆	CPP	
AllMerchants	⊆	CPP	

Due to space limitations, the following subsections will focus on the details of the incentive related taxonomies. The reader is referred to [6] for detailed description of the other taxonomies.

2.1.1 Incentive Type and Value Taxonomies

We first discuss a model of incentives. We then present the design of the incentive taxonomies. Marketers use different promotions techniques. We found the following techniques appropriate for mobile consumers: 1) Price reduction, 2) Happy hour (i.e., price reduction for a short time), 3) No payment for a specific period, 4) Payment in installments, 5) More items for free, 6) Bundle (which could be homogeneous — reduced price for second item, or heterogeneous — another product at a reduced price), 7) Premium (i.e., a free non-related product or service, e.g., free miles), 8) Prize, 9) Contest (i.e., based on a skill), 10) Sweepstakes (i.e., based on chance), and 11) Rebates or refunds (i.e., cash refund, coupon refund, or escalating refund).

We analyzed the above types of promotions and found that the incentive in these promotions has a type and a value where the type is one of the following classes: 1) monetary, 2) coupon, 3) time slack, 4) extra items, and 5) payment on installments. We have also found that monetary incentives can be instant (e.g., discount) or delayed (e.g., mail in rebate). Moreover, we found that promotions, in general, include conditions. For instance, a marketer may offer a coupon that is valid only with a minimum purchase. As such, a promotion model should include at least the following components:

- 1- Incentive type.
- 2- Incentive Value, e.g., face value of a coupon, reduction percentage, length of grace period, number of installments, etc.
- 3- Conditions; which can be: a) Temporal, e.g., date of event, b) Spatial, e.g., stores or locations of event, c) Product related, and d) Minimum purchase.

This model is not inclusive. If some application has other incentive types, valuations, or conditions, they can be added or at least classified under *other*.

We built two incentive taxonomies, *IncentiveType* and *IncentiveValue*, as terminologies in the preferences ontology. The DL description of the incentive type is as follows:

IncentiveType	⊆	T
Monetary	⊆	IncentiveType
Coupon	⊆	IncentiveType
TimeSlack	⊆	IncentiveType
Extraltems	⊆	IncentiveType
PayOnInstallments	⊆	IncentiveType
InstantRefund	⊆	Monetary

DelayedRefund \sqsubseteq Monetary

For each of these subclasses, an object property is defined where the domain is the `IncentiveType` class. For instance, for the class `Monetary`, the property `hasMonetaryIncentive` is defined. In addition, promotion conditions are expressed as property restrictions on the class `IncentiveType` and its subclasses where the common properties and restrictions are applied to the parent class and the remaining properties and restrictions are applied only to the appropriate subclass. For instance, the product condition is applied by a property `hasProduct` which has as range the class `AllProducts` with a restriction `allValuesFrom AllProduct` and with `cardinality = 1`.

The `IncentiveValue` taxonomy includes five subclasses (`PercentageReduction`, `ScalarReduction`, `Price`, `TimeSlack`, and `NumberOfInstallments`). The taxonomy also includes five datatype properties `hasPercentageValue`, `hasScalarValue`, etc. The range for these properties is the XML integer data type. Thus, a consumer can define a preference to receive discounts that are at least 20% as follows:

\geq_{20} `IncentiveValue.hasPercentageValue`

2.1.2 Time Taxonomy

The main class in the time taxonomy is `AllTimes` where its semantics is the set of all hours (in one year). Therefore, any subclass must be a valid subset of individuals (i.e., hours).

2.1.3 Location Taxonomy

The main class in the location taxonomy is `AllLocations` where its semantics is the set of all cities. However, since we are using class subsumption for reasoning, we represented cities as primitive classes instead of individuals.

2.1.4 Product Taxonomy

We use the United Nations Standard Products and Services Code (UNSPSC) [14] as the taxonomy for products. UNSPSC provides a five level taxonomy: Segment, Family, Class, Commodity, and Business Function. We imported the taxonomy from an XML format to an OWL ontology and merged it with the other taxonomies in the preferences ontology.

2.1.5 Merchant Taxonomy

We developed the `Merchant` taxonomy to be similar to the industry hierarchy in [7]. In that taxonomy, the main class is `AllIndustries` and subclasses are as in the US Census Bureau's industry classification. Merchants were related to the `Products` taxonomy by two properties: `producesProduct` and `offersService` where the domain of the properties is the `Merchant` taxonomy and the range is the `Products` taxonomy.

