METHOD FOR VISUALIZING RISK FACTORS OF SYSTEM FAILURES AND ITS APPLICATION TO ICT SYSTEMS

Takafumi Nakamura[1]*, Kyoich Kijima[2]

[1]Fujitsu FSAS Inc., Support Administration Group, Hamamatsu-Cho Support Center, 5-1, Hamamatsu-Cho 1-Chome, Minato-ku, Tokyo, 105-0013, JAPAN,
nakamura.takafu@jp.fujitsu.com

[2]Tokyo Institute of Technology, Graduate School of Decision Science and Technology,2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, JAPAN,
kijima@valdes.titech.ac.jp

*Correspondence to: Takafumi Nakamura

ABSTRACT

This paper proposes a method for visualizing risk factors of system failures. This method enables us to visualize risk factors and monitor them over time and compare them among systems. This is valuable for promoting system safety and reliability. First we introduce a methodology of holistically defining system failure then introduce our method for quantifying the risk factors of system failures with an interaction and coupling (IC) chart using normal accident theory (Perrow, 1999). Defining system failure is done using system of system failure (SOSF) (Nakamura and Kijima, 2007, 2008b, 2009a) with a meta-system frame called system of system methodology (SOSM) (Jackson, 2003, 2006). System of system failure enables us to understand system failure holistically. The IC chart is used to classify object systems – nuclear power plants, chemical plants, aircraft and air traffic control, ships, dams, nuclear weapons, space missions, and genetic engineering- using interaction (i.e. linear and complex) and coupling (i.e. tight and loose) between the components that constitute such systems. The IC chart (Perrow, 1999) is limited by the subjectivity in classifying target systems. We propose a method for quantitatively measuring risk factors (i.e. objective) from incidents that have occurred over time to complement the current IC chart shortcoming (i.e. subjective). This enables us to understand system features and the effectiveness of countermeasures quantitatively introduced to object systems. There have been several findings with this methodology. It enables us to quantify the risk factors in terms of the IC chart. Stock exchange, meteorological, and healthcare systems are located sequentially from linear to complex interaction and tight to loose coupling. Intel Architecture (IA) servers' quality control

measures (i.e. educating engineers for becoming hybrid engineers and altering server design goal) cause a shift in the linear interaction and tight coupling directions with less incident rates. Healthcare systems are migrating toward the complex interaction and loose coupling direction with deteriorating system quality. The Electronic Data Interchange (EDI) policy in the healthcare sector is one of the reasons for this migration. Application examples in information and communication technologies (ICT) engineering demonstrated that using the proposed method to quantitatively monitor risk factors will help improve safety and quality of various object systems.

Keywords: risk management; system failure model; normal accident theory (NAT); Interaction and Coupling Chart (IC chart); Information and Communication Technology (ICT)

1. INTRODUCTION

"The horror of that moment" the King went on,
"I shall never, never forget."
"You will, though," the Queen said, "if you don't make a memorandum of it."
Lewis Carroll, Through the Looking Glass (1871)

On 11th March, 2011 a large earthquake fiercely shook eastern Japan followed by devastating tsunamis. The disaster took more than ten thousand lives and even more people are still missing at the time of writing this paper. Japanese essayist Kenko Yoshida (1283-1350) wrote, "Death sneaks upon us from the back door not the front". After past disasters, stones with messages warning not to build houses below that spot were erected. However, this recent tsunami was far higher than those high-water marks. Perrow, in his book "Normal accident" (1999), placed nuclear plants at the complex interaction and tight coupling domain on the interaction and coupling (IC) chart. The argument is basically very simple. Perrow start with a plant, airplane, ship, biology, or other setting with a lot of components (parts, procedures, and operators). Then we need two or more failures among components that interact in some unexpected way. No one dreamed that when X failed, Y would also be out of order and the two failures would interact so as to both start a fire and silence the fire alarm. Furthermore, no one can figure out the interaction at the time thus know what to do. The problem is just something that never occurred to the designers. Next time they will put in an extra alarm system and a fire suppressor, but who knows, that might just allow three more unexpected interactions

among inevitable failures. This interacting tendency is a characteristic of a system, not of a part or an operator; we will call it the "interactive complexity" of a system. For some systems that have this kind of complexity, such as universities or research and development labs, the accidents will not spread and be serious because there is a lot of slack available, and time to spare, and other ways to get things done. But suppose the system is also "tightly coupled," that is, process happen very fast and can not be turned off, the failed parts can not be isolated from other parts, or there is no other way to keep the production going safely. The recovery from the initial disturbance is not possible; it will spread quickly and irretrievably for at least some time. Indeed, operator action or the safety systems may make it worse, since for a time it is not known what the problem really is. Probably many production processes started out this way - complexity interactive and tightly coupled. But with experience, better designs, equipment, and procedures appeared, and the unexpected interactions were avoided and the tight coupling reduced. Perrow argue that system failures cannot be managed properly in a complex-tight domain. The Fukushima No. 1 nuclear plant will require a long recovery from the disaster because it is a complex and tight system, as Perrow mentioned. There must be a better way than to only observe and accept horrible consequences of normal accidents. If interactive complexity and tight coupling – system characteristics – inevitably will produce an accident, we are justified in calling it a normal accident, or a system failure. The odd term normal accident is meant to signal that, given the system characteristics, multiple and unexpected interactions of failures are inevitable. Kenko is correct in that we should be cautious of our blind spots. Earthquakes and Tsunamis are complex interaction and tight coupling systems as are nuclear cooling systems and the critical status in atomic reactors. The effort to reduce risk seems to be endless and may not be completed within our own lifetimes. However, we hope this paper will shed some light on visualizing risk factors in various systems to pursue a better and happier life. We send our deepest condolences to the victims, particularly those who have lost loved ones, in the earthquake and tsunamis.

Risk management is a broader notion of system management science. We focused on risk management of system failures. Most current risk management methodologies in the engineering field are based on a reductionist approach (Nakamura and Kijima, 2008a; IEC60812 (2006); IEC61025 (2006)). This approach is not adequate for coping with complex environmental aspects that can not be predictable in the design phase. We need a holistic approach for managing risks of system failures. We argue that two prerequisites should be satisfied to develop such an approach. The first one is providing a common

language or framework for holistically understanding system failures. The second one is quantifying system failures expressed through the common language or framework. The quantification of risk enables us to understand system failures objectively and compare failures among different systems and share the best practices to remove risk to improve system safety and quality. These two prerequisites are introduced in the next section. The common language called System of system failure (SOSF) is introduced as a meta-methodology for preventing system failure. We propose a method for quantifying the risk factors of system failures using the close code of system failures over time. Every industry or organization managing system failures uses a close code system to improve system quality and performance. The close code system is classified into two dimensions, i.e., interaction and coupling, and a specific area in the SOSF is then quantified. The interaction and coupling classification was introduced by Perrow (1999) in a subjective manner. This quantification method serves as a metric in the SOSF space. Therefore, SOSF become a metric space. This enables us to objectively visualize the risk factors of system failures. We discuss the application of this method to ICT systems in Section 4 and present several findings as concluding remarks to answer the following research questions.

