International Journal of Innovative Computing, Information and Control
Volume **7**, Number **5(A)**, May **2011**

ICIC International ©2011 ISSN 1349-4198
pp. 2225–2234

A CONTEXT MODEL WITH A TIME-DEPENDENT MULTI-LAYER EXCEPTION HANDLING POLICY

Chia-Hung Shih[1,2], Nobukazu Wakabayashi[2], Saburo Yamamura[2] and Chen-Yuan Chen[1,3,*]

[1]Department and Graduate School of Computer Science
National Pingtung University of Education
No. 4-18, Ming Shen Road, Pingtung 90003, Taiwan

[2]Graduate School of Science and Technology
Kobe University
5-1-1 Fukae-minamimachi, Higashinada-ku, Kobe 658-0022, Japan

[3]Doctoral Program in Management
National Kaohsiung First University of Science and Technology
No. 2, Jhuoyue Road, Nanzih District, Kaohsiung 811, Taiwan
[*]Corresponding author: cyc@mail.npue.edu.tw

Received December 2009; revised April 2010

Abstract. *Context aware ubiquitous technology provides us with much more conveniences in our everyday life by defining contextual information in appropriate ways. Up to now, there has been a vast amount of researches and many context models proposed. Many characteristics have been incorporated into such models such as users, physical locations, available resources, etc. Some also take time factors into consideration. Most still focus on the interactions between context and users/groups rather than discussing follow-up operations deeply. In this study, we extend the scope of the context model to include metadata related to services and use it to describe the behaviours of general operations. Also, by further dividing the time factor into levels of importance, we can use time as the primary control factor to manipulate some specific operations so that operations could be triggered by real time constraints and handled properly. Some critical operations within the whole system may need to be handled by specific users or groups. The occurrence of unexpected situations when the default operator cannot proceed with the corresponding operation on time must be taken care of by a pre-designed time-dependent multi-layer exception handling policy to maintain the integrity and security of the system. We also propose a critical operation scenario, describe the processes of our model and explain the importance of time driven events to prove its applicability.*

1. **Introduction.** As e-communities grow in quality and quantity, online users require more appropriate tools designed to suit their needs and environments. Such tools are not needed in real-world communities where human beings interact with each other directly. Access to context aware ubiquitous content is facilitated by providing intuitive ways for accessing Web content based on the user's surrounding context [1]. Context, in this case, is any information that can be used to characterize the situation of an entity, where that entity can be a person, place or a physical/computational object [3]. Information can include where and when the web users are, as well as what the web content available nearby is, and so on. It is indeed the context that distinguishes ubiquitous web access from mobile access.

There has been extensive research effort directed at the development of context-awareness toolkits including HP's Cooltown project [4], Dey's Context Toolkit [2], the CB-SeC framework [11] and the Gaia middleware [7], to name a few. These toolkits provide functions to help service requesters obtain context-based services or enable content adaptations related to the user's contextual information. In order to promote context aware service provision Kouadri et al. [11] proposed a formal definition of service context to model contextual information. Lemlouma et al. [14] proposed a framework for obtaining metadata and web content information. They used Composite Capabilities/Preferences Profiles (CC/PP) as interoperable context representation to enable communication and negotiation of device capabilities and user preference in terms of a service invocation. There have been several Web Ontology Language (OWL)-based context models presented [5,6,13] to provide high-quality results of service discoveries beyond the expressive limitations of CC/PP. Ontology is used to describe contextual information including location, time, device, preference, network, etc. Semantic contextual information obtained with inductive or deductive techniques can be utilized to perform matches against both the user and service provider's context semantically. Also, the interactions within the u-learning environment are present [8-10] to provide timely content.

The popularity and growth of the ubiquitous environment has brought huge convenience to human society. Ubiquitous technology collects information about the surrounding environment (such as physical locations, available resources, communication equipment, etc.) and proceeds to apply the appropriate procedures based upon various judgment policies and automatic mechanisms. On the down side, this automation could lead to security problems, also a serious issue. Thus, some context models also incorporate security mechanisms to ensure the safety of the whole system. However, even though technology continues to progress, there are still some parts of services that need to be manipulated by human beings. In this paper, we call these kinds of operations critical operations. Critical operations need to be carried out or confirmed by a human being to make sure that the whole system is working correctly. The time factor is very important when dealing with critical operations, which are normally scheduled tasks or assigned to be carried out at a certain moment. In our service context model, we define several time factor elements within a critical operation and trigger the events in real time. If a critical operation cannot be successfully carried out following its default schedule, there are specific follow-up exception handling processes that should be taken relative to the ongoing situations. There could also be several levels defined to the exception handle policy depending on the reaction time the level of the emergency.

