

# Sequential Layered Approach for Optimized Context Integration

Yoosoo Oh\*

School of Computer and Communication Engineering, Information and Communication Research Center, Daegu University, Republic of Korea; yoosoo.oh@daegu.ac.kr

## Abstract

In this paper, we propose a sequential layered approach for optimized context integration. The proposed approach applies multiple fusion layers according to divided time intervals to accurately integrate contexts. Particularly, it is necessary to develop a generalized architecture for context integration that enables a decision to be made based on contextual data from large numbers of heterogeneous sensors. Thus, our approach is managed at an architectural level for dynamic changes in heterogeneous sensor environments. Also our approach is specified by the characteristics of contexts to decide high-level semantic information from different context inputs. Furthermore, our approach can serve as a fundamental study of context awareness in large-scale applications and the convergence among heterogeneous domains.

**Keywords:** Context-Aware Application, Context Fusion & Reasoning, Optimized Context Integration, Sequential Layer

## 1. Introduction

A huge number of sensors are deployed to access context-aware systems in heterogeneous computing environments. The deployed sensors, for instance, are physiological sensors, profiling sensors, and sensors measuring physical phenomena. For the maintenance of context-aware systems with the various sensors, intelligent analysis for context integration is necessary<sup>1-2</sup>. Moreover, the life-cycle support of the context-aware systems includes simulation, installation, debugging, adding new entities, removing old entities, and upgrading system components<sup>1-2</sup>. Especially context integration enables the provision of personalized services to multiple users by integrating inputted contexts for each user. Additionally, the context integration technology is platform-independent and it works for heterogeneous environments.

Several research activities on context integration have been developed<sup>1-8</sup>. The Context Fusion Network<sup>3</sup> allows context-aware applications to select distributed data sources and compose them with customized data-fusion operators into a directed acyclic information fusion graph. The Context Fusion Network supports context-aware mobile applications that need to aggregate data

from distributed sensors. The CoCo architecture<sup>4</sup> consists of key infrastructural components supporting the processes of context retrieval and context composition and a graph-oriented language describing the steps that need to be executed to compose context information. The CoCo architecture derives high-level context from low-level context information. The Software Engineering Framework<sup>5</sup> simplifies design and implementation tasks associated with context-aware software. The iQueue<sup>6-7</sup> enables applications to focus on the semantics of composition by facilitating the mechanics of composition. The sensor data fusion method<sup>8</sup>, using the Dempster-Shafer theory, collects all the relevant context information about each major entity in one central context data repository.

It is necessary to develop a generalized architecture for context integration that enables a decision to be made based on contextual data from large numbers of heterogeneous sensors. Unlike existing layered architectures, we apply multiple sequential layers by employing a characteristic of context integration based on the integration interval. Single-layered context integration collects contexts once when sensors deliver the sensory data and, so, it cannot have enough time to make a decision for semantic information in real time. Thus, it is difficult to achieve accurate integration

\*Author for correspondence

with high-level intelligence using the single-layered context integration. The sequential layered approach, however, improves the accuracy of the context integration that divides time intervals and reiterates contexts obtained from time buffers. Such a reiteration efficiently manages context history using the time buffers that have current and recorded contextual data. The sequential layered approach enables the extraction of meaningful contexts based on the fusion procedure for each of the contexts.

Therefore, we propose a sequential layered approach for optimized context integration. The proposed approach applies sequential fusion layers according to divided time intervals to accurately integrate contexts. The proposed approach is managed at an architectural level for high-level semantic analysis of large sensory data. The architecture accommodates dynamic changes (e.g., add, delete, modify, replace, etc.) in heterogeneous sensor environments. Furthermore, the proposed approach is specified by the characteristics of contexts to decide high-level semantic information from different context inputs. The proposed approach is designed as a generalized context integration architecture that supports mutual cooperation among domain-free applications, among heterogeneous sensors, and among user groups. In addition, the approach incorporates the real-time fusion of contexts which are obtained from multiple sensors and facilitates real-time maintenance by sequential context fusion for online dynamic changes of sensor environments (e.g., acceleration, profile, ambient light, outdoor noise, temperature, press, location). Accordingly, the proposed approach presents a way to integrate optimized contexts from any heterogeneous sensor.

