A framework for context-aware application programming on smart phone

Baek Gyoon Sung
Department of Computer Engineering,
Korea National University of Transportation,
50 Daehak-ro, Chungju-si, Chungbuk 380-702, South Korea
Fax: +82-43-841-5341
Email: bgsung@ut.ac.kr

Abstract: Our framework can be classified in six main groups. Sensor manager retrieves and updates a program component that collects sensor data from the library. Data generator generates current or statistic summary data, or statistic index. Context generator is for developing a program to generate automatically various data and context from existing program skeletons. Class library and API is for developing a service application program or a program to extract and infer various data and context. Service generator generates a program component for summary reports, inference chain reports, communication services and personalised services. Inference engine infers context by rule-based and case-based reasoning. The developed application by our framework is extracted and infers various context such as ‘moving’ or ‘stay home’ and offers users the personalised services about current user’s context.

Keywords: framework for context aware application; context awareness; personalised service; context aware system on smart phone; intelligent agent; smartphone sensor.


Biographical notes: Baek Gyoon Sung was a Senior Researcher in Korea Institute for Dense Analyses (1984–1992), an Associate Professor in Department of Computer Science, Wonju National Collage (1992–1997), and a Professor in Department of Computer Engineering, Korea National University of Transportation (1997). He became a member of Korean Institute of Information Scientists and Engineers in 1990 and of Korea Information Processing Society in 1994.

This paper is a revised and expanded version of a paper entitled ‘A development environment for context-aware application programming on smart phone’ presented at 2015 International Conference on Platform Technology and Service (PlatCon-15), International Convention Center Jeju Co., Jeju Korea, 26–28 January 2015.
1 Introduction

Smart phones have advanced incredibly in the past decade and the mobile technologies make research on the user-context awareness feasible.

Sensors in smart phones can collect the variety of user’s information and context-aware system understands and processes situation information from diverse sensors and devices. As a key technology of ubiquitous computing, various works of context-awareness have been conducted using that information (Lago, 2013).

In this paper, we propose a framework for context aware application programming which is applicable to smart phone.

Our system can be classified six main groups: sensor manager, data generator and manager, context generator and manager, class library and API, service generator and manager, and inference engine and knowledge manager.

Sensor manager retrieves and updates a program that collects or controls sensor data from existing program library. Data generator and manager generates and manages current or statistic summary data, or statistic index. Context generator and manager are for developing a program to generate automatically various data and context from those program and skeletons and to manage.

Class library and API is for developing an Android service program or a program to extract and infer various data and context. Service generator and manager, generates a program for summary reports, inference chain reports, communication services and personalised services. Inference engine is composed of a rule-based reasoning and case-based reasoning engine and knowledge manager manages the rule base or case base.

The developed application by our framework extracts basic context from log data such as ‘moving’ or ‘staying home’ and infers abstract context such as whether he has an unusual amount of calls today.

2 Related work

2.1 Context aware system

Alvarez et al. (2010) presented a semantic reasoning infrastructure for mobile devices, aimed at simplifying the creation of context-aware applications. This framework provides a sensing subsystem, which abstracts access to both physical and virtual sensors, and a context management subsystem, which using an ontology-driven model fuses and reasons over this context information.

The work by David et al. (2011) proposes a middleware for Android devices developing context-aware mobile applications easily. It adopts a context-oriented paradigm, which based in a context information publish-subscribe mechanism enables an agile application behaviour adaptation.

In Benitez-Guerrero et al. (2012), a model and an architecture for context-aware mobile collaborative systems are proposed. The basic idea is that users can assume a certain roles and carry out activities with the defined objects.

In Kim and Park (2013), a context awareness model for personalised services (CAMPS) is composed of four parts. Firstly, context data manager contracts primary contexts. Then, it restores and updates statistic information about user’s preferences as archive history data which includes calls and messages, uses of application, daily
schedules, phone directory, etc. It extracts and infers current user’s context from the statistical data and archive data.

Figure 1 shows an architecture for context awareness service implementation (Kim, 2010).

Figure 1  An architecture for context awareness service implementation (see online version for colours)

Source: The Knowledge Engineering Review, Cambridge Univ. Press

2.2 Target application architecture

The elements that compose the context aware application program (see Figure 2) is developed by our proposed framework which can be classified in five main groups (Kim and Park, 2013): sensor, sensor data manager, log data manager, context manager and service providers.

These components and functionalities are detailed below (Kim and Park, 2013, Sung, 2014):

1 Sensor: Sensor data acquisition is made from diverse sensors or various events on the smart phone, web service program on web servers and user’s direct input such as GPS, accelerometer, proximity, illuminometer, magnets, orientation, WiFi, Bluetooth, microphone, etc.

2 Sensor data manager is composed of streaming sensor data manager, event data manager, web service data manager and user profile data manager. Streaming sensor data manager collects streaming data from diverse sensors such as accelerometer on smart phones. Event data manager collects data related to daily events such as for example calls and messages on smart phone. Also, web service data manager collects
data from web servers like weather information and user profile data manager collects data about user profile such as home or office address from user.

3 Log data manager provides data logging by periods from sensor data manager. Streaming log data is logging data linked with sensor acquisition data; event log data is logging data which comes from calls or messages. Web service log data is logging data such as weather or temperature from web server.

