Improving Human Reliability On Checking

Toru Nakata
Research Institute of Secure System,
National Institute of Advanced Industrial Science and Technology (RISEC, AIST)
Umezono 1-1-1, Tsukuba, Ibaraki, Japan.
toru-nakata@aist.go.jp

ABSTRACT
Checks are fundamental operations to detect abnormalities and to ensure safety. Human reliability of checking is, however, insufficient to prevent issues being overlooked, resulting in many of the accidents that take place today. Even though people may attribute such instances of overlooking issues to a limitation in human attention, this paper concentrates on the reformation of checking procedures to improve reliability. The key concepts are: 1) 'staticization' of workflow, 2) independency triggering checks related to a job, and 3) objective questioning. These methods afford objectivity and stability of checking. The author states the mechanisms of these methods and then provides practical examples.

Author Keywords
Human Error; Check Methodology; Process Design; Staticization of Process.

ACM Classification Keywords
H.1.2 User/Machine Systems

INTRODUCTION

"It's a bad cook who can't lick his own fingers." — Shakespeare, *Romeo and Juliet.*

No accident can occur under proper scrutiny. Checks are therefore the most essential function for maintaining safety. The skill to detect errors could be said to be much more valuable than the skill associated with flawless job processing.

Human workers often undertake responsibility for checking. Although we employ automatic machines to conduct checks that are free from human errors, we still have to instruct the machines correctly by manual operations. Human errors are crucial in both automatic and manual checks.

Humans often make mistakes. According to the datasheet of human error probability complied by Gertman and Blackman [4], typical probabilities of general human errors are roughly estimated about 1% or so. In addition, performance becomes worse when the task becomes harder. People cannot complete their checks when the items being considered are numerous and when their layout is confusing. This means that human workers may no longer be equipped for undertaking checking tasks for the complex systems in modern industries. Gawande [3] emphasizes the importance of preparing and designing good checklists. He emphasizes the importance of a reduction in the number of checklist items. Gawande advises that checklists with more than ten items should not be used because of its complexity.

For these reasons, we should change strategy and concentrate on the method of checking rather than on improving human ability. A shortage in human performance can be overcome if a checking procedure that is robust and easy to understand is in place.

CONVENTIONAL STUDIES AND TECHNIQUES

We should assume that human operators generally commit a significant number of mistakes. Countermeasures for the unreliability of humans have been researched and developed for many years. Effort and implementations for this purpose can be summarized as follows:

Vulnerability of Human Checking

There are two major mistake patterns when checking: 1) omission and 2) subjective judgment.

1) People often overlook the necessity of checking. We may also skip some items on the checklist by mistake. Omission is a serious error pattern since it eliminates checks and leaves the situation unmonitored and uncontrolled. Remembering mandatory checks is basically easy when we use tools such as memos, checklists, or reminder software. Nevertheless, many accidents due to omitting checks do occur because we often forget (or are reluctant) to use the reminder tools from the beginning of the process.

2) Even though we succeed in remembering to undertake checking, we may still ultimately fail by becoming subjective. Our thought process is not completely logical, and it is influenced by our experiences. For example, we can read the sentence below [12]:

"I aulaclty uesdnatnrd waht I was rdanieg."

This phenomenon is called *typoglycemia*. Our cognitive process tends to find something meaningful in noise. In some cases, this tendency is harmful when conducting checking; we tend to assume that something which is nearly correct is in fact perfectly correct. Having prejudice that the situation is all right, we may overlook mistakes and fail to submit accurate reports. For efficient and effective checking, we therefore have to eliminate subjectivity from our consciousness.

Types of Human Error and Difficulty of Detection

Several conventional explanations exist concerning how people detect their mistakes. The probability of error detection largely depends on the degree of stableness of the operation.

According to Rasumussen's famous Skill-Rule-Knowledge (SRK) Model [11], all human behavior can be categorized into three levels, namely skill, rule, and knowledge. Skill refers to the ability to control movement that is hard to describe and is taught by language. Failures owing to inadequate dexterity are errors of skill. Rules are programmed directions concerning specific activities. Misunderstanding of rules produces errors of rule level. Knowledge is a set of unstructured information that may be useful for decisions. Shortage of knowledge hinders proper planning to solve complex problems.