1.1 Research questions

There are two research questions based on the topological presentation of system risk. The first is, “If a quantified expression of the IC chart is developed, is it possible to visualize system risk factors between systems and monitoring system quality improvement?”. The second is “If the plural-complex-class 3 domain (plural stakeholders – complex systems – emergent failure that can not be anticipated at design phase and there are few methodologies to cover this domain) in the SOSF space (details are explained in Section 2) corresponds to the complex-loose domain in the IC chart as well as the system failure occurrence ratio, could the effort to improve system quality be represented by the migration from the complex-loose domain to the linear-tight domain in the IC chart by reducing the system failure occurrence ratio?”. We attempt to answer these questions, especially in ICT systems, in the following sections.

2. SOSF META-METHODOLOGY AS COMMON LANGUAGE

System of system failure

The proposed SOSF meta-methodology for covering all system failure models (Nakamura and Kijima, 2007, 2008b, 2009a) is derived from system of system methodologies (SOSM) (Jackson, 2003, 2006) and system failure classes. System of system methodologies classifies the world of objects into two dimensions: systems and participants. The system dimension has two domains: simple and complex. The participant dimension has three domains: unitary, plural, and coercive. Therefore, SOSM classifies the world of objects into six (2 × 3) domains, and there is an appropriate methodology for each domain. System of system failure complementarily covers these domains on the basis of this worldview to enable viewing of objects system failures. System of system failure uses four domains (excluding the coercive domain because the main focus of this paper is technological systems rather than broader social domains) from SOSM. On top of these four domains, we add a third dimension to identify the person or factor responsible for the system failure. To identify the root causes of failures, we classify system failures on the basis of system boundaries and the responsible system level introduced with the viable system model (VSM) (Beer, 1979, 1981). Failures are classified in accordance with the following criteria (Nakamura and Kijima, 2008b, 2009ab).

Class 1 (Failure of deviance): The root cause is within the system boundary, and conventional troubleshooting techniques are applicable and effective.

Class 2 (Failure of interface): The root cause is outside the system boundary but is predictable in the design phase.

Class 3 (Failure of foresight): The root cause is outside the system boundary and is unpredictable in the design phase.

System safety can be achieved through the actions of various stakeholders. One such common language was developed by Van Gigch (1986) for the taxonomy of system failures. There are six categories of system failures, i) technology, ii) behavior, iii) structure, iv) regulation, v) rationality, and vi) evolution. In particular, SOSF was designed by allocating each type of failure from this taxonomy (Van Gigch, 1986) into an SOSM meta-methodology space. Figure 1 shows this space.

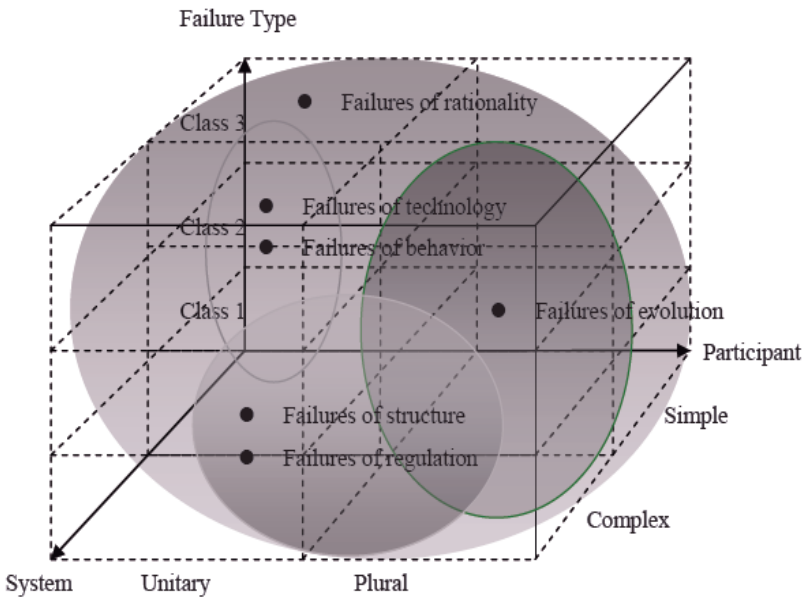

Figure 1 SOSF meta-methodology space

There are two widely used failure analysis techniques: failure mode effect analysis (FMEA: IEC 60812) and fault-tree analysis (FTA: IEC 61025). FMEA deals with single-point failures by taking a bottom-up approach, and is presented as a rule in the form of tables. In contrast, FTA analyzes combinations of failures in a top-down way, and is visually presented as a logic diagram.

Both the methodologies are mainly employed in the design phase. However, these methodologies are heavily dependent on personal experience and knowledge, and FTA in particular has a tendency to miss some failure modes in failure mode combinations, especially emergent failures.

The major risk analysis techniques (including FMEA and FTA) are explained in (Bell, 1989, 'pp 24-27'; Wang, J.X. et al, 2000, Chapter 4; Beroggi et al, 1994). Most failure analyses and studies are based on either FMEA or FTA. FMEA and FTA are rarely both performed, though, and when both are done they will be separate activities executed one after the other without significant intertwining. Current methodologies tend to lose the holistic view of root causes of system failures. And majority of them stay in the unitary-simple-class 1 domain. It is important to identify and cover the plural-complex-class 3 domain. In the next section, we introduce our quantification method of system failures identified using SOSF.

3. IC CHART AND QUANTIFICATION OF RISK FACTORS

3.1 Normal accident theory and IC chart

It is not unusual that several failures happen sequentially or simultaneously. Each is not a catastrophic failure in itself; however, the complex (i.e. unexpected) interaction of those failures may have catastrophic results. Tight coupling of a component involves a cascade of single-point failures that quickly reach a catastrophic end before safety devices come into effect. This is called system failure or normal accident as opposed to a single-point failure. Perrow analyzed system failures using interaction and coupling of system components. This is called normal accident theory. Tables 1 and 2 list the interaction feature and coupling tendencies respectively, according to Perrow's definition (1999). Table 1 explains linear and complex system interactions. Linear interactions are those in expected and familiar production or maintenance sequences and are quite visible even if unplanned. Linear systems have minimal feedback loops, and thus less opportunity to baffle designers or operators. And the information used to run the system is more likely to be directly received, and to reflect direct operations. Complex interactions are those of unfamiliar sequences, or unplanned and unexpected sequences, and either not visible or not immediately comprehensible. To summarize, complex systems are characterized by i) proximity of parts or units that are not in a production sequence, ii) many common mode connections between components (parts, units, or subsystems) not in a production sequence, iii) unfamiliar or unintended feedback loops, iv) many control parameters with potential interactions, v) indirect or inferential information sources, and vi) limited understanding of some processes.