2.2 Modeling of Preferences and Queries

In the `Preferences` ontology, two more taxonomies were created to represent the consumer privacy preferences (`ConsPref`) and the merchant queries (`MerchQuery`) as follows:

`ConsPref` \equiv `CPP` \sqcap
 $(\forall \text{hasLocation.L1} \sqcap$
 $\forall \text{hasTime.T1} \sqcap$

$\forall \text{hasProduct.P1} \sqcap$
 $\forall \text{hasMerchant.M1} \sqcap$
 $\forall \text{hasIncentiveType.IT1} \sqcap$
 $\forall \text{hasIncentiveValue.IV1}$

And

`MerchQuery` \equiv `CPP` \sqcap
 $(\forall \text{hasLocation.L2} \sqcap$
 $\forall \text{hasTime.T2} \sqcap$
 $\forall \text{hasProduct.P2} \sqcap$
 $\forall \text{hasMerchant.M2} \sqcap$
 $\forall \text{hasIncentiveType.IT2} \sqcap$
 $\forall \text{hasIncentiveValue.IV2}$

Individual preferences developed by the consumers are created under the `ConsPref` class as subclasses that uniquely identify the consumer and the preference number for that consumer. The naming convention for these classes is `ConsPref_CID_n` where `CID` is the consumer ID and `n` is the preference number for that consumer. Merchant queries were designed in a similar manner where incoming queries are given a unique name `MerchQuery_MID_m` which identifies the merchant and the query number for that merchant.

2.3 Enforcing Preferences by Reasoning

In order to enforce consumer preferences, we match the merchant query to the preferences. The matching is computed as the intersection of `ConsPref_CID_n` (denoted by `P`) and `MerchQuery_MID_m` (denoted by `Q`). Based on the intersection operations, the relation of the preference to the query can be:

Exact match: $P \equiv Q$ (that is `P` and `Q` are equivalent)

Disjoint: $P \sqcap Q \sqsubseteq \perp$ (that is the intersection is the Nothing concept).

Subsumes: $Q \sqsubseteq P$ (that is the query is a subset of the preference)

Subsumed: $P \sqsubseteq Q$ (that is the query is a super set of the preferences)

Exact matching is a clear case where the result should be "granted". Subsumes also is a situation where the result is granted because it indicates that for all the conditions, the query is a subset of the consumer preferences. On the other hand, disjoint is a clear case for "denied". The subsumed case is where the permission is non-determined. For instance, if the merchant's query specified the product as `FishingEquipment` while the consumer is only interested in `Rafts` (a descendant of `FishingEquipment`), it may not be accurate to consider that as a match. By considering it as *Denied*, the system is more conservative. However, as noted before, it can be handled by a generic consumer choice.

The matching algorithm consists of a loop that scans the consumer preferences and checks which of the cases above apply to the preference at hand. When a consumer's preference matches as `ExactMatch` or `subsumes`, that consumer's ID is added to the `ConsumerIDList` array which marks the consumers who should not be checked further. We also return subsumed matches in the `NonDeterminedList`.

ALGORITHM 1 Preferences Matching

```
INPUT: Merch_MID_Query_m //A query from a merchant
OUTPUT: ConsumerIDList, //A list of consumer IDs
        NotDeterminedList
ConsumerIDList = empty List
NonDeterminedList = empty List
Assert new query Merch_MID_Query_m
Retrieve All the Children of ConsPref to ChildrenList
WHILE NOT at end of ChildrenList DO
{
  IF Cons_CID_Pref_n NOT in ConsumerIDList THEN
    IF match(Merch_MID_Query_m Cons_CID_Pref_n) =
      ExactMatch OR match(Merch_MID_Query_m
        Cons_CID_Pref_n)= PlugIn THEN
      Extract CID
      ConsumerIDList.Append(CID)
    ELSE
      IF match(Merch_MID_Query_m Cons_CID_Pref_n)
        = Subsumes THEN
        Extract CID
        NotDeterminedList.Append(CID)
      ENDIF
    ENDIF
  ENDIF
}
RETURN ConsumerIDList, NotDeterminedList
```

3. IMPLEMENTATION

In this section, we describe our implementation of the enforcement mechanism. We also describe the tools we used and their influence on the modeling of the ontology.

