Journal of the KSQC
Vol. 20, No. 2, Dec. 1992

1992년 12월 韓國品質管理學會誌 제20권 제2호

Experiment design and human reliability in software quality control system

Peom Park*

Abstract

This study involves an experiment for the cognitive experiment design and the human reliability in software engineering. Its overall objectives are to analyze common-cause human domain error and reliability in human-software interaction.

A laboratory study was performed to analyze software engineers' task behavior in software production and to identify software design factors contributing to the effects in common cause failure redundancy. Common-cause model and its function were developed, then the main experiment using programming experts was conducted in order to define a new cognitive paradigm, in the aspects of identification, pattern recognition, and behavior domain for human reliability and quality control in software development.

The results and analytical procedures developed in this research can be applied to reliability improvement and cost reductioin in software development for many applications. Results are also expected to provide guidelines for software engineering quality control and for more effective design of human-software interface system.

I. Introduction

In these days common-cause failure studies(Musa, 1987 ; Dhillon, 1983 ; Watson, 1981) in the human-system area have been receiving wide attention especially in the software systems area. This is because the assumption of

* Incheon University

statistically independent failure of redundant systems is easily violated in real human-software interaction processing systems. Since the software components are not independent of each other in regard to failure behavior, software redundancy does not improve reliability except in multiversion software development. That is often requested to improve of reliability, especially in ultra-high reliability systems such as nuclear power control, air traffic control, space shuttle missions, and war games. The major common-cause human domain errors found in this research can contribute strongly to internal common -cause failure effects in a multi-version software development project. The common-cause model includes three analytical reasoning categories and a common-cause function established in terms of human-software information processing systems, human error mechanisms, and cognitive control domains. It is used to characterize the human factors mechanisms behind typical categories of errors considered as occurrences of human-software task mismatches.

Scope of the Problems : This study deals with the problem of common cause human domain error in human-software interaction, that is, the major causal factors in common-cause failure effect on the multiversion software development. More qualified expert subjects in software development will give a better performance, but the major portion of common cause error properties in human-software interactions will not differ significantly among all subjects who have different categorical conditions such as level of programming expertise, knowledge of programming language, and type of task requirements.

Ⅱ. A Common Cause Model and Experimental Design

Common-cause model : The common-cause model can be used to define internal common-cause human-based error and to develop a common cause error control mechanism for human- software interaction. It can be explained in the schematic and systematic experimental design as illustrated in Figure 1. The stages of common-cause model are as follows : (1) human-software interaction componets, (2) common-cause error protocol, (3) common cause error function, and (4) common-cause analysis, representation, and system redesign.

Common-cause function : The common-cause function is shown in the existence and in performance allocation of common-cause error protocol with its identification(I_i), pattern recognition(P_j), and behavior domain(B_k) of common-cause error mode. Each allocated common-cause error mode is evaluated by performance variables

using common-cause error frequency(F_i, j, k), error correction time (C_i, j, k), point of error occurrence in time (O_i, j, k)during the software development period.

