

# An Adaptive Error Event Control Model for Smart Festival Management System running on Situation-Awareness Environments

Soongohn Kim<sup>1</sup> and Eungnam Ko<sup>2\*</sup>

<sup>1</sup>*Department of Computer Science, Joongbu University, 101 Daehakro, Chubu-Meon, GumsanGun, Chungnam, 312-702, Korea*

<sup>2</sup>*Division of Information & Communication, Baekseok University, 115 Anseo-dong, Dongnam-gu, Cheonan, Chungnam, 330-704, Korea*

*sgkim@joongbu.ac, sskn@daum.net*  
*\*Corresponding Author*

## Abstract

*This paper proposes a new adaptive error event control model for smart festival management running on situation-awareness environments by analyzing the window and attributes of the object error event. This paper presents the design of the adaptive error event control model for sharing control, which is running on RCSM (Reconfigurable Context Sensitive Middleware) for smart festival management. RCSM provides standardized communication protocols to interoperate an application with others under dynamically changing situations for the smart festival management.*

**Keywords:** *adaptive error event control model, situation-awareness environments, smart festival management*

## 1. Introduction

Currently, the regional local festivals, which are a part of community development projects that utilize the regional cultural resources of the local self-governing entities, contribute in the nurturing of local cultures. And, because of the advantages of festivals -- directly and indirectly -- in terms of their increasing the incomes of the local residents, raising the potential for regional development and being suitable for the acceptance by the dynamic forms of the modern tourism, lately these festivals have been utilized quite a lot. The smart festival management system is a management system that, for the various festivals that are operated by the local self-governing entities and agencies, enables a direct operation of all the process phases from the advance preparation phase to operation phase and the post management and the administrative tasks, etc. of the planning operational headquarters and agencies [1]. Context awareness (or context sensitivity) is an application software system's ability to sense and analyze context from various sources; it lets application software take different actions adaptively in different contexts [2]. In a ubiquitous computing environment, *computing anytime, anywhere, any devices*, the concept of situation-aware middleware has played very important roles in matching user needs with available

computing resources in transparent manner in dynamic environments [3 - 5]. Although the situation-aware middleware provides powerful analysis of dynamically changing situations in the ubiquitous computing environment by synthesizing multiple contexts and users' actions, it is difficult to avoid a problem of the seam in the ubiquitous computing environment during data collection for smart festival management. This paper proposes a new adaptive error event control model based on situation-awareness environments such as RSCSM (Reconfigurable Context-Sensitive Middleware) for smart festival management by analyzing the window and attributes of the object error event. RSCSM provides an Object-based framework for supporting context-sensitive applications. It shows how all of RSCSM's components are layered inside a device. All of RSCSM's components are layered inside a device. However, it did not include an adaptive data control and platform for multimedia collaboration in the architecture. In the Context Toolkit, a predefined context is acquired and processed in context widgets and then reported to the application through application-initiated queries and callback functions. In this Reconfigurable Context-Sensitive Middleware(RSCSM), Stephen S. Yau *et al.* [2] proposed a new approach in designing their middleware to directly trigger the appropriate actions in an application rather than have the application itself decide which method(or action) to activate based on context. Section 2 describes related works. Section 3 denotes an adaptive collaboration platform based on RSCSM for festival event control. Section 4 present system results. Section 5 present conclusions.

## 2. Related Works

The field of fault-tolerant computing has evolved over the past twenty-five years [8]. In spite of this current trend, however, study on fault- tolerance of application software has not actually been enough. Generally, fault-tolerance system can be classified as software techniques, hardware techniques and composite techniques [9]. The tolerance of software faults is in most cases more difficult than dealing with hardware faults since most software-fault mechanisms are not well understood and do not lend themselves readily to "nice" techniques such as error coding [10].

