

시험단계에서의 철도부품 고장 데이터 수집 및 분석 체계 지원을 위한 S/W개발

김시연*, 김종운
네모시스 주식회사

Development of Failure Reporting, Analysis, and Corrective Action System Support S/W for Railway Components in Test Stage

See Yeon Kim*, Jong Woon Kim
Nemosys Corporation

요약 본 논문에서는 시험단계에서 철도부품 시험 시 발생한 데이터를 기록하고 관리할 수 있는 FRACAS(Failure reporting, analysis and corrective action system) S/W에 관한 개발과정을 설명한다. FRACAS는 주로 운영단계에서 활용된다. FRACAS와 관련된 대부분의 연구 및 S/W는 운영자가 운영단계에서 사용하며 운영단계의 고장 및 신인성 관리에 초점이 맞추어져 있다. 하지만 양산 이전 단계인 시험단계에서 시정조치를 도출하여 적용할 경우 비용, 시간 측면의 효율성을 올릴 수 있다. 따라서 운영단계가 아닌 시험단계에서 활용할 수 있는 FRACAS S/W 개발에 관한 연구를 수행하였다. 해당논문에서 수록된 FRACAS S/W는 유사 S/W의 기능을 분석하여 사용자가 시험단계에서 사용할 수 있도록 구성하였으며 도출한 기능을 토대로 구현하였다. 시험단계에서 사용하는 S/W이기 때문에 어떤 시험/검사를 하는지 정의할 수 있는 기능 또한 추가되었다. 해당 S/W는 사용자가 단계에 따라 시험 내용 및 시험 결과를 입력할 수 있도록 구현하였으며, S/W를 통해 관리한 데이터는 추후 새로운 시스템 개발 시 활용하여 신뢰성 향상에 도움이 될 것으로 판단된다.

Abstract This paper describes the development process for FRACAS (failure reporting, analysis, and corrective action systems) S/W that can record and manage data generated in the test stage. FRACAS is mainly utilized in the operational stage. Most of the studies related to FRACAS are used by operators in the operation stage and focus on failure and reliability management in that stage. However, if corrective measures are derived and applied in the test stage, which is a pre-production stage, trial and error can be reduced, and cost and time efficiency can be increased. Therefore, a study on the development of FRACAS S/W that can be used in the test stage rather than the operation stage was conducted. The FRACAS S/W analyzed the functions of similar S/W, configured them to be used in the test stage, and implemented them based on the derived functions. Since the S/W is used in the test stage, a function to define which tests/inspections are performed was also added. The S/W was implemented so that a user can input test contents and test results according to stages, and the data managed through S/W can be used to improve reliability in the future when developing a new system.

Keywords : FRACAS, Railway Components, System Development, Test Sage, Data Management

본 논문은 국토교통과학기술진흥원의 지원으로 수행되었음. (과제번호 22RTRP-B148873-05)

*Corresponding Author : See-Yeon Kim(Nemosys Corp.)

email: sykim@nemosys.kr

Received January 5, 2023

Accepted March 3, 2023

Revised March 2, 2023

Published March 31, 2023

1. 서론

시스템 개발 시 개발될 제품이나 서비스에서 발생할 수 있는 유사 고장 재발을 방지하고 신뢰도 성장을 향상시키기 위해서는 데이터 수집 및 분석을 통해 얻은 정보가 중요하다. FRACAS(Failure reporting, analysis and corrective action system)는 주로 운영단계에서 고장 데이터를 수집하고, 고장원인을 결정하는 절차를 제공하며, 수행한 시정조치를 문서화하는 프로세스이다. FRACAS는 신뢰도 성장을 향상시키는데 필요한 활동 중 하나이며 FRACAS 과정 중 하나인 고장 보고 및 분석은 제품의 신뢰성과 정비성을 달성하고 유지하는 데 필요하다. FRACAS는 고장을 정확하게 보고하고 분석하며 재발을 줄이거나 방지하기 위해 시정조치를 적시에 취하는 조항을 포함해야 한다. 또한 FRACAS는 설계 또는 제조와 관련된 고장의 기본원인을 결정하고 시정조치를 이행하는 closed-loop 방법을 제공한다[1].