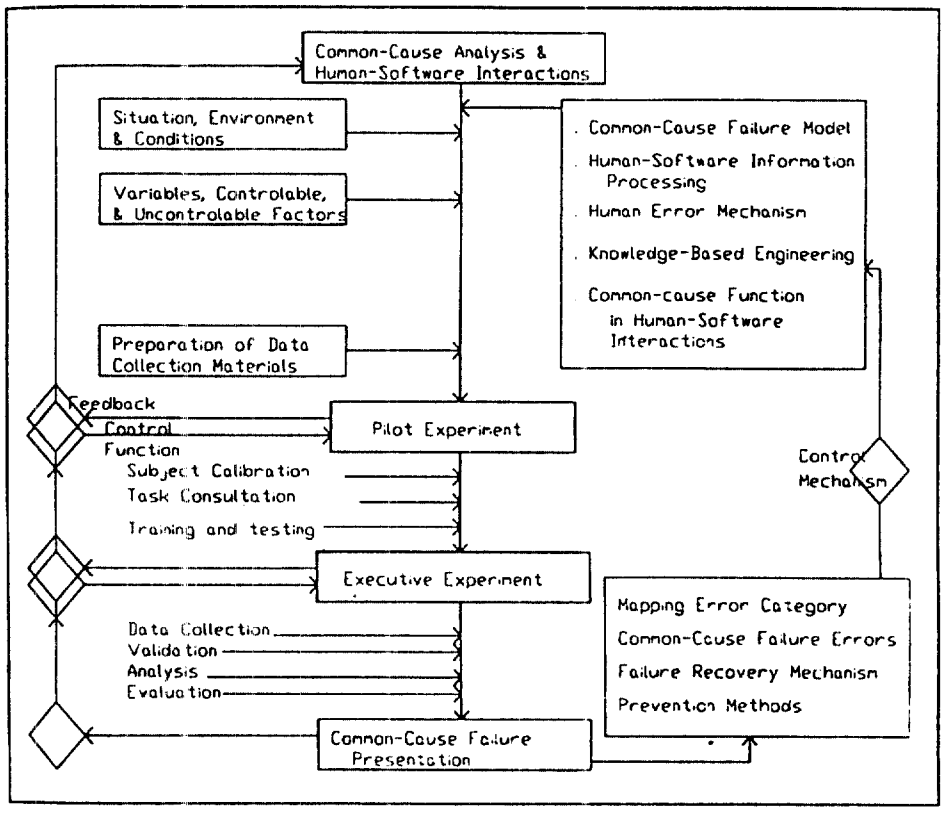


Figure 1. Schematic Experimental Design in Human Software Interaction

The common-cause function, C_r is :

C_r=C_C(I_i(F_i, C_i, O_i), P_j(F_j, C_j, O_j), B_k(F_k, C_k, O_k)), here, the summation of all I, P, B, F, C, O variables is equal to 1, and the range of them is 0 to 1.

There are three features of internal common--causes, that can be used in determining the caracteristics of common-cause failure effects in human-software intercation.

Identification of common-cause error protocol : There are eight identification modes(:I_i) categories of typical human-based programming error from common-cause error protocols, which are used in the determination of the common-cause error that caused the failure. Each error protocol mode means the actual location of common-

cause error and contributes to the common-cause effect at each stage of human-software interaction(Park, 1992 ; Thayer, 1978). They are : I-1 System design and requirement error, I-2 Variable setting and program handling error, I-3 Program input and data base error, I-4 Computation based error, I-5 Program logic error, I-6 Human-software system interface error, I-7 System operation error, I-8 Output and output formatting error.

Pattern recongnition c-c error modes : Common-cause reasoning patterns(:P-j) can be recongnized with causal characteristics which implicate the identical elements of reason, perception, control mechanism, occurrence processing, stimulus response requirement, etc. Each identical property or reason matches a pattern recognition for the common-cause human domain error mode(Park, 1992 ; Youngs, 1981 ; Andres, 1975). They are as follows : P-1 Knowledge deficiency, P-2 Design deficiency, P-3 Operation and maintenance error, P-4 Functional deficiency, P-5 Syntax error, P-6 Semantic error, P-7 Logical error, P-8 Clerical error, P-9 System complexity.

C-c behavior domain error modes : There is a common-cause error category in terms of the programmer's behavioral aspects(:B-k) or point of view. Such common-cause error factors may be representative of from the human information processing, knowledge based design, error control mechanism, and human behavioral science (Park, 1992 ; Rasmussen, 1987 ; Reason, 1987). They are as follows : B-1 Skill-based behavior domain : B-1. A Perception and sensing, B-1. B Automated sensory-motor reaction systems, B-2 Rule-based besed behavior domain : B-2. A Pattern matching and recognition, B-2. B Representation and association, B-2. C Working memory and rule interpreter, B-3 Knowledge based behavior domain : B-3. A Task identification and domain principle, B-3. B Object orientation and concurrent design, B-3. C Integration and optimization, B-4 Model-based behavior domain.