Table 1 Linear vs. Complex Systems

Linear Systems	Complex Systems
● Spacial segregation	● Proximity
● Dedicated connections	● Common-mode connections
● Segregated subsystems	● Interconnected subsystems
● Easy substitutions	● Limited substitutions
● Few feedback loops	● Feedback loops
● Single purpose, segregated control	● Multiple and interacting controls
● Direct information	● Indirect information
● Extensive understanding	● Limited understanding

Table 2 explains the nature of coupling (i.e. tight and loose). Coupling is particularly germane to recovery from the inevitable component failures that occur. One important difference between tightly and loosely coupled systems deserves a more extended comment in this connection. In tightly coupled systems the buffers and redundancies and substitutions must be designed in; they must be thought of in advance. In loosely coupled systems there is a better chance that expedient, spur-of-the –moment buffers and redundancies and substitutions can be found, even though they are not planned ahead of time. What is true for buffers and redundancies is also true for substitutions of equipment, processes, and personnel. Tightly coupled systems offer few occasions for such fortuitous substitutions; loosely coupled ones offer many.

Table 2 Tight and Loose Coupling Tendencies

Tight Coupling	Loose Coupling
● Delays in processing not possible ● Invariant sequences ● Only one method to achieve goal ● Little slack possible in supplies, equipment, and personnel ● Buffers and redundancies are designed-in, deliberate ● Substitutions of supplies, equipment, limited personnel and designed-in	● Processing delays possible ● Order of sequences can be changed ● Alternative method available ● Slack in resources possible ● Buffers and redundancies fortuitously available ● Substitutions fortuitously available

The IC chart is a table for classifying object systems by interaction and coupling. Figure 2 shows the IC chart developed by Perrow. Topological expression was done subjectively by Perrow. By combining the two variables in this way, a number of conclusions can be made. It is clear that the two variables are largely independent. Examine the top of the chart from left to right. Dams and nuclear plants are roughly on the same line, indicating a similar degree of tight coupling. But they differ greatly on the interaction variable. While there are few unexpected interactions possible in a dams and there are many in nuclear plants. Or, looking across the bottom, university and post offices are quite loosely coupled. If something goes wrong in either of these, there is plenty of time for recovery, nor do things have to be in a precise order. But in contrast to universities, post offices

do not have many unexpected interactions - it is a fairly well laid out (linear) production sequence without a lot of branching paths or feedback loops. The IC chart defied two key concepts, the types of interaction (complex and linear) and the types of coupling (loose and tight). There variable has been laid out so that we can locate organizations or activities that interest us and show how these two variable, interaction and coupling, can vary independently of each other. The next section introduces a method for using a metric in the IC chart. The metric is a close code system of an object system's failures. This enables us to quantitatively monitor a target system's safety and quality.

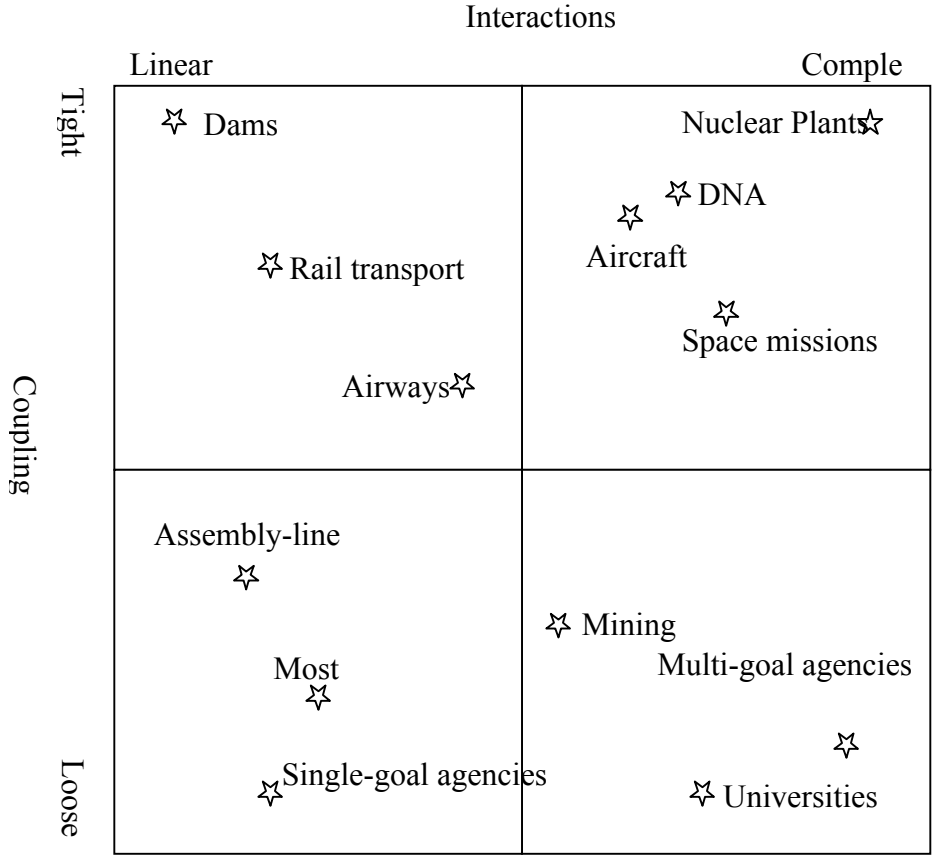

Figure 2 Interaction/Coupling Chart

3.2 Close code metrics

In every industry or organization, system failures are classified using close codes based on the root cause analysis of failures. In this chapter, we use the close code system as a metric to objectively represent risks. Generally, close codes can be classified into two

dimensions. The first dimension consists of phases for creating an object system (i.e. design, configure, and operate in time sequence) and the second is the nature of the stakeholders (i.e. simple or complex) relating to system failures. The close code system is a so called filter of the root causes of system failures. Figure 3 shows the general concept of close code classification. The loop represent learning cycles, and in complex cases, the learning cycles spread over multiple phases and stakeholders. Most industries use the close code system reactively for single-system failures. However, it is important to monitor the close codes accumulated for any arbitrary amount of time and to confirm effective countermeasures. To achieve this, it is necessary to introduce metrics to quantitatively represent the risk status.