FRACAS에 관한 기존 연구 사례는 다음과 같다. 국방 분야에서는 군 무기체계의 운용/유지를 위하여 국방 분야에 대한 FRACAS 모델을 구축하였으며 이를 바탕으로 군 무기체계의 수명주기 관리에 최적화된 FRACAS 워크플로우와 실제 적용방안을 연구하였다[2]. 또한 한국의 무기체계와 운용 및 정비 환경에 적합하게 구축된 FRACAS를 연구 및 개발하는 연구 또한 수행되었다[3]. 국내 방산분야에서 수행된 FRACAS 연구의 대부분은 운용/유지 단계에만 집중되어 있기 때문에 국방 무기체계의 전 수명주기에 걸쳐 FRACAS를 효과적으로 적용하기 위한 연구 또한 수행되었다[4]. 철도분야에서는 납품한 차량의 보증 기간동안 고장, 문제와 관련된 시정조치에 대한 기록을 제공하며 수명 주기 비용에 입력할 수 있는 고장 수를 예측할 수 있는 FRACAS 시스템을 연구하였다[5]. 또한 철도 운영법의 개정에 따라 차량 제작사가 직접 신뢰도를 검증하는 규정이 요구되면서 차량의 신뢰성에 대한 검증 요구가 높아져 차량 제작사와 철도 운영사의 신뢰성 검증 수행을 지원하기 위한 국내 도시 철도 차량을 위한 FRACAS 솔루션을 개발하는 연구를 수행하였다[6]. 기존에 사용자가 사용할 수 있는 FRACAS 시스템에서는 프로세스가 구축당시 설계한 내용으로 고정되어 있어 수정 또는 추가적인 프로세스의 개발이나 적용이 불가하기 때문에 업무 프로세스 관리도구인 워크플로우 관리 시스템 기반의 FRACAS 프로세스 개발 방법론을 제시하기도 하였다[7]. FRACAS와 관련된 국내 연구는 주로 국방 분야에서 이뤄지며 연구의 대부분이 운영자

관점에서 시스템의 운영/유지보수 단계의 고장 및 신인성 관리에 초점이 맞춰져 있음을 알 수 있다.

운영단계의 경우 고장 데이터를 시스템 사용자에게서 받기까지 그 기간을 예상할 수 없으나, 시험단계에서 고장 발생 시 고장 데이터를 설계 담당자가 확인이 가능하며 확인 후 고장분석 및 시정조치 도출까지 빠른시일내로 가능하다. 따라서 시험단계에서 고장데이터를 받아 분석하기까지 시간측면에서 효율적이며, 개발단계에서 설계 변경이 이뤄질 경우 운영 시 발생할 수 있는 고장이 줄어들기 때문에 비용측면에서도 효율적이기 때문에 중요하다. 그러나 현재 사용자가 사용할 수 있는 FRACAS S/W는 운영단계에서의 시스템의 데이터 수집 및 분석 절차만을 지원한다.

따라서 본 논문에는 시스템의 운영단계가 아닌 시스템의 시험단계에서의 시험 시 발생하는 데이터 수집 및 고장 분석을 지원하는 FRACAS S/W 개발과정 및 결과를 수록하였다. 본 논문은 2장에서는 FRACAS의 수행 절차에 관해 설명하고, 3장에서는 FRACAS S/W 개발과정에 대해서 설명하며 4장에서는 결론을 제시하였다.

2. FRACAS 개요

FRACAS는 장비에서 결함을 식별하고 해결하여 해당 결함의 재발을 방지하기 위해 개발된 관리 도구이다. FRACAS는 제조, 검사, 시험 및 운영 중에 발생한 고장의 체계적 보고와 분석에 기초한다. FRACAS의 closed-loop 특징은 고장 분석 중 얻은 정보를 프로그램에 속한 모든 의사 결정 엔지니어와 관리자에게 제공한다.

FRACAS 수행 절차별 설명은 다음과 같다.

