Mobile In-Store Personalized Services

Jun Li, Ismail Ari, Jhilmil Jain, Alan H. Karp, Mohamed; Dekhil

HP Laboratories
HPL-2009-48

Keyword(s):
mobile shopping assistant, cross-organizational service. personalized service, event-based service delivery. authorization-based access control

Abstract:
The Mobile Shopping Assistant (MSA) is a mobile application platform to deliver real-time, in-store, and personalized services, such as personalized product offerings and in-store customer advisory support, to improve the shopping experiences of in-store customers. The service delivery network that powers the MSA involves retail stores and their business partners such as manufacturers. This paper presents the core technologies that we developed in this cross-organizational service network to support the MSA and its personalized services, with focus on service delivery, customer behavior understanding and information sharing. Real-time, asynchronous, event-based service delivery allows customers, stores and manufacturers to deliver and consume the services in a loosely coupled manner. Service response tracking enables the stores to construct a comprehensive view of a customer's in-store shopping behavior. Finally, the cross-organizational authorization-based access control mechanism effectively enforces information sharing between the stores and their partners.

External Posting Date: March 6, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: March 6, 2009 [Fulltext]

Mobile In-Store Personalized Services

Jun Li, Ismail Ari, Jhilmil Jain, Alan H. Karp, and Mohamed Dekhil
Hewlett-Packard Laboratories, Palo Alto, California
{jun.li, ismail.ari, jhilmil.jain, alan.karp, mohamed.dekhil}@hp.com

Abstract

The Mobile Shopping Assistant (MSA) is a mobile application platform to deliver real-time, in-store, and personalized services, such as personalized product offerings and in-store customer advisory support, to improve the shopping experiences of in-store customers. The service delivery network that powers the MSA involves retail stores and their business partners such as manufacturers. This paper presents the core technologies that we developed in this cross-organizational service network to support the MSA and its personalized services, with focus on service delivery, customer behavior understanding and information sharing. Real-time, asynchronous, event-based service delivery allows customers, stores and manufacturers to deliver and consume the services in a loosely coupled manner. Service response tracking enables the stores to construct a comprehensive view of a customer's in-store shopping behavior. Finally, the cross-organizational authorization-based access control mechanism effectively enforces information sharing between the stores and their partners.

1. Introduction

For retailers, a rich and engaging in-store customer shopping experience is key to retaining loyal customers and increasing the value of customers' shopping baskets. Variety in product offerings, in-store assistance, and other factors shape the customer's perception of how much attention they receive from the store and whether the offers they receive are among the best. Due to increased proliferation, mobile devices can potentially become the ubiquitous personalized service delivery platforms that retailers can leverage. Unlike traditional online shopping from the desktop-based environment, the in-store mobile application platform has the potential to deliver personalized, store-specific, and advisory-based features in real-time for in-store customers when they stand in the aisles next to the products that they want to learn more about or buy. We call such a mobile application platform the Mobile Shopping Assistant (MSA).

Let's assume that Alice is a customer visiting a store to find a gift for her brother. She uses her MSA to scan a barcode at the entrance to the store labeled "Promotions", and is told she'll get a 10% discount if she buys a TV today because she is a frequent buyer. She'd been thinking about buying one, so she goes to that department. She scans the barcodes of several TVs to check features and instantaneously sees the product descriptions. She doesn't understand such technical terms as "progressive scanning", so she invokes the MSA product advisory service and is told that a store employee named Joe will be there in three minutes to help her. After Joe answers her questions, Alice scans a 40-inch HP LCD TV and finds out that she's entitled to a $100 rebate because her company buys HP servers. Although she had no intention of buying a TV when she entered the store, this deal is just too good to pass up. On the way home, she remembers that she forgot all about her brother's gift.

Illustrated above are the two exemplar features of MSA, personalized product offering and advisory support. The retailer requires a comprehensive and holistic understanding of its customers to deliver useful personalized services. At the backend, the service platform includes not only the store but also the manufacturers that have their products shelved in the store. The service platform is primarily owned by the store and the store opens it up to allow the manufacturers to be connected to the store's customers indirectly. More specifically, manufacturers can provide to the store, not only updated product information, but also their registered customers' information from which the store can better know the customers (especially when they are newcomers without a purchase history). The store in return shares customer purchase intent, customer shopping behavior, and other information, such that manufacturers can provide real-time, personalized, location-specific product offerings to the store's customers, and closely monitor their campaign progress at the store level.