## 2. Sequential Layered Approach

Context integration is a process of extracting context information that has high-level meaning, achieved by collecting low-level contexts from heterogeneous sensors<sup>9</sup>. The integrated context, which is produced by context integration, is obtained by fusing inputted contexts. In order to optimize the output of the context integration, we apply sequential fusion layers according to divided time intervals to accurately integrate contexts. Parallel layered approach concurrently processes context inputs according to the system function, but sequential layered approach sequentially processes context inputs according to time. Accordingly, we apply sequential layered approach, because our context integration approach manages context inputs in real time. The applied approach is

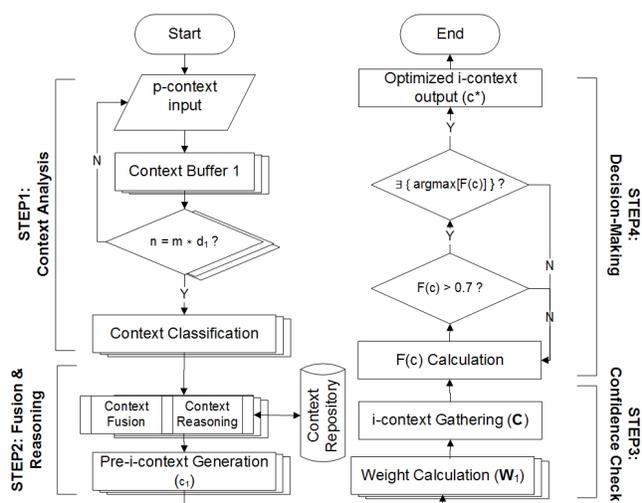
managed at an architectural level for high-level semantic analysis of large sensory data. In particular, our approach is specified by the characteristics of contexts to decide high-level semantic information from different context inputs. Our approach stores contexts from sensors into the time division buffer and then applies results to several context fusion process layers.

Our approach considers different characteristics of heterogeneous sensory data. Some sensory data changes violently; others change very little. For instance, there are a sensor generating data in every 1 Hz and a sensor generating data in every 1 KHz. In this case, our approach has a direct path that collects the sensory data from 1 Hz sensor in the first layer and collects the sensory data from 1 KHz sensor in the second layer. According to those different characteristics, our proposed approach efficiently integrates contexts from heterogeneous sensors. The proposed approach collects context input from those sensors in each context buffer according to time interval of each buffer. Using separated time buffers can induce a time gap between time buffers. To cover the gap, we apply a time-moving window in each buffer. Each layer is divided by the integration interval and generates an integrated context as the divided time interval. Accordingly, the proposed approach integrates contexts when the context input changes, even though it collects contexts periodically. By moving the time window between the time intervals, our method dynamically fuses current context inputs and past inputted context history.

In Figure 1, represents the designed architecture inside each integration layer. The architecture is composed of the analysis, fusion, and reasoning steps. Contexts from various kinds of sensors are collected periodically. The collected contexts are preliminarily integrated according to characteristics of each sub-context of 4W1H (Who, What, Where, When, and How context)<sup>9</sup>. The pre-integrated contexts adopt the appropriate fusion method for each characteristic. Then, the complete integrated contexts are generated by adding the inferred context (Why context). The integrated contexts are stored and managed as the context history in context repository for the future reasoning step.