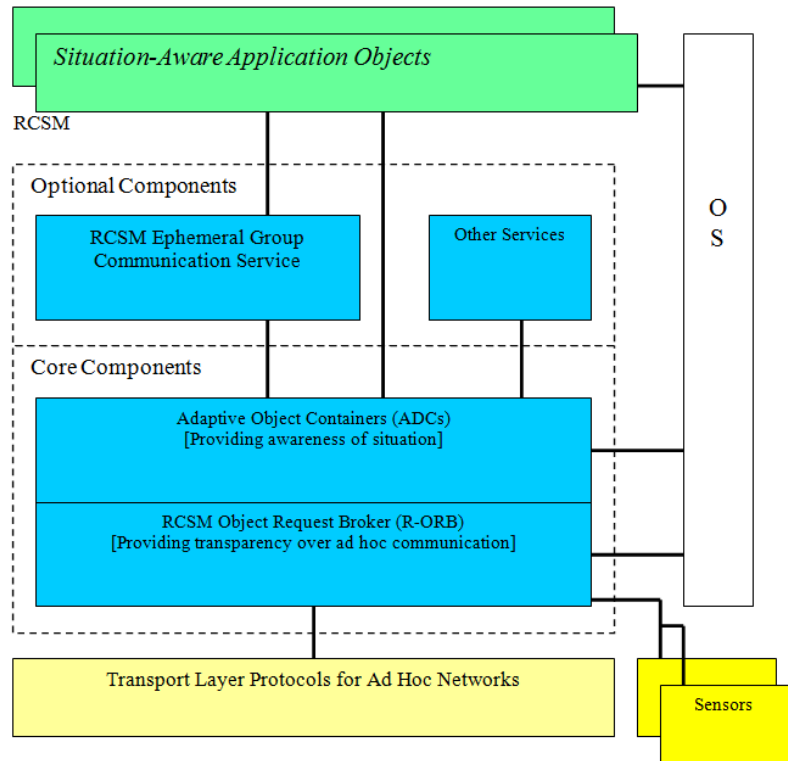
## 3. RSCSM

### 3.1. RSCSM

A conceptual architecture of situation-aware middleware based on Reconfigurable Context-Sensitive Middleware (RSCSM) is proposed in [2]. Ubiquitous applications require use of various contexts to adaptively communicate with each other across multiple network environments, such as mobile ad hoc networks, Internet, and mobile phone networks. However, existing context-aware techniques often become inadequate in these applications where combinations of multiple contexts and users' actions need to be analyzed over a period of time. Situation-awareness in application software is considered as a desirable property to overcome this limitation. In addition to being context-sensitive, situation-aware applications

can respond to both current and historical relationships of specific contexts and device-actions.

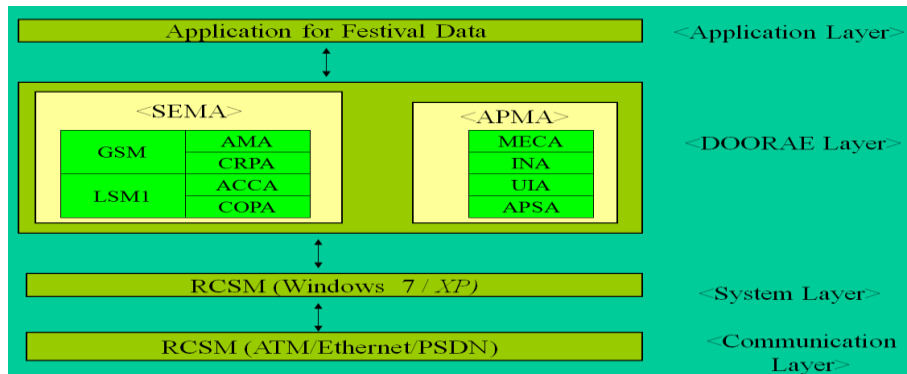
However, it did not include an adaptive error event control and platform for multimedia collaboration in the architecture.