- 1) 고장 관측: 아이템의 가동이나 시험에서 고장 관측
- 2) 고장 기록: 최소한 다음의 사항을 포함하여 관측된 고장 기록
 - ① 고장 부위/위치
 - ② 고장일시
 - ③ 고장 난 부품의 부품번호
 - ④ 고장 난 시스템의 일련번호
 - ⑤ 고장 난 시스템의 모델번호
 - ⑥ 관측된 고장 현상(symptom)
 - ⑦ 고장 발견자 이름
 - ⑧ 고장 발생 당시의 모든 중요한 환경 여건
- 3) 고장 검증: 고장 검증 수행
- 4) 고장 분리: 시스템 내부에서 고장으로 의심되는 가

- 장 하위수준의 부품을 분리
- 5) 고장 의심 아이템 교체: 분리한 부품을 정상 부품으로 교체한 후, 시스템을 재시험하여 최초에 보고된 고장이 수리되었음을 확인
- 6) 고장 의심 아이템 교체: 고장으로 의심되어 분리한 부품을 제품 수준 또는 그 하위수준에서 시험하여 의심되는 부품이 고장임을 확인
- 7) 고장 분석: 관측된 고장의 원인이 된 내부고장 메커니즘을 찾기 위해 고장 난 부품을 대상으로 고장 분석
- 8) 데이터 검색: 동일 또는 유관 부품에서의 유사한 고장 발생을 찾기 위해 기존 데이터를 검색
- 9) 고장의 근본 원인 확인: 7단계 및 8단계에서 도출된 데이터를 사용하여 관측된 고장의 선행원인 또는 근본 원인을 결정
- 10) 시정조치 결정: 고장 재발 방지를 위해 필요한 시정조치, 설계, 공정, 절차변경 등을 결정. 시정조치의 결정은 여러 관련분야가 참여하는 설계팀에 의해 이루어져야 함
- 11) 시정조치 적용: 애초의 시험 시스템에 대해 제안된 시정조치를 적용

- 12) 성능 확인 시험: 시정조치를 적용한 시스템을 재 시험
- 13) 시정조치의 효과 확인: 모든 데이터 검토와 재시험을 마친 후 제안된 시정조치의 효과를 확인
- 14) 시정조치를 모든 아이템에 적용: 제안된 시정조치의 효과가 확인된 후 인도할 모든 시스템에 대해 시정조치 적용

3. FRACAS S/W 개발

3.1 FRACAS S/W 기능 분석 및 정의

현재 시중에 판매가 되고있는 FRACAS S/W의 기능을 분석하였다. 현재 판매가 되고있는 FRACAS S/W는 Favoweb FRACAS, Relyence FRACAS, XFRACAS 등이 있다. 시스템의 시험단계에서 적용할 수 있는 FRACAS S/W 개발에 앞서 세가지 FRACAS S/W의 기능을 분석하였다. 기능별 설명은 아래의 Table 1과 같다. 각 S/W에서 제공하고 있는 기능은 아래의 Table 2과 같다.

Table 1. Feature description

Feature	Description
Configurable	<ul style="list-style-type: none"> • Adjust built-in fields and preferences • Create a new custom field • Set permissions to control what features a user or group can access
Personal notification	<ul style="list-style-type: none"> • Displays tasks that need to be completed, faults that need to be analyzed, problems or projects, and frequently used reports by the corresponding user
Incident (failure) reporting	<ul style="list-style-type: none"> • Report and troubleshoot the issues from multiple locations, suppliers, dealers • Capture findings from the failure analysis on returned parts
Failure analysis history (for replaced parts)	<ul style="list-style-type: none"> • Capture the detailed findings for parts that were removed/replaced and returned for failure analysis, such as the item number, sales order number or work order number, findings from the visual inspection and detailed analysis, fault history or sensor data, or other test results.
Part tracking for the system	<ul style="list-style-type: none"> • Track the details of the systems, such as the system installation information, related events, operating time or usage, system configuration, current state of the system (running, waiting for parts).
Assigned job completion tracking	<ul style="list-style-type: none"> • Assign the tasks to users • Track progress and completion
Root cause analysis	<ul style="list-style-type: none"> • Provides file attachment feature • Register root cause analysis results • Assign persons for specific tasks and track completion • Review and follow up approval by the Fault Review Board (FRB)
Incidents or problem status tracking	<ul style="list-style-type: none"> • Show the number of open/closed issues, the status of the corrective action process on one screen
Queries and reports	<ul style="list-style-type: none"> • Show the customizable report tables
Dashboard	<ul style="list-style-type: none"> • Show the customizable dashboard and charts
Integration with other SW	<ul style="list-style-type: none"> • Integrate the collected data for further analysis (e.g., FMEA, Weibull analysis)
XML import and export	<ul style="list-style-type: none"> • Data import and export