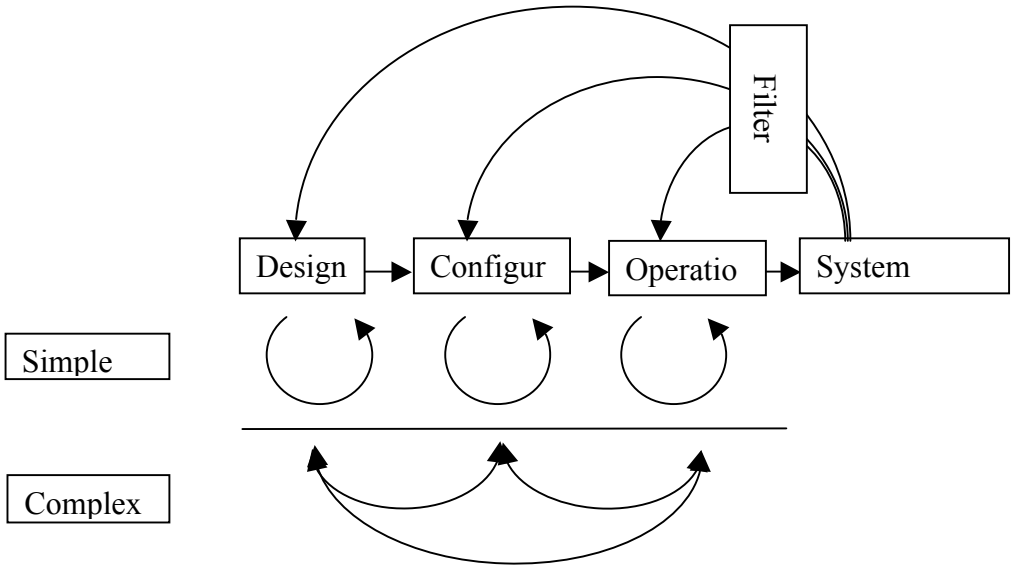

Figure 3 Classification of close codes and learning cycle from system failure

The close code system should be confirmed in terms of the taxonomy of system failures (Van Gigch; 1986, 1991). This is to verify the mutually exclusive and collectively exhaustive close code system. Table 3 is an example of mapping a close code system onto a close code matrix with regards to the ICT industry. The close code system varies by system or industry. However, it is possible to classify the close code system into a

close code matrix with the two dimensions. The two-tuple number (X, Y) represents the area in the close code matrix. For example (3, 2) represents the operation-complex domain in the close code matrix. The symbols A, B, P, and N in Table 3 represent the causes of system failures. They are A: hardware malfunctions, B: human behavioral error, P: maintenance period expiration, and N: future consideration to implement new features (i.e. to avoid further system failures). Causes A and B have subcategories. The subcategories of A are A1: CPU, A2: memory, A3: channel, A4: power, A5: disk, AB: hardware setup mistakes, A6: other IOs, and AU: unknown causes. The subcategories of B are BA: network setup mistakes, BB: IO setup mistakes, BC: parameter setup mistakes, BD: Installation mistakes, BE: operation mistakes, BF: application coding mistakes, and BG: other mistakes.

The close code matrix is related to the IC chart in terms of classification of system failures. The first dimension of the close code matrix (i.e. design, configure, and operate) corresponds to the interaction axis of the IC chart. The second dimension of the close code matrix (i.e. simple and complex) corresponds to the coupling axis of the IC chart. The next section introduces the metric into the SOSF space derived from the close code matrix.

Table 3 Mapping close code system onto close code matrix

<table>
<tr><td rowspan="2"></td><td rowspan="2">Close Codes</td><td>1 (Design)</td><td>2 (Configure)</td><td>3 (Operation)</td></tr>
<tr><td>Failure of Technology and Structure</td><td>Failure of regulation</td><td>Failure of behavior and evolution</td></tr>
<tr><td colspan="4">Failure of Rationality, Evolution</td></tr>
<tr><td rowspan="4">1 (Simple)</td><td>A (Hardware)</td><td>A(1~5)</td><td></td><td>A(B)</td></tr>
<tr><td>B (Behaviors)</td><td></td><td>B(A~D,F)</td><td>B(E)</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>P (Obsolete)</td><td></td><td>P</td><td></td></tr>
<tr><td rowspan="4">2 (Complex)</td><td>A (Hardware)</td><td>A(6)</td><td></td><td>A(U)</td></tr>
<tr><td>B (Behaviors)</td><td></td><td></td><td>B(G)</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>N (Future plan)</td><td>N</td><td></td><td></td></tr>
</table>

3.3 Topological presentation of system failure risk factors

We first introduced the system failure space (i.e. SOSF) in Section 3.1, then introduced quantification of risk factors of the close code system in Section 3.2. If we use the close codes as the metric in the SOSF, we can topologically present the risk factors of system failures. Every single-system failure can be located in the SOSF space. Object system risk location is presented topologically within the SOSF space with this metric. An object system's risk factor is represented quantitatively in the SOSF space with the metric, which monitors the transition of the risk factor over time. The risk factor location at any arbitrary time of an object system is represented by a three-tuple number. The object system risk location in the SOSF space can be represented by System Risk Location (SRL) (X, Y, Z), where X represents the metrics of system interactions, Y represents the metrics of system coupling, and Z represents the annual call rate (ACR): incidents/100 shipments per year). There are several steps for introducing these metrics into the SOSF space, as shown in Fig. 4. The first step is defining a system failure group at any arbitrary time. This group is the bases of calculating system failure risk factors and is expressed in the SOSF space. The second step is mapping the corresponding close code system onto the closed code matrix. The third step is corresponding the close code matrix with the IC chart. The X (Y) axis corresponds to the interaction (coupling) axis. The (3, n) (i.e. n = 1: Simple or 2: Complex) area in the close code matrix corresponds to the right side of the interaction axis (i.e. complex area) in the IC chart. The (m, 2) (i.e. m=1: Design, 2: Configuration or 3: Operation) area in the close code matrix corresponds to the lower area of the coupling axis (i.e. loose area) in the IC chart. The quantification of risk factors can be achieved using the (m, n) notation in the close code matrix. The complex interaction risk factor can be represented by the number α: (3, n)/ number of system failures at any arbitrary time. The loose coupling risk factor can be represented by the number β: (3, 2)/ number of system failures at any arbitrary time. Figure 4 shows that β is the area inside α; therefore, β is defined as (3, 2)/number of system failures, not (m, 2)/number of system failures. The reason of measuring β in α is the risk of an object systems should be measured during the operation phase. The complex and loose risk factors of an object system can be represented as a two-tuple number $\gamma = (\alpha, \beta)$. This is the quantitative coordinate point in the IC chart.