**Figure 1. Overview of Situation-Aware Middleware**

### 3.2 Collaboration Works based on RCSM for Festival Data

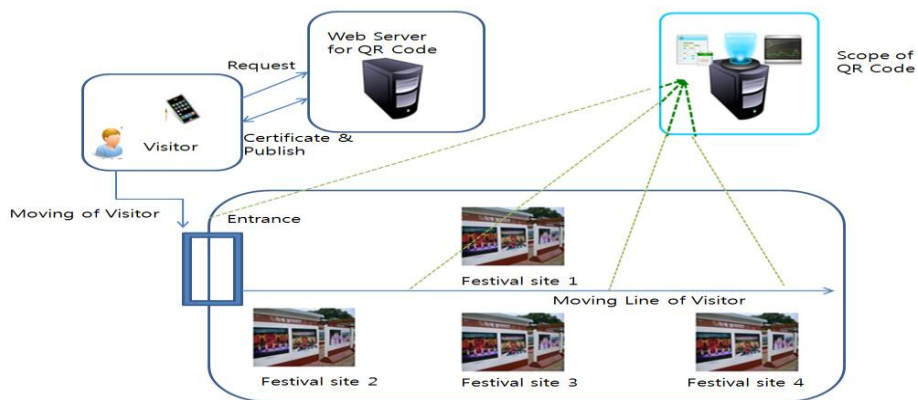
DOORAE is made up of a distributed architecture for a collaborative multimedia distance education system and forms several levels according to service functions. DOORAE is a foundation technology for computer collaborative work that allows development of required application by combining many agents composed of units of functional module, provided by DOORAE, when user wishes to develop new application field. As it can be seen on Figure 2, DOORAE's basic structure consists of DOORAE application program, DOORAE agents, operating system and communication subsystem. DOORAE agents are composed of SEMA and APMA. SEMA is a session management agent that appropriately control and manages session and opening / closing of sessions, even in the case of several sessions being generated at the same instant. APMA is an application management agent that handles request of application. SEMA consist of GSM, LSM, AMA, CRPA, ACCA, and COPA. APMA consist of MECA, INA, UIA, and APSA. AMA is an agent that has functions of application management. CRPA is an agent that has functions of managing formation control of DOORAE communication protocol. ACCA is an agent that has functions of managing floor control and concurrency control. COPA is an agent that has functions of providing participants same view.



**Figure 2. DOORAE Platform based on RCSM for Festival Data**

### 3.3 Application for Festival Data Visitor Event Flow

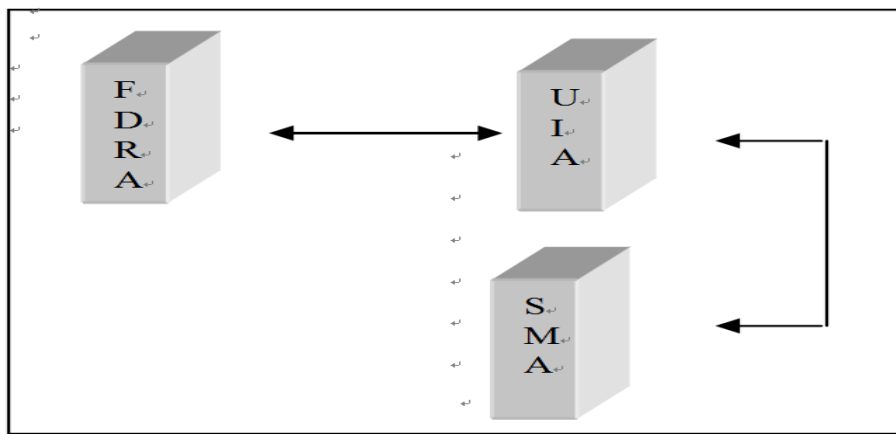
For a festival, based on the characteristics of the festival and in accordance with separation into a preparation phase, operation phase, post management and reporting phase, the requirements for each of the phases are defined and a management system is developed for -- in a logical order -- planning, design and development. For the requirements of the festival operation phase, what is needed are the assessment data needed for operation such as punctuality check, smoothness check, surveys, etc. and the management side requirements such as schedule reporting, visitor feedback, event management, risk management, visitor traffic flow trace, etc. During the festival operation, systems for the design and construction of festival site facilities and festival risk management are important requirements as well. For smooth feedback during the festival operation phase, the interoperability between the management that applied QR Code (Quick Response Code) and automatic visitor processing system is essential. The concept drawing for automatic visitor processing system is shown in Figure 3.



