PACIS 2010 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2010

Ontology-Based Privacy Protection in Location Commerce

Seng-cho T. Chou
National Taiwan University

Timon C. Du
The Chinese University of Hong Kong

Chia-ho Yu
E.Sun Bank

ONTOLOGY-BASED PRIVACY PROTECTION IN LOCATION COMMERCE

Seng-cho T. Chou, Department of Information Management, National Taiwan University, Taiwan, R.O.C.

Timon C. Du, Department of Decision Sciences and Managerial Economics, The Chinese University of Hong Kong, Hong Kong

Chia-ho Yu, E.Sun Bank, Taiwan, R.O.C.

Abstract

Location commerce extends e-commerce through the provision of location-related activities, but this gives rise to greater concerns about privacy invasion. To encourage the smooth growth of location commerce, it is suggested that control over the sharing of intimate information be given back to the consumer. This study proposes an ontology-based privacy protection (OPP) framework that allows consumers to specify their own privacy preferences and then uses these preferences to determine whether or not a message from a merchant can be delivered to a consumer. We use ontology to structure the knowledge to simplify the framework and allow for the possibility of automation. The system is believed to be context-aware, as the location, time, service type, information type, and other contextual data are taken into consideration. We develop a prototype system for demonstration and experiment, and show that the framework design is feasible and has a reasonable performance.

Keywords: location commerce; privacy and security; ontology; context-aware system

1 INTRODUCTION

Location commerce evolved from mobile commerce as a means of defining mobile services for targeted customers in specific locations at specific times (Turban et al., 2008). Such services can be personalized, as they are based on knowledge about particular individuals (Tewari et al., 2003). This knowledge can be static or dynamic. Static knowledge includes information on age, salary, and gender, whereas dynamic knowledge includes information on location and time. However, distinct from mobile commerce and conventional e-commerce, location commerce focuses on location-based events, movement tracking, and positioning (Ngai et al., 2007).

There are many players involved in a location commerce application. For example, content providers supply messages to end users, service providers track and position the location of individuals, infrastructure providers offer interfaces to convert the protocols of different operators, network providers support wireless connection, and mobile devices afford identification and mobility. Location commerce therefore offers merchants a new business opportunity to provide services that are closely related to the location of consumers.

Unfortunately, many successful new commercial applications cause concerns about privacy invasion, which may subsequently withhold their development (Sarathy and Muralidhar, 2006). Privacy refers to the right to be left alone, and the protection of privacy involves maintaining individual seclusion from the public or society (Richard and Anita , 1992). Westin categorized privacy rights into political, personal, and group rights (Westin , 1996). Political rights ensure that an individual has space outside of any political involvement. Personal rights ensure an individual’s dignity, existence, and individualism and allow the individual release from social constraints. In a group, rights relate to the relationships and communication between individuals.

The right of privacy is enshrined in and used to protect the completeness of an individual from invasion by society or individuals (Goffman , 1959). It is passive. In fact, an individual can decide how to present him/her and what information he/she would like to release to others actively. In this way, he/she and can present differently to different parties to maintain his/her individualism and privacy. The right of privacy refers to control. For example, H. T. Laurence in *American Constitution Law* considers that an individual deciding whether to release information to allow a search of their personal space is a decision about the control of their openness (Laurence , 1978).

This study proposes an ontology-based framework for privacy protection in location commerce. This framework creates a privacy-aware environment in which a consumer can set privacy preferences through a mobile device for storage in the databases of Internet Service Providers (ISP). When a merchant wants to send a message to a consumer, it submits the message content and recipient (consumer) information to the ISP serving the consumer, as identified through a mobile device (such as a phone). The ISP checks the recipient’s privacy preferences to determine whether to grant permission to the merchant to convey the information, and then delivers the message (such as a coupon) to the consumer’s mobile device, if authorized. This design allows consumers to control the way in which they appear to merchants, while at the same time allowing location commerce to progress smoothly.

In the design, ontology is used to model the knowledge for simplicity and automation, and contextual information is acquired to facilitate location commerce. The privacy preferences are written in rule form to allow the construction of a rule-based system and the adoption of declarative logic reasoning.

The remainder of the paper is organized as follows. Section 2 reviews the literature on ontology and context. The framework, which is called the ontology-based privacy protection (OPP) framework, is discussed in Section 3. Section 4 demonstrates the system and presents some experiments to determine system performance. The conclusion is presented in Section 5.

2 ONTOLOGY

Ontology is a way of representing concepts and the relationship between concepts. It can be used to structure current knowledge and to reason new knowledge. It is also a means of improving machine automation (Berners-Lee, 2001). Ontology is defined by classes, following the concept of object-oriented languages. The relationships between classes can take the form of associations and specifications. Associations build the relationship between classes, whereas specifications describe the explicit attributes among classes (Du and Li, 2007).

