Advanced Software Protection Architecture by Using Systematic Innovation Method

Song-Kyoo Kim, *Senior Member, IACSIT*

***Abstract*—This paper is dealing with improving the software reliability by using the systematic innovation method that is recently invented. The mathematical modeling shows the theoretical software protection scheme in the security perspective. If software application modules are represented as backups under proposed architecture, the system can be solved by using the stochastic maintenance models with replacement policies. This practical approach of technology enhancement in software engineering is demonstrated in the framework of optimized software allocation problems with unreliable backups.**

***Index Terms*—Systematic innovation, security system, stochastic architecture design, duality principle, TRIZ, TIPS.**

I. Introduction

In light of the recent acts of cyberterrorism, it becomes imperative not only to provide a software security, but to offer a paradigm of a "reliable system" which can be applied to software applications for the business continuity such as operating systems, databases and other computer applications. Availability [11]-[12] is one of major subjects of security [19]. Availability of software modules such as database and their applications is one of critical issues in software engineering because protecting the information is so important. A natural assumption is that the entire system functions stochastically, i.e., it is subject to attacks at random times, the recovery of individual module is random, and even the information about attacks is limited and observed at random epochs of time.

Systematic innovation [18] is a structured process and set of practical tools anyone can use to create (or improve) products, process or services that deliver new value to customers. It is also a set of continuous evolving tools that will improve ability to solve the problems. TRIZ is the most powerful methods for systematic innovation methodologies. The substance-field model [9], [16] and 76 Inventive Standard [4]-[5], [16] were conceptualized by the founding father of TRIZ, Genrich Altshuller [1], [2]. Even though, 76 Inventive Standards do not provide graphic models for every standard and the standards are not new to the TRIZ community, they can help the TRIZ specialist find solutions concepts for many kinds of problems as a collection of methods to identify [5]. The Standard Solutions are grouped by constraints, so they can help the specialists find appropriate solution concepts [16].

They are more accessible to TRIZ newcomers than ARIZ [8], [20] because the user is liberated from the ARIZ dictum of mastering every step before using any step. The 76 Inventive Standard Solutions are among the fundamental techniques that provide the foundation for most of commercial major TRIZ softwares but they are not currently being used widely [5]. There are several reasons why the Inventive Standards are not applied widely and two main reasons need be addressed. First, people learning TRIZ still must do a lot of case studies that illustrate the principles of TRIZ using terms and technologies before using Inventive Standard correctly. Second, the standards are categorized by physical interactions. The Inventive Standards (76 Standard Solutions) are well defined and organized [4]. But it is still difficult to learn and complicated even for TRIZ specialists. More importantly, the 76 Inventive Standards are not intuitive [16].

Manuscript received July 17, 2012; revised September 6, 2012.

Song-Kyoo Kim is with Asian Institute of Management, Paseo de Roxas, Makati City 1260, Philippines (e-mail: skim@aim.edu)

II. Concept Design by Using Systematic Method

Innovative notation schema is classified the Inventive Standards more simple way and users can be guided to the candidate solutions from the problems based on Su-Field model with the minimal knowledge of 76 Inventive Standard solutions. The notation for Su-Field model (Su-Field notation) is applied (aka. Amang's notation) [14].

The Su-Field model for Inventive standard solution can exhibits the summarized main characteristics of a Su-Field model [14].

$$(a/b/c):(d/e/f) \tag{1}$$

where the symbols *a*, *b*, *c*, *d*, *e* and *f* stand for basic elements of the model as follows:

a = arrivals distribution,
b = service time distribution,
c = number of servers (*c*=1, 2, 3, …)
d = service properties (i.e., FCFS, LCFS, SIRO)
e = capacity of the system
(a waiting room and servers)
f = population of input resources.