**Figure 3. An Example of QR Code and Automatic Visitor Event Flow for Festival Data**

### 3.4 Error Event Flow of Application for Festival Data Visitor Event Flow

As shown in Figure 4, you can see the message flows in relationship between FDRA (Fault Detection and Recovery Agent) and the application software for festival system. UIA is an agent which plays a role as an interface to interact between the user and FDRA for festival system. UIA is a module in FDRS. UIA has functions which receive user's requirement and provide the results for the user. SMA is an agent which plays a role in connection of UIA and FDRA and as management for the whole information for festival system. SMA consists of GSM (Global Session Manager), Daemon, LSM (Local Session Manager) and PSM (Participant Session Manager) for festival system. FDRA is an agent which plays a role in detecting error and recovering it for festival system. That is, FDRA becomes aware of error occurrence after it receives requirement of UIA. FDRA informs SMA of the results of detected errors for festival system. Also, FDRA activates application software for error event automatically. It informs SMA of the result again for festival system.



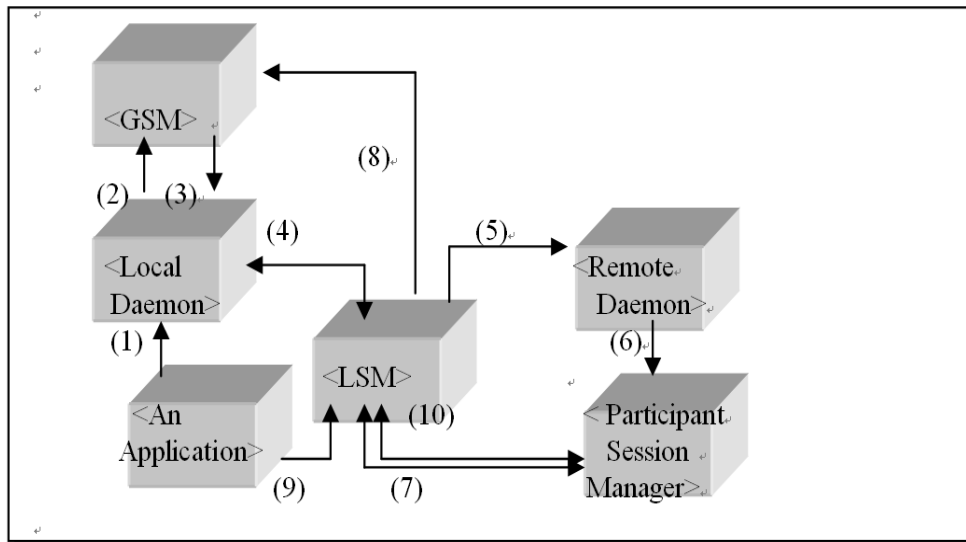
**Figure 4. An Error Detection Event Flow of Automatic Visitor Event Flow for Festival Data**

Step1. (Error detection for festival system): We are first in need of a method to detect an error for session's recovery for festival system. One of the methods detects an error by using "hooking techniques in MS-windows API (Application program Interface)". When an error occurs, a hook is a point in the Microsoft Windows message-handling mechanism where an application can install a subroutine to monitor the message traffic in the system and process certain types of message before they reach the target window procedure. Windows contains many different types of hook for festival system.

Step2. (Error recovery for festival system): After a system is detected, it processes recovery for festival system. Session management can create the sequence below and you can see the message flow in Figure 5.