Ontology provides the foundation for the Semantic Web, which is a new form of knowledge representation on the Internet originally proposed by W3C. The Semantic Web describes knowledge using ontological language and tools. A well-known ontology that uses XML to depict semantic knowledge (http://www.w3.org/TR/xml/) is the Resource Description Framework (RDF), but unfortunately it cannot be used for knowledge inference (Manola and Miller, 2004). To enhance the reasoning capability of the Semantic Web, W3C proposed a descriptive logic language called Ontology Web Language (OWL) (Smith, 2004). OWL adds the entity properties, operators, and concepts of classification, such as disjointedness, cardinality, and enumerated class, to the RDF. The constructs of OWL thus provide the elements for both reasoning and the detection of inconsistency.

In general, there are three sub-languages of OWL with incrementally descriptive capabilities. OWL-Lite is used to define simple binary statements, OWL-DL (Description) Logic is used for formal expressions when the outcome can be derived (computational completeness) and determined (decidability), and OWL-Full is designed for RDF users who require full freedom to describe knowledge. The computational completeness of OWL-Full is not guaranteed. In this research, OWL-DL is selected due to the need for complex knowledge expression and structured knowledge inference capability.

Context is the information that describes the specific circumstance of an object (Dey and Abowd, 2000). The object can be a human, a location, an entity, or the interaction between a user and an application. A system is said to be context aware if the program action can adapt to the context (Baldauf et al., 2007). A context model should include the five dimensions of who, where, what, when, and how by collecting information on actors, locations, time, activities, and devices, respectively (Gu et al., 2004)(Wang et al., 2004).

There are three ways to model contextual information. The first is the application-oriented approach, in which contextual information is acquired for designated applications. For example, Context Toolkit transfers low-level environmental data into XML and HP's Cooltown develops a Web context and links to a corresponding URL (Kindberg and Barton, 2001). The second is the model-oriented approach, which normally conceptualizes circumstances into a canonical model that it then fits into an entity relationship (ER) model in a relational database (Henricksen et al., 2005). The third is the ontology-oriented approach, which builds ontology for a specific domain. For example, (Chen and Finin, 2003) defines an OWL-based ontology for ubiquitous computing on campus, and (Ranganathan and Campbell, 2003) uses middleware for DAML and OIL platforms.

In location commerce, privacy preferences may be changed according to the circumstances. A location commerce system needs to be able to model these preferences and capture the contextual data, and then use both to determine whether or not a message can be delivered or a service provided. This study uses ontology to model the preferences and circumstances. A similar approach was taken in (Chen et al., 2004), in which a Context Broker Architecture was constructed to protect privacy by allowing users to define policy rules using OWL and descriptive logic. In the architecture, each policy rule has a creator and can permit or forbid actions. An action includes actors, the recipients of the action, the target of the action, and additional information such as location and time. We use the same basic approach, but with some modifications, such as the separation of knowledge types and the enhancement of the inference engine, to improve the design.

There are several instances of the applications of ontology using contextual data, such as the well-known examples of the E-Wallet project and PeCAN. E-Wallet, which was developed by (Gandon and Sadeh. 2003), selects and deposits Semantic Web services automatically in an open environment. E-Wallet uses OWL to define three layers of knowledge. The internal layer is the core that includes static information and some basic rules. The second layer comprises a service to provide matching functions. The outer layer contains the privacy rules that determine access rights and abstract the query. E-Wallet has many task-specific agents that work collaboratively under different circumstances. For example, an agent can help to make appointments or plan a trip, but to complete the task needs to access information from others, the retrieval of which is confined by privacy regulations. Both static information and dynamic information need to be shared through the Internet, which is achieved in several steps. These include asserting the query context, identifying the information elements, pre-checking the access rights, fetching the static knowledge, initiating external services, post-checking the access rights, granting authorization, and so on (Gandon and Sadeh. 2003). In this way, E-Wallet can evaluate the right of person A to access the position of person B. However, it does not clearly define the information hierarchy, and is unable to link to the Platform for Privacy Preference (P3P) set up by the World Wide Web Consortium (W3C).

PeCAN (Personal Context Agent Networking), which was developed by (Jutla et al, 2006), co-opts both server and client to enhance trust and privacy protection. Similar to E-Wallet, PeCAN uses OWL to define the ontology and data structure, but out-performs E-Wallet because it is compatible with the P3P. On the P3P, privacy protection is context aware, and both preferences and beliefs are considered (Cranor , 2003). Beliefs can pertain to organizations, regulations, transactions, data, roles, sectors, stakeholders, and so on. In PeCAN, beliefs guide the comprehension of external entities, and help software agents make a decision as to whether or not to precede with an action. The PeCAN framework includes internal agents and external agents. The former manage personal preferences, whereas the latter are responsible for monitoring external events, such as network liability and privacy laws. However, PeCAN was developed for conventional e-commerce applications, which means that that it does not consider the information that is critical to location commerce, such as time and location.