The attributes of the substance *s* are as follow:

S^* = general terms of the substance that can solve the problems

S^+ = +1 substance from basic structure to solve the problems

S' = modify the substance (tool) to solve the problems without changing the number of components from basic structure

DOI: 10.7763/IJET.2012.V4.449

S^- = -1 substance from basic structure (i.e., tool is missed)

S^{∞} = substance (tool) is divided infinitely (Technical System Evolution)

S'' or S^2 = adding the clone of the substance (+1)

The attributes of the field *f* are similar with substance attributes:

F^* = general terms of the field that can solve the problems

F^+ = +1 field from basic structure to solve the problems

F' = modify the field to solve the problems without changing the number of components from basic structure

F^- = -1 field from basic structure

F^{∞} = field is divided infinitely (Technical System Evolution)

F'' = adding the clone of the field (+1)

$\overleftarrow{F}$ = reverse direction of the field

The attributes for fields and substances indicate how to modify the substances and the fields.

A. Problem Clarifications

The problem for enhancing the reliability of the software architecture with minimized complicity can be described as Su-Field models. Object (S1) is applications on the top of the platforms and Tool (S2) is the platform by itself. Based on Su-Field Model, Problem Type-2 [14] as Su-Field Notation is the problem that contains the harmful action and the candidate solution is basically removing the harmful function:

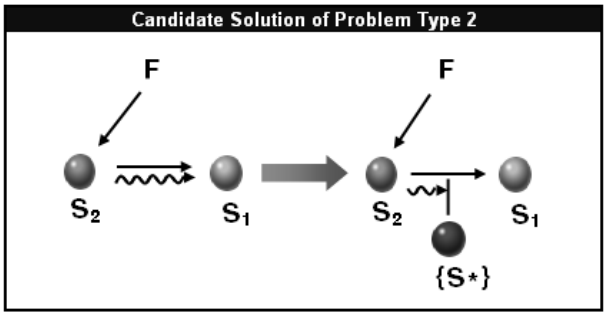

Fig. 1. Su-Field Solution Diagram of 2/S*/F

From Fig. 1, the candidate solution of Problem Type 2 can be determined as follow:

$$2/S/F\{/0\} \rightarrow \begin{cases} 2/S^*/F, & S^* = S^+ \text{ or } S' \\ 2/S/F,^+ & \\ 2/S/F/a, & 0 < a < 1 \end{cases} \quad (2)$$

Even the formula (4.4) gives the concept solution by adding a new substance (S^*=S^+) or modifying the substance (S^*=S^+), not limited. The candidate attribute of substance for Type-2 Solution can be:

$$S^*=\{S^*: S', S^+, S^2, S, S^n\}$$

According to (2) and (3), the concept solution for enhanced software architecture is

$$2/S/F\{/0\} \rightarrow 2/S^{\infty}/F$$

where S^{∞} indicates that the substance is evolving based on technical system evolution.

III. Multi-Layered Software Architecture

Enhanced software architecture is designed based on the concept solution. S^{∞} in the concept solution indicates that the substance can be modified based on the technical system evolution. The platform S2 is divided as two layers and one of layers (common module) is adapted with application modules for recovery after application crash. The layered architecture consists there layers: core module, common module and application module. A core module directly communicates the hardware and locates the low-level. The core module should be reliable and we assume that core module is fault free. The common modules are laid on the core module and interworking between the core and the application modules. It is a medium layer module that can improve the compatibility of applications. Application modules are general application programs for users such as MS-Word, IE and so on. Fig. 2 shows the multi-layered software architecture. Common modules are unreliable but need to be protected. This paper deals with common modules for protection.

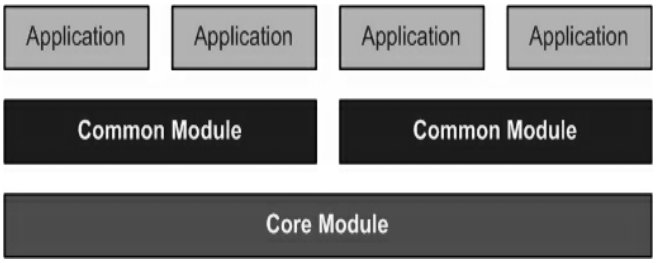

Fig. 2. Multi-layered Software Architecture

There are four assumptions for the further process.