The probability that human operators detect their own errors depends on the SRK level of their activities. Errors of knowledge are the hardest to detect. People who are confident as to the correctness of their thinking find it difficult to detect their mistakes. In an experiment by Rizzo [14], people were found to be more easily able to detect their errors of skill level than those of rule level. Allwood [1] reported a similar result that negative and suspicious attitudes towards results leads to a strict application of the verification process, and so improves the error detection rate.

Newell and Simon [10] state that people commit fewer mistakes when they execute an operation in a highly mechanical way. Through training, people will acquire an understanding of the step-by-step program of the activity, and they can follow that program almost automatically without fully comprehending the meaning behind it.

Summarizing the discussion above, it can be concluded that ad hoc operations with large degrees of freedom result in errors being harder to detect. However, such ad hoc operations are not suitable for machines to govern. Human operators are often given authority to manage those operations, despite the risk of human error.

Verification by Repetition

To detect errors, we may collate two or more results which are duplicated in the same process. This strategy has been employed from the 1940s with computers [19]. The specific machine for this purpose was called a "verification machine" and it accepted two sets of punched tapes or cards and detected the difference between them. Today, we are still employing the duplication method to secure important data entry like email addresses and passwords.

The advantage of this doubling strategy is universality and convenience. We can apply it for almost every variety of checking. Moreover, we do not need to know the correct answers beforehand because concordance of doubled results assures correctness.

Strictly speaking, there is a possibility that the same mistakes will be committed in the same process, even if a different human operator processes it. Notably, non-random mistake patterns like *typoglycemia* may be steadily reproduced in spite of a variety of operators. According to VanLehn's research on children's misconceptions while solving arithmetic problems [18], some stable types of errors are repeated with 34 % probability. Minton [8] reports that 10—15 % of errors in typed data occur despite double inputting for verification. So this verification method is neither perfect nor wholly reliable for practical use.

In addition, duplication requires both doubled cost and labor. Only when checking limited data of exceptional importance can the duplication (and even triplication) be justified [5]. In general, duplication is criticized for its repetition, low efficiency, and limited reliability.

Error Detection Based on Additional Information

Additional information concerning data may help humans to detect errors. In communication technology, the check-digit method is a basic tool to detect errors. However, human cognitive ability is not suitable to verify numerical check-digits.

For humans, the meaning behind data will support error detection. Newell and Simon [10] state that meaningfulness concerns error recovery. When the checker understands the meaning of the target, checking will be more reliable. On the other hand, the checking of abstract symbols is very confusing. Computer systems that accept user's inputs should display not only the typed inputs but also additional information to express meaning. For example, a display of "20004: ZIP code of Washington DC" is preferable when checking than simply the digits "20004".

Error Detection by Feedback

When an operator commits a mistake, the system should provide him with some informational feedback to warn of the error. Most modern computer systems accept human commands through dialogue [5] because dialogue is necessary and efficient for detecting errors in human commands.

The systems usually have some expectation regarding command input. For example, input to designate the month should be a number less than or equal to 12. If the user typed a number over 13, the system alerts the user immediately and refuses to accept the number.

Barfield [2] proposes that helpful feedback should provide the user with instructions to correct the operation. In the example above, the system should say "The number you have inputted is too large". A message like "Error code 19" is useless because it lacks instruction.

Most conventional studies assume that such feedback is required when the inputted commands seem abnormal and incorrect. In other words, warning or questioning messages are regarded as unnecessary for ordinary inputs. This paper reconsiders the use of questioning feedback.

Deterministic Estimation and Quantitative Estimation

In conventional studies, there are two means of estimating risk of human errors: the deterministic method and the quantitative method.

Human reliability, which is the probability of a human performing a certain function without failure, is hard to predict quantitatively. Since the abilities of people are different and variable over time, universal and accurate values of human reliability do not exist.

For the same reason, it is very hard to estimate performance of error detection techniques quantitatively. This paper will therefore not present quantitative evidence of the performance of each technique for error detection.

Instead of the quantitative method, many industrial standards ([6] and [7], for example) for human factor risk management employ deterministic risk estimation. Reproducibility of values assigned to human reliability can be observed in some cases. There exist tasks that almost everyone succeeds at and tasks that almost everyone fails at. In such cases, we can evaluate the risk with sufficient accuracy without numerical calculation.