This paper presents a service-oriented solution architecture that enables in-store customers, the store, and the manufacturers to participate in real-time personalized service delivery. In this cross-organizational and dynamic service environment, we focus on the core technologies that include real-time event-based channels to deliver personalized services to the customers, mobile session-based customer response tracking that provides data to gain insights into customer shopping behavior, and cross-organizational authorization-based access control on information sharing that aims to enforce service integrity, *i.e.,* what can be shared is what was agreed in business contracts, no more and no less.

The rest of the paper is organized as follows. Section 2 introduces the solution architecture with cross-organizational service interactions. Sections 3, 4, 5 detail asynchronous event-based service delivery, customer response tracking and cross-organizational information sharing through an authorization-based access control mechanism respectively. We present the system prototype in Section 6, and compare it with related work in Section 7. Conclusions and future work are provided in Section 8.

2. System architecture to support personalized services

The MSA service platform is a service-oriented solution architecture owned by the store and shared with its manufacturer partners. Shown in Figure 1, each personalized service feature of MSA has its back-end application service counterpart. Each application service further makes use of core services that include publish/subscribe based event notification service, session management service, personalization related service, and security and privacy related service that involves authentication (for the device application to access its back-end application service), access control and privacy control on information held in the service platform. The service platform also includes composite web services from well-defined business processes, such as *Product Inventory Management* and *Product Offer Lifecycle Management*.

The service platform relies on *User Identity* (me, holding the device), *Location* (here, at this store), and *Timeliness* (now) to deliver personalized services. The customer reveals her registered identity to the store when she launches MSA device application. For MSA related service delivery, we only need the store identifier that uniquely identifies the store. Such information can be determined by a GPS sensor on the device, mobile operator's location service, or inferred from the device IP address assigned by the in-store Wi-Fi network. The customer's interest can be communicated to the MSA when she scans the product barcode or keys in the barcode number. In summary, the three parameters, *user identifier*, *store identifier* and *product identifier*, are key inputs to the service network.

Next we detail how individual personalized services can be developed by leveraging the core services that are illustrated in Figure 1.

2.1. Personalized product offerings

For the store to provide personalized product offerings, apart from the above three identified inputs, it requires customer profile information from *Customer Profile Management*. The *Customer Profile Management* deals with a wealth of information that can be either provided by customers voluntarily (such as phone number, residential address, *etc*.), or constructed from historical purchase transactions, historical MSA interactions with the store, *etc*. *Customer Intention Tracking* and *Response Tracking* help provide customer-store interaction information by tracking product interest expressed from the customers and their reaction to and interaction with the personalized service contents.

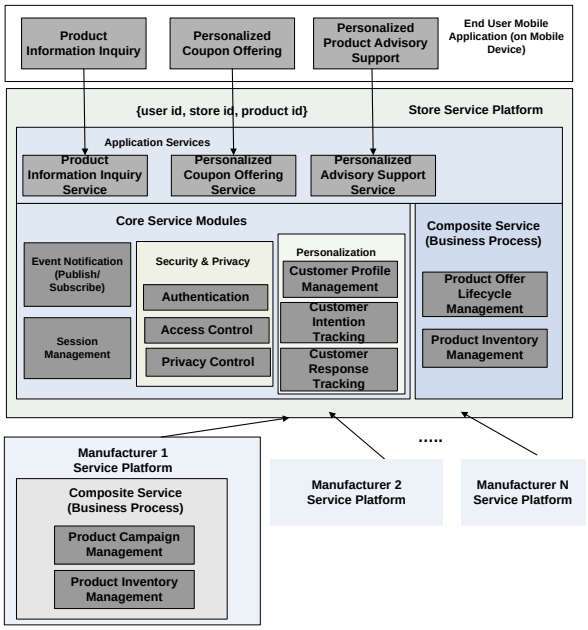

Figure 1. The MSA service platform that supports personalized services

As the participant to the service platform, the manufacturer would like the store to release as much customer profile information as possible (*e.g*., a phone number or zip code), and correlate such information with its own customer database (constructed, *e.g*., through product registration and warranty support). Furthermore, the *Customer Intention Tracking* and *Customer Response Tracking* information, if revealed with the necessary customer privacy protection, can help the manufacturers make personalized offers and tune their campaigns accordingly.