1) Core module is reliable and the fault-free.
2) The operation of the common modules.
3) Reliability of application modules is depended by reliability of common modules
4) Common modules as machines (main, super-reserve).

The analytic solution gives the mathematical guidelines for simulated solution and the actual system.

IV. Stochastic Software Protection Analysis

As known, closed queueing model also called *Repairman Problem* [6-7] and this model can be categorized by repairman's problem with super-reserve backups [11-12]. If software application modules are represented as "machines" under proposed architecture, the protection system can be solved by closed queueing analysis. To approach from practical application to mathematical model *m+1* main machines are represent common modules and backup

modules that are copied for common module backup are represented by *S* super-reserve machines that are used during the idle periods.

Instead of starting the proposed model which is called model 1, we will refer as to Model 2 and Model 3. Model 2 is similar to Model 1, except that it does not have the backup facility and idle periods. We rather associate it with repairman's vacations, which are distributed as regular repairs. However, upon his return, the repairman brings a brand new machine (the machine in Model 2 represents to the common module in Model 1), which replaces any one that breaks down during his vacation trip if any such available. Otherwise, the new machine he brings in substitutes any other machine and in both cases the old machine is disposed. Model 2 is directly connected with yet another model, which we will call Model 3. Model 3 is a regular multi-channel queueing system, in notation, *$(GI_0$ /GI)/M/m/0* [7]. Duality principle [6] is applied to analyze Model 1 by using the results of Model 2 and Model 3.

Denote by Z_t^1 the total number of intact main machines at time *t* in Model 1. If a repairman fixes all machines completely, the total number of main machines is restored to *m+1*. Then he goes on vacation until *S+1* machines break down, after which the repairman resumes his work. In other words, a busy period begins. As we already mentioned, backup machines are replaced when main machines fail during repairman's idle period. Denote by S the random number of external backups that are used during idle periods. Since total *N* super-reserve (common) modules are also unreliable with the availability *p*, the PMF (Probability Mass Function) of S (the available number of super-reserve modules for backup) is

$$S(n)=\binom{N}{n}p^n(1-p)^{N-n} \tag{5}$$

with the mean $r=\mathrm{E}[S]=Np$. Let $\tau_0(=0),\tau_1,\ldots$ be the successive moments of repair completions. The random variable $\boldsymbol{\tau_{n+1}-\tau_n}$ is supposed to have a PDF (probability distribution function)

$$A(x)=P\{\tau_{n+1}-\tau_n\},x\geq 0, \tag{6}$$

with the mean $\boldsymbol{a}=\mathrm{E}[\boldsymbol{\tau_{n+1}-\tau_n}\}<\infty$. If upon the service completion, the total number of intact machine is greater than *m*, the PMF of next service completion period is $A_0(x)$ with the mean $a_0=\int_{R+}xA_0(dx)$. Each of the main machines breaks down independently of each other and of repairs, and according to the exponential distribution with parameter $0<\mu<\infty$.

In Model 2, there is a maximum of main served by a single repairman. The total number of main machines at time t s denoted by Z_t. Unlike Model 1, there are no idle periods even when all main machines become intact. Let T_n be the n-th repair completion. If the line of defective machines is nonempty, the repairman continues to repair a next machine immediately. When the cumulative number of intact machines becomes m at T_n. the repairman leaves the system and returns at T_{n+1} with a brand new machine. We assume that his vacation is distributed as his regular service time. The new machine replaces a defective one, if by then available, or otherwise-refund the new machine. We suppose that the last action does not affect the status of the system in this particular problem setting. The random variable T_{n+1}-T_n is stochastically equivalent to $\tau_{n+1}-\tau_n$ of Model 1. The assumption about the failures distribution is the same as of Model 1 (i.e., exponential distribution with parameter μ). Let

