SOLE: Context-Aware Sharing of Living Experience in Mobile Environments

Jian Zhu
National University of Singapore, Singapore
zhujian@comp.nus.edu.sg

Mohammad Oliya
National University of Singapore, Singapore
oliya@nus.edu.sg

Hung Keng Pung
National University of Singapore, Singapore
dcspkh@nus.edu.sg

Wai Choong Wong
National University of Singapore, Singapore
elewwcl@nus.edu.sg

ABSTRACT
In this paper we present the design of an application framework for context-aware Sharing Of Living Experience (SOLE) in mobile environments. With SOLE, users can share their experience of daily activities. They can specify where to store their information - on their mobile devices to have full access control over their data, or in the distributed SOLE application servers for ease of access by others. SOLE leverages on our service-oriented middleware (Coalition) to achieve context-awareness. To cater for resource constraints of mobile devices, the concept of “proxy” is introduced. As a proof of concept, we develop a prototype of SOLE for mobile Android phone users. Preliminary experiments have been carried out to test the performance of the framework.

Categories and Subject Descriptors
H.3.4 [Information Storage and Retrieval]: Systems and Software—Current awareness systems, User profiles and alert services

General Terms
Design, Human Factors

Keywords
experience sharing framework, context-aware, mobile service

1. INTRODUCTION
The recent success of ubiquitous technologies (e.g. RFID tagging, GPS and wireless sensors) enables the provision of computing and communication services anywhere, anytime. Context-awareness, as one of the key concept to achieve ubiquitous computing, makes IT invisible and yet ever closer and seamless to our daily living. Our recent effort focuses on the development of a service-oriented middleware — Coalition [1] to support context-oriented application development.

In this paper, we investigate the design, integration and deployment of the Sharing Of Living Experience (SOLE) application in the Coalition environment. More specifically, we demonstrate how Coalition: (i) facilitates the deployment of (mobile) application services; (ii) provides the flexibility in resource management and protection of user privacy; (iii) enables applications to achieve context-awareness. SOLE will have real mobile users playing the roles of Application Service Providers (ASPs) and Application Service Consumers (ASCs). The ASPs share experiences regarding real-world entities in daily life scenarios (e.g. shopping and traveling); while ASCs could access this information ubiquitously.

The rest of this paper is organized as follows: Section 2 presents an overview of the related work. Section 3 describes our proposed SOLE and its constituent components. Section 4 shows a prototype of SOLE and presents preliminary experimental results. Finally, Section 5 concludes the paper and highlights our future work plan.

2. RELATED WORK
The recent advance of social computing has promoted the idea of interactive information sharing. Inspired by the concept, context-aware systems catering for user experience sharing have been developed. Examples can be found in RevisiTour [2], eXspot [3], APriori [4] and Sentient Graffiti [5]. However, one major limitation of these systems is that they are designed for a specific application domain, e.g. for museums and exhibitions, and hence tend to be smaller in scale. Besides, they assume all the entities are identified with RFID tags, which may be restrictive in practice.

Existing Web 2.0 applications like Flickr [6] and foursquare [7], provide a platform for large-scale user interaction. Nevertheless, these applications require the users to upload their content to a central repository. A user may not be able to do so due to technical issues such as low bandwidth, or intermittent availability in network connectivity. Also, he may not trust the hosting service for the storage of his private data. Most importantly, as the user may be using several Web 2.0 services (e.g. using several photo sharing services), management of the contents residing in multiple repositories can be problematic [8].
Our SOLE application framework differs from the existing work in the following aspects: (i) it is generic and scalable, by not assuming a specific application scenario. It allows users to share and retrieve experiences about any entity at anywhere, anytime (without requiring RFID tags); (ii) it is flexible, by allowing the user to choose where to store the experience data and to specify his audience. When the data is stored on mobile devices, it could be offered through mobile services from these devices; (iii) it is context-aware by considering the location, time, and preferences of the user in discovery and delivery of experience information.