2.2. Personalized advisory support

To provide real-time in-store advisory support, once receiving the request from the mobile device, the store first uses customer profile information to evaluate customer importance (based on amount spent, items purchased, *etc*.), and accordingly schedules a store staff member to serve the customer. Second, the customer's past transactions related to the product that he or she intends to purchase now, at the store or directly from the manufacturer, can provide the store staff member an aggregated view of how the product will be used along with

historical purchases (*e.g.*, the LCD monitor on sale doesn't have a refresh rate fast enough for the game system she recently bought). This also provides the store opportunities to cross-sell and up-sell. Third, historical transactions can reveal brand preferences, thereby enabling the store to provide suggestions that better match the customer's preferences.

2.3. Our unique service delivery capabilities

The following sections will present the detailed techniques that highlight the uniqueness of the MSA solution as compared to other mobile-commerce applications that also provide personalized services. Our core techniques include:

- **Asynchronous Event-based Service Delivery**. A bi-directional but loosely coupled service channel allows (a) customers to express service interests, (b) the store or the manufacturers (or both) to subscribe to these interests, and (c) the store and the manufacturers to publish service responses based on their understandings of the customers, which are eventually routed to the customers.
- **Session-based Product Intent and Service Response Tracking.** It provides a rich set of raw information captured from the personalized services to facilitate customers' shopping behavior characterization.
- **Authorization-based Cross-organizational Access Control.** Its addresses information and service sharing in the dynamic and large-scale store-manufacturer service network, in which information sharing is dictated by contractual business relationships. Such relationships are further complicated due to sub-contracting [8].

3. Real-time event-based service delivery

Personalized coupon offering or product advisory support is delivered through an asynchronous event-based publish/subscribe system [6]. Following [12], in a publish/subscribe event-based system, a subscriber subscribes its interest to the event notification broker. A publisher publishes its content to the broker. Upon receiving the published content, the broker matches the subscriber' interest with the published content and the subscriber is notified of the result. In our service platform, once chooses to use asynchronous delivery, each personalized application service is actually decomposed into two publish/subscribe based *event notification services* that are chained. Each notification service itself is an event notification broker, that supports a particular content type (*e.g.*, customer intent, product offer, *etc*.) and allows property-based filtering (*e.g.*, only product id = "100200"). Shown in Figure 2, the generic event notification service interface in our service platform is defined as:

```
interface EventNotificationService {
        void AddSubscriber (string subcriberId);
        void RemoveSubscriber (string subscriberId);
        void AddSubscription (string subscriberId, string subscrip-
tion);
        void RemoveSubscription (string subscriberId,
                                                string subscription);
        void Publish (XmlElement content);
        XmlElement GetNotification (string subscriberId,
                                                string sessionId);
}
```

Figure 2. Generic service interface for event notification services

Such an event-based system allows customers, the store and its business partners to be loosely coupled. Neither the store nor the manufacturers is required to listen to the customer's intent. Neither the store nor the manufacturer is obligated to respond with offer to the captured intent.

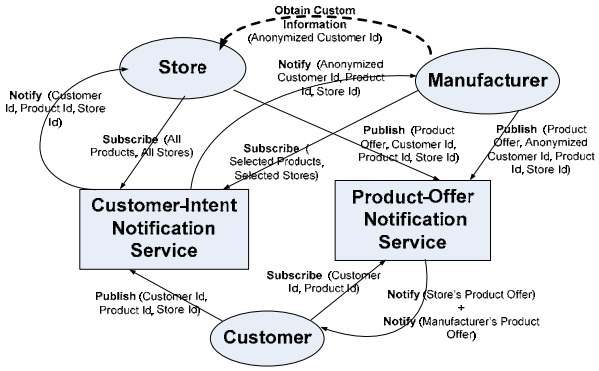

Figure 3. The two chained notification services

The personalized product offering service consists of two notification services. The first service is called *customer-intent* (shown in Figure 3). Customers publish their intentions, which represent the particular products that they are considering purchasing, through the *Publish* method defined in Figure 2. The store and the manufacturers subscribe to the expressed intents through *AddSubscription*. The store is the owner of the service platform and has no subscription restrictions. However, each manufacturer is subject to subscription restrictions on product categories and geographical regions where the stores are located (See Section 5 for a discussion of enforcement). An administrator sets these per-subscriber restrictions, which are stored in a policy database associated with the event notification services, before the administrator invokes the *AddSubscriber* method of the service.