2. **Architecture.** The concept of metadata is well known in computer science. A general definition is "data about data", i.e., data that describes the original data. It improves the value of the operational data by giving applications and users additional information on the data's origin, its precision and its staleness.

Context models store data on the user's context: location, surroundings (what is around him and close-by), intentions, activities, etc. So far, most context models have focused on providing only the data that reflects the current situation in the real world. In this study, we go one step further and extend the scope of context models to also contain data about data, the so-called metadata.

The objective of this study is to provide a contextual architecture with time event driven facility. We use a context model to describe the behaviours of some services, a time trigger to manipulate critical operations, an role-based access control (RBAC) model to deal with user permissions and a multi-layer exception handle policy to deal with unexpected situations.

We not only use the role-based access control (RBAC) [15] architecture to provide permission management for the user/group, but also extend our context model to have the

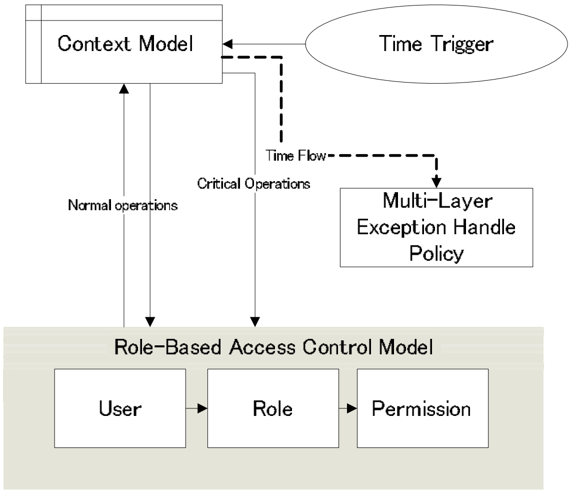

FIGURE 1. Overview of the architecture of the proposed model

ability to deal with critical operations by applying a time-dependent multi-layer exception handling policy.

On the security level, although authentication is an important step toward authorization, specific authentication mechanisms are not the focus of this paper; any authentication mechanism that securely transmits the credentials to the user in such a manner that credentials cannot be tampered with, stolen or forged, is used. Given these properties, the access control can rely on the credential to identify the requester.

Figure 1 shows an overview of our system. It is composed of the context model, the RBAC model, a time trigger and an exception handler.

Through the RBAC model, a person can be authenticated as a regular user, and obtain permissions according to the role they are assigned. Normally, users are free to connect to the resources for which they have permission, at any time and any place. However, when a critical operation's initial condition matches a moment in real time, the time trigger will initiate corresponding services and ask related users to take a reaction default designed into the system. If the relevant users cannot accomplish the operation as set up in the conditions, the exception handler will be initiated to ask for advanced processes. Our exception handling policy can be divided into several levels depending on the passage of time. After the passage of time, if the critical operation is still in an abnormal situation, then a higher-level exception policy will be applied.

3. **Context Model.** Metadata helps various applications to assess the quality and select the desired data in a fine-grained fashion. Storing the metadata explicitly in the context model increases its flexibility and facilitates the integration of many local context models into a global one. This in turn, enables many applications and data providers to share a lot of data, which offers applications a larger-than-ever pool of data and cuts down deployment times and costs.

In this study, we use our context model to describe the behaviours of services. The most important feature of our context model is that it extends time features from normal context models so then we can apply a background time trigger (according to real time) and deal with some time-dependent exceptions.

The proposed model requires three key elements.

(1) The user is assigned as a role with specific permission to access resources under his/her privilege limits within the system.

(2) Critical operations can be triggered automatically in comparison with real time.