Fig. 1 presents an example of reproducibility in human reliability. In the pattern on the left, it is very easy to find the letter 'O' among the other letters. In contrast, the task of finding the 'O' in the right-hand pattern is very difficult for everyone. The pattern in the center has an intermediate level of complexity and difficulty. Human performance in this case may diverge depending on personal ability and situation.

With adequate reasoning, it is permissible to approximate human reliability as 100 % for the easiest tasks. Probabilistic and quantitative estimation of the human error rate may therefore be unnecessary. In such cases, qualitative and deterministic assessments of human factors can be regarded as sufficient.

For situations of intermediate complexity, human reliability cannot be estimated by design considerations in the absence of concrete data on human performance. In case human reliability in such a situation is limited, compensation measures are required to offset human errors. Methods of quantitative and probabilistic risk assessment should be employed in such situations.

Fig. 2 illustrates the criterion for selection between deterministic (qualitative) assessment and probabilistic (quantitative) assessment. The knee point of the human reliability curve may indicate the threshold for acceptable deterministic assessment.

This paper employs the deterministic method to evaluate techniques for human error detection, since most of the discussion stems from the deterministic area. Instead of presenting numerical data, the author offers practical examples of error detection.

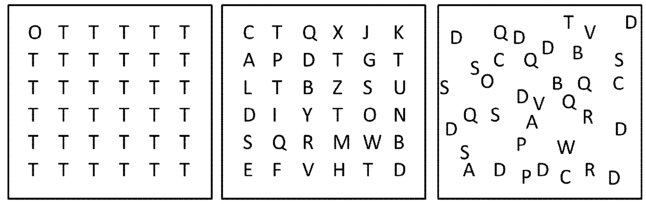

Figure 1. Test images for visual searches of differing complexity.

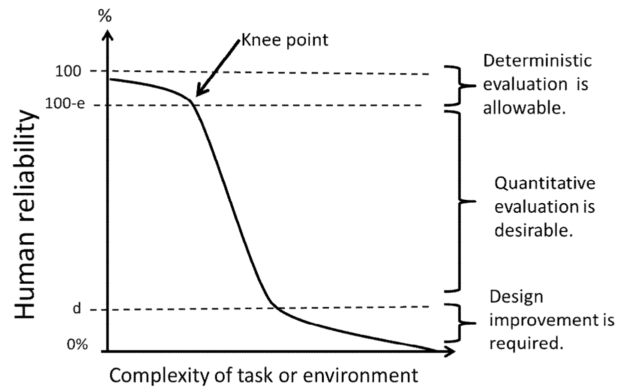

Figure 2. Change in human reliability according to the complexity of task or environment.

IMPROVEMENT OF THE CHECKING PROCESS AGAINST OMMISION

Static Process and Its Merits

In general, processes can be classified into two categories: *static* and *dynamic.* Static processes are free from dynamical constraints. In other words, even though a static process is stopped in a particular moment, it can restart again without any damage. In contrast, dynamic processes will change their behavior if they are interrupted even once. For instance, our walking behavior is a typical dynamic process. We would fall down if we stopped our body while walking.

Now consider industrial processes and business workflow processes. Static processes in such fields allow standstills at any time. During the intermission, we can carry out checking or take time-outs to have meetings or to rest. Intermissions are, therefore, very important in keeping the processes safe and ensuring that personnel are calm and well-informed.

Staticization of Dynamic Processes

Without particular consideration, we tend to design workflow processes to have unnecessary constraints related to timing and velocity and that therefore become dynamic.

Imagine two workers are fixing a beam up between walls as shown in Fig. 3. Each of them has to hold the beam during the process; otherwise, the beam falls down. Applying pillars to support the beam, the workers become free from that constraint (Fig. 4). They can take an intermission at any time, and they need no longer work simultaneously. Thus the process can be reformed as static.

In this example, the reason for dynamicity was concentrating tasks on the workers. The workers had to handle the beam for two purposes: holding and fixing. The supporting pillars separate those two tasks.

Breaking down task concentration is one of the key methods by which to render a process static. Badly planned workflows often involve a concentration of tasks. Consequently, they cannot become static. In general design theory [16, 17], tasks with two or more purposes are considered to be faults. Unnecessary concentrations should be eliminated by reviewing the designs to make the processes static.