3 ONTOLOGY-BASED PRIVACY PROTECTION FRAMEWORK

This study proposes an ontology-based privacy protection (OPP) framework for location commerce. The framework is composed of a knowledge base, a decision module, and several other components, as shown in Figure 1. The data about privacy preferences, context, and privacy rules are stored in the knowledge base, and the decision module comprises an inference engine and evaluation functions. The framework allows a user to specify general privacy rules through a form-based GUI. The rules describe situations in which users will be willing to accept messages from merchants. The inference engine considers more complex analogous situations. An action will be performed only when the current situation satisfies both the preferences and the context. Although the system is designed to maintain the fewest possible rules, new rules for unspecified circumstances can be added to improve system performance.

The decision support module performs the functions of rule inference and rule evaluation. The inference engine and the knowledge base are separated in the system design to differentiate the deduced knowledge from the stored facts to save computation time. In practice, this means that the evaluation function first checks the contextual information about a location against the facts before deducing any new knowledge.

The data access layer is a unified layer to maintain consistent data retrieval and independence from the rule inference and knowledge update functions. The timer function is the heart of the framework, and activates the whole process by checking whether or not any inquiries need to be transmitted. If inquiries are pending, then the system checks the contextual data, privacy preferences, and rule templates through the data access layer. The information is then sent to the rule transfer function, which converts it into the right format for the decision-making module. If permission is given to send the message to the consumer, then the timer transfers it from the merchant to a push function, which then delivers the message to the consumer. If the consumer does not want to receive such messages in the future, then the preferences can be changed using the Feedback function. A new rule is then generated and stored in the consumer's privacy preferences.

The contextual data are obtained from an individual's circumstance, and can be divided into static and dynamic portions. The static data are data that cannot be changed over the time, such as location, information provider, and message type. In contrast, the dynamic data are data that are updated frequently, such as position, request time, and service type. The static data can be pre-stored and optimized for a given query system, whereas the dynamic data need to be processed in real time.

Consumers can indicate their privacy preferences to the system. At the initial stage, the system deposits these preferences into the knowledge base and optimizes them for later inquiry. When a request from a merchant is received, the system then loads the personal privacy rules and decomposes them into different contextual factors, converts the privacy rules into Ontology Web Language-Query Language (OWL-QL) using a rule template, and assesses the contextual factors using both OWL-QL and the evaluation function. If the conditions of a rule (or rules) are satisfied, then the rule is supported. If the rule is positive (negative), then permission will be granted (denied) to the merchant.

The evaluation function that determines whether or not a merchant can access a consumer's information involves many steps. First, the privacy rules of a consumer in the rules set are classified based on the circumstances. The parameters for each circumstance (e.g., position) are only recorded once. When a situation is encountered, the system checks whether or not the circumstances match those recorded. If they do not match, then the parameters of the current circumstances will be compared against the rules in the knowledge base. To accomplish this step, the system first sorts the privacy rules based on personal privacy preferences, and then verifies whether or not all of the current parameters satisfy the privacy rules. If the rules are satisfied, then permission can be granted to the merchant if the rules are positive, and denied if the rules are negative.

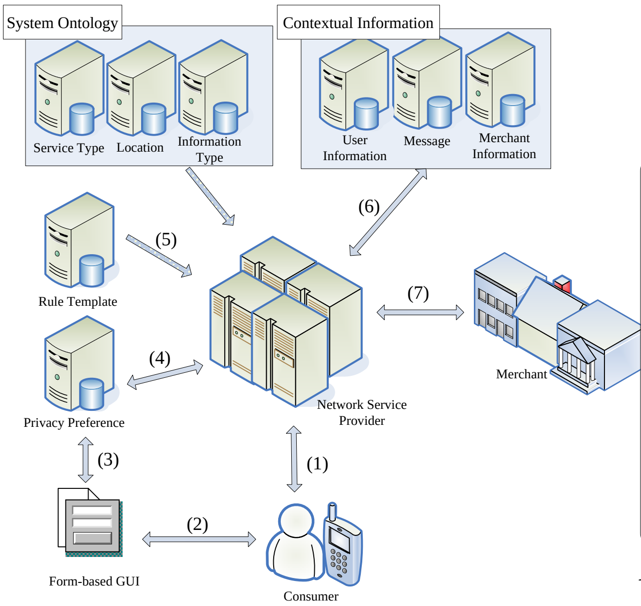

Figure 2 An Example of Location Commerce.