Table 1. Context model architecture

Contextual Element		**Content**
Name		Context (service/information) name.
Entity		If the context is a service, this should address where the entity of service actually exists.
Description		Context content description could be used to announce necessary information to users/groups.
Owner		The owner (highest permission user) of this context.
Physical Location		Limitation of where this context can be accessed. If there is no limitation, then the value should be "null".
Attribute		For ordinary operations this attribute can be set to "normal". For critical operations this attribute can be set to "critical", which can be triggered by real time.
Property		Define the property as "public" or "private"; if the context is private, it could be further divided into groups.
Current State		Shows whether this context is normal or abnormal.
Relevance		The context could be related to specific users/groups; only legal users/groups with permission are allowed to access it.
	Informing time	Before the initial context, the time that the related member is informed.
	Operation's initial time	The starting time of a critical operation.
Time	Exception's initial time	The time the exception handling policy begins to be applied.
	Exception handling time interval	The time interval when different exception policies will be applied.
	Operation destruction time	The time this context fails.

(3) Critical operations require specific users/groups to have permission to handle them.

By following the above principles, each user is assigned a role subset from the entire role set. Similarly, each resource is assigned a permission subset from the entire permission set for each role that has the privilege to access it. Table 1 illustrates the features of our context model: name, entity, description, owner, physical location, attribute, property, current state, relevance and time. The time factor is used to trigger and manage critical operations, including informing time, initial operation time, initial exception time, exception handling time interval and operation destruction time. A schematic representation of our context model is shown in Figure 2.

This model expresses the ordinary workings of the system. However, as mentioned earlier, a critical operation is started by a time trigger, affected by the time factor and should be applied following the exception handle policy. They will be described in next section.

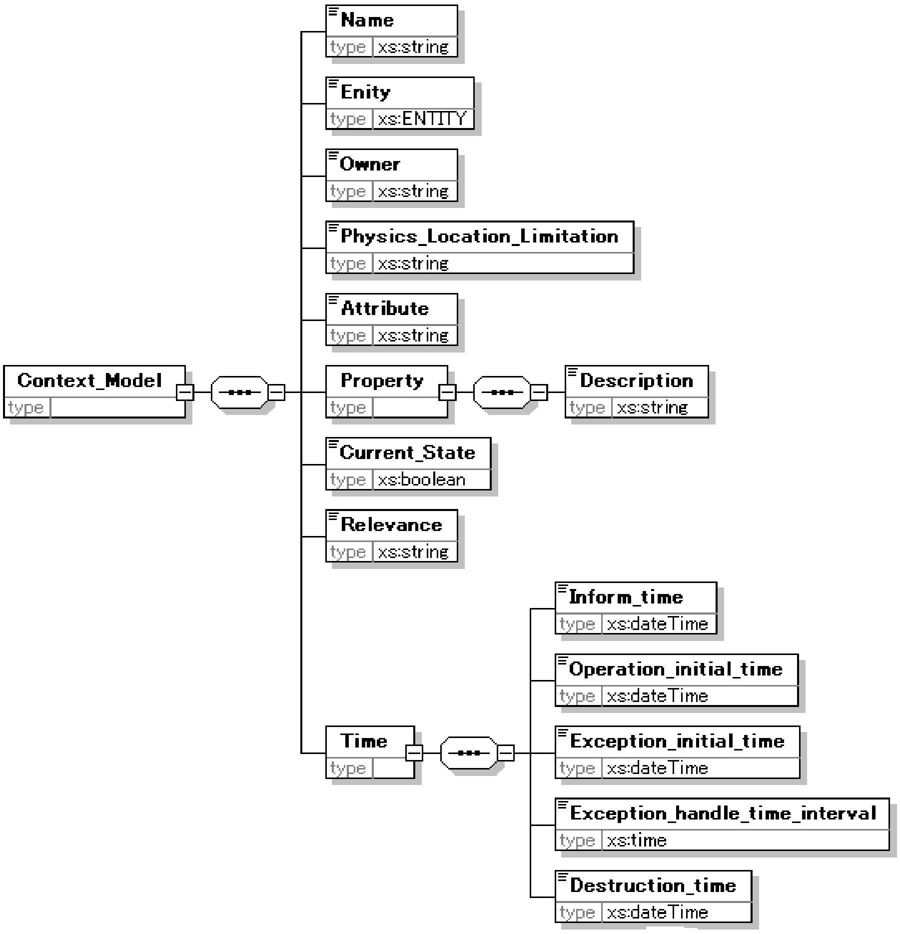

FIGURE 2. Schematic diagram of the context model

4. **Time Trigger and Exception Handle Policy.** Within our context model, the time factor is a very important control factor to make sure that critical operations operate correctly. The facilities of the time factor elements in our model are described as follows.