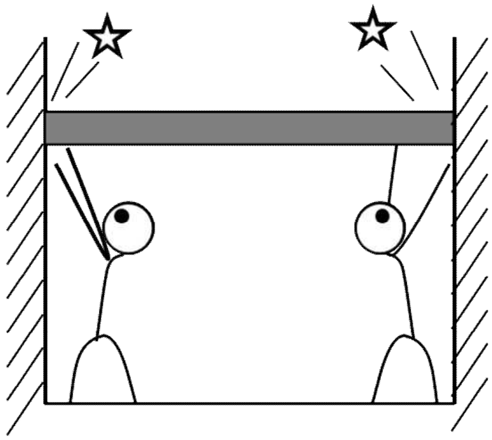

Figure 3. Mounting a beam without supporting pillars is a dynamic process (i.e. not interruptible).

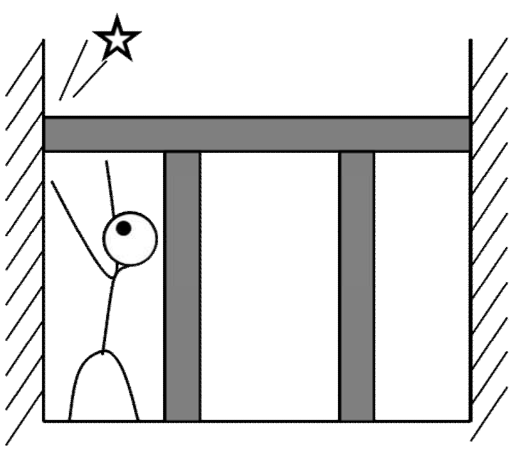

Figure 4. Supporting pillars make the process static, easy, and safe.

Dynamicity Evokes Human Errors on Checking

In general, work consists of several parallel workflows for subordinate tasks. Fig. 5 provides an example of a job process with three streams of sub-tasks. To supervise the process, it seems normal to insert checks for each stream like those shown in Fig. 5. This scheme, however, has two serious weaknesses.

The first problem concerns appointing the person responsible for the checks. People can conduct very strict checks when they verify the results made by others. In contrast, self-checking is far less reliable. Therefore, the role of checker should be assigned to someone who does not participate in the work. The scheme in Fig. 5 is not suitable for such assignment. When each of the streams is moving forward independently, it is rather troublesome for a worker on a certain stream to switch to another stream.

The easiest way to overcome this issue is to ask workers to check the stream that they are working on, because switching of personnel is therefore not needed. Thus problematic self-checking is implemented.

The second problem is that those parallel schemes tend to lead to checks being omitted. In Fig. 5, each check is triggered by the end of the preceding action. If the action is forgotten, the check is also omitted. If the check itself is not monitored by other checks, there is no longer a chance to notice its omission. We must employ a check on checks, but such a check on checks will require another check to observe itself.

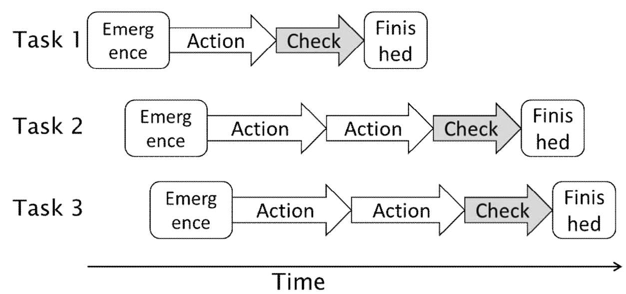

Figure 5. Checks belonging Actions.

Triggering Checks Independent from Preceding Job

To solve these two problems, we should reform the triggering of check steps. A typical solution involves executing checks over all streams simultaneously (Fig. 6). In other words, a kind of “stage-gate schedule” planning, which settles check gates across all streams.

At the check gate, all workers are released from their jobs, so we can assign the workers to check each other’s work streams. This exchange makes the checking process more objective.

In order to avoid omissions, the cues for starting checks should be independent from the end of a particular preceding job. Instead, we should stop every job stream at the check gates and wait for other streams to reach the same gate.

We can find many examples of independent and simultaneous check triggering in traditional situations. The manner of a tea ceremony in Japan involves many stage gates in the process. The positions and orientations of the tools are strictly designated for each stage gate, and the tea master processes the ceremony with frequent checking of tool positions.