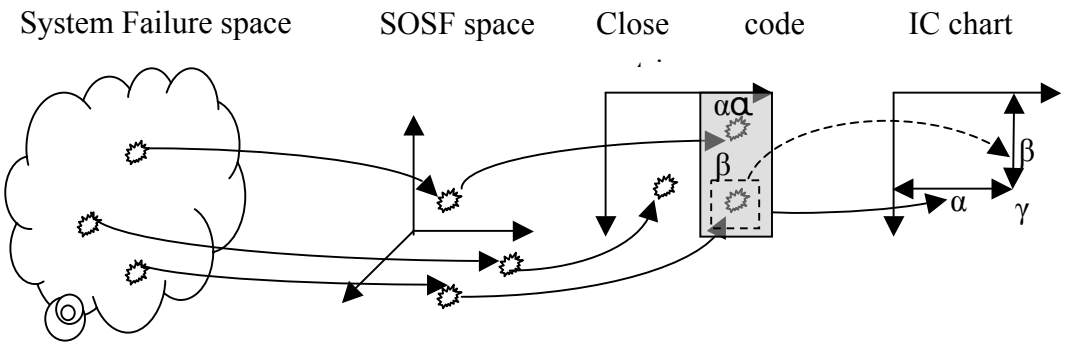

Figure 4 General Sequence of introducing metrics into SOSF

We define γ as the representation of an object system's risk factor and objectively place it in the IC chart (i.e. with a metric). Figure 5 gives a detailed explanation of γ from the system failure group at any arbitrary time. Therefore, γ = (α, β) in the IC chart. Adding a new dimension (i.e. Z axis representing ACR) to γ produces an SRL (α, β, ACR). The γ can only represent the looseness and complexity of the target system. The frequency of system failures should be incorporated in the system-failure metric. This is the reason for introducing a 3rd dimension of ACR. ICT development engineers use the annual failure rate (AFR) for monitoring system component quality rather than the system as a whole. ICT users who encounter the problems of the products report the incident to the help desk, and the help desk provides them with a solution. The help desk then identifies the cause of the incident, and, if it was caused by faulty product design, the help desk escalates it to the development section for further investigation. The development section designs new products on the basis of data for the escalated incidents that the help disk believes were due to product defects. This is mainly because the user-related incidents are screened at the help desk so that the development section can concentrate on product-related issues. The development section measures product quality by AFR (Annual Failure Rate) using only the incidents escalated from the help desk, not by ACR (Annual Call Rate) using all the incidents received directly from the users. The metric for product quality is the AFR and system quality that includes product quality is the ACR, which are calculated as shown in Fig. 6. Corresponding the 3rd axis of the SOSF area (i.e. system failure classes 1, 2 and 3) with the ACR is straightforward because the AFR stands for class 1 failure, and the difference in the ACF and AFR stands for classes 2 and 3 failures. Figure 5 shows the transition of SRL over time. To emphasize the magnitude of the ACR, the size of the black circles in this figure changes according to the ACR value. Figure 5 also shows that the initial SRL at t_0 could shift to the SRL at t_1 with increasing ACR (large

circle) or to the SRL at t_2 with decreasing ACR (small circle). The black Circles in the metric SOSF space change size to represent the ACR. Table 4 summarizes the metrics introduced into the SOSF space and the relation with the closed code matrix and IC chart.