The notification to the store contains a customer identifier, product identifier and store identifier. The manufacturer will receive the anonymized customer identifier

along with the other two identifiers identical to what the store receives. *GetNotification* retrieves a notification result from a queue that belongs to the subscriber. Note that the two different notification results by the store and the manufacturer can be encapsulated by the same generic XML element type definition in *GetNotification*.

The second service is called *product-offer* (also shown in Figure 3). The store and the manufacturer that supplies the product both publish their offers. The customer subscribes to the offers specific to the product she had scanned (*i.e.,* her purchase intent). The offers from the store and the manufacturer are presented to the customer. The notification result from the store carries the explicit customer identifier. However, regarding the manufacturer's offer, the personalized offering service will have to rely on the session management service shown in Figure 1 to map the anonymized identifier sent from the manufacturer back to the customer identifier that it knows about.

A customer's subscription to the *product-offer* event notification service is dynamic and happens only after the customer expresses an intent. A customer's subscription is automatically removed by the session management service when the session terminates. The lifecycle of the mobile session will be detailed in Section 4. Thus customers can only receive offers during their store visits.

Anonymized customer identifiers protect customers' identities from the manufacturer, as they have no meaning to the manufacturer. Furthermore, because they are constantly changed from one customer interaction to another, the manufacturer will not be able to identify who the customer is by tracking an identity over time. Nevertheless, the manufacturer could still obtain customer profile information (*e.g.*, customer segmentation) by following its contractual agreement with the store. The manufacturer can invoke the *Customer Profile Management* service shown in Figure 1, through the dashed line shown in Figure 3, by presenting the anonymized customer identifiers that it knows about. Certain customer profile information, such as telephone number (with which the manufacturer can correlate the customer information it owns), is privacy sensitive. The privacy-protection framework that we developed, MUPPET, can handle such customer-controlled information release [4], by having privacy-control policies defined for each manufacturer or each manufacturer category. With selective customer profile information access, the manufacturer can make personalization decision more effectively. Because the manufacturer has no customer tracking capability due to identity anonymization, policies on service usage control need to be enforced at the event notification services to prevent the customers from abusing the services.

Similarly, two chained notification services are used to realize personalized advisory support.

4. Session-based service tracking

MSA helps the store construct a comprehensive view of the customers' in-store shopping behavior, to evaluate the effectiveness of its product campaign strategies, and to monitor service quality delivered by the store staff, *etc*. The store can improve customer satisfaction, which leads to improved retention and increased loyalty. Such capabilities become feasible because the MSA serves as an intermediary between the customer and the store, and all bi-directional interactions are well captured.

A mobile session is established when an in-store customer activates the MSA. The session terminates either due to time out, or when the session management service detects that the customer has left the store from location information that the backend platform can receive (e.g., GPS, or location service) or can be inferred (from the mobile client IP address). In each application service, a session management component [1] is deployed to each service-side access point for message interception. It logs each service request/response from and to the mobile device applications, along with time stamps and session identifiers. The log information is persistently stored for offline data analysis once the mobile session is terminated.

Data analysis on different personalized services can offer different insights. From analysis on information captured from advisory support service, the store can determine which products are frequently require assistance, how responsive the store staff members are to the inquiry, how the customers value the assistance, the rate of purchases from the store staff's assistance (by linking subsequent customer purchase transactions with the mobile session in which advisory support occurs), *etc*.

On the other hand, by tracking the personalized product offering service, the store can determine what promotions entice customers and in what customer segments, what product offers are accepted by customers and in what customer segments, *etc*. Furthermore, with a unique identifier (GUID) assigned to each offer, the store can track the entire lifecycle view of that particular offer (shown in Figure 4). Such tracking information can be shared with the manufacturers for offers they issue.

The lifecycle starts with offer issuing (by the store or manufacturer), to offer viewing (by the customer), to offer accepting (to be digitally stored by the customer), and finally to offer redeeming (at store checkout). Each offer state can be further annotated with time and store identifier. Such personalized offer lifecycle tracking, which is encapsulated in the *Product Offer Lifecycle Management* service shown in Figure 1, is far more precise and effective than traditional paper-based coupons and in-store kiosks shared among in-store customers.