Time-driven triggering is another useful way of cueing checks.

To offer an example of this method: In a Japanese food industry factory, the workers were required to conduct too many checks in a day, checks that they sometimes forgot to do. In general, food processing requires frequent checks. For example, some points have to be checked once every three hours, and other checks have carried out in a different cycle.

The workers were not aware of the problem of check triggering. They were working under a conventional workflow design like in Fig. 5. There were few tools to remind the workers of necessary checks.

Advised to adopt the strategy of check gates, they changed the way of triggering: now the checks are started on the hour. Such a time-driven reminder involves less risk of forgetting something important.

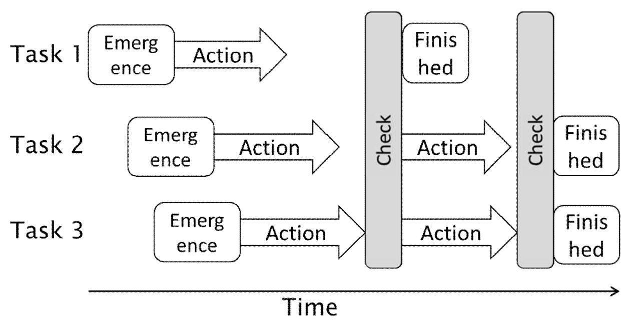

Figure 6. Triggering checks with independence from actions.

Final Check Gate to Move Away from Hazards

The last check gate has a special role in addition to the functions stated above. It also serves to guide workers from dangerous locations to safe areas.

There is a tendency for major accidents to occur at the end of a job. Realizing that they are nearing the end of their job, workers often relax their attention. They are standing at the position where final jobs are conducted.

The last working position, however, is rather hazardous because it is still inside the workplace. The machine that the workers took care with is near, so they should not relax their attention until they have moved to a safe place.

We can guide the workers to move to safe places by designing the last check gate well. The last check gate should demand that workers stand in designated positions. These positions should be outside of the workplace, so that the workers become remote from hazards.

In most Japanese airline companies, ground crews have to trail aircrafts to give the customary final salute to the passengers (Fig. 7). Their substantial tasks are already finished, so they could retreat from the job like other airline companies worldwide. However, they instead move alongside the airplane and retreat to a safe distance. They form a row and watch the airplane again, so that they have an additional chance to detect abnormality. Thus, this well-designed last check gate ensures safety for the workers and reliability of checks.

Figure 7. A trailer team forms a row and offers a salute to the passengers at the end of their job.

Comparison between Staticization and Time-Driven Methods

There are particular advantages and disadvantages associated with the staticization method and with the time-driven method (Table 1).

In general, the staticization method has greater advantages in terms of increasing the accuracy of checking. The time-driven method should be mainly used to prevent omission of checking. By combining both methods, a certain element of the shortcomings will be mitigated.

	Staticization	Time-driven
Ease of checking	Checking becomes more accurate owing to stand-still of situation.	It interrupts the workflow and may confuse it.
Inhibiting omission	It involves a certain risk of forgetting and omission.	Robust against forgetting.
Subjectivity of checker	It enables rotation of operators.	N/A
Time efficiency	It delays the workflow due to waiting.	It delays the workflow due to interruption.

Table 1. Advantages and disadvantages of the Staticization and Time-driven methods.

IMPROVEMENT IN OBJECTIVITY OF CHECKS

Problem of Yes Bias in Ordinary Checks

Comprehending the situation and making logical judgments can be hard for humans. Even when processing checks that are guided by written checklists, many checking failures are still reported [9].

This paper proposes three patterns of questioning for checks (Table 2).

Questioning Type	Merit	Demerit
Yes/No Interrogation	Simple. Process quickly.	Risk of answering 'yes' mistakenly.
Reporting Unpredictable Things	Checker will look at the target carefully.	More time consuming.
Conversational Pair Checking	Very high accuracy.	More time consuming. Training of role play is required.

Table 2. Three patterns of questioning.

First, an ordinary and straightforward form of checklist is an array of yes/no interrogative sentences. The checkers only have to fill out the check boxes, so the checking can be conducted quickly, without ambiguity and deep consideration.