Table 2. Comparison of FRACAS S/W

FRACAS S/W \ Feature	FavoWeb FRACAS	Relyence FRACAS	XFRACAS
Configurable	○	○	○
Personal notification	×	×	○
Incident (failure) reporting	○	○	○
Failure analysis history (for replaced parts)	×	×	○
Part tracking for the system	×	×	○
Assigned job completion tracking	○	○	○
Root cause analysis	○	○	○
Incidents or problem status tracking	○	○	○
Queries and reports	○	○	○
Dashboard	○	○	○
Integration with other SW	○	○	○
XML import and export	○	○	○

세 가지 FRACAS S/W를 비교하였을 때 Table 1에서 정의한 12가지 기능 중 9가지 기능을 공통적으로 제공하고 있다. 세 가지 FRACAS S/W에서 제공하고 있는 공통된 기능 9가지를 토대로 본 연구에서 제시하는 FRACAS S/W는 아래의 Table 3과 같은 기능을 제공할 수 있도록 구성하였다.

3.2 FRACAS S/W 개발 및 구현

‘3.1 FRACAS S/W 기능 분석 및 정의’에서 정의한

기능을 토대로 시험단계에서 사용할 수 있는 DCACAS S/W의 상세기능을 정의하였으며 아래 Table 4 와 같다.

기존의 세 가지 FRACAS S/W에서 공통적으로 제공하는 기능 외에 시험단계에서 사용할 수 있도록 시험 정의 및 시험 결과 관리 기능을 추가로 정의하였다. 정의한 기능을 토대로 FRACAS S/W를 구현하였으며 시험단계에서 사용하는 S/W이기 때문에 주 사용자를 설계 담당자와 시험 담당자로 하여 구현하였다. 아래의 Fig. 1은 DCACAS S/W의 데이터 흐름을 나타내었다.

Table 3. Features provided by FRACAS S/W

Feature	Description
PBS configuration	Register the PBS (Physical Breakdown Structure) of the corresponding system under test
Failure code management	Register and manage the failure codes based on the user specification
Dashboard	Inquire the failure records and problem analysis status
Failure registration	Register the failure information
Customization	Set the user assignment and menu management

Table 4. Feature specification

Category	Feature Description
Project and test management	Register and manage the information related to the project, system and failures
Test information/definition	Register the test information which are performed in the test stage
Test result management	Register and manage the test result, failure, and action records
Problem and corrective action management	Register the problems encountered during test stage and provide the management of the corrective actions to resolve the corresponding problems
Status inquiry	Show the status of records entered by the users
User Management	Manage the new user registration and the role/permission setting for each user