## 3. Optimization of Context Integration

The aim of our method is to improve the reliability of the integration output. Our approach features



**Figure 1.** Sequential layered context integration procedure.

multiple-layered capability. Our proposed approach is the extended version based on the architecture of the single-layered context integration. To build this architecture, we assume that a minimum sensing period is 50 ms and applied the other intervals in an orderly fashion. The determined interval is for the short-term integration of context. The proposed architecture allows the evaluation of semantic information by integrating various contexts from heterogeneous sensors and improving the reliability of the integration result. The architecture integrates homogeneous and heterogeneous inputs. If the input is homogeneous, the architecture processes the input in the first layer for fast integration. If the input is heterogeneous, then the architecture applies multiple processing layers to increase output accuracy. This procedure consists of context analysis, fusion, reasoning, confidence check, and decision-making steps. Each step, as shown in Figure 1, is described by the following steps.

- **Step 1: Context Analysis:** Context analysis. For context analysis, we assume that the approach obtains a variety of contexts from heterogeneous sensors in discrete time, and each layer has a duration window. The applied moving window has a duration  $d$  at the  $k$ th observation. We can observe context inputs with a duration  $d$  by the following equation:

$$F_{kd} = \sum_{n=k-d+1}^k [\bigcup_{i=0}^{m_n-1} CO_n(i)], \quad d \geq 0, n \geq 0, d \leq k \leq \infty \quad (1)$$

where  $d$  is fusion duration time,  $n$  is discrete time sequence,  $m_n$  is a size of  $CO_n(i)$ , and  $k$  is discrete observation time. Here,  $CO_n(i)$  is a set of discrete context inputs;  $F_{kd}$  is the  $k$ th observation with a duration  $d$ . The equation (1) represents inputted contexts which are stored in each time buffer. In other words, the approach collects  $F_{kd}$  context inputs in each context buffer according to the duration  $d$ . Using (1), we can count  $|p[n]|$ , the number of contexts in a moving window at  $k$ , where  $p[n]$  is a sequence to be represented as a sum of scaled and delayed impulses ( $\delta[n]$ ).

$$p[n] = \sum_{i=k-d+1}^k m_i \delta[n-i] = m_{k-d+1} \delta[n-k+d-1] + m_{k-d+2} \delta[n-k+d-2] + \dots + m_k \delta[n-k] \quad (2)$$

- **Step 2: Context Fusion and Reasoning:** Context integration has two functionalities: context fusion and context reasoning which are key properties of our context integration. Context fusion is a process of integrating context information from multiple sensors to produce a semantic unified context. Precisely, the context fusion integrates each component of the 5W1H context<sup>9</sup> as a characteristic. Context reasoning is a context process of looking for reasons for the current situation. The 5W1H context has a hierarchy which consists of sub-contexts. As described in the previous research<sup>9</sup>, the 5W1H context fusion method is selected according to each characteristic of the sub-contexts. The context reasoning has alternative methods: rule-based reasoning, or statistical reasoning. Rule-based context reasoning sets rules and conditions in a knowledge base. We use JESS<sup>10</sup> or CLIPS<sup>11</sup> according to a development environment for rule-based context reasoning. The defined rule invokes a high-level semantic context output from low-level collected contexts. We assume that there are incidental and essential rule conditions which have different weighting ratios. By using the weighting ratio between both conditions, each weighting factor is calculated, and, then, the rule is executed.
- **Step 3: Confidence Check:** The next step is a confidence check that calculates the weighting factor ( $W_c$ ) for each layer. The calculated confidence makes a reliable output of context integration. To make a reliable decision by means of context integration, a confidence factor of the integrated context should be calculated. Figure 2 shows a diagram of the confidence

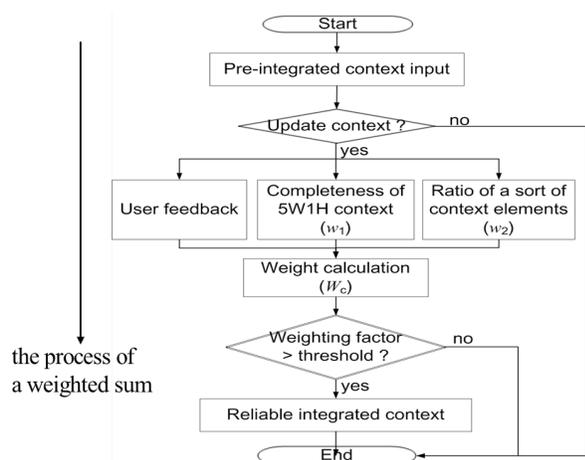


Figure 2. The confidence check in context integration.