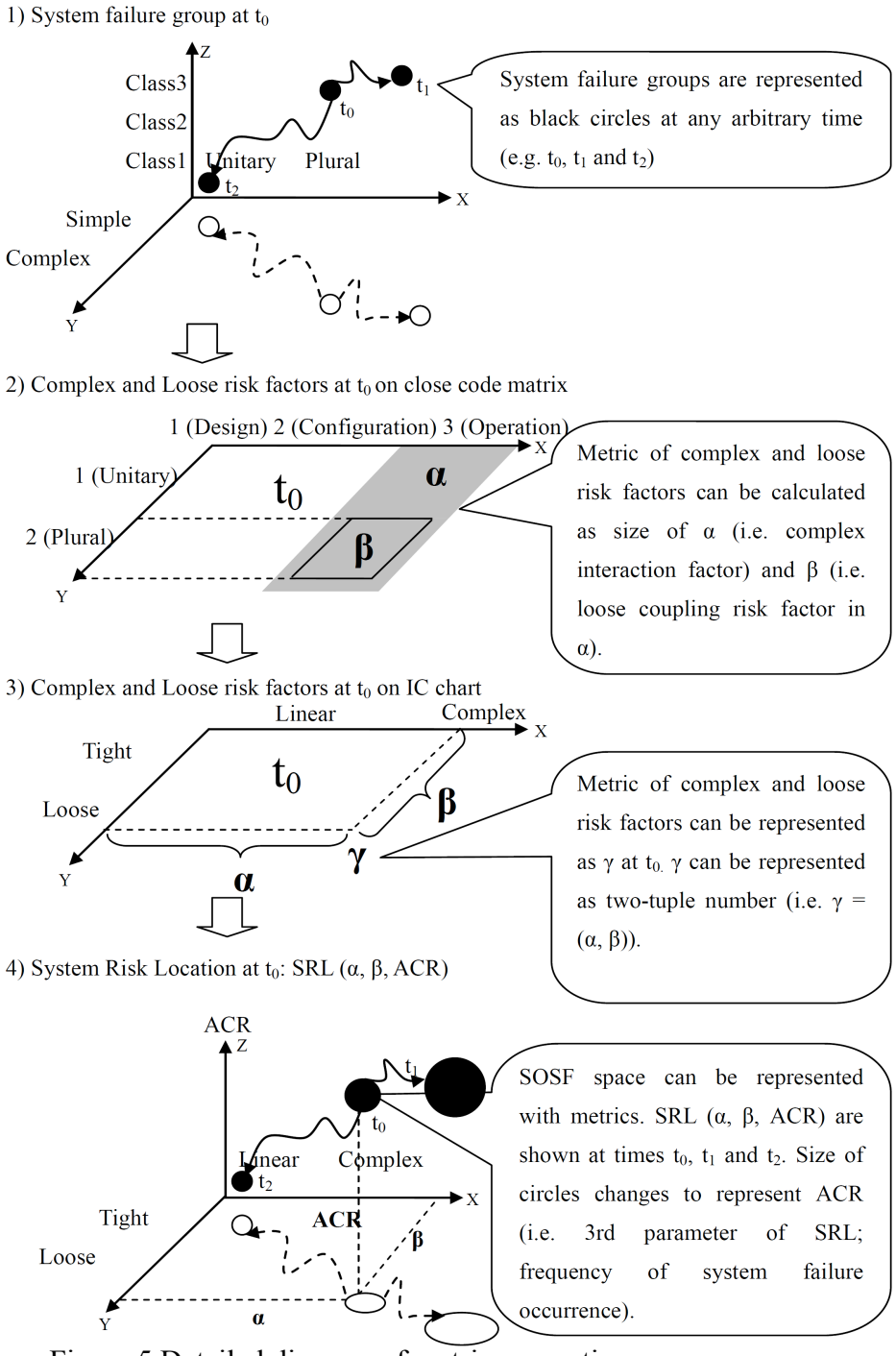

Figure 5 Detailed diagram of metric generation

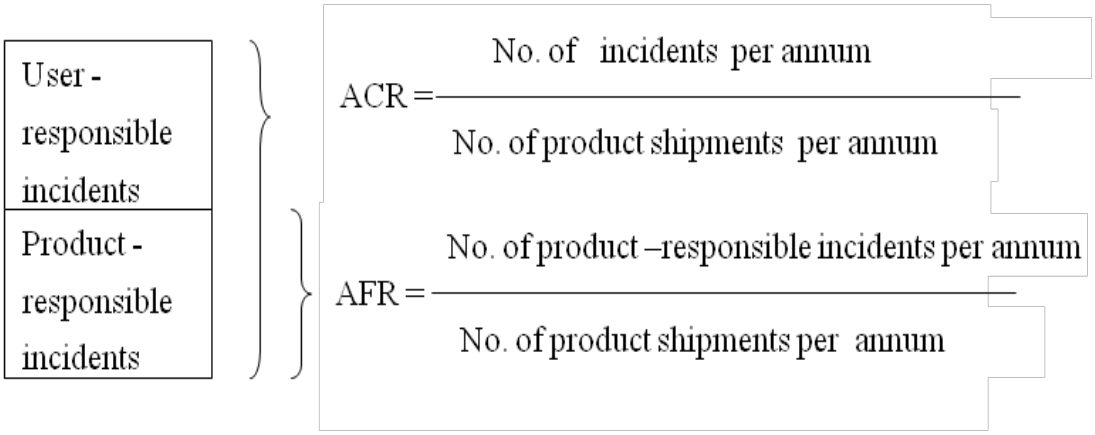

Figure 6 Calculation of annual failure rate (AFR) and annual call rate (ACR)

Table 4 Summary of SOSF to SOSF with metrics via close code matrix and IC chart

Axis	SOSF	Close code matrix	IC chart	Metric SOSF SRL (X,Y,Z)
X	Stakeholders (i.e. Unitary and Plural)	System Creation Phase (i.e. Design, Configuration and Operation)	Interaction (i.e. Linear and Complex)	Interaction Metrics(α): (3,n)/all incidents
Y	System feature (i.e. Simple and Complex)	System feature (i.e. Simple and Complex)	Coupling (i.e. Tight and Loose)	Coupling Metrics (β): (3,2)/all incidents
Z	Failure class (i.e. Class1,2 and 3)	N/A	N/A	ACR (including AFR)

According to the discussion of an SRL (α, β, ACR), a larger α:(3，n) indicates that an object system has more complex interaction, a larger β:(n, 2) indicates that an object system has more loose coupling, and a larger (3，2) indicates that an object system has more complex and loose properties. Figure 7 shows the transition of the second research question; the transition to linear and tight transition would decrease ACR (i.e. small circle).

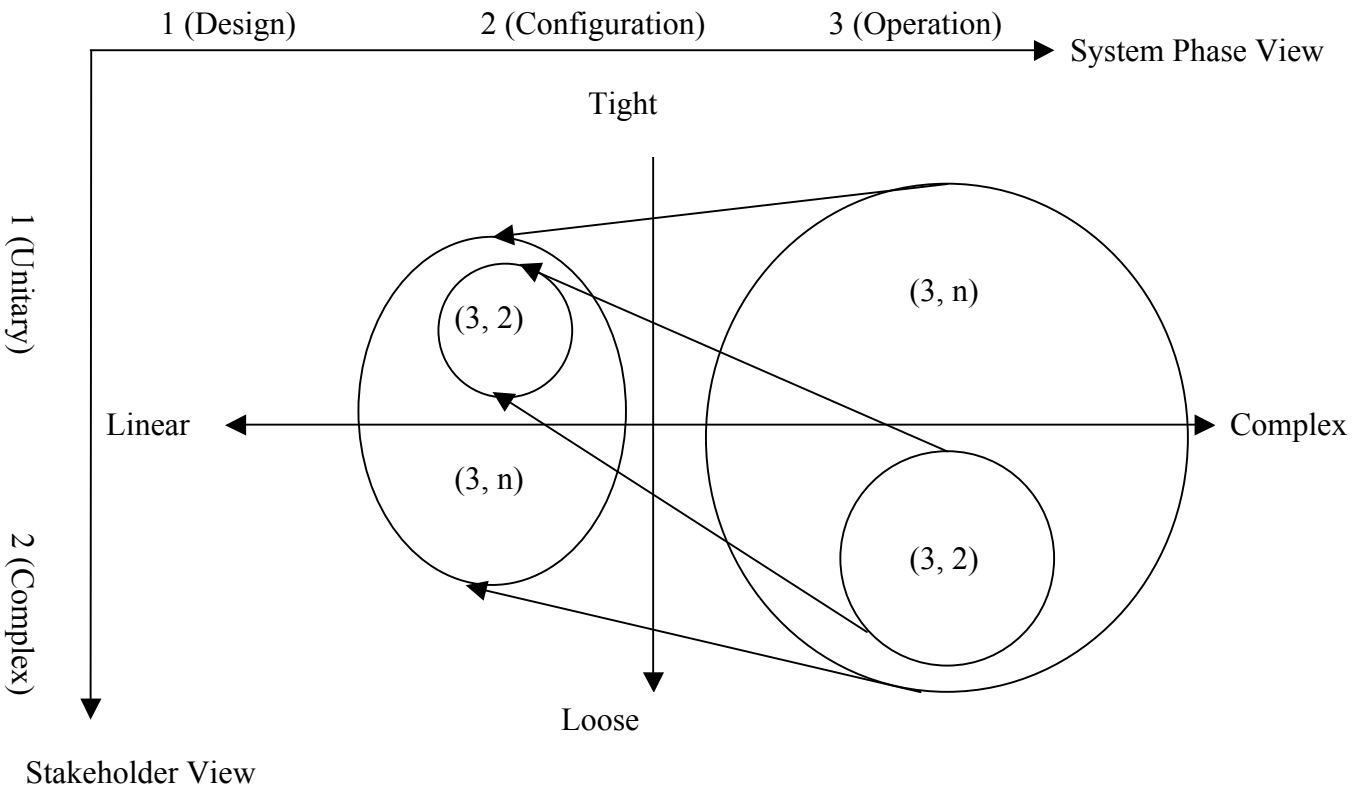

Figure 7 Transition to linear and tight with decreasing ACR

4. APPLICATION TO ICT SYSTEMS

4.1 Topological presentation of SRF for various ICT systems

Figure 8 shows the system risk factor distribution between several ICT systems. Table 5 lists the data related to specific ICT systems. The SRL (α, β) is calculated based on the incidents that occurred in a system. The third component of the SRL was not considered because there were no reasons to compare the ACR between the different systems. All ICT systems in Fig. 8 have correlation between the two factors (i.e. interaction and coupling). The stronger the interaction (i.e. linear) becomes the stronger the coupling (i.e. tight). The number of jobs or tasks in an object system could affect the results. Stock exchanges are more single-goal agencies than other object systems. They tend to reside in the linear-tight domain, on the other hand multi-goal agencies (i.e. meteorological and healthcare) have the opposite tendency. This result is in accordance with the design concept of each systems. This qualitative argument was confirmed with this quantitative measure.