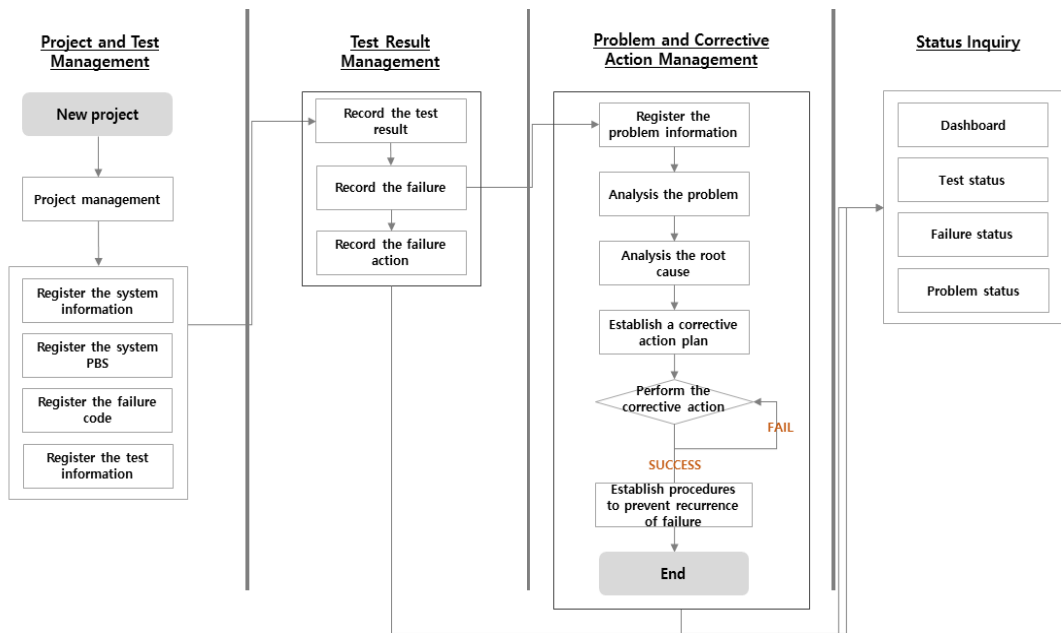


Fig. 1. Data flowchart in FRACAS procedure

3.2.1 단위 프로젝트 및 기본정보 관리 (Project and information management)

프로젝트 및 기본정보 관리는 설계 담당자가 시험 전에 입력을 완료해야 하는 메뉴이며, S/W로 관리하고자 하는 프로젝트와 관련된 정보를 등록할 수 있다. 프로젝트 및 기본정보 관리는 프로젝트 관리, 시스템 개체 등록, 시스템 PBS 등록, 고장 코드 등록, 시험 종류 및 항목 등록 화면으로 구성되어 있다.

- 1) 프로젝트 관리: FRACAS S/W로 관리하고자 하는 프로젝트를 등록 및 관리할 수 있으며 입력항목은 다음과 같다.
 - 프로젝트명: 관리하고자 하는 프로젝트 명칭
 - 시스템명: 관리하고자 하는 프로젝트에서 사용하는 시스템 명칭
 - 시작 일자/종료 일자: 프로젝트가 시작/종료되는 일자
- 2) 시스템 개체 등록: 프로젝트에서 사용되는 시스템 개체를 등록 및 관리할 수 있으며 입력항목은 다음과 같음
 - 시스템 개체 명: 해당 프로젝트에서 시험을 시행하는 대상인 시스템의 명칭
 - 시스템 개체 일련번호: 시스템 개체의 고유(일련) 번호
 - 제작일: 시스템 개체가 제작된 날짜
 - 제작사: 시스템 제작사

3) 시스템 PBS 등록: 시스템을 구성하는 품목을 등록 및 관리할 수 있으며 입력항목은 다음과 같음

- PBS: 시스템의 구성 품명
 - 구성 수량: 시스템 1개당 수량 정보
- 4) 고장 코드 등록: 프로젝트별 고장 코드(고장 원인, 고장 조치, 고장 범주)를 등록 및 관리할 수 있으며 입력항목은 다음과 같음
- 고장 원인/조치/범주 코드: 고장 원인/조치/범주를 구분할 수 있는 코드(식별자)
 - 고장 원인/조치/범주 코드명: 시스템에서 발생할 수 있는 고장 원인/조치와 고장 범주

5) 시험 종류 및 항목 등록: 프로젝트별로 수행하는 시험 정보를 등록 및 관리할 수 있으며 입력항목은 다음과 같음(Fig. 2 참고)

- 시험 종류 명: 시험 항목의 그룹과 같은 개념(ex. 제품시험, 현차시험 등)
- 시험 항목 명: 시스템에 실제 행해지는 시험
- 시험 목적: 시험 항목을 수행하는 목적
- 시험 방법: 시험 항목을 수행하는 방법
- 시험 기준: 시험 항목 수행 시 결과를 판단할 수 있는 기준 작성

시스템 PBS 등록, 고장 코드 등록의 경우 사용자가 기존에 S/W를 사용한 이력이 있다면 해당 내용을 불러올 수 있다.