$$\pi_k^1=\lim_{t\to\infty}P^x\{Z_t^1=k\},$$

and

$$\varphi_{\mathrm{k}}^1(\mathrm{n})=\lim_{\mathrm{t}\to\infty}\mathrm{P}^{\mathrm{x}}\{\mathrm{Z}_{\mathrm{t}}^1=\mathrm{k}\big|\mathrm{S}=\mathrm{n}\},\mathrm{k}=0,1,\ldots,\mathrm{m}$$

be the limiting probabilities and conditional probabilities of the process Z_t^1. These probabilities exist, under the same conditions as those for the embedded process. Then from duality principle [6], it follows that

$$\varphi_{\mathrm{m+1}}^1(\mathrm{n})=\lim_{\mathrm{t}\to\infty}\mathrm{P}^{\mathrm{x}}\{\mathrm{Z}_{\mathrm{t}}^1=\mathrm{m}+1\big|\mathrm{S}=\mathrm{n}\}=\frac{\mathrm{P_m}(n+1)}{m\mu(\bar{a}+P_m a_0 n)+P_m(n+1)}$$

and

$$\varphi_{\mathrm{k}}^1(\mathrm{n})=(1-\varphi_{m+1}^1(\mathrm{n}))\pi_{\mathrm{k}},\mathrm{k}=0,1,\ldots,\mathrm{m}$$

where P_k and π_k are subjects to be dealt in the related research [12]. Because of conditional expectations, we have

$$\pi_{\mathrm{m+1}}^1=\mathrm{E}\left[\frac{\mathrm{P_m}(n+1)}{m\mu(\bar{a}+P_m a_0 n)+P_m(n+1)}\right] \tag{7}$$

and

$$\pi_{\mathrm{k}}^1=\mathrm{E}\left[(1-\varphi_{m+1}^1(\mathrm{n}))\right]\pi_{\mathrm{k}},\mathrm{k}=0,1,\ldots,\mathrm{m} \tag{8}$$

Model 3, as mentioned, is the conventional *$(GI_0$ /GI)/M/m/0* (multi-channel) queue. Recall that such a system is characterized by the general-independent input (i.e. a renewal process), *m* parallel channels without any buffer A customer enters a free channel available with his service demand distributed exponentially with parameter Inter-renewal times are distributed in accordance with the PMF *A(x)* and *A_0(x)*. Model 2, as we see it, is congruent to Model 3, while Model 1 is dual with Model 2, so also with Model 3. The latter is a classical system investigated by Takacs [13].

The stationary probabilities $P=(P_0,P_1,\ldots,P_m)$ for the embedded process are known to satisfy the following formulas:

$$P_k = \begin{cases} \sum_{r=k}^{m} B_r \binom{r}{k} (-1)^{r-k}, k = 0, \ldots, m-1, \\ \left[a_0 \sum_{r=0}^{m} \frac{\binom{m}{r}}{a_r} \right]^{-1}, k = m, \end{cases}$$

where

$$B_n = \left(a_n \sum_{r=n}^{m} \frac{{}_m C_r}{a_r} \right) \Big/ \left(a_0 \sum_{r=0}^{m} \frac{{}_m C_r}{a_r} \right),$$

$$a_r = \begin{cases} 1, r = 0 \\ \prod_{i=0}^{r} \frac{a_i}{1-a_i}, r \geq 1, \end{cases}$$

$$a(\theta) = \int_0^\infty e^{-\theta u} A(du),$$

$$a^0(\theta) = \int_0^\infty e^{-\theta u} A_0(du),$$

$$a_r = a(r\mu), a_r^0 = a^0(r\mu).$$

Now, the continuous time parameter queueing process of Model 3 is considered. By using the Kolmogorov differential equations and the semi-regenerative techniques, the limiting distribution $\pi = \{\pi_0, \pi_1, \ldots, \pi_{m+1}\}$ is :

$$\begin{cases} \pi_0 = 1 - \sum_{n=1}^{m} \pi_n & , k = 0, \\ \pi_k = \frac{P_{k-1}}{k\mu\bar{a}} & , k = 1, \ldots, m. \end{cases} \quad (9)$$