- (1) Create a session with initial configuration information for festival system
- (2) Request port ids for A/V servers to build-up a Local Session Manager for festival system
- (3) Assign port ids for A/V servers of an application for festival system
- (4) Invite to the session and build-up a session instance monitor for festival system

- (5) Send invited messages to start build-up of session instance monitor for festival system
- (6) Build-up Session Instance Monitor using the configuration information from LSM for festival system
- (7) Send joint message to the Local Session Manager for festival system
- (8) Send session information to Global Session Manager for set-up of GSMT for festival system
- (9) Begin a session for festival system
- (10) Exchange message or command between LSM and PSM and media data between media server based on interpretation of message handler for festival system



**Figure 5. An Error Recovery Event Flow of Automatic Visitor Event Flow for Festival Data**

#### 4. Simulation Results

To evaluate the performance of the proposed system, an error detection method was used to compare the performance of the proposed model against the conventional model by using DEVS formalism [11-15].

(Simulation 1 for festival system)

In the first simulation, we have considered composition component as shown in Table 1 for festival system. The atomic models are EF, RA1, and ED1. The combination of atomic models makes a new coupled model. First, it receives input event, *i.e.*, polling interval. The value is an input value in RA1. An output value is determined by the time related simulation process RA1. The output value can be an input value in ED1. An output value is determined

by the time related simulation process ED1. We can observe the result value through transducer for festival system.

**Table 1. Atomic Model and State Variable for festival system**

Com-ponent	State Variable	Contents
EF (genr)	Poll_int	Polling interval
RA1	Ra_re_time App_cnt  Ra_re_t_a	Response time The number of application program Accumulated response time
ED1	Tat_t_a	RA accumulated response time

(Simulation 2 for festival system)

In the second simulation, we have considered composition component as shown in Table 2 for festival system. The atomic models are EF, RA2, and ED2. The combination of atomic models makes a new coupled model. First, it receives input event, i.e., polling interval. The value is an input value in RA2. An output value is determined by the time related simulation process RA2. The output value can be an input value in ED2. An output value is determined by the time related simulation process ED2. We can observe the result value through transducer for festival system.

**Table 2. Atomic Model and State Variable for festival system**

Com-ponent	State Variable	Contents
EF (genr)	Poll_int	polling interval
RA2	Ra_re_time App_cnt  Ra_re_t_a	Response time The number of application program Accumulated response time
ED2	Tat_t_a	RAaccumulated response time

We can observe the following for festival system. The error detected time interval is as follows. Conventional method:

$$2 * \text{Poll\_int} * \text{App\_cnt}$$

Proposed method:  $1 * \text{Poll\_int}$

Therefore, proposed method is more efficient than conventional method in error detected method because of  $\text{App\_cnt} \geq 1$ . We have compared the performance of the proposed method with conventional method for festival system.

As shown in Table 1, conventional multimedia distance systems are Shastra, MERMAID, MMconf, and CECED. you can see the characteristic function of each system function for multimedia festival system.

**Table 1. Function of Distance System for Festival Data**

Function	Shastra	MERMAID	MMconf	CECED
OS	UNIX	UNIX	UNIX	UNIX
Development Location	Purdue Univ. USA	NEC, JAPAN	Cam-Bridge USA	SRI, International
Development Year	1994	1990	1990	1993
Structure	Server /client	Server /client	Centralized or Repliated	Repliated
Other Service based on RCSM	No	No	No	No
Error Event Control for Festival Data	No	No	No	No

## 5. Conclusions

This paper proposed a new model adaptive error event control model based on situation-awareness environments for smart festival management by analyzing the window and attributes of the object for error event. As a result, it offers a seamless view without interfering with concurrency control is also suggested for smart festival management. In the future work, fault-tolerance system for smart festival management will be generalized to be used in any environment, and we will progress the study of domino effect for distributed multimedia environment as an example of situation-aware applications for smart festival management.