(1) Informing time: before the operation is initiated, the system can send information instantly to related users to remind them.

(2) Initial operation time: the starting time of a critical operation.

(3) Initial exception time: after a critical operation is initiated, this is the time the exception handler is initiated if the operation is not handled in time.

(4) Exception handling time interval: the time interval for the different exception policies being applied. After the passage of a specified time, if a critical operation is still in an abnormal situation, the corresponding exception policies will be applied, appropriate to the different time intervals.

(5) Operation destruction time: the time that a critical situation fails. If the ongoing real time passes this destruction time, the situation will be treated as a serious system error. All services in the system will be temporarily paused, and the system administrator informed to deal with the error immediately.

There are different levels to the exception handling policy depending on the passing of time. The architecture of our time-dependent multi-layer exception handle policy is illustrated in Figure 3, and its algorithm is shown in Figure 4.

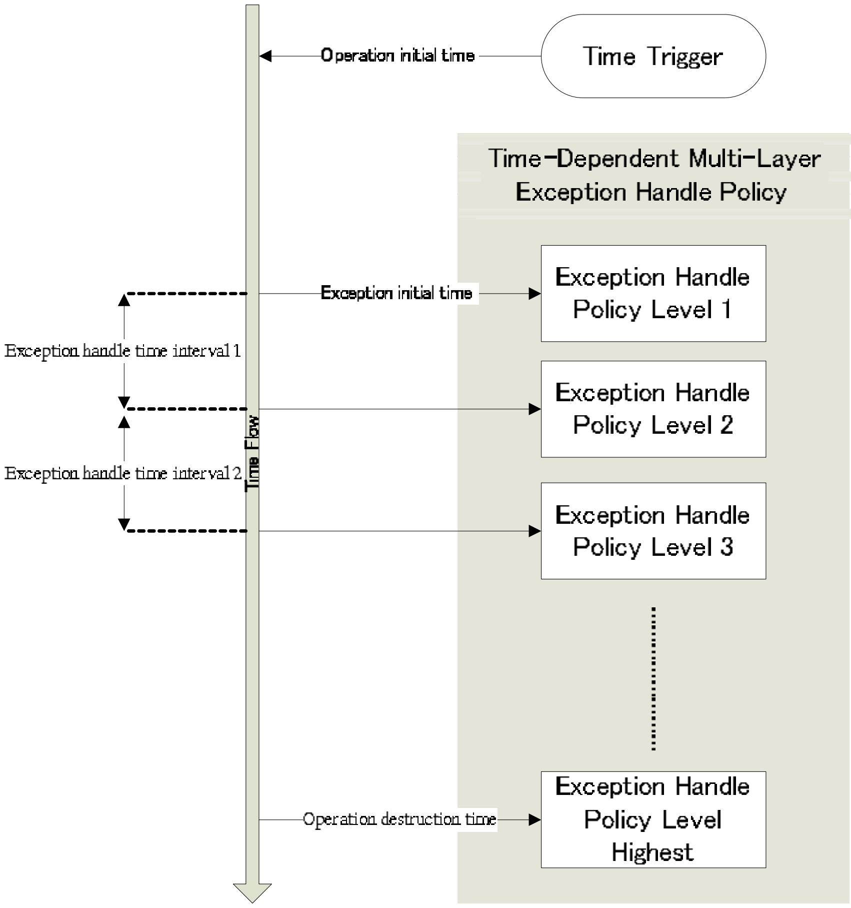

Figure 3. Architecture of the exception handling policy

The exception handling policy can be divided into several levels, depending on the passing of time. The more time that passes, the more serious the exception handling policy that will be applied. If the requested reaction is implemented within the exception handling procedure, the follow-up exception handling processes will be cancelled, and the system will return to its normal working situation. However, the reason for the exception and the handling processes should be stored for future reference and improvement.