3. PROPOSED APPROACH

3.1 SOLE Application Scenarios

Alice often strolls along the world famous shopping belt along Orchard Road. The goods and services provided by the ION Orchard are excellent and the store is giving further discount to commemorate this opening occasion. She could not help but to share her impression of the newly opened centre through her Nexus One with the newly installed SOLE application. Alice is given a couple of options to share her impression of the shop: (a) to store her data in a default server for ease of access by others or; (b) to store the data, if deemed private, in her handphone for a full control over the data. She can share the information publicly or provide a friend list. All options could be configured with the touch of her finger tip.

After almost one hour of shopping, Alice takes a break to decide on what to do next. She starts to explore SOLE and realizes that: (a) SOLE automatically updates experience information of around her proximity when she changes her preference of types of experience; she could turn “off” or “on” this “push” mode; (b) she could query (pull) “information” by searching certain keywords or specifying an area on the map for browsing the available information. Having decided to take a coffee, she checks for available feedbacks about nearby coffee shops on the map. The Starbucks next to her seems to be of satisfying service and so she opts to go there.

3.2 SOLE in Coalition

Coalition is a context-aware middleware which was previously named as CAMPH [1] for pervasive homecare application. The middleware is designed in a layered architecture. Through its context data management layer, Coalition is capable of locating and extracting relevant context data from large numbers of heterogeneous data sources (e.g. sensors) distributed over many different operating spaces (e.g. shops). Above the data management layer is its service management layer, which facilitates Location-Aware Service Organization and Discovery (LASOD) [9] for both system services and third-party services. LASOD implements a two-tier structure of peers for service management (Figure 1). The 1st-tier service management peers (or superpeers) define and manage geographical scopes for each area under an administration and the routing of queries in the overlay-network of 1st-tier peers, while 2nd-tier service management peers (or service peers) of a geographical area register application services for that local area and route queries in the overlay-network of 2nd-tier peers.

We consider SOLE as a potential application service that can be deployed on any service peer in LASOD as illustrated in Figure 1. To do so, three logical software components are defined: (i) service management specific component which interacts with the underlying service management functions of LASOD to complete tasks such as service registration and query routing; (ii) application specific component which performs specific business logics; and (iii) user interface specific component which interacts with the user to allow functions of the application service to be invoked. The three components may reside in one or be distributed to more than one computing devices, e.g. static machines or mobile devices. With respect to SOLE, we will refer component (i) and its hosting device as SOLE Application Service Media-tor (SOLE-ASM); component (ii) and its hosting device as SOLE Application Server (SOLE-AS); and component (iii) and the device through which a user can provide or consume experiences as SOLE Application Service Provider or Application Service Consumer (SOLE-ASP/ASC).

The SOLE-AS can be performed on any service peer in LASOD. Alternatively, when the users opt to store the experience data locally, the SOLE-AS is resides in the mobile device of the SOLE-ASP and the SOLE-ASCs will retrieve the information directly from the SOLE-ASP’s device. However, for the latter case, issues such as limited memory, processing power, and battery life of the mobile device should be considered for the provisioning of mobile services. Therefore, we introduce the proxy to perform the following functions for mobile devices playing the roles of SOLE-ASP/ASC: (i) performing SOLE-ASM functions; and (ii) communicating with SOLE-AS and other SOLE-ASP/ASC. The proxy can be hosted in a dependable computing platform such as a service peer in the 2nd-tier. Figure 2 shows the functional components of a proxy. The bottom level defines the necessary LASOD components such as service indexing and query routing, while the top level indicates the services deployed locally in the proxy. Services offered by the mobile devices (e.g. a SOLE-ASP) shall register with the proxy and be bound to the respective ASM of the proxy to communicate with its underlying LASOD components. With the introduction of the proxy, the architecture of LASOD is thus extended to three functional layers as shown in Figure 1.

3.3 SOLE Component Design

SOLE facilitates sharing of experiences through distributed SOLE-As. The main task of the SOLE-AS is to maintain an index to the shared data of experiences. The index key

Figure 1: The three-tier LASOD architecture. Multiple application services can be hosted in the same service peer. The 3rd-tier is extended by the concept of “proxy” to support mobile service provider.
represents the entity of interest (such as a shop), and the value describes certain aspects of the experience, e.g., who shared it, when and where it was shared (as specified by the data schema in Section 3.4). We adopt a Distributed Hash Table (DHT) technique to choose a SOLE-AS for storing an index. Each SOLE-AS takes the ID of its hosting service peer (assigned from LASOD) and is responsible for the corresponding segment of the keyword space. For storing the index of the experience data, the name of the entity of interest is hashed to a key and is assigned to the first SOLE-AS whose identifier is equal to or greater than the key value. To allow public access, the SOLE-AS joins LASOD by registering its Web service so that it can be discovered by SOLE-ASP/ASC. The Web service exposes functions such as experience indexing and retrieval by manipulating on a Experience Database. In the process of experience retrieval, if the actual data is not stored on the SOLE-AS, the SOLE-ASP’s ID where the data is stored will be returned instead.