check for each layer. By using the pre-integrated context, this procedure ensures reliability of the integrated context that resolves unexpected errors of contexts. To this end, the procedure should compare the current context with the context history and then evaluate whether user feedback, completeness of context, or ratio of context sorts. The user feedback is to check the pre-integration result by a user. The completeness of context is a weight that is a degree of completeness of six (5W1H) contexts. The ratio of context sorts is a weight which indicates how many different context elements are invoked. Accordingly, we calculate a weighted sum from those evaluation factors and verify the weight by the predefined threshold. This confidence check is used in the process to recommend items from multiple contexts that are operated by the weighted sum. This calculation obtains the recommended list by the numerical integration of rule outputs. We assume that user feedback has the highest confidence ( $W_c=1$ ), because user feedback is the most important input. The completeness of the 5W1H context ( $w_1$ ) and the ratio of a sort of context elements ( $w_2$ ) are given. As described in Figure 2, we calculate the weight  $W_c$ :

$$W_c = \begin{cases} 1, & \text{if (user\_feedback)} \\ w_1 \cdot w + w_2 \cdot (1 - w), & \text{else} \end{cases}, 0 \leq W_c \leq 1,$$

$$E = \{e_k \mid e_{who}, e_{what}, e_{where}, e_{when}, e_{why}, e_{how}\}$$

$$w_1 = \sum_{e_k \in E} \frac{r_1(e_k)}{a}, \quad w_2 = \sum_{e_k \in E} \frac{r_2(e_k)}{b},$$

$$(a = 6, b = 1 + 2 + \dots + 6 = 21), 0 \leq w_1, w_2 \leq 1 \quad (3)$$

The weight  $w_1$  is calculated by a sum of a prior probability. The  $e_k$  is an element of six context elements and  $r_1(e_k)$  is a predefined weight of each element. The predefined weight, from zero to six, is decided by a developer, according to the importance of each context element. The weight  $w_2$  is calculated by a sum of a prior probability where  $\{r_1(e_k)/a\}$  is a prior probability and  $a$  is a constant which is the number of context elements. The total elements of the 5W1H are six ( $a = 6$ );  $\{r_2(e_k)/b\}$  is a prior probability,  $b$  is the number of a total sort by the sum of six context elements, and  $r_2(e_k)$  is the number of a sort of context elements.

- **Step 4: Decision Making:** The final step is decision making, which is used to determine the optimized integrated context among a set of pre-integrated contexts produced from multiple layers. In this step, the proposed approach first calculates  $F(c)$  which is a weight function of an integrated context, then evaluates it based on the threshold. Finally, the approach decides the optimized output. To create this decision-making process, we assume that each integrated context has confidence and a different priority. Given data are the number of fusion layers ( $N$ ), a set of integrated contexts from  $N$  layers ( $C = \{c_k \mid c_1, c_2, \dots, c_N\}$ ), a set of context elements, the weight of current integrated context, and a size of each context element. The rate of the number of context elements ( $f(v_c)$ ) is calculated by the size of each element. The function  $f(v_c)$  determines whether contexts are abundant or not. The weight of  $v_c$  is expressed in (4):

$$f(v_c) = \begin{cases} 1, & v_c = e_{who} \\ \frac{n(v_c)}{n(V_c)}, & o.w. \end{cases},$$

$$w(v_c) = \begin{cases} s_1, v_c = e_{who} \\ s_2, v_c = e_{why} \\ s_3, v_c = e_{how} \\ s_4, v_c = e_{what}, e_{where}, e_{when} \\ 0, o.w. \end{cases} \quad (4)$$

where  $v_c$  is an element of  $V_c$ ,  $V_c$  is a sub-set of  $E$ , and  $n$  is a size;  $s_1, s_2, s_3$ , and  $s_4$  are constants. We set the values as  $s_1=0.4, s_2=0.3, s_3=0.2$ , and  $s_4=0.1$ .