5. **Case Study.** In this section, we look at an example to describe the behaviours of our context model. The example describes a bank security guarding system that includes some critical operations, in this case, a scheduled after business hours night patrol. Banking is an industry that needs high security. Every operation and person needs to be authenticated to as having permission to access related resources.

Here is the scenario. Assume that the bank will be patrolled every two hours. The time is now is AM 00:00, and the next scheduled patrol will be at AM 01:00. The system default will send information to remind the guards in the next shift 30 minutes before the initiation of critical operations. In this case, they receive a reminder at AM 00:30. The exception handling time intervals are set to be 10, 15 and 30 minutes for three different levels. During each patrol, two guards must use their PDAs to sign into the system when entering the money storage area. The guards should then enter the area to check the situation. If everything is fine, they confirm that all is clear to the system. The operation destruction time is set to be one hour after the initial scheduled patrol time, in this case, AM 02:00.

```
Time-Dependent Multi-Layer
Exception Handle Policy Algorithm

operation initial time matches with real time {
    operation initiate ;
}

If ( operation destruction time matches with real time &&
_system.current_state == abnormal ) {
    system.false==true;
    inform system.administrator;
    break;
}

exception initial time matches with real time {
    if (corresponding reaction is taken ){
        system.current_state = normal;
        break;
    } else {
        system.current_state = abnormal;
        exception policy level i = 1;
        exception policy level i . apply ();
        next exception check time = exception initial time +
        _exception handle time interval i ;

        while ( system.current_state != normal ) {
            if ( next exception check time matches with real time ) {
                if (corresponding reaction is taken ){
                    system.current_state = normal;
                    break;
                } else {
                    exception policy level i = exception policy level i + 1;
                    exception policy level i . apply();
                    next exception check time = exception initial time +
                    _exception handle time interval i ;
        }
    }
}
```

FIGURE 4. Algorithm for the time-dependent multi-layer exception handling policy

In this example scenario, if any exception to the procedure happens, the exception handler will be applied. Possible situations are described below:

(1) Exception situation 1: If the time is AM 01:00 and both guards are not there to sign in before AM 01:05 (the initial exception handling time), then the first level of the exception handle policy will be applied. Information will be sent instantly to the guard's PDAs to remind them to show up. At this moment, the money storage area will be locked-down and entry is not allowed.

(2) Exception situation 2: If another 10 minutes pass (at AM 01:15) and both guardsmen are still not there to sign in, the second level of the exception handling policy will be applied. The guards' superiors will be notified to check on the shift and deal with this abnormal situation. If the guards' superiors determine that this is just a simple case of carelessness, they can sign into the system and cancel the abnormal situation, send new guards or just unlock the area to allow the guards on the scene to entering for checking.

(3) Exception situation 3: If another 15 minutes passes after the application of the second exception handle policy at AM 01:30, and the state of the treasury is still not cleared, then the area is locked down and can only be opened by the president of the bank. The president can cancel this lock-down through internet with proper authentication.

(4) Exception situation 4: If another 30 minutes passes after the application of the second exception handling policy at AM 02:00 (the default operation destruction time),