Users use SOLE-ASP/ASC to share or retrieve user experience information. To simplify the sharing process, SOLE uses the service browsing and discovery features of LASOD. If an entity of interest has a corresponding service registered with LASOD, the service is visible on the map and users can associate their experience with the representative service. Similarly, to retrieve the experience information, the users can select a service to check whether the corresponding entity has any experience information associated with it or not. SOLE can easily support proximity accessing (or sharing) of experiences or a range-based access by SOLE-ASCs due to the location-awareness of LASOD. Alternatively, the users may also search for the relevant experience in all areas using keywords. The interactions among SOLE components for experience retrieval are illustrated in Figure 3. Note in Step 4, the functions to support experience retrieval on the SOLE-ASP’s mobile device are all exposed as Web services, which is similar to the case for the SOLE-AS.

3.4 SOLE Experience Representation

In SOLE, each experience data shared by the SOLE-ASP is characterized by a data schema to enable context-aware experience sharing. The schema is composed of two segments: Meta-data and Experience Data. “Meta-data” captures three contextual elements, namely Where (location), When (time), and Who (user profile). These data can be used in experience filtering or aggregation. Note that all the meta-data is either filled once (e.g., ask the user explicitly) or captured implicitly (e.g., location and time). “Experience Data” corresponds to What the user is sharing. It is further divided into three aspects: Entity of Interest, Tags and Data. “Entity of Interest” specifies the object, place, or event of interest. If the entity has a corresponding service registered with the LASOD, it can be browsed on the map and be selected. If the entity cannot be browsed on the map, due to being mobile (e.g., a taxi) or not having a corresponding service in LASOD, we will use the entity’s name together with the location information associated with the experience to identify it. The “Tags” part consists of the keywords that can help in providing high level knowledge representations such as tag clouds. For the “Data” part, the representation format could be texts, images, or videos.

The decision for the storage of the experience data must take security, privacy, and performance into consideration. In SOLE, we offer three storage schemes for the experience data: Public, Restricted (visible to a friend list) and Private. With the “Public” scheme, the experience data is encouraged to be stored on the distributed SOLE-ASs to maximize the efficiency of retrieval, filtering, and aggregation of experience data. This will also prevent excessive load on SOLE-ASP’s device, if the shared content becomes popular.

3.5 SOLE Context-Awareness

To achieve context-awareness in SOLE, we consider each SOLE-ASP/ASC as a mobile Physical Space Gateway (PSG) for the person who carries it. By registering the PSG in the context data management layer of Coalition, the following context-aware applications can be realized.

1) Mobility of the SOLE-ASP: If the experience data is stored on a mobile device (instead of a distributed server), efficiently locating the device by the SOLE-AS or SOLE-ASC can be challenging. In Coalition, such a problem can be handled in two phases: (i) retrieving the current location of the mobile PSG by issuing the query “SELECT location FROM PERSON WHERE id = asp_id”, where asp_id is the SOLE-ASP’s ID; (ii) sending a service discovery query to LASOD limiting the search scope to the proximity of SOLE-ASP’s location. Once the reference of the SOLE-ASP is retrieved, its service can be invoked, but if the reference corresponds to a proxy, the proxy will instead take care of the request by directing it to the right SOLE-ASP.

2) Experience “Pushing” to the SOLE-ASC: The context of an SOLE-ASC can be exploited for pushing relevant experience information. The pushing of information can also be personalized by taking the person’s preference into consideration. For instance, the SOLE-AS can subscribe with...
Coalition to detect the event of users entering a mall, e.g. 
"SELECT name FROM PERSON WHEN LocationChange 
WHERE new_location = 'ION Orchard'". Later when the 
LocationChange event occurs, Coalition will notify the SOLE-
AS with the name of the person and then the SOLE-AS 
can get the preferences of the mobile users by issuing an-
other query “SELECT preference FROM PERSON WHERE
name = person_name AND app_type = ‘SOLE’”.