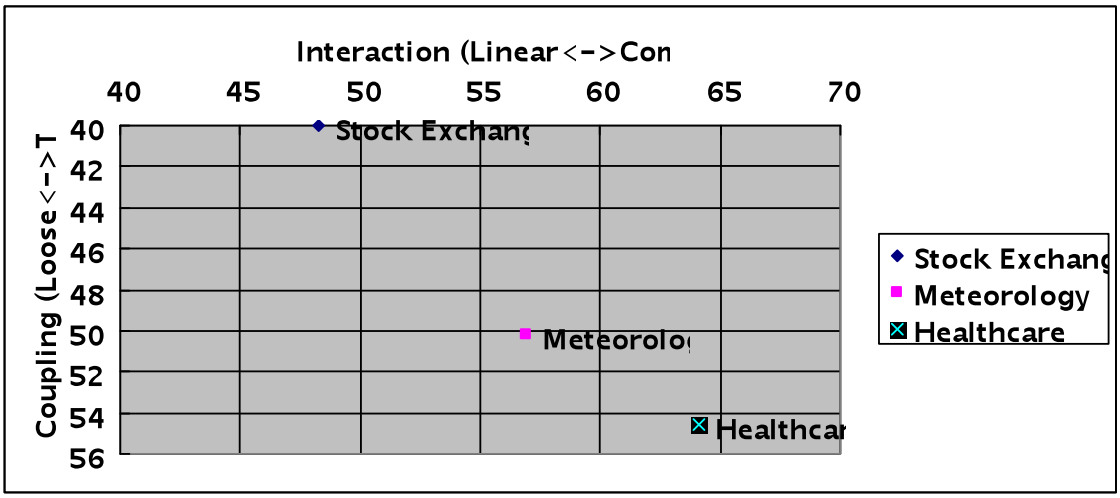

Figure 8 distribution of SRL between various ICT systems

Table 5 Data from ICT systems (Figure 8)

System	A	β	Incidents	duration
Stock Exchange	48.3	40	145	2010
Meteorology	56.9	50.2	239	2008-9
Healthcare	64.1	54.6	940	2008

4.2 IA server systems shift to linear interaction and tight coupling

Figure 9 shows the SRL transition of IA server systems (all systems that uses Intel Architecture servers). Table 6 lists the data used in the SRL calculation. During this transition period, IA server systems migrate toward the linear interaction and tight coupling domain with decreasing ACR. During this period, two counter measures are used (Nakamura and Kijima, 2008a, 2008b, 2009b) in these systems. The first countermeasure is to educate engineers to become hybrid engineers who can handle hardware and software. This countermeasure does not require cumbersome communication between engineers and reduces long outstanding incidents due to lengthy communication. The second countermeasure is to alter the noise design goal to achieve an acceptable noise level to use even in an office environment. The ACR decreased by 69% compared to that in 2004. The first change is contributed to a tight coupling shift

due to the removal of such cumbersome communication and the second is contributed to the linear interaction shift due to the reduction in stakeholders in an office environment.

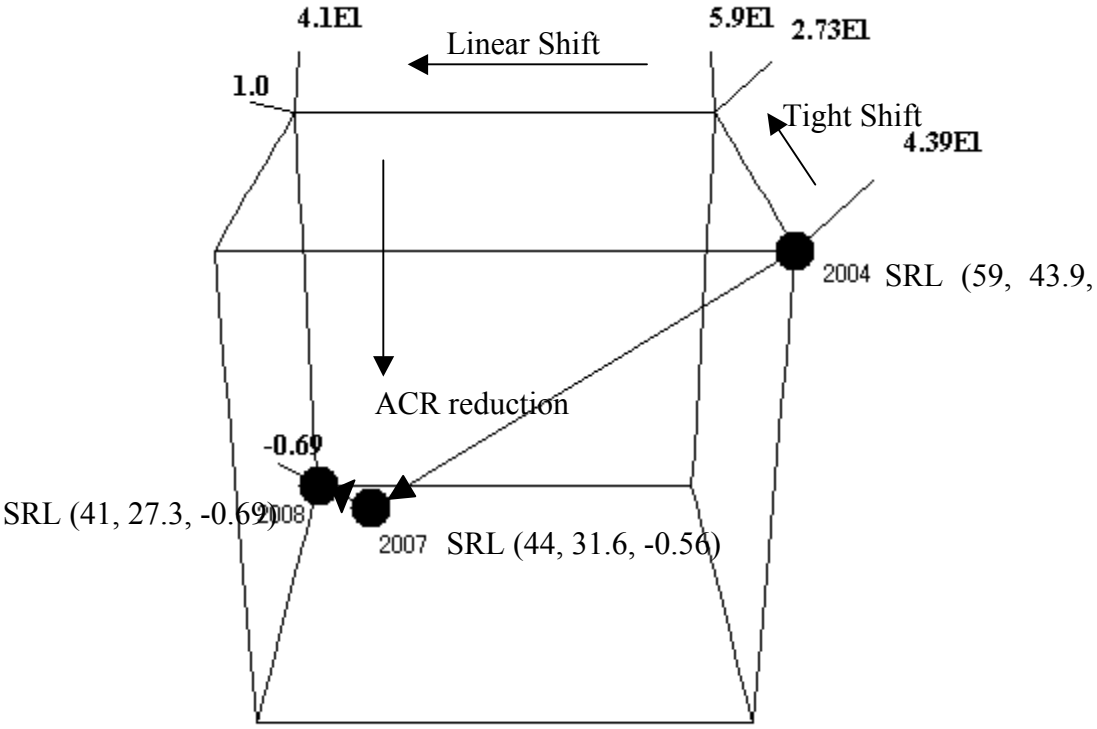

Figure 9 SRA transitions of IA server systems

Table 6 Data from IA server systems (Figure 9)

	α	β	ACR	Incidents
2004	59	43.9	1	82
2007	44	31.6	-0.56	250
2008	41	27.3	-0.69	242

4.3 Healthcare systems shift to complex interaction and loose coupling

Figure 10 shows the SRL transition of healthcare systems. Such systems include various tasks or jobs. Typical systems are electric health record systems and those for processing medical practitioners' receipts for health insurance claims. They are closely related to national policy to comply with the Electronic Data Interchange (EDI) policy. Table 7 lists the data used in the SRL calculation when healthcare systems are migrating toward the complex interaction and loose coupling domain with increasing ACR. The main factor for this complex interaction shift is the EDI policy. In accordance with this change, new stakeholders (i.e. medical equipment vendors and politician) participate in the new EDI processes, which is a main factor for the loose coupling shift of systems. This shift requires the introduction of a countermeasure to more clearly define system boundaries to remove stakeholders' misunderstandings or to overlap objectives between systems. This requires further research to confirm the outcome.