Ⅲ. Experimental Design and Procedure

After three prior pilot experiments conducted 3 leveled subjects and all materials, such as subject selection, specifications of requirement, experimental procedures, data collection sheets, and analytical materials for the experiment were prepared and subject life data were gathered. For reliable subject calibration, subjects were trained using the actual requirements and overall experimental procedures in an initial session and consultation session. Then, the main experiment was conducted with data collection according to frequency of common-cause error oc-

currence, error correction time, and point of error occurrence in time for each of the categorical factors : I_i, P_j, and B_k. During the experiment, the contents of common-cause human error in subject programming failure were, first, recorded with an explanation of the reasons for those failures, correction time, and point of error occurrence in time. Then, at the representational interview session held every 30－45 minutes, the common-cause error protocol was allocated to each of the common-cause factors, identification, pattern recognition, and behavior domains, by directed definition and cooperative decision with the supervisor and the subject.

Experimental data was then validated and analyzed by statistical methods and a geometrical method using vector analysis and mapping designed for use in analyzing common-cause errors in human-software reliability and interactions.

Results were derived using the following analytical methods : common-cause error mode data and table, mapping and geometric vector evaluation in hexahedron contours, value of common-cause function with simulated rating, historical common -cause error recovery time zone, transition relationship diagram, grouping of major common-cause factors, and correlation and regression analysis of categorical factors.

Verification of the results using expert subjects was intended to identify clearly those factors related to the design of software development as distinguished from conditional factors associated with level of subject, type of language, and type of requirement. Finally, the characteristics and the properties of common-cause failure modes in human-software interaction were determined by the analysis of experimental data collected on the ten expert subjects and compared with data from each of the categorical conditions.

Ⅳ. Common-Cause Analytsis and Result Representation

In the main experiment, two different required tests were conducted using two respective languages with subjects averaging 24 years in age, and 5.8 years of programming experience as shown in Table 1. Results consisted of a 32.1 average(9.4 standard deviation) total common-cause error frequency, and 255.3 minutes average total error correction time during 523 minutes total computing time per each version of software development. Time spent in understanding and problem solving was 109 minutes, and design time for programming was 170 minutes.

Weight rating factors were developed from interviews with the subject programming experts concerning the ex-

Table 1. Subject Task Data in a Common-Cause Model Experiment

S-ID[b]	Reqt[c]	Exp[d]	Freq[e]	Ct-T[f]	Com-T[g]	Sol-T[h]	Des-T[i]	Tot-T[j]
P4C01	P4-B	5	24.0	256.0	452	120	300	872
P4C02	P4-B	7	27.0	242.5	494	60	90	644
P4C03	P4-A	8	25.0	119.5	256	60	60	376
P4C04	P4-A	5	32.0	432.5	781	180	270	1231
P4C05	P4-A	5	52.0	487.5	886	180	210	1276
P4C06	P4-B	5	38.0	312.5	535	120	240	955
P4C07	P4-B	6	43.0	123.0	482	70	50	602
P4C08	P4-B	6	29.0	158.0	403	60	60	523
P4C09	P4-A	6	23.0	144.5	255	120	120	495
P4C10	P4-A	5	28.0	277.0	686	120	240	1046
MEAN		5.8	32.1	255.3	523	109	170	802
S.D.:		1.0	9.4	127.8	208.2	46.3	104.2	318.9

[a]Expressed as time in min.

[b]S-ID : Subject Identification No.

[c]Reqt : type of requirement(A : dynamic programming, or B : inventory control).

[d]Exp : programming experience(years).

[e]Freq : frequency of common-cause error mode.

[f]Ct-T : correction time of error.

[g]Com-T : computing time of program.

[h]Sol-T : problem solving time.

[i]Des-T : program design time.