For the process $\boldsymbol{Z}_t^1$ the corresponding formulas yield (from the previous section)

$$\pi_{m+1}^1 = \mathbb{E}\left[\frac{P_m(S+1)}{m\mu(\bar{a} + P_m a_0 S) + P_m(S+1)} \right] \quad (10)$$

and

$$\pi_k^1 = \mathbb{E}\left[(1 - \varphi_{m+1}^1(S)) \right] \pi_k \; k = 0, 1, \ldots, m. \quad (11)$$

V. Optimality and Implementation of Software Module Protection

In this section we will deal with a class of optimization problems that arise in reliability. Let us formalize a pertinent optimization problem. Let a strategy, says Σ, specify, ahead of the time, a set of acts we impose on the system, such as the choice of repair distribution, the number of main and external backups, statistics of failure rates dependent on the number of backup machines that enables us to spend more or less time on the maintenance and so on. On the other hand, a system can be subject to a set, say C , of cost functions. Denote by $\phi(\Sigma, C, t)$ the expected costs within, due to the strategy Σ costs C and define the expected cumulative cost rate over an infinite horizon:

$$\phi(\Sigma, C, t) = \lim_{t \to \infty} \frac{1}{t} \phi(\Sigma, C, t)$$

Now we turn to convergence theorems for semi-regenerative, semi-Markov, and Markov renewal processes [3],

$$\text{(i)} \lim_{t \to \infty} \frac{1}{t} R^i(t) = \frac{1}{PM}$$

$$\text{(ii)} \lim_{t \to \infty} \frac{1}{t} \int_0^t P^i\{Z_s^1 = k\} ds = \pi_k^1$$

$$\text{(iii)} \lim_{t \to \infty} \frac{1}{t} \int_0^t P^i\{Y_u^1 = k\} du = \frac{P_k M_k}{PM}$$

to arrive at the objective function $\phi(\Sigma, C, t)$, which gives the total expected rate of all processes over an infinite horizon. We arrive at the following expression for the sample objective function [7]:

$$\phi(\Sigma, C) = c \frac{P_m M_m}{PM} + r \frac{1}{PM} + \sum_{k=0}^{m+1} \left[kh + (m+1)l \right] \pi_k^1. \quad (12)$$

where *h, l* are relevant (cost) constant coefficients. We restrict the initial strategy of this model to one, which includes only the control level of backup modules (super-reserve machines).

$$\phi(\Sigma(n), C) = \min\{\phi(\Sigma(n), C) : N = 1, 2, \ldots\}. \quad (13)$$

As an demonstration, we take *c=2, h=1* and *r=4*. Repair time distribution is exponential with mean *a=1.2* and the parameter μ=0.2. Take the total number of operating common modules as 7. Now, we calculate $\phi(\Sigma(N), C)$ and N_0 that gives a minimum for $\phi(\Sigma(N), C)$. In other words, the control level N_0 stands for the available number of backup modules (super-reserve machines) which minimizes the total cost of this system. The calculation yields that $N_0=5$ for which the minimal cost equals 9.470. It means that we allocate operating resources to 7 operating *(m+1)* common modules and obtain the decision value $N_0=5$ which is the number of external backup modules under availability *p(=0.2)* for common module protection.