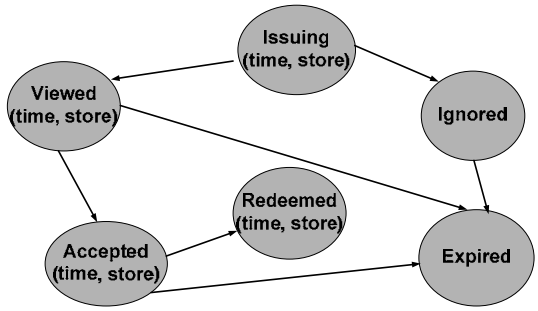

Figure 4. Personalized offer's lifecycle tracking

5. Cross-organizational information sharing

Data and services are shared across multiple organizations. The business contractual relationship agreed between the store and the manufacturer dictates the rights of one party to access the information owned by the other. Rights to access the store's data and services differ among the manufacturers. First of all, each manufacturer can only issue offers for its own products. Secondly, each manufacturer may opt into a different level of partnership (platinum, gold, regular, *etc*.), each of which is entitled to different rights. For example, customer profile information is only shared with platinum and gold partners; a platinum partner can provide offers to stores anywhere within the US, whereas a gold partner is only entitled to selected regions.

In a dynamic business environment, the partner relationship between the store and the manufacturers is changing, driven by factors like demands, service quality, *etc*. Sub-contracting relationships are also common. A manufacturer may outsource its capabilities and have subcontractors deliver products to the store on its behalf. Consequently, the service network naturally embraces sub-contracting partnership networks. Each subcontractor is authorized to issue offers and receive customer profile information. But the contractor's rights cannot exceed the ones granted to the manufacturer having a direct contract with the store, because the sub-contractor's rights are further restricted by its signed contract with the manufacturer. For example, a manufacturer that has a direct partner relationship with the store can provide offers to the entire US, but its sub-contractor located at California can only provide offers to the customers located in California.

We assume that all parties involved in this dynamic service network are trusted to not intentionally leak information. But we still like to preserve the service integrity such that information is appropriately shared and consumed based on what is agreed upon in the business contracts.

We developed an authorization-based access control mechanism for a service-oriented computing environment [11]. The core technique that we developed uses the SAML certificate [13] to encode authorization decision about each service method within a service, along with constraint specifications to either the entire service interface or certain service methods. The service access decision is local to the service and is based on the authorization token presented to the service. We extended the SAML token to support delegation that crosses organizational boundaries, such as delegation of rights due to subcontracting. This mechanism is well-suited for decentralized access control, because granting/ delegating authorization is a decision local to each organization. This mechanism addresses the issues such as fine grained access rights management, ambiguity of rights enforcement, and rights delegation that commonly exist in identity-based access control mechanisms [9].

Next we show how we can apply this access control mechanism to the MSA personalized service delivery network that spans the store and its manufacturer partners. We choose the event notification services detailed in Section 3 to be the primary example for illustration. The same mechanism can be applied to the other services that we have mentioned before, including *Customer Profile Management*, *Customer Intention Tracking* and *Customer Response Tracking*, *Product Offer Lifecycle Management*, *etc*.

5.1. Access right granting process

Each service is the root of its access granting chain [11]. It creates a SAML token in which it grants itself full access rights. Each service then further delegates its full access right to the *Domain Access Controller* (DAC) in its administrative domain. When two organizations (*e.g.*, the store and the manufacturer) establish a business contract, the store's DAC will delegate a subset of its rights to access *customer-intent* and *product-offer* event notification services to the manufacturer's DAC, which later further delegates the restricted (or full) set of rights to services in the manufacturer's administrative domain. Such a manufacturer's services will invoke the event notification service in the store's administrative domain eventually, presenting the granted authorization in the service request message.

Section 5.2 and Section 5.3 detail how rights restriction and delegation are expressed in SAML tokens, through method-level and attribute-level restrictions. A complete SAML token example demonstrating these two schemes can be found at [16].

5.2. Method-level permission granting

In an event notification service, the right to access *Add/Remove Subscriber* (shown in Figure 2) are reserved to the store's service administrator and never granted out.

The DAC of the manufacturer (and later delegated to the manufacturer's personalized offering service) is granted the right to access *Add/Remove Subscription*, *Publish* and *GetNotification*. Furthermore, these methods are not uniformly granted. For instance, in the *customer-intent* notification service, the manufacturer's DAC is only granted the right to use *Add/Remove Subscription*, *GetNotification*, but not *Publish*, because the manufacturer is a subscriber but not publisher in this particular event notification service. Similarly, to the *product-offer* notification service, the manufacturer's DAC is only granted for *Publish*, but not *Add/Remove Subscription* and *GetNotification*, because it is a publisher but not subscriber.