This simplest method of yes/no checking, however, poses a serious disagreement to human psychology. People tend to answer "yes" even for items that are not right. This phenomenon stems from an unbalance of yes/no distribution. In industrial or business settings, there are usually many more correct things than incorrect things. Checkers therefore might answer "yes" many times. Such biased distribution evokes the preconception that most of the answers will be yes, so we easily overlook the checklist.

Avoiding Yes Bias by Reporting Unpredictable Things

To prevent Yes Bias, we should improve the rule of reporting. We ask that checkers report not only yes/no but also provide more detailed and unpredictable information concerning the targets.

For example, a yes/no question like *"Is the light on?"* can be transformed as *"How much lux does the light emit?"* or *"When was the light turned on?"*

When focused solely on preconceptions, we cannot answer about changeable things or numerical data only based on guessing. In this case, we have to watch the targets more carefully to fill up the check list.

This method requires more time and cognitive effort than simple yes/no checking. We should select either of the methods according to the required level of reliability and speed.

Conversational Pair Checks against Stubborn Preconceptions

In spite of efforts to avoid faulty preconceptions, there still exists a certain possibility of overlooking incorrect things, especially when the checking is conducted by only one person.

Checking by a pair of workers will improve reliability, but mere repeats of the same checking twice do not increase trustworthiness. We have to assign a different role to each of the pair.

The first person, the checker, fills up the checklist as normal. The second person, the observer, stands by the checker, reviews the checking results, and asks some skeptical questions of the checker, like in the following example.

Checklist: *"When was the light turned on?"*

Checker: *"Nine o'clock."*

Observer: *"Why isn't it eight o'clock?"*

The checker now has to state the logical grounds for his judgment. Imagine that the rule requires turning the light on before eight, but the checker has misunderstood the timing as being before nine o'clock. Such a preoccupation cannot be solved by the checker himself. The observer provides chances to notice the error by asking questions.

This conversational pair checking method is the most rigorous method by which to conduct checks. Although this method requires a rather high personnel and time cost, we should choose it for the most critical jobs. The important thing is to select the proper method whilst remembering the tradeoff between reliability and cost.

Case Study on Checking by Teamwork

There is no guarantee that the reliability of checking is in proportion to the number of checkers. In some cases, teams with too many operators fail severely since, as the proverb says, "Too many cook spoil the broth".

The disastrous collision between two large airplanes at Tenerife airport in 1977 was caused by a failure of communication and cooperation among the crews (Fig. 8). In spite of the presence of another airplane on the runway, the captain and co-pilots were convinced that the runway was clear, and they hastened to take off. Communication among the crew had received little criticism, so they did not doubt their optimism. In this case, the captain had the highest authority in the crew to decide everything. This kind of top-down power structure is often commonplace but is not appropriate for ensuring that checks are objective.

In industrial history, we can find good examples of teamwork, which involve intentional allocation of the criticism role. In the cockpit of a stream locomotive, for example, there are two operators: a driver (formally called "driving engineer") and an assistant ("stoker"). During the procedure of receiving a token for entering railway section, the driver does not communicate with station officers directly. Instead of the driver, the assistant accepts the token and asks a question of the driver about the shape of hole perforated in the token. The driver has to answer based on his time schedule chart without seeing the token. This questioning verifies the concordance between the thinking in the cockpit and that at the station. The allocation of the criticism role to the assistant is intentionally designed for safety.

As shown above, the number of checkers is not the decisive factor for accuracy of checking. Objectivity based on criticism instead controls the soundness of checking (Table 3).

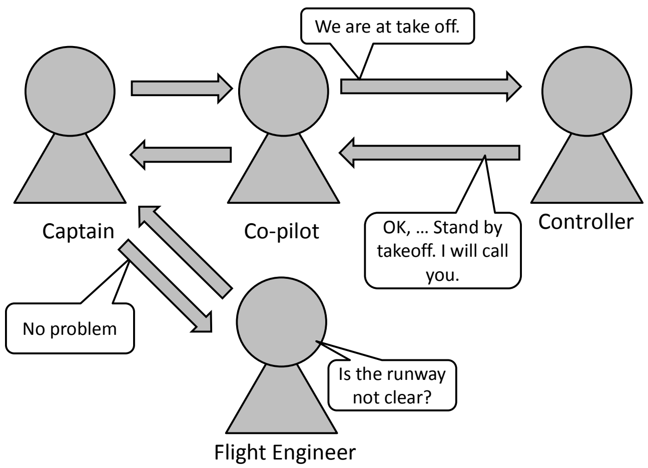

Figure 8. Problematic communication occurred in Tenerife airport disaster 1977.