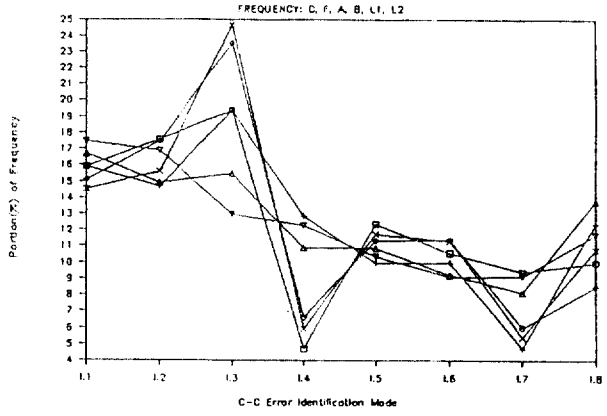
[j]Tot-T : total spent time.

periment in human software interaction.

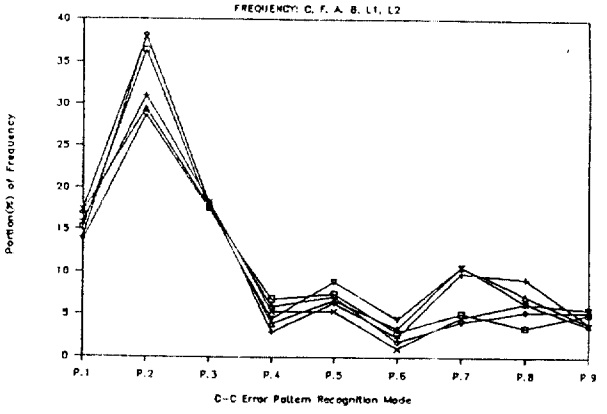
With the error occurrence frequency factor, the major reasoning categories in each common-cause error mode are : in the identification mode, I.3(19.4%), I.2(16.2%), and I.1(15.9%) ; in the pattern recognition mode, P.2 (33.7%), P.3(18.0%), and P.1(15.7%) ; in the behavior domain mode, B.3(43.6%) and B.2(36.5%). When the error correction time factor is applied, I.1(26.2%), I.5(16.6%), and I.8(13.9%) in the I_i mode ; P.2(44.8%) and P.1(21.2%) in the P_j mode ; and B.3(62.7%) and B.2(28.1%) in the B_k mode.

Figure 2 shows plots of proportional mean frequency based on six criteria for characteristics in each common cause error mode. All trends are similar except for I.3 and I.4 in the common-cause identification mode. Figure 3 shows plots of proportional mean correction time based on six criteria for characteristics. There are no significant differences in pattern recognition and behavior domain. However, I.4 and I.5 in the identification mode have a little difference in correction time. Figure 4 shows plots of proportional mean occurrence time based on the same characteristics. P.4, P.5 and P.6 in pattern recognition and B.4 in behavior domain result in different relative proportions but the remaining common cause error modes show a strong trend for comparison among the various proportional means.