All the service methods permitted are explicitly listed in the SAML authorization token.

5.3. Attribute-level constraint specifications

Our authorization token can further constrain the method-level rights to achieve even finer granularity control, by using SAML attribute statements to express the desired constraints.

For the personalized offering service, constraints can include product categories and geographical regions allowed. For example, in the *customer-intent* notification service, a manufacturer can only subscribe to the TV category (even though it might also deliver the printer-related products) and receive notifications on the consumer intent from the stores within a particular region.

An SAML token that expresses geographical region related constraints is illustrated below. The attribute statement contains only a unique identifier (such as *Regions392*) as a handler, corresponding to an enumerable set on the allowed geographical regions at the policy repository accessible by the service. Alternatively, the set enumeration can be explicitly specified in the statement.

```
<saml:AttributeStatement>
<saml:Attribute
   AttributeName="AllowedRegions" AttributeNamespace
      ="http://www.futurestore.com/MobileCouponOffering.asmx">
    <saml:AttributeValue>Regions392</saml:AttributeValue>
</saml:Attribute>
…..
</saml:AttributeStatement>
```

The enforcement on attribute statements typically is service-dependent. Specific to the event notification services that we introduced in Section 3, the enforcement can be carried out in one of two ways. The first one is to ensure that the subscription supplied to *AddSubscription* matches the constraint specification, such that only the matched notification result will be distributed accordingly. The second one is to perform pre-filtering of the published data on *Publish* and post-filtering of the notification data on *GetNotification*. Note that the first approach does static conformance checking whereas the second one enforces dynamic runtime filtering. The first approach is only applicable to subscription (and thus notification), but not for publishing. Furthermore, static conformance checking is not easy to perform, especially when multiple constraints are involved. The second approach ensures that even if a subscriber over-subscribes to what it is entitled to, the notification will be filtered accordingly. Our implementation took the second approach.

5.4. Handling sub-contracting

A sub-contracting agreement can be translated into method-level or attribute-level constraints, or both. Attribute-level constraints are more complex to express and enforce than the method-level ones. A SAML token granted to the manufacturer's DAC can be further constrained on product category and geographical regions when delegated to the sub-contractor's DAC. A delegation certificate is schematically shown in Figure 5. The inner SAML certificate (assertion) issued by the store's DAC to the manufacturer's DAC, becomes the *evidence* field inside the authorization statement of the outer SAML certificate, which is issued by the manufacturer's DAC to its counterpart at the subcontractor. Thus, a SAML token with delegation is actually a SAML token chain.

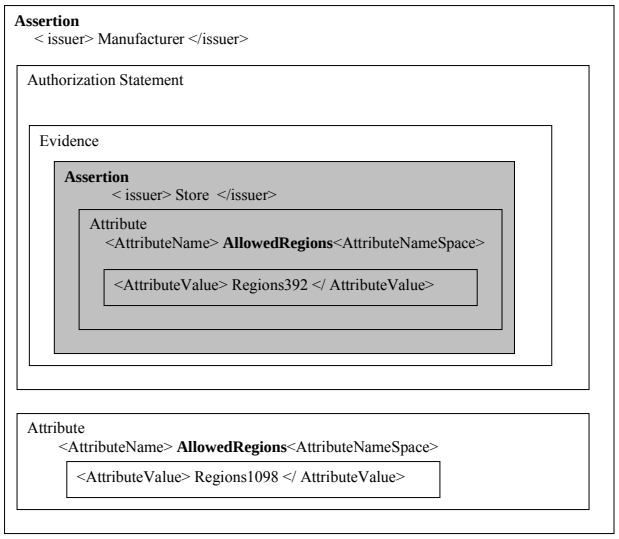

Figure 5. SAML token with delegation

The subcontractor can use this delegated certificate to participate in the personalized offering service owned by the store, as if it is the store's direct partner. The filtering is done at *Publish* and *GetNofication* and takes into account the attribute-level constraints with the identical

attribute name in the full chain, starting from the innermost one, and applying the constraints outwards.