Figure 2 A target application architecture (see online version for colours)

4 Context manager is composed of production rules, current summary data, statistic summary data, basic context, context index, abstract context, and monitoring and audit trail. Basic context is that is mainly extracted from streaming log data such as ‘moving’ or ‘staying home’. Current summary data is defined as event log data. It is an accumulated data up to the present for a day. Statistic summary data is the average data that is extracted from current summary data and maintains by weekday or holiday for a certain period of time. Context index is the stress index extracted from rule-based reasoning and emotion index presented by valence-arousal index. Abstract context is the context that is inferred from the rule-based reasoning using the production rules presented by if-then. Abstract context is the high level context such as if he has an unusual amount of calls today, or if he gets more stress today. Monitoring and audit trail is the trace data that is collected for monitoring and audit.

5 Service provider is composed of personalisation service provider, monitoring service provider and communication service provider. Personalisation service provider recommends the results appropriately according to the current user’s context or emotion (see Figure 3). Monitoring service provider displays the various reports
A framework for context-aware application programming on smart phone

about his current contexts on the smart phone and also, communication service provider synchronises the data of smart phone based on user’s usages type compared with others.

Figure 3 An example of developed personalise services (see online version for colours)

2.3 Target application data flow

Our context awareness application is developed on the proposed framework. The data flow of developed application is shown by Figure 4 (Sung, 2014).

1. The developed application performs sensor data acquisition from the diverse sensors.
2. The sensor data manager collects and stores streaming data in database and logs data from sensor by periods.
3. The summary data manager accumulates the various sensor data to the current summary data and the statistical summary data.
4. The basic context is extracted from the streaming log data. Stress index and emotion index is extracted by rule-based reasoning. Abstract context is inferred by from rule-based reasoning using production rules.
5. Service provider generates personalisation services, monitoring services and communication services.
6. Context is modified by user’s feedback or other’s phone context.
3 A framework for context aware application programming

3.1 A proposed framework

The elements that compose a system architecture of proposed development environment (see Figure 5) can be classified in six main groups.

Android integrated development environment is a cross development environment that is composed of Java Development Kit, Android software development Kit, eclipse, and Java native interface, etc.

Services generator/manager generates and manages summary reports, inference chain reports, communication services and personalised services. Inference engine is composed of a rule-based reasoning and case-based reasoning engine.

Sensor manager retrieves and updates a program that collects or controls the streaming data, events, web service data and user profile from existing program library and insert new program. Operating system is Windows or Linux.
Our proposed framework components are shown on the Figure 6.

**Figure 6** A framework components

<table>
<thead>
<tr>
<th>Service Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Library &amp; API</td>
</tr>
<tr>
<td>CLAS</td>
</tr>
<tr>
<td>APIW</td>
</tr>
<tr>
<td>Inference Engine</td>
</tr>
<tr>
<td>Rule Based Reasoning</td>
</tr>
<tr>
<td>Case based Reasoning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data &amp; Context Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Data Manager</td>
</tr>
<tr>
<td>Summary Data Manager</td>
</tr>
<tr>
<td>Context Manager</td>
</tr>
</tbody>
</table>

| Sensor Manager |

### 3.2 Class library and API

Class library and API is composed of four parts (see Figure 7) and is for developing an Android service program or program components to extract and infer various data and context. Class library for Android services (CLAS) is classes for monitoring services, personalised services and communication services.

Class library for data management (CLDM) is classified sensor data collection, log data management, summary data management, context management, and audit trail management. Class library for knowledge management (CLKM) is organised to manage production rules and case base. APIW is API for data acquisition from web services.

**Figure 7** Class library and API of proposed framework

<table>
<thead>
<tr>
<th>Monitoring Services</th>
<th>Personalized Services</th>
<th>Communication Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Management</td>
<td>Class Library for Data Management</td>
<td></td>
</tr>
<tr>
<td>Audit Trail Management</td>
<td>Summary Data Management</td>
<td></td>
</tr>
<tr>
<td>Sensor Data Collection</td>
<td>Log Data Management</td>
<td></td>
</tr>
<tr>
<td>API for Data Acquisition from Web Services</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class Library for Knowledge Management

- Production Rules
- Case base
3.3 Context and data generator/manager

Context and data generator/manager is composed of five parts (see Figure 8). Also, it is also used to develop a program in order to automatically generate various data and context from existing program skeletons and to manage the generated programs and program skeletons.

**Figure 8** Context and data generator/manager

Sensor data manager is components to store and delete streaming data log. Summary data generator is components to generate current summary data and accumulates statistic summary data. Summary data manager is components to update and delete its and statistic index.

- Context generator is components to generate basic context, high level context, context index and abstract context and context manager is components to update and delete its
- M&A manager is components to store and monitor monitoring trail and audit trail and
- knowledge manager is components to collect and store production rule and case base.

4 Conclusions

In this paper, we propose a framework for context aware application programming on smart phone.

Our system can be classified in six main groups. Sensor manager retrieves and updates a program that collects or controls sensor data from existing program library. Data generator and manager generates and manages current or statistic summary data, or statistic index.

Context generator and manager is for developing a program to generate automatically various data and context from existing program skeletons and to manage generated programs and program skeletons. Class library and API is for developing an Android service program or a program to extract and infer various data and context. Service generator and manager, generates a program for summary reports, inference chain reports, communication services and personalised services. Inference engine is composed of a rule-based reasoning and case-based reasoning engine and knowledge manager manages rule base or case base.
The developed application by our framework is extract and infers various context such as ‘moving’ or ‘stay home’ and offers users the personalised services about current user’s context. Extensions of our work are focused on developing additional framework classes for context aware application program on smart phone.

References