C-C Error Identification Mode



C-C Error Pattern Recognition Mode



C-C Error Behavior Domain Mode

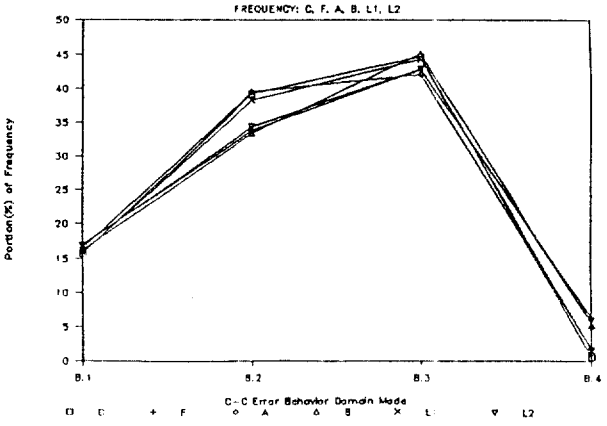
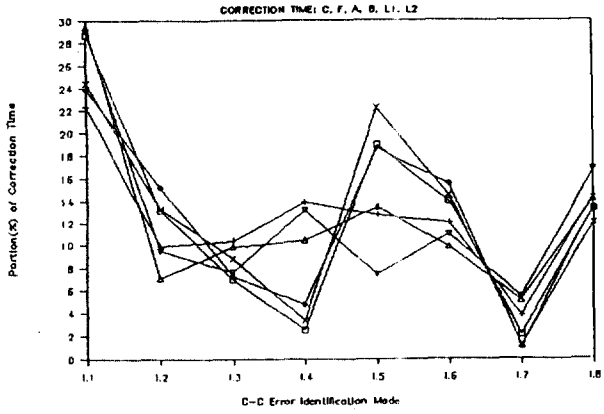
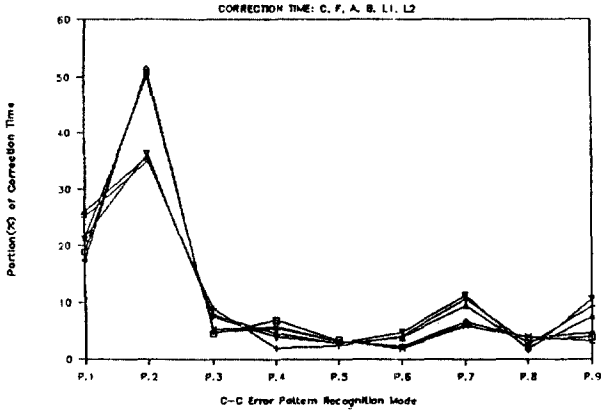


Figure 2. Portion of Frequency in each Common-Cause Error Mode

C-C Error Identification Mode



C-C Error Pattern Recognition Mode



C-C Error Behavior Domain Mode

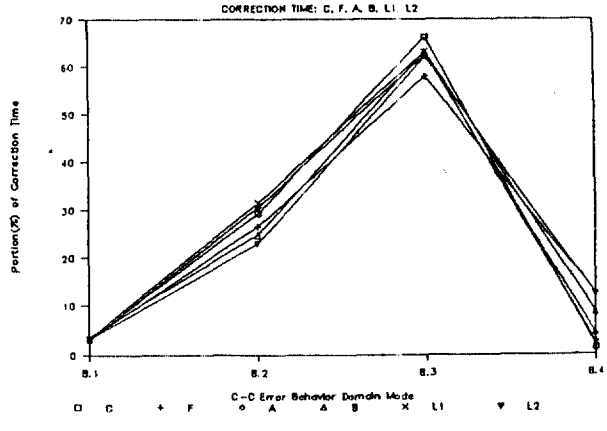
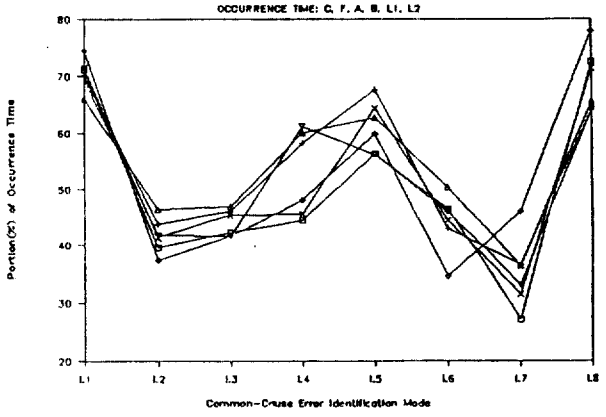
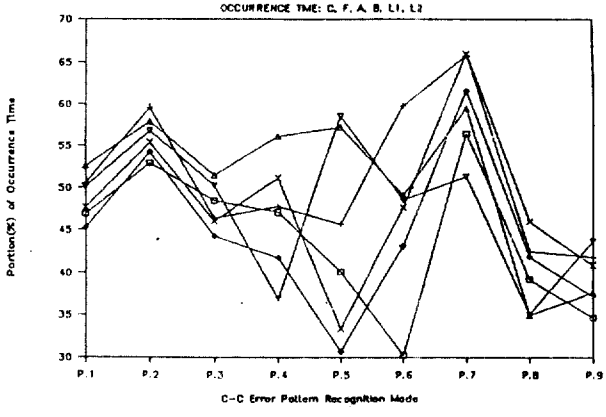