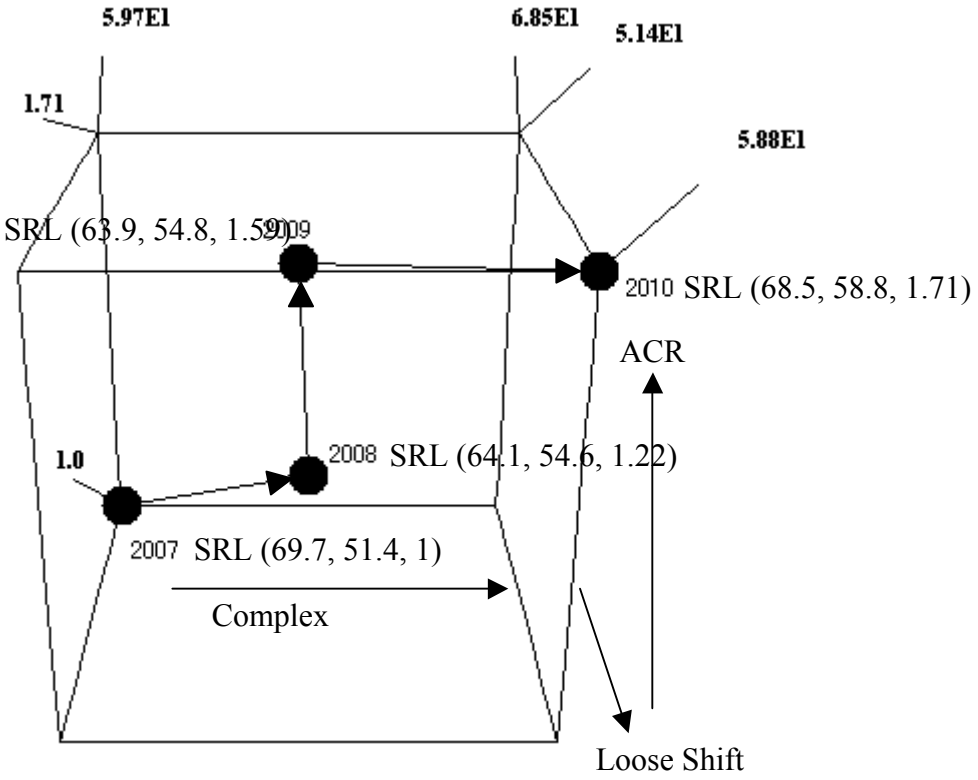

Figure 10 SRA transitions of healthcare systems

Table 7 Data from healthcare systems (Figure 10)

	α	β	ACR	Incidents
2007	59.7	51.4	1	769
2008	64.1	54.6	1.22	940
2009	63.9	54.8	1.59	1220
2010	68.5	58.8	1.71	1317

5. CONCLUSION

We obtained several findings from the application of our risk quantification method to several ICT systems. The results of this application confirmed the two research questions introduced in Section 1.1. For the first research question, we confirmed that the risk factors for system failures can be quantified and presented as a topological space (i.e. SOSF space with IC metrics). According the comparisons of various ICT systems, stock exchange systems are more linear and tight than meteorological or healthcare systems. Stock exchanges are single-goal agencies. On the other hand, healthcare systems are migrating toward the complex-loose domain due to electric medical record systems with various stakeholders being introduced to EDI policy. If migration toward complex interaction is inevitable to adapt to environmental change, other countermeasures for preventing migration in the tight coupling direction could be a challenge for healthcare systems. Clarification of job goals or seeking loose coupling between systems within healthcare could promote system safety. However, this requires more research to reach concrete results.

As for improving system safety for IA servers, educating engineers to become hybrid engineers results in tight coupled migration, and decreasing noise in the design goal results in linear interaction migration. Along with these two shifts, ACR decreased by 69% in four years. This confirms the second research question which stipulates that migration toward the tight-linear domain of IA server systems enhances system safety.

We verified that the linear-tight domain is safer than the complex-loose domain for ICT systems, especially IA server systems. Healthcare systems are migrating toward the complex-loose domain as the ACR increases. Further research is required to confirm whether the countermeasures for preventing migration in the tight coupling direction would reduce the ACR in healthcare systems.

The proposed method for visualizing risk factors by introducing metrics in the SOSF space is effective because it complements the shortcomings of the subjective IC chart. Complex and tight shifting could be prevented by periodically monitoring the SRL trajectory in the SOSF space. This would enable us to objectively compare various systems in terms of risk management, and assure that countermeasures will be introduced to migrate toward the ideal domains.

REFERENCES

Beer, S. (1979). The Heart of Enterprise. John Wiley & Sons: London and New York.

Beer, S. (1981). Brain of the Firm, 2nd edition. John Wiley & Sons: London and New York.

Bell, T.E., ed. (1989). 'Special Report: Managing Murphy's law: engineering a minimum-risk system,' IEEE Spectrum, June, pp 24-57

Beroggi, G.E.G. and Wallace, W.A. (1994). 'Operational Risk Management: A New Paradigm for Decision Making,' IEEE Transactions on Systems, Man and Cybernetics, Vol.24, No.10, October, pp.1450-1457

IEC 60812 (2006). Procedure for failure mode and effect analysis (FMEA)

IEC 61025 (2006). Fault tree analysis (FTA)

Jackson, M.C. (2003). Systems Thinking: Creative Holism for Managers. John Wiley & Sons: London and New York.

Jackson, M.C. (2006). Creative Holism: A Critical Systems Approach to Complex Problem Situations. Systems Research and Behavioral Science Vol. 23, Issue 5, September/October 2006: 647-657.

Lewis, C. (2000). Through the Looking Glass: And What Alice Found There, Oxford Bookworms Library

Nakamura, T., Kijima, K. (2007). Meta system methodology to prevent system failures. Proceedings of the 51st Annual Meeting of the ISSS in Tokyo (Aug. 2007).

Nakamura, T., Kijima, K. (2008a). A Methodology for Learning from System Failures and its Application to PC Server Maintenance. Risk Management 10.1, 2008: 1–31.

Nakamura, T., Kijima, K. (2008b). Failure of Foresight: Learning from System Failures through Dynamic Model. Proceedings of the 52nd Annual Meeting of the ISSS in Madison (Jul. 2008).

Nakamura, T., Kijima, K. (2009a). System of system failures: Meta methodology for IT engineering safety. Systems Research and Behavioral Science Vol. 26, Issue 1, January/February 2009: 29–47.

Nakamura, T., Kijima, K. (2009b). A methodology to prolong system lifespan and its application to IT systems. Proceeding of the 53rd Annual Meeting of the ISSS in Brisbane (Jul. 2009).

Perrow, C. (1999). Normal Accidents: Living with High-Risk Technologies. Princeton Paperbacks: New York.

Van Gigch, J. P. (1986). Modeling, Metamodeling, and Taxonomy of System Failures. IEEE trans. on reliability, vol. R-35, no. 2, 1986 June: 131-136.

Van Gigch, J. P. (1991). System Design Modeling and Metamodeling. Plenum: New York.

Wang, J.X. and Roush, M.L. (2000). WHAT EVERY ENGINEER SHOULD KNOW ABOUT RISK ENGINEERING AND MANAGEMENT. Marcel Dekker, Inc.

Yoshida, K. (1985). Ture-Zure-Gusa, Chapter-155, Iwanami-Shoten (in Japanese)