## References

- [1] B. C. Lim, S. G.Kim and B. C. Lee, "Requirements Analysis for Smart Festival Management System", In Proceedings of the 37<sup>th</sup> Conference of the KIPS, vol.19, no. 1, (2012) April, pp. 1004-1006.
- [2] S. Yau, F. Karim, Y. Wang, B. Wang and S. Gupta, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing", IEEE Pervasive Computing, vol. 1, no. 3, (2002) July-September, pp. 33-40.
- [3] S. S. Yau and F. Karim, "Adaptive Middleware for Ubiquitous Computing Environments", Design and Analysis of Distributed Embedded Systems, Proc. IFIP 17th WCC, vol. 219, (2002) August, pp. 131-140.
- [4] S. S. Yau and F. Karim, "Contention-Sensitive Middleware for Real-time Software in Ubiquitous Computing Environments", Proc. 4<sup>th</sup> IEEE Int'l Symp. on Object-Oriented Real-time Distributed Computing (ISORC 2001), (2001) May, pp. 163-170.
- [5] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21<sup>st</sup> century", IEEE Computer, vol. 36, Issue 3, (2003) March, pp. 25-31.
- [6] V. P. Nelson and B. D. Carroll, "Fault-Tolerant Computing: Chapter 1 Introduction to Fault-Tolerant Computing", IEEE Computer Society Order Number 677, Library of Congress Number 86-46205, IEEE Catalog Number EH0254-3, ISBN 0-8186-0677-0.
- [7] D. K. Pradhan, "Fault-Tolerant Computer System Design", Prentice Hall Ptr, (1996).
- [8] E. Wold, T. Blum, D. Keislar and J. Wheaton, "Content-Based Classification, Search, and Retrieval of Audio", IEEE Multimedia, vol. 3, no. 3, (1996) Fall, pp. 27-36.
- [9] H. Zhang A. Kankanhalli and S. Smoliar, "Automatic Partitioning of Full-motion Video", A Guided Tour of Multimedia Systems and Applications, IEEE Computing Society Press, (1995).
- [10] V. P. Nelson and B. D. Carroll, "Fault-Tolerant Computing: Chapter 5: Software Fault Tolerance", IEEE Computer Society Order Number 677, Library of Congress Number 86-46205, IEEE Catalog Number EH0254-3, ISBN 0-8186-0677-0.
- [11] B. P.Zeigler, T. H. Cho and J. W. Rozenblit, "A Knowledge-Based Simulation Environment for Hierarchical



- Flexible Manufacturing”, IEEE Transaction on Systems, Man, and Cybernetics-Part A: System and Humans, vol. 26, no. 1, (1996) January, pp. 81-90.
- [12] T. H. Cho and B. P. Zeigler, “Simulation of Intelligent Hierarchical Flexible Manufacturing: Batch Job Routing in Operation Overlapping”, IEEE Transaction on Systems, Man, and Cybernetics-Part A: System and Humans, vol. 27, no. 1, (1997) January, pp. 116-126.
- [13] B. P. Zeigler, “Object-Oriented Simulation with hierarchical”, Modular Models, Academic Press, (1990).
- [14] B. P. Zeigler, “Multifaceted Modeling and Discrete Event Simulation”, Orlando, FL: Academic, (1984).
- [15] B. P. Zeigler, “Theory of Modeling and Simulation”, John Wiley, NY, USA, (1976), reissued by Krieger, Malabar, FL, USA, (1985).

## Authors



### Soongohn Kim

He received Ph.D. degrees from Chonbuk National University, Seoul Korea, in Computer Engineering in 1999. He has been working as a Professor in Joongbu University from March 1995. His research interests include Ubiquitous Computing, Distributed Computing, Database Integrity, Cryptographic Protocol, A methodology of Software Development, Software Evaluation, Networks and so on.



### Eungnam Ko

He received Ph.D. degrees from Sungkyunkwan University, Seoul, Korea, in Computer Engineering in 2000. He has been working as a Professor in Baekseok University from March 2001. His research interests include Web Services and Multimedia, Ubiquitous Computing, Distributed Computing, Fault Tolerance, Distance Education and so on.