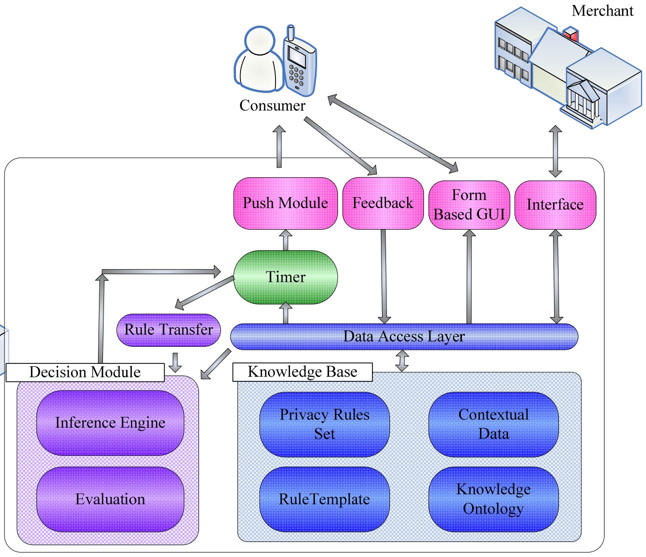

Figure 1 Ontology-based Privacy Protection (OPP) Framework.

Once permission is granted, the action plan allows the merchant to collect information about the consumer or to send the consumer a message. Consumers can also edit their privacy preferences through the form-based GUI. As mentioned, these preferences are facts that are converted into OWL format, and include many privacy rule sets, each of which is a collection of privacy rules that specify the preferences in certain contextual circumstances, including position, service provider, message type, time, and so on. The privacy rules are presented in order of priority, with the more general rules gaining higher priority. The system verifies the conditions starting from the first rules. In brief, the arrangement of the privacy preferences can be presented as follows.

Personal privacy preference = {Privacy rule set}*

Privacy rule set = {Privacy rules}* + Actions

Privacy rules = {Sequence, Contextual conditions}*

Contextual condition = {Position, Message type, Service type, Time, …}.

The privacy rules within the privacy rules set are disjointed, which means that as long as one rule is satisfied, the privacy rule set is supported. However, the predicates in the proposition of a privacy rule are conjunctional.

A scenario in which ontology-based contextual-aware location commerce is applied as shown in Figure 2 and explained in detail as follows.

(1) A consumer connects to the network using a mobile device and registers with the network service provider. The network service provider verifies the privacy preferences of the consumer, or user, and grants permission for messages to be delivered to the user. If the consumer decides that the delivered message is not of interest, then he or she can ask the network service provider not to deliver a similar message again.

(2) The consumer can access and edit his or her own current privacy preferences through the form-based GUI. When the consumer specifies the rules, he or she needs to provide contextual data, such as location, service type, message type, and time.

(3) The system retrieves the privacy preferences from the knowledge base and accepts the instruction from the consumer.

(4) Before transmitting a message to the consumer, the system verifies the consumer’s preferences. Based on these preferences and the contextual data, the system determines whether or not the message should be delivered to the consumer. If the consumer refuses to receive the message, then a new privacy rule is generated by the system to prevent the delivery of similar messages to the user in the future.

(5) The system transfers the implicit privacy rules and the consumer’s contextual data into a query language and predicates by referring to the rule template.

(6) The merchant gives the necessary information about the message to the network provider, including time, location, and content. The system deposits this information into a contextual database for later use. The system refers to the location of the consumer’s mobile device and the transmitted information between the network provider and the device to update the contextual data.

(7) The merchant informs the network service provider of the message parameters, such as time, location, and content. The network service provider then transmits the message based on the instructions from the merchant. The system informs the merchant whether or not the message has been accepted or rejected by the user, and the merchant can change the content or contextual information accordingly.

4 DEMONSTRATION

We have developed a prototype for demonstration and experimentation. The inference engine and the knowledge base were developed using RacerPro (now renamed ABox and Concept Expression Reasoner Professional, http://www.racer-systems.com/), which uses descriptive logic in a similar way to OWL. Compared with other commonly adopted engines, such as FaCT++ (http://owl.man.ac.uk), Pellet (http://www.mindswap.org), and KAON2 (http://kaon2.semanticweb.org), RacerPro is the only commercial inference engine that can optimize ontology entities and provide a socket for HTTP. This gives the system better scalability, which allows distributed and remote servers to be added to the system. RacerPro can process documents in OWL Lite and OWL DL and edit the knowledge base, and can also be used to query and infer the relationship between classes and the entities within a class. The built-in interface language of RacerPro, JRacer, allows automatic communication between the OPP system and the inference engine using Java. The Push Module is written by the push registry in Java.