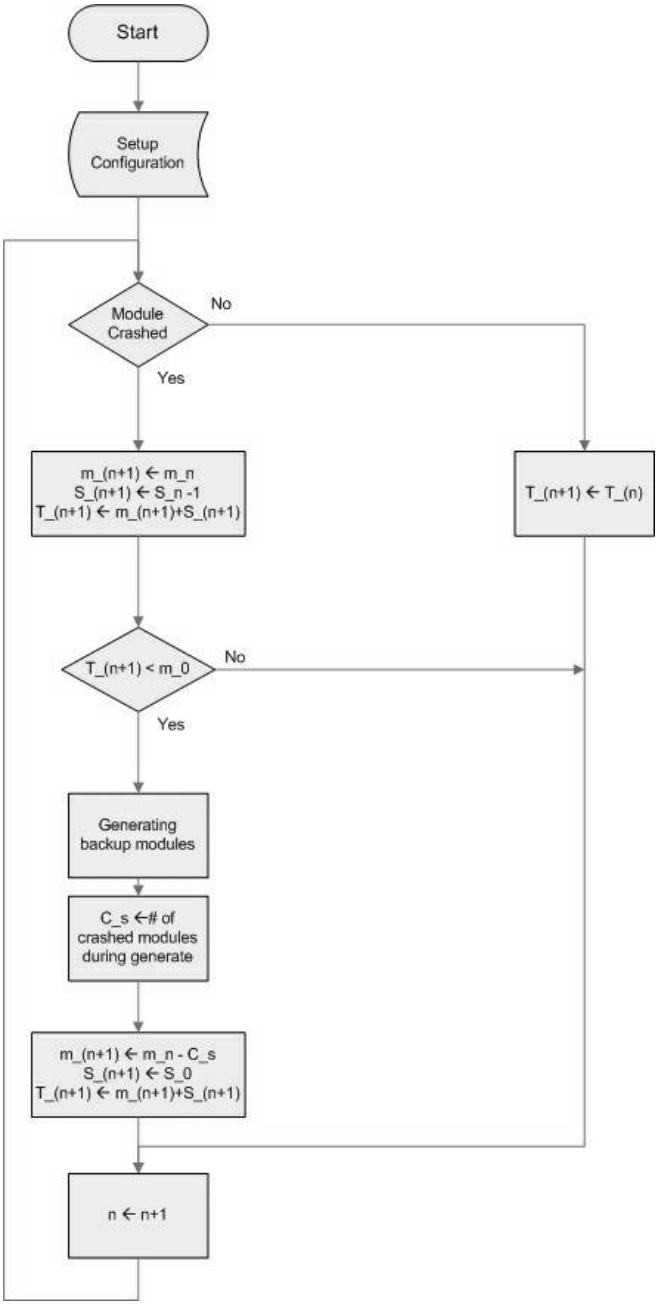

Fig. 3. Implementation guidelines

The system can be configured based on the configuration values from the mathematical modeling in the previous session. Basically, the moment of generating the common modules as a group is the time when the number of intact modules is smaller than initial modules. During the generating periods, the crashed modules are also recovered. The initial configurations are as follow (See Fig. 3):

Initial Conditions:

. m_0 : Initial number of running common module

. S_0 : Optimal number of backup modules

. T_0 m_0 + S_0

. n 0 : Iteration value

Values:

. m_n : Number of running common modules at n-th step

. S_n : Number of backup common modules at n-th step

. T_n : Total number of intact common modules at n-th step

. C_s : Number of crashed modules during backup generation

. G_s : Number of generating running modules during backup generations

VI. Conclusion

The new approach of systematic innovation method is applied for solving the problem in software architecture. Su-Field Notation provides a user to be guided even with minimal knowledge of Theory of Inventive Problem Solving (TRIZ) method. Even though systematic innovation method included in TRIZ is focused on concept design, the pattern of this approach can be also applied to other industries. The paper shows the analytical approach of the software protection method to improve the availability by using the closed queueing system with flexible conditions and enhanced model of the previous research. This approaches theoretical and designed for software architecture, but feasible to apply real-world applications such as networked server allocation [12], VoIP unit protection [11] and many other applications. Implementation of the model and comparison between the model and the actual data will be the further direction of this research.