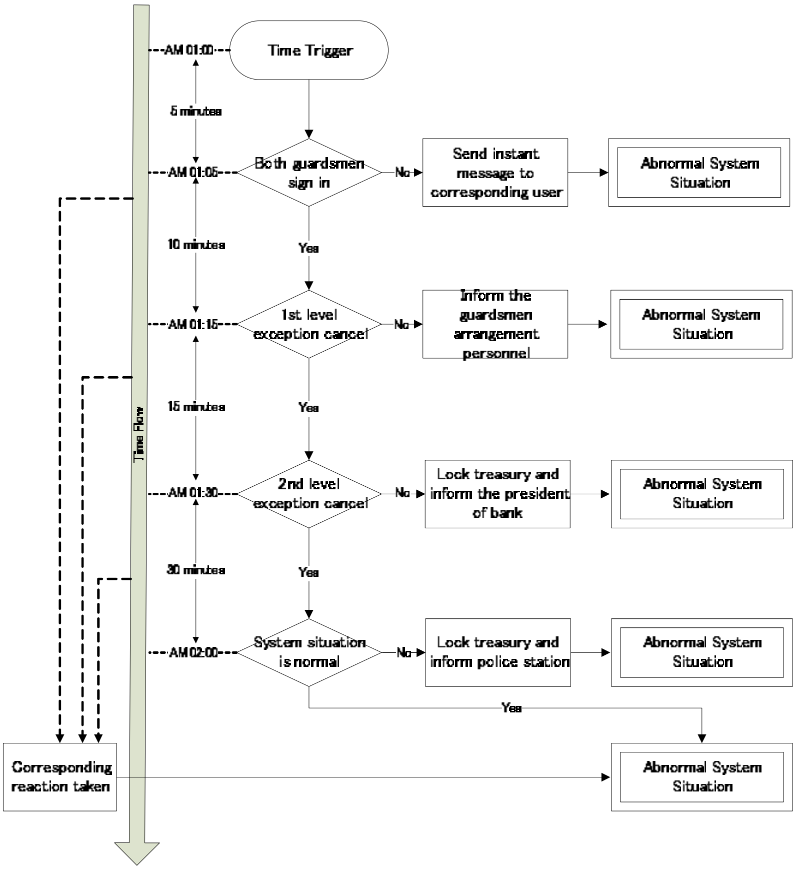

FIGURE 5. Flow-chart of the example scenario

and nobody has come to deal with the problem to remedy the abnormal situation, then the treasury is locked-down and can only be opened by the president of the bank during daytime business hours. Simultaneously, a message is sent to inform the police to come to check the actual situation.

A bank scenario is used in the example above because the services need to be highly secure and illustrate a critical operation to explain how our time-dependent multi-layer exception handling policy works. Not every operation needs to be triggered by this type of time driven event, however, some important critical operations need to be considered when applying this model to implement the necessary security for the whole system. The flow-chart for this exception handling example scenario is illustrated in Figure 5.

6. **Conclusions.** In the real world, even though computer technology is already so highly developed [16-44], there is still some need for artificial operations. In this paper, we describe an extended context model for service driven by the time factor to deal with critical operations. According to the time trigger mean and corresponding multi-layer exception handling policy (sub-divided by time), we can ensure that some missions with specific importance can be handled correctly and according to the level of emergency to eventually maintain the integration and high security of whole system. However, this model does limit the degree of freedom, therefore, we suggest that only some operations (those with high necessity of security) should be set as critical operations. General operations can still be operated by the RBAC model. This would help to maintain the flexibility of the whole system. The implementation of this model would be valuable in some industries where high-security is necessary.

Acknowledgment. The authors are appreciative of the financial support in the form of research grants to Dr. Chen-Yuan Chen from the National Science Council, Taiwan under Grant Nos. NSC 98-2221-E-153-004 and NSC 99-2628-E-153-001. The authors are also most grateful for the kind assistance of Editor of IJICIC, and for the constructive suggestions from the anonymous reviewers, all of which has led to the making of several corrections and have greatly aided us to improve the presentation of this paper.