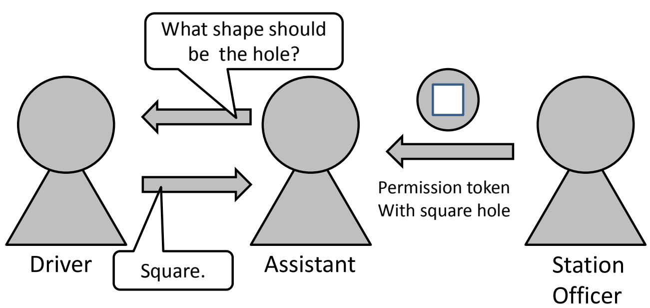

Figure 9. Safer checking by teamwork with criticism.

Checker Team Organization	Reliability
Team without criticism maker	Double-check does not work when the leader is assertive.
Verify by mere repetition	A certain portion of the errors cannot be removed [8].
Team with proper role assignment	It produces more chances to verify the results.

Table 3. Reliability of checking by teamwork.

CONCLUSION

This paper argued the two origins of human errors concerning checking. One is dynamicity of workflows, and the other is subjectivity of checkers induced by improper questioning. Dynamicity and subjectivity are very common when job processes are designed without particular consideration.

The solutions identified in this paper, namely process staticization, independently triggering checks, and objective questioning, will improve the trustworthiness of checking.

In future work, automatic planning of checks should be studied. In general, workflows are described as Petri net diagrams. It seems possible to build algorithms that detect the vulnerabilities involved in checks and other human errors by analyzing the job diagrams with respect to process dynamicity and checking style.

REFERENCES

1. Allwood, C. M., "Error Detection Processes in Statistical Problem Solving," *Cognitive Science, 8,* pp.413-437, 1984.

2. Barfield, L., *The User Interface: Concepts and Design,* Addison-Wesley, 1993.

3. Gawande, A., *The Checklist Manifesto*, Picador (2011)

4. Gertman, D. I., Blackman, H. S., *Human Reliability and Safety Analysis Data Handbook,* Wiley-Interscience Publication (1994)

5. Gilb, T., Weinber, M., *Humanized Input: Techniques for Reliable Keyed Input,* Winthrop Pub., 1976.

6. International Organization for Standardization, *ISO11064—Ergonomic design of control centres,* 2013.

7. Ministry of Defense of United Kingdom, *Defense Standard 00-250: Human Factors for Designers of Systems: Part 4 HFI Method, Tools and Techniques,* 2008

8. Minton, G., "Inspection and correction error in data processing," *J. of American Statistical Association,* pp.1256-1275, 1969.

9. NASA Aviation Safety reporting System, *Callback (Issue 410)*, NASA (2014).

10. Newell, A., Simon, H. A., *Human Problem Solving*, Prentice-Hall, 1972.

11. Rasmussen, J., *Information Processing and Human-Machine Interaction,* Elsevier, 1986.

12. Rayner, K., White, S. J., Liversedge, S.P., Raeding Wrods With Jubmled Lettres, *Psychological Science 17*(3), pp.192-193 (2006)

13. Reason, J., *Human Error,* Cambridge Univ. Press, 1990.

14. Rizzo, A., Bagnara, S., Visciola, M., "Human error detection processes," *Intl. J. of Man-Machine Studies, 27,* pp.555-570, 1987.

15. Stanton, N. A., et al, *Human Factors in the Design and Evaluation of Central Control Room Operations,* CRC Press, 2010.

16. Suh, N. P., *Axiomatic Design,* Oxford Univ. Press (2001)

17. Van der Aalst, W., Van Hee, K., *Workflow Management —Models, Methods, and Systems,* MIT Press (2004)

18. Van Lehn, K., *Mind Bugs: The Origins of Procedural Misconceptions,* MIT Press, 1990.

19. Von Neumann, J., *Theory of Self-reproducing Automata,* Univ. Illinois Press, 1966.