Figure 3. Portion of Correction Time in each Common-Cause Error Mode

Common-Cause Error Identification Mode



C-C Error Pattern Recognition Mode



C-C Error Behavior Domain Mode

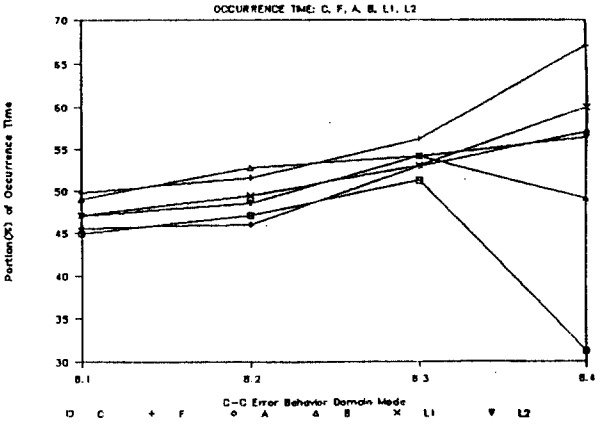


Figure 4. Portion of Occurrence in Time in each Common-Cause Error Mode

Table 2. Vector Evaluation with Rating Simulation

CCM[b]	-	-	-	V1[c]	V2	V3	V4	V5	V6	V7	V8
-	F[d]	C[e]	O[f]	1:1:1	1:2:3	2:1:3	3:2:1	1:3:2	3:1:2	2:3:1	2:2:1
I.1	15.9	26.2	70.7	77.1	219.1	216.1	100.1	162.6	151.5	110.4	93.6
I.2	16.2	11.9	41.6	46.2	128.1	129.5	68.3	92.0	97.0	63.7	57.9
I.3	19.4	8.4	44.1	48.9	134.8	138.1	74.9	93.8	106.0	63.9	61.1
I.4	8.8	7.1	50.9	52.1	153.6	153.9	59.1	104.4	105.4	57.9	55.7
I.5	11.2	16.6	61.3	64.5	187.2	186.0	77.4	132.8	128.2	82.1	73.2
I.6	10.3	13.3	45.0	48.0	138.0	137.0	60.7	99.0	96.1	63.6	56.2
I.7	7.0	2.7	31.2	32.1	94.0	94.7	38.0	63.3	65.9	35.1	34.6
I.8	11.2	13.8	68.7	71.0	208.2	207.8	81.3	143.9	142.1	83.3	77.4
P.1	15.7	21.2	48.6	55.3	152.6	150.6	79.9	117.2	110.1	86.0	71.7
P.2	33.7	44.8	55.9	79.2	193.1	186.2	146.2	178.0	157.2	160.4	125.3
P.3	18.0	6.1	47.6	51.3	144.4	147.4	73.0	98.6	109.6	62.4	60.9
P.4	4.9	5.1	47.9	48.4	144.1	144.1	51.1	97.1	97.1	51.2	49.9
P.5	6.8	3.0	42.7	43.3	128.4	128.9	47.7	86.1	87.9	45.7	45.2
P.6	2.6	2.9	45.7	45.9	137.2	137.2	46.7	91.8	91.8	46.8	46.4
P.7	7.4	7.9	60.6	61.6	182.6	182.6	66.4	123.7	123.5	66.7	64.4
P.8	6.3	3.1	39.7	40.3	119.4	119.8	44.4	80.2	81.7	42.7	42.1
P.9	4.6	5.9	37.9	38.6	114.4	114.2	42.0	78.0	77.3	42.8	40.7
B.1	16.3	3.2	47.1	49.9	142.4	145.0	68.2	96.1	106.2	58.1	57.6
B.2	36.5	28.1	49.2	67.4	162.1	167.0	132.5	134.6	149.9	121.9	104.4
B.3	43.6	62.7	53.5	93.2	208.3	193.1	188.9	220.8	180.2	214.1	161.8
B.4	3.6	6.0	55.3	55.7	166.4	166.2	57.6	112.1	111.3	58.6	57.0

[a]Ii : identification of common-cause error mode, Pji : pattern recognition of common-cause error mode, Bk : behavior domain of common-cause error mode.