The enforcement mechanism (Figure 1) is composed of three modules that interact among each other and with the RACER reasoner [15]. The first module is the *ontology loader* which reads the ontology from an OWL file, converts it to DIG format [9] using the OWL API [10], requests an empty knowledge base from RACER[15], and submits the converted ontology as a set of *Tells*. DIG is a set of specifications for standardizing the interface to a DL reasoner. The second module, the *query loader*, receives a query, asserts it, and initiates the third module, the *query processor*. In the current implementation, the query loader reads the query from a file or from a screen field; however, a full implementation would run as a Web service where queries are received through the SOAP protocol. The third module, the query processor, performs the algorithm above and retrieves the list of consumer IDs.

There are several ontology development tools that have been mainly developed by academia including Protégé [11] by Stanford University, Swoop by the University of Maryland, and OilEd by the University of Manchester. We used Protégé for importing the UNSPSC ontology and for developing the rest of the taxonomies. Protégé provides several wizards and plug-ins that are very useful in saving time and checking correctness of work. Protégé can connect to any reasoner that supports the DIG

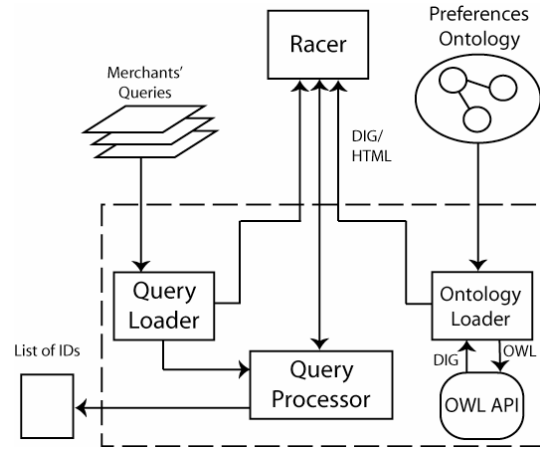


Figure 1: Semantic enforcement mechanism

interface. We used the RACER reasoner to check the consistency of the ontology during development time and as part of the enforcement mechanism.

While testing our implementation, we faced a problem with the DIG interface that led to several changes in the ontology. Although both OWL language and RACER support datatype properties and reasoning with them, we found that the interface to the reasoner, the DIG protocol, does not support datatype properties. As such, we had to change the modeling of the ontology to avoid using datatype properties.

4. PERFORMANCE EVALUATION STUDY

We conducted a performance evaluations study to examine the impact of the ontology size and the query size on the performance of the matching program. In designing the experiment, we used the preferences ontology with different sizes of consumer preferences. We also performed the experiment on different ontology sizes by reducing the preference ontology. In order to ensure the accuracy of measurements, the merchant queries were asserted dynamically and RACER was restarted between the different measurements to avoid loading multiple ontologies in its memory which could influence the accuracy of the results. The study was conducted on a Windows-based machine with Intel Xeon processor running at 2.4GHz and with 2 GB of RAM.

It is important to notice that, first, the performance of reasoning algorithms is not comparable to the highly optimized DBMSs. Even though most existing studies found that the performance of these algorithms is a function of the size of the knowledge base [4], we found the query size to be more significant (Figure 2). Therefore, it is important that the matching algorithm be designed to avoid any unnecessary matching.

5. RELATED WORK

Most of the recent work in this area has addressed the use of semantics to address the implications of the Semantic Web technology on data representations and the inadequate protection that needs to be improved. Stoica and Farkas [1] discussed the problem of unauthorized inferences in applications based on the Semantic Web where equivalent concepts in unclassified taxonomies can be used to draw conclusions about other terms in classified taxonomies. While this work is one of few that use

inferences and reasoning as part of the solution, it clearly addresses a different problem.

Damiani et al. [2] proposed extending policy languages by utilizing Semantic Web languages. They addressed the limitations of employing simple identifiers as policy components, e.g., to refer to subjects and objects. They extended XACML [8] and presented an architecture to illustrate the use of a semantically aware policy. While this work is a good first step towards full utilization of the Semantic Web power, it is based on the syntactical capabilities of RDF. The data retrieval is based on XQuery which does not perform any reasoning. The authors presented another work [3] on enhancing the P3P privacy policy. Since P3P does not utilize semantic knowledge, they extended it by building a consumer preferences ontology. However, they did not discuss how this ontology could be used in the reasoning process.