3.2.2 시험 및 결과 관리 (Test and result management)

시험 결과 관리 메뉴는 시험 담당자와 설계 담당자가 시험 중, 시험 후에 입력하는 메뉴이다. 시험 전 설계 담당자가 '프로젝트 및 기본정보 관리'에 입력한 내용을 토대로 시험 시 발생한 시험, 고장, 고장 조치 기록을 등록 및 관리할 수 있다. 시험 및 결과 관리는 시험 기록, 고장/고장 조치 기록 화면으로 구성되어 있다.

- 1) 시험 기록: 시험수행 시 시험 이력 및 결과를 등록 및 관리할 수 있으며 입력항목은 다음과 같다. (Fig. 3 참고)
 - 시험 시작 일자: 시험을 시작한 일자
 - 시작/종료 시간: 시험을 시작/종료한 시간
 - 시험 장소: 시험을 수행한 장소
 - 날짜: 시험수행 시 날짜
 - 시스템 개체 명: '시스템 개체 등록'에서 입력한 시스템 개체 중 시험을 수행한 개체 선택
 - 시험자: 시험을 수행한 사람의 이름
 - 운용 시간(개체당/hr): 시험 시작 시각에서 종료 시간까지 시험 대상인 시스템 개체가 운용된 시간
 - 운용 거리(km): 시험 시작 시각에서 종료 시간까지 시험 대상인 시스템 개체가 운용된 거리
 - 합격 여부: '시험 종류 및 항목 등록'에서 작성한 시험 기준에 따라 결정된 합격 여부(Y/N) 선택
 - 시험 현황: 시험 전, 시험 중, 시험 완료 중 선택
 - 시험 결과 내용: 시험수행 결과
- 2) 고장/고장 조치 기록: 시험수행 시 발생한 고장 이력 및 고장 조치 이력을 등록 및 관리할 수 있으며 입력항목은 다음과 같음
 - 고장 일시: 시험수행 중 고장이 발생한 일시
 - 고장 발생 장소: 시험수행 중 고장이 발생한 장소
 - 시스템 개체 명: '시스템 개체 등록'에서 입력한 시스템 개체 중 고장이 발생한 시스템 개체를 선택
 - PBS: 'PBS 등록'에서 입력한 PBS 중 고장이 발생한 구성품을 선택
 - 고장번호: 사용자가 정의한 고장번호
 - 고장 원인: '고장 코드 등록'에서 입력한 고장 원인 코드 중 해당하는 원인 코드를 선택
 - 기타 원인: 고장 원인 코드에서 정의한 원인 외의 고장 원인을 작성
 - 고장발견 코드: 고장을 발견한 방법을 선택
 - 고장 범주: '고장 코드 등록'에서 입력한 고장 범주 코드 중 해당하는 범주 코드를 선택

- 고장 현상: 발생한 고장 현상
- 고장 조치: '고장 코드 등록'에서 입력한 고장 조치 코드 중 해당하는 조치 코드를 선택
- 조치 시작/종료일: 고장 조치를 시작/종료한 날짜
- 조치 소요 시간(hr): 고장 조치 때 소요된 시간 작성
- 담당자: 고장 조치 완료 여부를 확인한 담당자 이름
- 조치 내용: 고장 발생 시 수행한 조치 내용

3.2.3 문제 및 시정조치 관리 (Problem and corrective action management)

문제 및 시정조치 관리 메뉴는 설계 담당자가 시험 중, 시험 후에 입력한 내용을 분석하여 문제를 도출하고 문제를 해결하기 위한 시정조치를 등록 및 관리할 수 있는 메뉴이다. 문제 및 시정조치 관리는 문제 관리, 시정조치 관리 화면으로 구성되어 있다.