In Figure 3, a consumer sets up a preference such as “I can receive 3C promotion at Song [a location] between 3:00 pm to 4:00 pm” through form-based GUI.

Rule 1: {15:00~16:00, Song, 3C promotion}.

The preference is transformed and converted into an OWL-QL format rule. The declarative rule is

Individual(a:Rule1 type(a:PrivacyRule)

value(a:sequenceNumber a:”1”)

value(a:startDateTime a:”15:00”)

value(a:endDateTime a:”16:00”)

value(a:locateIn a:”Song”)

value(a:serviceType a:”3C promotion”).

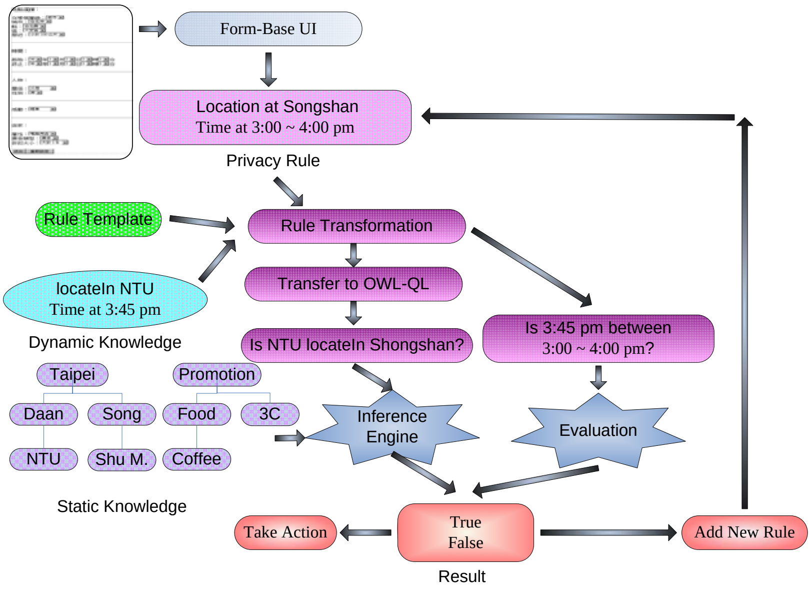

Figure 3 An Example of Sending a Promotion to a Consumer.

We assume that a merchant wants to send a coffee promotion to the consumer at 3:45 pm at Song, and that the consumer is at the NTU campus (dynamic knowledge). The OPP system checks whether or not 3:45 pm is within the time frame of 3:00-4:00 pm and finds out from the ontology that “NTU is located at Daan”.

Individual(a:CurrentContext type(a:Context)

value(a:currentTime a:”3:45”)

value(a:locateIn a:”NTU campus”)

value(a:serviceType a:”Coffee promotion”)).

In this case, the request is rejected, as NTU is located at Daan, not Song. The consumer can add a new rule to allow the promotion to be delivered in the future if required.

We further assume that two more rules are specified.

Rule 2: {09:00-20:00, NTU Campus, 3C promotion}.

Rule 3: {Shu Market, Coffee promotion}.

The system pre-processes these rules and incorporates the same contextual data.

Time {15:00-16:00, 09:00-20:00}.

Location {Song, NTU Campus}.

Service type {3C promotion, Coffee promotion}.

The contextual data are then converted into the knowledge base format for inquiry using a rule template.

（(x) ((NTU Campus) locateIn (Daan))） ➔ True

（(x) ((NTU Campus) locateIn (Song))） ➔ False

（(x) ((Coffee) subclassOf (Food))） ➔ True

（(x) ((Coffee) subclassOf (3C))） ➔ False

The predicates are written as follows.

If (DataTime.Compare(“15:45”, “15:00”) > 0 and (DataTime.Compare(“15:45”, “16:00”) < 0) ➔ True

If (DataTime.Compare(“09:00”, “12:00”) > 0 and (DataTime.Compare(“20:00”, “12:00”) < 0) ➔ True

Based on the contextual data, the following results are obtained.

Time {15:00-16:00 = True, 09:00-20:00 = True} Current value: 15:45.

Location {Daan = True, Song = False} Current value: NTU campus.

Service type {Food= True, 3C = False, Coffee= True}, Current value: Coffee.

To evaluate the privacy preferences, we first check the support for each contextual parameter, and then map the parameters to the privacy rules. If all of the contextual parameters of a rule are found to be true, then we consider the rule to be supported and service is granted. If the consumer happens to move to Shu Market, for example, the location parameter becomes

Location {Song = True} Current value: Shu Market.

The third rule is still supported, as Shu Market is located at Song, coffee is a food, and the time requirement is satisfied. Protégé-OWL can provide a GUI interface to create OWL individuals using classes and properties, and can also implement external inference engines to verify the correctness of ontology. Moreover, the Protégé-OWL library contains an open-source Java library that can be used to process OWL and the Resource Description Framework (RDF), thus providing interoperability between applications that exchange machine-understandable information.