REFERENCES

[1] E. Bagheri, R. Zafarani and M. Barouni-Ebrahimi, Can reputation migrate? On the propagation of reputation in multi-context communities, *Knowledge-based Systems*, vol.22, no.6, pp.410-420, 2009.
[2] A. Dey, *The Context Toolkit*, http://www.cs.berkeley.edu/˜dey/context.html, 2004.
[3] B. Schilit, N. Adams and R. Want, Context-aware computing applications, *Proc. of the Workshop on Mobile Computing System and Applications*, pp.85-90, 1994.
[4] H. Packard, *Cooltown Project*, http://www.cooltown.com/cooltown/index.asp, 2004.
[5] M. Khedr and A. Karmouck, Negotiating context information in context-aware systems, *IEEE Intelligent Systems*, vol.19, no.6, pp.21-29, 2004.
[6] M. Khedr, A semantic-based, context-aware approach for service-oriented infrastructures, *Proc. of the 2nd IFIP International Conference on Wireless and Optical Communications Networks*, pp.584-588, 2005.
[7] M. Román, C. Hess, R. Cerqueira et al., A middleware infrastructure to enable active spaces, *IEEE Pervasive Computing*, vol.1, no.4, pp.74-83, 2002.
[8] S. J. H. Yang, B. C. D. Wu and N. W. Y. Shao, A context-based adaptation rule design for mobile commerce, *Proc. of the 1st International Conference on Information Management and Business*, 2005.
[9] S. J. H. Yang, I. Chen and N. Shao, Ontological enabled annotations and knowledge management for collaborative learning in virtual learning community, *Journal of Educational Technology and Society*, vol.7, no.4, pp.70-81, 2004.
[10] S. J. H. Yang, J. J. P. Tasi and C. C. Chen, Fuzzy rule based systems verification using high level petri nets, *IEEE Trans. on Knowledge and Data Engineering*, vol.15, no.2, pp-457-573, 2003.
[11] S. K. Mostefaoui and B. Hirsbrunner, Context aware service provisioning, *Proc. of the IEEE International Conference on Pervasive Services*, pp.71-80, 2004.
[12] S. K. Mostefaoui, A. T. Bouzid and B. Hirsbrunner, Using context information for service discovery and composition, *Proc. of the 5th Conference on Information Integration and Web-based Applications and Services*, pp.129-138, 2003.
[13] T. Broens, S. Pokraev, M. van Sinderen et al., Context-aware, ontology-based service discovery, *Proc. of the 2nd European Symposium on Ambient Intelligence, LNCS*, vol.3295, pp.72-83, 2004.
[14] T. Lemlouma and N. Layada, The negotiation of multimedia content services in heterogeneous environments, *Proc. of the 8th International Conference on Multimedia Modeling*, pp.187-206, 2001.
[15] M. J. Moyer and M. Abamad, Generalized role-based access control, *Proc. of the 21st International Conference on Distributed Computing System*, pp.391-398, 2001.
[16] T. Wang, Y. Chen and S. Tong, Fuzzy improved interpolative reasoning methods for the sparse fuzzy rule, *ICIC Express Letters*, vol.3, no.3(A), pp.313-318, 2009.
[17] T. Shang, S. Y. Wang, J. W. Liu and X. B. Wan, Knowledge radius: A selective knowledge use strategy for distance-type fuzzy reasoning method, *ICIC Express Letters*, vol.3, no.3(A), pp.331-336, 2009.
[18] M. Shimakawa, A proposal of extension fuzzy reasoning method, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2603-2616, 2008.
[19] T. Wang, Y. Chen and S. Tong, Fuzzy reasoning models and algorithms on type-2 fuzzy sets, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2451-2460, 2008.
[20] C. W. Chen, W. L. Chiang and F. H. Hsiao, Stability analysis of T-S fuzzy models for nonlinear multiple time-delay interconnected systems, *Mathematics and Computers in Simulation*, vol.66, no.6, pp.523-537, 2004.
[21] C. Y. Chen, R. C. Hsu and C. W. Chen, Fuzzy logic derivation of neural network models with time delays in subsystems, *International Journal on Artificial Intelligence Tools*, vol.14, no.6, pp.967-974, 2005.

[22] C. W. Chen, W. L. Chiang, C. H. Tsai et al., Fuzzy Lyapunov method for stability conditions of nonlinear systems, *International Journal on Artificial Intelligence Tools*, vol.15, no.2, pp.163-171, 2006.

[23] C. W. Chen, Stability conditions of fuzzy systems and its application to structural and mechanical systems, *Advances in Engineering Software*, vol.37, no.9, pp.624-629, 2006.

[24] C. Y. Chen, J. R. C. Hsu, C. W. Chen and M. H. Cheng, Numerical model of internal solitary wave evolution on impermeable variable seabed in a stratified two-layer fluid system, *China Ocean Engineering*, vol.20, no.2, pp.303-313, 2006.

[25] T. Y. Hsieh, M. H. L. Wang, C. W. Chen, C. Y. Chen, S. E. Yu, H. C. Yang and T. H. Chen, A new viewpoint of S-curve regression model and its application to construction management, *International Journal on Artificial Intelligence Tools*, vol.15, no.2, pp.131-142, 2006.

[26] C. W. Chen, K. Yeh, W. L. Chiang, C. Y. Chen and D. J. Wu, Modeling, H^∞ control and stability analysis for structural systems using Takagi-Sugeno fuzzy model, *International Journal of Innovative Computing, Information and Control*, vol.1, no.4, pp.635-642, 2005.