6. SUMMARY

In this paper, we presented an approach to represent consumer preferences using semantic data models and to reason about their matches to merchant queries using DL reasoning techniques. It is important to notice that the approach presented in this paper can be generalized to different information retrieval purposes especially when retrieval involves semantic matching. Finally, our implementation shows that there is much work needs to be completed in term of the interface to the reasoners, and in optimizing reasoning algorithms.

7. REFERENCES

- [1] A. G. Stoica and C. Farkas, "Ontology guided Security Engine", In *the Journal of Intelligent Information Systems: 23(2) pp.209-223*, 2004.
- [2] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, "Extending Policy Languages to the Semantic Web", In *Proceedings of the Fourth International Conference on Web Engineering*, Munich, 2004.
- [3] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, "Semantic-Aware Privacy and Access Control: Motivation and Preliminary Results", In *Proceedings of the first Italian Semantic Web Workshop: Semantic Web Applications and Perspectives (SWAP)*, Ancona, Italy, 2004.
- [4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, "The Description Logics Handbook: Theory, Implementation and Applications", *Cambridge University Press*, UK, 2003.
- [5] M. K. Smith, C. Welty, and D. L. McGuinness, Editors, "OWL Web Ontology Language Guide" A *W3C Recommendation*, Latest version available at <http://www.w3.org/TR/owl-guide/>, February 2004.
- [6] M. Youssef, "Semantically Enhanced and Efficient Location Services for Preserving Mobile Consumer's privacy", A Ph.D. thesis, Rutgers University -Newark, Available at <http://cimic.rutgers.edu/~youssefm/dissertation.pdf>, 2005.

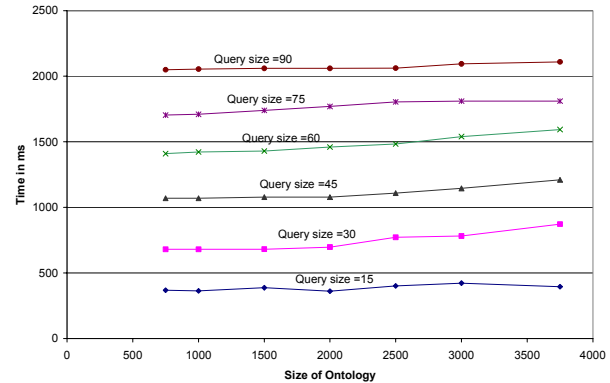


Figure 2: Performance of matching using RACER for query sizes from 15 to 90

- [7] M. Youssef, V. Atluri, and N. R. Adam, "Preserving mobile customer privacy: an access control system for moving objects and customer profile", In *Proceedings of the 6th International Conference on Mobile Data Management (MDM05)*, Ayia Napa, Cyprus, May 2005.
- [8] Organization for the Advancement of Structural Information Standards (OASIS), "eXtensible Access Control Markup Language", Available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, 2004.
- [9] S. Bechhofer, "The DIG Description Logic Interface: DIG/1.1", Available at <http://dl-web.man.ac.uk/dig/2003/02/interface.pdf>, 2003.
- [10] S. Bechhofer, P. Lord, R. Volz, "Cooking the Semantic Web with the OWL API", In *Proceedings of the 2nd International Semantic Web Conference, ISWC*, Sanibel Island, Florida, October 2003.
- [11] Stanford University, "The Protege Project", Available at <http://protege.stanford.edu>, 2005.
- [12] The World Wide Web Consortium, "RDF Primer", Available at <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, 2004.
- [13] The World Wide Web Consortium, "Semantic Web", Available at <http://www.w3.org/2001/sw/>, 2001.
- [14] United Nations Development Program, "United Nations Standard Products and Services Code (UNSPSC)", Available from <http://www.unspsc.org/>, April 2005.
- [15] V. Haarslev, R. Moller, "RACER Users's Guide and Reference Manual", Available at <http://www.cse.concordia.ca/%7Ehaarslev/racer/racer-manual-1-7-19.pdf>, 2000.