We validate the performance of the system through experiments that estimate the computation time of the OPP system with different settings, including different numbers of privacy rules and different contextual situations involving the three contextual dimensions of location, service type, and message type. To conduct the experiments, a program called *CouponReceiver* was developed in Java and installed in a mobile device to allow communication between the device and the OPP system. The user is asked through *CouponReceiver* whether or not permission is granted to send a message from the OPP system. If the user declines, then the system adds a new privacy rule to remember the current circumstance and stores it in OWL format.

A rule is composed of a hierarchy of ontology entities. Figure 4 shows an example of locations, message types, and service types represented in a hierarchical structure. The higher levels of the hierarchy contain more general entities, whereas the lower levels contain more specific entities. In the experiments, the ontology is automatically generated and can be arranged into three levels of privacy preferences. The privacy rules in level 1 are general entities, whereas the rules in level 2 are relatively more specific, and the rules in level 3 highly specific. Thus, rule level 1 is more general than level 2, and level 2 is more general than level 3.

The simulation is repeated over 1,000 runs. In each run, we set up different numbers of ontology entities and one of five different numbers of rules, that is, 10, 20, 40, 80, and 160. We first determine the number of ontology entities and privacy rules in the system, and then generate the entities (location, service type, and message type). Next, we simulate various circumstances with different contextual parameters, such as the locations of the user, the services provided to the user, and the type of messages sent to the user, and then generate different levels of rules accordingly. We then determine whether or not the message will be accepted in the current context based on the predefined privacy rules. If permission is granted, then we consider the privacy rules to match the current context, and record the computation time. The experiments were run using an Intel Pentium 4 2.4 GHz processor, 1.25 GB of RAM, Windows Server 2003 Enterprise Service Pack 2, MySQL 5.0.41, Protégé 3.2.1, RacerPro 1.9.0, and Java SE Runtime Environment 6 Update 1.

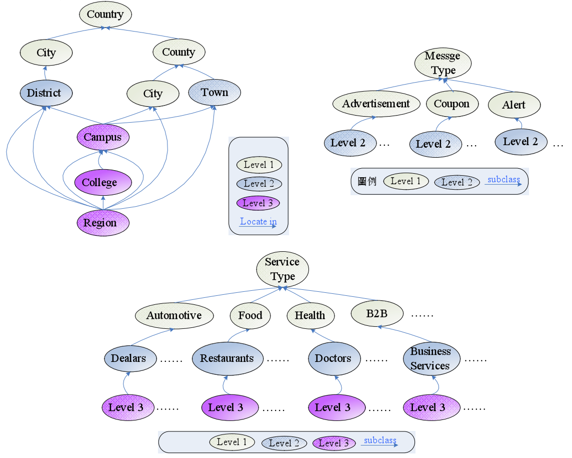

Figure 4 Ontological Hierarchies of Locations, Message Types, and Service Types.

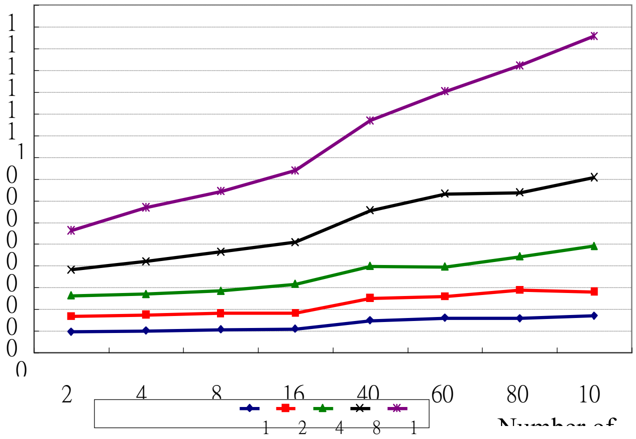

Figure 5 Computation Time in Seconds for Different Numbers of Privacy Rules with Increasing Number of Ontology Entities.

Figure 5 shows the computation time in seconds that the system took to implement five different numbers of privacy rules (10, 20, 40, 80, and 160) for different numbers of ontology entities (200, 400, …, 10000). The computation time is taken as the average over 1,000 experiment runs. The time increases proportionally as the number of ontology entities increases. When the number of ontology entities reaches 10,000, the computation time for the case with 160 rules is 1.5 seconds, which is acceptable for implementation in a real-world situation.

It is also noticed that the computation time is smaller when the rules are more general, as shown in Figure 6. This is probably because the system optimizes the rules to prevent duplication and keeps the general rules at the front. The general rules cover more circumstances than the specific rules, and are also evaluated earlier. The computation time is consequently decreased, because the outcome can be determined more quickly.