[b]CCM : common-cause error mode.

[c]V : rating weight for simulation.

Value of the common-cause function and simulated rating : Each value listed in the common-cause function parameters can be produced by three factors, F_i, J, K, C_i, J, K, O_i, J, K. The common-cause function can be simulated with different weighting of variables, rating as in Table 2. The final evaluation value of these common cause functions can be produced using the geometrical vector evaluation method. Each common-cause mode can be evaluated by the calculation of a geometrical vector value from the geometric origin(F,C,O) = (0,0,0).

By different emphasis or weighting on the evaluation factors, a different orientation stress for representing development cost or effort, frequency of error occurrence, error correction time, and point of error occurrence time, using evaluation by geometric vector can be calculated to produce varying shapes. Such simulation has shown trends of differences in identification modes among different ratings, the major reasoing common-cause error modes being I.1, I.8, and I.5. In pattern recognition mode, the same trend results with major reasoning patterns, P.2, P.7, and P.1 in simulation V1, V2, V3, V5, V6, but different order results with P.2, P1, P.7 in V7 and V8.

Correlation and regression analysis : From the correlation analysis result, a more experienced programmer had a better performance in the programming task with less frequent error, and less time taken in programming design and error correction. More spent time in the design phase resulted in a lower frequency of error occurrence during the computing phase. The hypothesis in this experiment, that there was no significant difference in common-cause error properties among different categorical conditions such as specification requiremants, programming languages, and subject expertise levels, was tested according to F statistics using the variance analysis test with SAS.

Using factorical experimental analysis, there is no significant difference in frequency as a dependent variable for three independent variables involving two levels for each variable. The probabilities values associated with the F value are 0.9770 for requirement, 0.3085 for expert level, and 0.9770 for programming language. No significant difference occurred for correction time as a dependent variable associated with the F value probabilities, 0.4285 for requirement, 0.2689 for expert level, and 0.7867 for language. There is only a significant difference in the expert level with computing time as a dependent variable. The F value probabilities are 0.3705 for requirement, 0.0400 for expert level, and 0.3617 for programming language.

For an estimate of linear regression equation of the straight line that best fits the points between two variables

involving frequency and correction time, the result has the coefficients and intercepts for the linear regression equation describing frequency as a dependent variable :

$F = (0.0391)C\_t + (22.1197)$,

where F : frequency, C_t : correction time(Units : time in min.), The linear regression line of design time as a dependent variable(correlation coefficient : 0.7019) is :

$D\_t = (0.5727)C\_t + (23.7929)$,

where D_t : design time, C_t : correction time(Units : time in min.). The acceptance confidence probabilities for the best fit regression line between the variables are significantly enough with t test statistics except design time.

General observations and some symptoms of common-cause error were discovered during the experiment.

(1)Per-existing knowledge was a major diagnostic symptom for completing the programming task.

(2)Human memory, recognition, and availability were associted with some effects in the rule-based behavior domain.

(3)Human attention and perceptual ability can be affected by subject sensory-motor variability, recent physical and psychological events, and external environments.

(4)Incomplete of knowledge was a major common-cause in the area of system operation, programming language, design method, and requirements specification.

(5)Uncertainty of information was associated with knowledge-based, model-based, and skill-based behavior domain groups.

(6)There was no significant difference in common-cause error properties between the two levels among three of the subject factors : two programming language, two requirements specification, and two programming expertise level, the exception bein computing time in level of expertise.

(7)The major common-cause error modes arose form system design and requirement error, output and output formatting error, and program logic error.