Using the weighted sum, we finally decide the optimized integrated context ( $c^*$ ), as in (5):

$$c^* = \underset{c \in C}{\operatorname{argmax}} [F(c)],$$

$$F(c) = W_c \sum_{v_c \in V_c} f(v_c)w(v_c) \quad (5)$$

for all  $c$ , such that  $F(c) > 0.7$ . This constraint means that an optimized output should be greater than the given threshold. We heuristically defined that the minimum threshold is the 70% confidence. This threshold can be adjusted by a developer.

As shown in Figure 3, we developed 20 simulated sensors and 8 real sensors as the context input and we also implemented the optimization process based on the sequential layered context integration. As an experimental

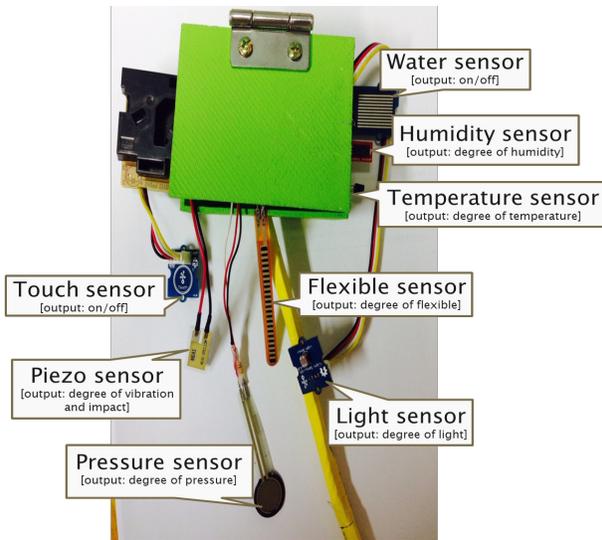


Figure 3. Implemented sensors.

```
[Service>maxFc:optimized_ICI 0.739869 1st iteration
MH<PUBLIC:Body:STRING:NULL:socun
[Service>maxFc:optimized_ICI 0.8624 2nd iteration
MH<HO:PUBLIC:Body:STRING:NULL:socun
MH[Service>maxFc:optimized_ICI 0.88 3rd iteration
MH<HO:PUBLIC:Body:STRING:NULL:socun
MH[Service>maxFc:optimized_ICI 0.894667 4th iteration
MH<HO:PUBLIC:Body:STRING:NULL:socun
MH[Service>maxFc:optimized_ICI 0.917714 5th iteration
MH<HO:PUBLIC:Body:STRING:NULL:socun
MH[Service>maxFc:optimized_ICI 0.926933 6th iteration
MH<HO:PUBLIC:Body:STRING:NULL:socun
MH<HO:PUBLIC:Gender:STRING:NULL:female
MH<HO:PUBLIC:NAME:STRING:NULL:SMin
MH<HO:PUBLIC:Peculiarity:STRING:NULL:student
MH<HEN:PUBLIC:TimeStamp:STRING:NULL:Now
MH<HERE:PUBLIC:Location:STRING:NULL:Home
HO<HAT:PUBLIC:Object:STRING:NULL:NULL
HO<HAT:PUBLIC:SENSOR:STRING:NULL:PhysicalSensor
HO<OW:PUBLIC:Flexi:STRING:NULL:552
HO<OW:PUBLIC:Humid:STRING:NULL:28.22
HO<OW:PUBLIC:Light:STRING:NULL:591
MH<OW:PUBLIC:Piezo:STRING:NULL:Press
HO<W:PUBLIC:Tempe:STRING:NULL:25.66
HO<W:PUBLIC:Touch:STRING:NULL:ON
WHY:PUBLIC:Reasoning:STRING:NULL:NULL
```