We also attempted to determine the probability of contextual information being defined in the privacy rules. This is the *coverage probability* that a circumstance is incorporated in a rule or that a rule is evaluated as "true." The coverage probability is defined as the ratio of instances of a privacy rule being supported to the total number of experiment runs. In Figure 7, it can be seen that the coverage is proportional to the number of rules but not the number of ontology entities. When the number of privacy rules is 160, all of the circumstances are covered, and more than 90% coverage is achieved when there 80 rules are defined.

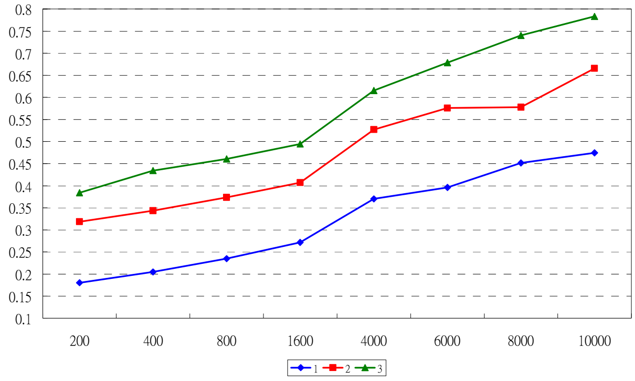

Figure 6 Computation Time with Different Levels of Privacy Preferences.

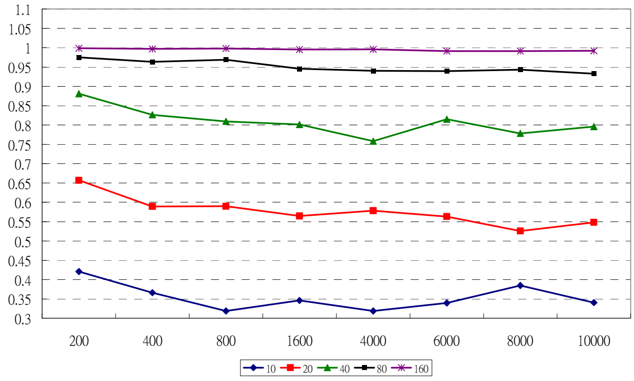

Figure 7 Coverage Probability with Different Numbers of Rules

Figure 8 shows that the coverage varies with the generalness of the rules: the more general the rules, the higher the coverage. For the most general rules, that is, the level 1 rules, the coverage ranges from 82% to 87%, but for the most specific rules (level 3) the coverage is still higher than 60%.

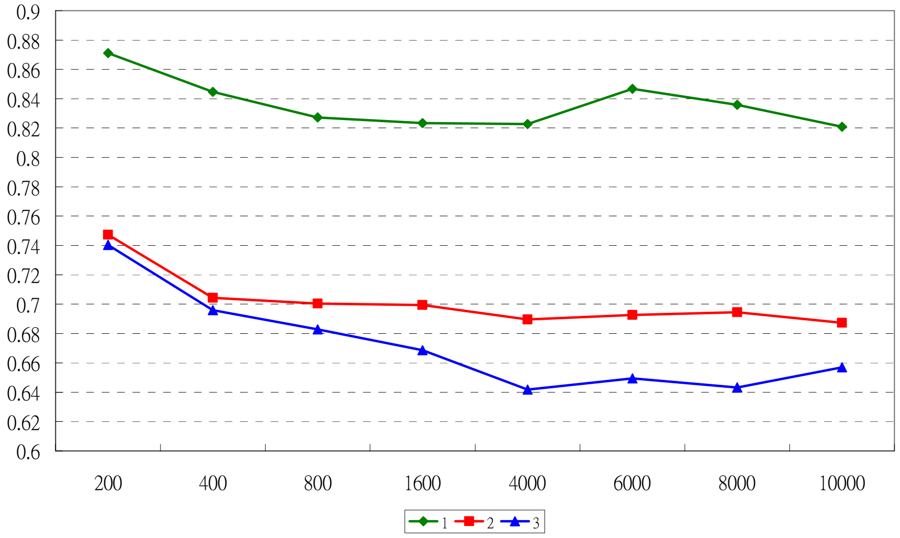

Figure 8 Coverage Probability with Different Levels of Rules.

5 CONCLUSION

This study proposes an ontology-based privacy protection (OPP) framework for context-aware location commerce. In the design, we integrate dynamic context information with privacy rules and convert them into an ontology query language for checking against the preloaded static information. In this way, the computation cost of static information inquiries is lowered. The differentiation of deductive cases (those in which the results need further deduction) and cases in which the privacy rules are confirmed eases the loading on the inference engine. The framework is validated with a prototype, and several experiments have been conducted to verify the system performance. The system is reasonably efficient, and takes 1.5 seconds to process an action when the number of ontology entities is 10,000 and the number of privacy rules is 160.