(8)The knowledge-based behavior error domain was associated with the most significant error mode group in each of the common-cause function factors which involved identification and pattern recognition error modes.

(9)Frequency and correction time in each common-cause error mode have a more consistent trend than point of occurrence in time among different error modes and over different task criteria.

V. Conclusions

Conclusions derived form this research are as follows. First, two major common cause reasoning groups exist in human-software interaction : (a) a major group consisting of knowledge-based behavior related errors indicated by design and knowledge deficiencies ; (b) another major group consisting of rule-based behavior related error indicated by logical errors, functional deficiencies, and system complexity. Second, in training education sessions, consideration should be given to common-cause reasoning characteristics to eliminate the common-cause human error in human software interaction. These characteristics include : (a) human mind robustness(pre-existing incorrect knowledge and information) ; (b) pattern recognition in human memory ; (c) human attention and perceptual ability ; (d) incompleteness of knowledge and information uncertainty. Third, design with intelligence and concurrence by the knowledge-based processing : (a) konwledge acquisitions ; (b) knowledge representation ; (c) knowledge utilization.

The results and analytical procedures showed during the study were to analyze common-cause of software development related to human domain error and to identify software design factors contributing to common types of error occurring in human-software interaction.

Therefore, this can be applied to improving reliability of software development and to providing guidelines for design of software development.

References

1. Askren, W. B., T. L. Regulinski.,(1969) "Quantifying Human Performance for Reliability Analysis of Systems" Human Factors, 11, pp. 393~396.
2. Boehm, B. W., TRW., "Improving Software productivity"(1987), Computer, Sep. pp. 43~57.
3. Curtis, B.,(1981) "Human Factors in Software Development." IEEE, Cat. No. EHO 185~9.
4. Dhillon, B. S.,(1983) Reliability Engineering in Systems Design and Operation, Van Nestrand Reinhold Co., New York.
5. Endres, Albert.,(1975) "An Analysis of Errors and Their Causes in System Programs" IEEE Transactions on

Software Engineering, June, pp. 140~149.

6. Jones, T. C.,(1986) Programming Productivity, McGraw-Hill, New York.
7. Meister, D.,(1962) "The Problem of Human-initiated Failures." Proceedings of the 8th National Symposium on Reliability and Quality Control, IEEE, New York, pp. 234~239.
8. Mitta, Deborah.,(1991) "A Methodology for Quantifying Expert System Usability." Human Factors, 33(2), pp. 233~245.
9. Musa, J.D., A.Iannino, K. Okumoto.,(1987) Software Reliability Measurement, Prediction, Application. Mc Graw-Hill Com., New York,pp. 77~101.
10. Park, Peom.,(1992) "Common-Cause Analysis in Human-Software Interaction: System Design, Error Control Mechanism, and Prevention." Doctoral dissertation, Iowa State U, Ames, Iowa, U.S.A.
11. Peters, G.,(1966) "Human Error: Analysis and Control." Journal of the ASSE, Jan.
12. Rasmussen, J., K. Duncan, J. Leplat,(1987) "Cognitive Control and Human Error" New Technology and Human Error. John Wiley & Sons, p. 53~61.
13. Reason, James.,(1987) "Generic Error-Modelling System(GEMS): A Cognitive Framework for Locating Common Human Error Forms." New Technology and Human Error, Ed., by J. Rasmussen, K. Duncan and J. Leplat, John Wiley & Sons. Ltd pp. 63~83.
14. Watson, I. A.,(1981) "Review of Common Cause Failure." National Centre of Systems Reliability, Report NCSR R27.
15. Woods, D. D.,(1990) "Modeling and Predicting Human Error." Human Performance Models for Computer-Aided Enginerring, Ed. by J. I. Elkind, Academic Press, Inc.
16. Youngs, Edward A.,(1981) "Human Errors in Programming." Human Factors in Software Development: COMPSAC81, Ed. by Bill Curtis, L. A., C. A. pp. 383~392.