Suppose the filtered result from *GetNotification* in the *Customer-Intent* notification service is held in a database table called *IntentsFiltered*. The following SQL query can perform filtering with respect to the attribute *AllowedRegions*.

```
select u.* from IntentsFiltered u, Store store
    where u.StoreId = store.StoreId and store.zip in
    (select cf.LocationCategory from
   LocationCategoryFiltering cf where cf.FilterCode = handler)
```

The *LocationCategoryFiltering* database table stores the enumerable set of zip codes corresponding to the *AllowedRegions* attribute (the handler) specified in the SAML token. The above SQL query can be performed in a chain of these handlers specified in the SAML token that represents a delegation chain involving the manufacturer, its sub-contractor, and potentially its sub-sub-contractors as well. Each intermediate query (i.e., filtering) result is stored in the table *IntentsFiltered*. The final filtered notification result can be retrieved from the same *IntentsFiltered* table.

6. Prototype implementation

We have implemented a prototype of the MSA as an iPhone browser-based application. The screenshots in Figure 6 show two offers issued by the store (named *First Store*) and the manufacturer (HP). They both arrive after the customer expresses interest in buying the HP Photosmart R717.

The MSA backend was developed using .NET C# web services and the web front-end was developed using ASP.NET. The notification services were developed with the Microsoft Notification Service supported by SQL 2005. The personalized offering business processes for the store and the manufacturers were developed with Microsoft Windows Workflow Foundation. Business rules for product offerings involved customer segmentation, product price and store location. In our implementation, customer segmentation is pre-populated but it can be constructed, *e.g.*, based on the Recency-Frequency-Monetary Value approach [3] from customer historical purchases. The SAML authorization-based access control mechanism has been developed separately as a package and applied to the *customer-intent* and *product-offer* event notification service presented in Section 3. This access control mechanism has not been integrated with the rest of the MSA service backend.

7. Related work

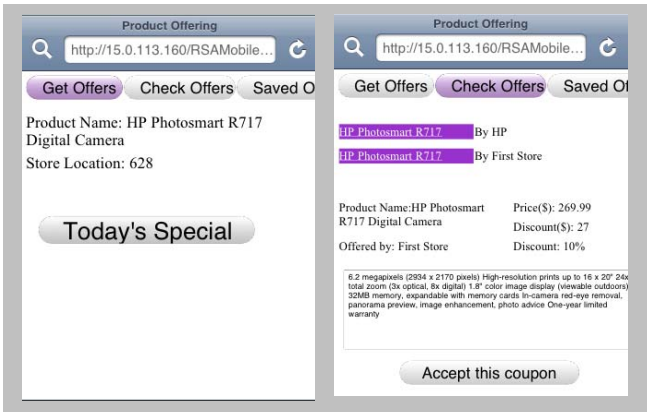

Figure 6. The Screenshots on product offerings

M-commerce has been defined as "e-commerce for users on the move" [18]. Personalized services should be seen as a prerequisite for m-commerce applications due to the imperfect usability (limited screen and keyboard size) of handsets and the situations and settings in which m-transactions are likely to be made [7]. Mobile commerce services are typically personalized based on time, location of the user and the user profile.

Most m-commerce applications require user location, which is typically obtained by GPS tracking and continuous location monitoring [5, 19, 10]. In contrast, our solution needs only a unique store identifier and does not require intrusive continuous location monitoring. Some applications, such as AURA [17], provide location-based services with static information (*e.g.* price comparisons, product information) that is not personalized for users.

Most m-commerce applications that personalize content based on user profiles, expect users to input their preferences [5, 19]. In contrast, in our solution, the offers and advisory services are personalized based on the user profile generated from the retail ecosystem − both the store in which the service is requested and the related manufactures. We believe that our solution enables a richer contextual personalization and provides a holistic view of user preferences. Finally, applications like Zagme [19] or SMMART [10] require users to navigate through a slew of available offers, whereas our solution is based on the specific user intentions captured automatically.

Tracking of user web browsing activity through cookies or click-streams is common in online shopping sites. Click-stream data, such as the one from Omniture [14], provides comprehensive activity recording but it is difficult to distinguish individual users precisely and pinpoint exact user location (typical precision is at city level [15]). Consequently, it is difficult to define a matching pattern over web user activities to characterize shopping behavior. Our MSA is running on a personalized device and provides personalized services in a confined shopping

environment. Therefore each click on the device reveals the user's true intent at that moment.