(a) Implementation Result of Context Integration

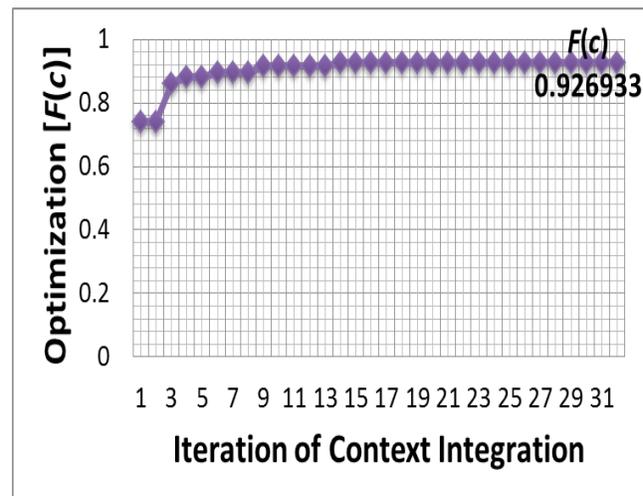
result in Figure 4, we obtained the optimized integrated context( $C^*$ ) which was converged with the threshold value over the 92% confidence (at the 6<sup>th</sup> iteration). From the above result, the proposed approach produces a reliable context output by optimizing an integrated context. The proposed approach reduces superfluous context data by reiterating inputs and improves the accuracy of context integration.

## 4. Conclusion

In this paper, we proposed a sequential layered approach for optimized context integration. The proposed approach applied multiple fusion layers according to divided time intervals to accurately integrate contexts. As the experimental process, we presented that the proposed approach reduced superfluous context data by reiterating inputs and improved the accuracy of context integration. In addition, our proposed approach can serve as a fundamental study of context awareness in large-scale applications and the convergence among heterogeneous domains. Also, our approach can be applied to the management of applications (e.g., smart home, healthcare, meeting space) in large systems.

## 5. Acknowledgement

This research was supported by the Daegu University Research Grant, 2014.



(b) Optimized Output Result ( $F(c)$ )

Figure 4. Experimental result.

## 6. References

1. Huebscher C, McCann A. An adaptive middleware framework for context-aware applications. *Personal and Ubiquitous Computing*. 2006 Feb; 10(1):12–20.
2. Karypidis A, Lalis S. Automated context aggregation and file annotation for PAN-based computing. *Personal and Ubiquitous Computing*. 2006 Oct; 11(1):33–44.
3. Chen G, Li M, Kotz D. Design and implementation of a large-scale context fusion network. *Proceeding of 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*; 2004 Aug 22–26. p. 246–55.
4. Buchholz T, Krause M, Linnhoff-Popien C, Schiffers M. CoCo: dynamic composition of context information. *Proceeding of 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*; 2004 Aug 22–26. p. 335–43.
5. Henricksen K, Indulska J. A software engineering framework for context-aware pervasive computing. *Proceeding of 2nd IEEE International Conference on Pervasive Computing and Communications*; 2004 Mar 14–17. p. 77–86.
6. Cohen NH, Purakayastha A, Wong L, Yeh DL. iQueue: a pervasive data composition framework. *Proceeding of 3rd International Conference on Mobile Data Management*; 2002 Jan 8–11. p. 146–53.
7. Cohen NH, Lei H, Castro P, Davis II JS, Purakayastha A. Composing pervasive data using iQL. *Proceeding of 4th IEEE Workshop on Mobile Computing Systems and Applications*; 2002 Jun 20–21. p. 94–104.
8. Wu H. *Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory* [PhD Thesis]. Pittsburgh, PA, USA: Carnegie Mellon University; 2004.
9. Oh Y, Han J, Woo W. A context management architecture for large-scale smart environments. *IEEE Communications Magazine*. 2010 Mar; 48(3):118–26.
10. Sandia National Laboratories. JESS. Available from: <http://herzberg.ca.sandia.gov/>
11. Sourceforge.net. CLIPS. Available from: <http://clipsrules.sourceforge.net/>