[27] C. W. Chen, C. Y. Chen, H. C. Yang and T. H. Chen, Analysis of experimental data on internal waves with statistical method, *Engineering Computation*, vol.24, no.2, pp.116-150, 2007.

[28] C. W. Chen, C. L. Lin, C. H. Tsai, C. Y. Chen and K. Yeh, A novel delay-dependent criteria for time-delay T-S fuzzy systems using fuzzy Lyapunov method, *International Journal on Artificial Intelligence Tools*, vol.16, no.3, pp.545-552, 2007.

[29] C. Y. Chen, J. R. C. Hsu, C. W. Chen et al., Generation of internal solitary wave by gravity collapse, *Journal of Marine Science and Technology*, vol.15, no.1, pp.1-7, 2007.

[30] C. Y. Chen, J. R. C. Hsu, C. W. Chen et al., Wave propagation at the interface of a two-layer fluid system in the laboratory, *Journal of Marine Science and Technology*, vol.15, no.1, pp.8-16, 2007.

[31] C. Y. Chen, C. W. Shen, C. W. Chen, K. F. R. Liu and M. J. Jeng, A stability criterion for time-delay tension leg platform systems subjected to external force, *China Ocean Engineering*, vol.23, pp.49-57, 2009.

[32] C. W. Chen, Modeling and control for nonlinear structural systems via a NN-based approach, *Expert Systems with Applications*, vol.36, pp.4765-4772, 2009.

[33] C. W. Chen, The stability of an oceanic structure with T-S fuzzy models, *Mathematics and Computers in Simulation*, vol.80, pp.402-426, 2009.

[34] P. C. Chen, C. W. Chen and W. L. Chiang, GA-based modified adaptive fuzzy sliding mode controller for nonlinear systems, *Expert Systems with Applications*, vol.36, pp.5872-5879, 2009.

[35] C. Y. Chen, J. W. Lin, W. I. Lee and C. W. Chen, Fuzzy control for an oceanic structure: A case study in time-delay TLP system, *Journal of Vibration and Control*, vol.16, no.1, pp.147-160, 2010.

[36] C. Y. Chen, H. C. Yang, C. W. Chen and T. H. Chen, Diagnosing and revising logistic regression models: Effect on internal solitary wave propagation, *Engineering Computations*, vol.25, no.2, pp.121-139, 2008.

[37] C. W. Chen, H. C. Yang, C. Y. Chen et al., Evaluation of inference adequacy in cumulative logistic regression models: An empirical validation of ISW-ridge relationships, *China Ocean Engineering*, vol.22, no.1, pp.43-56, 2008.

[38] K. Yeh, C. Y. Chen and C. W. Chen, Robustness design of time-delay fuzzy systems using fuzzy Lyapunov method, *Applied Mathematics and Computation*, vol.205, no.2, pp.568-577, 2008.

[39] C. H. Tsai and C. W. Chen, An earthquake disaster management mechanism based on risk assessment information for the tourism industry-a case study from the island of Taiwan, *Tourism Management*, vol.31, no.4, pp.470-481, 2010.

[40] C. W. Chen, M. H. L. Wang and J. W. Lin, Managing target the cash balance in construction firms using a fuzzy regression approach, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol.17, no.5, pp.667-684, 2009.

[41] C. W. Chen, K. Yeh and F. R. Liu, Adaptive fuzzy sliding mode control for seismically excited bridges with lead rubber bearing isolation, *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol.17, no.5, pp.705-727, 2009.

[42] M. L. Lin, C. W. Chen, Q. B. Wang et al., Fuzzy model-based assessment and monitoring of desertification using MODIS satellite imagery, *Engineering Computations*, vol.26, no.7, pp.745-760, 2009.

[43] T. H. Chen and C. W. Chen, Application of data mining to the spatial heterogeneity of foreclosed mortgages, *Expert Systems with Applications*, vol.37, no.2, pp.993-997, 2010.

[44] P. C. Chen, C. W. Chen and W. L. Chiangh, GA-based fuzzy sliding mode controller for nonlinear systems, *Mathematical Problems in Engineering*, vol.2008, 2008.