WS-event notification [12] defines publish/subscribe based service interfaces but addresses little on access control to arbitrate information flow. Role-based access control has been recently applied to a publish/subscribe system [2]. In a cross-organizational environment, identity-based or role-based access control suffers various identity management related problems, such as granularity of rights granting, ambiguity of rights enforcement, difficulty of right delegations, *etc* [9]. Moreover, the mechanism such as [2] is focused on data confidentiality, whereas our focus is more on service integrity, ***i.e***., to enforce the rights of information access based on what is stated in business contracts.

8. Conclusions

MSA is a mobile application platform that aims to provide a broad range of personalized services offered by the store to improve in-store customer experiences. This paper focuses on two specific application features: personalized product offering and personalized advisory support. But other features, such as customer front-door greeting, product recommendation, and bridging online/in-store multi-channel shopping, can be easily provided using our platform. We presented the three key techniques, namely, event-based service delivery network, mobile service session tracking and cross-organizational information sharing, which together facilitate personalized service delivery over the business network that involves the store and its business partners in the shopping ecosystem. We have developed a prototype to provide a proof of concept and are in the process of designing a user study to validate it in a real-time in-store setting.

9. References

[1] I. Ari, J. Li, R. Ghosh and M. Dekhil, "Session Management as a Service", Poster Presentation at WWW'07.

[2] J. Bacon, D. Eyers, J. Singh and P. Pietzuch, "Access Control in Publish/Subscribe Systems," Proceedings of 2nd International Conference on Distributed Event-Based Systems, pp. 23-34, July 2008.

[3] M. Chen, A. Chiu, H. Chang, "Mining changes in customer behavior in retail marketing," *Expert Systems with Applications*, (2005) 773–781.

[4] W. Cheng, J. Li, K. Moore and A. H. Karp, "A Customer-Centric Privacy Protection Framework for Mobile Service-Oriented Architectures," Proceedings on International Conference on Services Computing, Vol. 2, pp. 13-20, 2008.

[5] J. de Castro, H. Shimakawa, "Mobile Advertisement System Utilizing User's Contextual Information," International Conference on Mobile Data Management, pp. 91, 2006.

[6] P. Eugster, P. Felber, R. Guerraoui, A. Kermarrec, "The many faces of publish/subscribe," ACM Computing Surveys (CSUR), Vol. 35, Issue 2, 2003.

[7] J. Fink, J. Koenemann, S. Noller and I. Schwab, "Putting Personalization into Practice," Communications of the ACM, 45(5), 41-42, 2002.

[8] Y. Karabulut, F. Kerschbaum, F. Massacci, P. Robinson and A. Yautsiukhin, "Security and Trust in IT Business Outsourcing: A Manifesto," Proceedings of the Second International Workshop on Security and Trust Management (STM 2006), pp. 47-58.

[9] A. H. Karp, "Authorization Based Access Control for the Services Oriented Architecture", Proc. 4th Int. Conf. on Creating, Connecting and Collaborating through Computing (C5 2006), Berkeley, CA, IEEE Press, January 2006.

[10] S. Kurkovsky and K. Harihar, "Using ubiquitous computing in interactive mobile marketing," Personal Ubiquitous Comput. 10, 4 (Mar. 2006), 227-240.

[11] J. Li and A. Karp, "Access control for the services oriented architecture," Proceedings of the ACM workshop on Secure Web Services, pp. 9-17, 2007.

[12] P. Niblett and S. Graham, "Events and service-oriented architecture: The OASIS Web Services Notification Specifications," IBM Systems Journal, Vol. 44, No. 4, pp. 869-886, 2005.

[13] OASIS, "Security Assertion Markup Language (SAML) 2.0 Technical Overview, Working Draft 05", May 2005.

[14] Omniture, http://www.omniture.com.

[15] R. Richmond, "We Know Where You Are," *Wall Street Journal*, Sept. 29, 2008.

[16] SAML token example for event notification service, http://opra.hpl.hp.com/Mercado/PartnerCertificateMobileOffering.xml.

[17] M. Smith, D. Davenport, H. Hwa, "AURA: A mobile platform for object and location annotation," Ubicomp 2003.

[18] P. Vittet-Philippe, J. Navarro, "Mobile E- Business (M-Commerce): State of Play and Implications for European Enterprise Policy," European Commission Enterprise Directorate-General E-Business Report, 2000.

[19] ZagMe - http://www.beepmarketing.com.