The system allows consumers to determine whether a message is delivered to them or whether their information can be provided to a merchant. Giving control to the consumer in this way eases concerns about privacy, and should allow the further development of location commerce.

References

[1] E. Turban, J. K. Lee, D. King, J. McKay, and P. Marshall, *Electronic Commerce 2008*, New Jersey: Prentice Hall, 2008.

G. Tewari, J. Youll, and P. Maes, “Personalized Location-based Brokering Using an Agent-based Intermediary Architecture”, *Decision Support Systems*, Vol. 34, no. 2, January 2003, pp. 127-137.

E. W. T. Ngai, T. C. E. Cheng, S. Au, and Kee-hung Lai, “Mobile Commerce Integrated with RFID Technology in a Container Depot”, *Decision Support Systems*, Vol. 43, no. 1, February 2007, pp. 62-76.

R. Sarathy and K. Muralidhar, “Secure and Useful Data Sharing”, *Decision Support Systems*, Vol. 42, no. 1, October 2006, pp. 204-220.

C. T. Richard and L. A. Anita, *Privacy: Cases and Materials*, John Marshall Publishing Company, Houston, TX, 1992.

A. F. Westin, “Science, Privacy, and Freedom: Issues and Proposals for the 1970s.”, *Columbia Law Review*, vol. 66, no. 7, 1966, pp. 1205-1253.

E. Goffman, *The Presentation of Self in Everyday Life*, Anchor-Doubleday, New York, 1959.

H. T. Laurence, *American Constitution Law*, 1978, pp. 886-990.

T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, May 2001.

T. Du and E. Li, “Building Dynamic Business Process in P2P Semantic Web,” *Advances in Electronic Business II*, edited by Eldon Li and Timon Du. 1st ed. pp. 180-198. PA, Hershey: IDEA Production Co., 2007.

F. Manola and E. Miller, “RDF Primer”, World Wide Web Consortium, 2004, http://www.uazuay.edu.ec/bibliotecas/conectividad/pdf/RDF%20Primer.pdf.

M. K. Smith, C. Welty, and D. L. McGuinness, “OWL Web Ontology Language Guide”, World Wide Web Consortium, 2004, http://www.w3.org/TR/owl-guide/.

A. K. Dey and G. D. Abowd, “Towards a Better Understanding of Context and Context-awareness”. *Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness,* affiliated with the CHI 2000 Conference on Human Factors in Computer Systems, 2000.

M. Baldauf, S. Dustdar, and F. Rosenberg, “A Survey on Context-aware Systems”, *Int. J. Ad Hoc and Ubiquitous Computing,* Vol. 2, No. 4, 2007, pp. 263-276.

T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, “An Ontology-based Context Model in Intelligent Environments“, *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004.

X. H. Wang, D. Q. Zhang ,T. Gu, and H. K. Pung, “Ontology Based Context Modeling and Reasoning using OWL”, *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW’04)*, 2004.

T. Kindberg and J. Barton, “A Web-based Nomadic Computing System”, *Computer Networks*, vol. 35, no. 4, 2001, pp. 443- 456.

K. Henricksen, R. Wishart, T. McFadden, and J. Indulska, “Extending Context Models for Privacy in Pervasive Computing Environments”, *PerCom 2005 Workshops*, IEEE Press, 2005, pp. 20- 24.

H. Chen and T. Finin, “An Ontology for a Context Aware Pervasive Computing Environment”, *IJCAI Workshop on Ontologies and Distributed Systems*, Acapulco MX, August 2003.

A. Ranganathan and R. H. Campbell, “A Middleware for Context-Aware Agents in Ubiquitous Computing Environments”, *Lecture Notes in Computer Science*, 2003, ISSU 2672, pp. 143-161.

H. Chen, T. Finin, and A. Joshi, “A Pervasive Computing Ontology for User Privacy Protection in the Context Broker Architecture”, TR-CS-04-08, University of Maryland, 2004.

F.L. Gandon and N. M. Sadeh, “A Semantic E-Wallet to Reconcile Privacy and Context Awareness”, ISWC 2003, Springer Berlin, 2003, pp. 385-401

D. N. Jutla, P. Bodorik, Y. Zhang, “PeCAN: An Architecture for Users’ Privacy-aware Electronic Commerce Contexts on the Semantic Web”, *Information Systems*, vol. 31, no. 4, 2006, pp. 295-320.

L. F. Cranor, “P3P: Making Privacy Policies More Useful”, *Security and Privacy Magazine*, IEEE Press, vol.1, issue 6, 2003, pp. 50- 55.