References

[1] G. Altshuller, *And Suddenly the Inventor Appeared: TRIZ, the Theory of Inventive Problem Solving*, Technical Innovation Center, Worecester, MA (1996).

[2] G. Altshuller, *40 Principles, Technical Innovation Center*, Worcester, MA. 1997.

[3] E. Cinlar, *Introduction to Stochastic Processes*, Prentice Hall, Englewood Cliffs, N.J. 1975.

[4] E. Domb, "The Seventy-Six Standard Solutions: How They Relate to the 40 Principles of Inventive Problem Solving," *TRIZ Journal*, May 1999.

[5] E. Domb, "Using the 76 Standard Solutions: A case study for improving the world food supply," *TRIZ Journal*, April 2003.

[6] J. H. Dshalalow, "On a duality principle in processes of servicing machines with double control," *J. of Appl. Math. & Sim.* vol. 1, no. 3, pp. 245-251, 1988.

[7] J. H. Dshalalow, "On single-server closed queues with priorities and state dependent parameters," *Queueing Systems* vol. 8, pp. 237-254, 1991.

[8] Grace, Frank, *et al.*, "A New TRIZ Practitioner's Experience for Solving an Industrial Problem using ARIZ 85C," *TRIZ Journal*, January 2001.

[9] L. Haijun, "Substance-field Models for Fourth Class Standards," *TRIZ Journal*, February 2009.

[10] X. Mao, *et. al*., "Generalized Solutions for Su-Field Analysis," *The TRIZ Journal*, August 2007.

[11] S. -K. Kim, "Stochastic optimization method of the low-bandwidth VoP unit redundancy by using the closed queueing model," Korea Patent 0501321, 2005.

[12] S.-K. Kim, "Enhanced Management Method of Storage Area Network (SAN) Server with Random Remote Backups," *Mathematical and Computer Modelling*, vol. 42, pp 947-958, 2005.

[13] S.-K. Kim, "Enhanced User Experience Design based on User Behavior Data by Using Theory of Inventive Problem Solving," *IEEE Proceedings of IEEM*, pp 2076-2079, 2010.

[14] S.-K. Kim, "Innovative Design of Substance-Field Notations for Reformulating the Seventy-six Standard Solutions in TRIZ," *International Journal of Systematic Innovation*, 2011, pp. 19-26.

[15] R. Orfali and D. Harkey, *Client and Server Programming with JAVA and CORBA* 2nd Ed., John Wiley & Sons Inc., New York, NY.1998.

[16] P. Soderlin, "Thoughts on Su-Field Models and 76 Standards: Do we need all the standards?," *TRIZ Journal*, March 2003.

[17] L. Takacs, *Introduction to the Theory of Queues*, Oxford University Press, New York, NY 1962.

[18] J. Terninko, A. Zusman, *et. al., Systematic Innovation: An Introduction to Theory of Inventing Problem.*

[19] D. Russell and G. T. Ganemi, *Computer Security Basics*, O' Reilly and Asso. Inc., Sebastopol, California, 1991
[20] B. Zlotin and A. Zusman. (March 1999). ARIZ on the Move. *TRIZ Journal*. [Online]. Available: http://www.triz-journal.com

Song-Kyoo (Amang) Kim is an Asian Institute of Management faculty member as the Associate Professor. He had been a technical manager and TRIZ specialist of mobile communication division at Samsung Electronics. He is involved in IT industries more than 10 years. Dr Kim has received his master degree of computer engineering on 1999 and Ph.D. of Operations Research on 2002 from Florida Institute of Technology. He is the author of more than 20 operations research papers focused on stochastic modeling, systematic innovations and patents. He had been the project leader of several 6 Sigma and TRIZ projects mainly focused on the mobile industry.