3) General Context-Awareness: The location-awareness is 
already embedded in SOLE by leveraging on LASOD. More 
specifically, the SOLE-ASC/ASP may browse for entities 
within the local range or issue keyword-based queries to 
select an entity of interest. Other types of context-awareness 
can be realized by using the experience meta-data. For in-
stance, “Time” can be used to filter old experience data; 
"Friend List" can be used to filter irrelevant SOLE-ASCs. 
Note that the proposed data schema is extensible so that 
applications may define their own attributes to be matched 
in the SOLE-AS. These attributes can also be opted not to 
be shared publicly.

4. PROTOTYPE IMPLEMENTATION AND 
PRELIMINARY EXPERIMENT

We have developed a prototype of SOLE in our laboratory. 
The prototype consists of several SOLE-ASs which rely on 
MySQL for the data management. The SOLE-ASP/ASC is 
implemented using HTC Hero. Figure 4 presents the UI for 
the SOLE-ASP/ASC. The user relies on the SOLE-AS for 
access control, i.e. by uploading friend list. He may also 
keep the whole experience data (especially large data) on 
his device for future reference or retrieval.

For a more rigorous study and evaluation of SOLE per-
formance, we have run the SOLE-AS on desktop PC (In-
tel Dual-Core E8400) and the SOAP-ASP on mobile device 
(HTC Hero) to study the performance (see Table 1). Specif-
ically, we measure “SOAP Deserialization/Serialization” 
and the overhead occurred by the application (“Application Ex-
ecution”). The current study is based on simple texts that 
possibly issued by people. Though compared to the con-
ventional desktops, applications running on mobile devices 
incur more overhead due to their slower CPU and smaller 
memory, the performance is still acceptable.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Time (Milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop</td>
<td>Mobile Device</td>
</tr>
<tr>
<td>SOAP Deserialization</td>
<td>12</td>
</tr>
<tr>
<td>SOAP Serialization</td>
<td>5</td>
</tr>
<tr>
<td>Application Execution</td>
<td></td>
</tr>
<tr>
<td>Experience Insertion</td>
<td>17</td>
</tr>
<tr>
<td>Experience Deletion</td>
<td>2</td>
</tr>
<tr>
<td>Experience Search</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1: Performance comparisons for SOLE run-
ning on desktop and mobile device in real-time.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a generic Sharing Of Living 
Experience (SOLE) framework, which fits in the vision of 
ubiquitous computing to let people communicate and share 
information at anywhere, anytime. SOLE has been used as a 
test case for application development on our context-aware 
middleware — Coalition. By leveraging on the Location 
Aware Service Organization and Discovery (LASOD) layer 
of Coalition, SOLE can be deployed as an application ser-
vice which utilizes the resources (e.g. registered services) 
and mechanisms (e.g. location-aware service discovery) of 
LASOD. Furthermore, the context data management layer 
of Coalition enhances the quality of service for SOLE, and 
provides context-awareness in the experience sharing or re-
trieval process. As a proof of concept, we have implemented 
a prototype of SOLE. We plan to deploy SOLE for a field 
trial in the university campus, to do a more rigorous eval-
uation of its performance.

6. ACKNOWLEDGMENTS

This work is partially support by project grant NRF2007IDM-
IDM002-006 on “Life Spaces” from the IDM Project Office, 
Media Development Authority of Singapore.

7. REFERENCES

27, No. 4, pp. 510-524, 2009.
experience with user-generated content,” in Proceedings 
experience at the Exploratorium,” in Journal of ACM 
recommendation system interacting with tagged 
2.0 platform to enable context-aware mobile 
distributed personal web-service data,” in Proceedings 
approach to location-aware service organization and 

Figure 4: Screenshots for SOLE-ASP/ASC: (a) 
Upon selecting an entity (star) on the map, the user 
can share or retrieve experience about the entity; 
(b) the user can view the list of experiences shared 
by others and he may add in his experience directly; 
(f) the details for the user selected experience.