- 1) 문제 관리: 시험수행 시 반복적으로 발생하는 여러 고장을 하나의 문제로 등록하고 문제 원인 분석 및 근본 원인 분석을 등록하여 문제를 관리할 수 있으며 입력항목은 다음과 같다.
 - 문제명: 시험수행 중 발생한 문제
 - 담당자: 시험수행 중 발생한 문제를 도출한 담당자 이름
 - 문제 등록일: 문제를 등록한 날짜
 - 관련 고장: '고장 기록'에서 입력한 고장 중 문제와 관련된 반복적으로 발생하는 고장을 선택
 - 문제 설명: 문제에 대한 설명
 - 문제 발생 단계: 문제가 발생한 단계
 - 문제로 인한 영향: 문제로 인해 발생한 영향
 - 문제 분석일: 문제를 분석한 날짜
 - 문제 분석자: 문제를 분석한 사람의 이름
 - 근본 원인: 문제에 대한 근본 원인을 분석한 내용
 - 근본 원인 설명: 근본 원인에 대한 설명
 - 근본 원인 검증 방법: 근본 원인을 검증할 방법
 - 근본 원인 분석자: 근본 원인을 분석한 사람의 이름
 - 근본 원인 분석일: 근본 원인을 분석한 날짜
- 2) 시정조치 관리: 등록된 문제를 해결하기 위해 해당 문제에 대한 시정조치 정보를 등록 및 관리할 수 있으며 입력항목은 다음과 같음
 - 시정조치 번호: 사용자가 정의한 시정조치 번호
 - 시정조치: 문제를 해결할 수 있는 시정조치 방안
 - 시정조치 검증 방안: 시정조치가 문제를 해결할 수 있는지 검증할 수 있는 방안
 - 등록일: 시정조치를 등록한 날짜

시험 종류 및 항목
✕

○ 시험 항목

시험 항목 명	기계적 내구성 시험
시험 목적	기계적 내구성 파악
시험 방법	제동장치를 차량에 장착하여 운영 후 상태확인
시험 기준	마모한계에 도달할때까지 변형이 없을 것
비고	

저장 취소

Fig. 2. Project and information management_Test information record

시험 기록
✕

○ 시험 항목 정보 저장

시험 항목명	기계적 내구성 시험	시험 목적	기계적 내구성 파악
시험 방법	제동장치를 차량에 장착하여 운영 후 상태확인	시험 기준	마모한계에 도달할때까지 변형이 없을 것
합격 여부	Y	시험 현황	시험 완료
시험 결과 내용	마모한계에 도달할때까지 변형이 없음		

○ 시험 기록 저장 | 상세

시작 일자	2022.07.18	종료 일자	2022.07.18
시험장소	00역	시험자	김시연
시스템 개체명	제동장치1, 제동장치2, 제동장치3, 제동장치4, 제동장치5	첨부파일	파일 선택 선택된 파일 없음
운용시간(개체당)(hr)	3	운용거리(km)	30

Fig. 3. Test and result management_Test result record



Fig. 4. Status inquiry_dashboard

- 등록자: 시정조치를 등록한 사용자 이름
- 시행일: 시정조치를 시행한 날짜
- 상태: 시정조치가 진행되고 있는 상태(완료/수행 중/실패) 중 선택

3.2.4 현황조회 (Status inquiry)

현황조회 메뉴에서는 설계 담당자, 시험 담당자 시험 중 또는 시험 후 입력한 고장, 고장 조치, 문제, 시정조치 현황을 조회할 수 있는 메뉴이다. 프로젝트별로 사용자가 기록한 내용을 조회할 수 있으며, 시험, 고장, 문제 현황별로 상세 조회 또한 가능하다. 프로젝트별로 사용자가 기록한 내용을 조회할 수 있는 대시보드 화면은 다음 Fig. 4와 같다[8].

3.2.5 사용자 관리 (User management)

사용자 관리 메뉴는 프로그램에서 최고 권위자로 할당된 사용자가 사용자 등록 및 권한을 할당할 수 있는 메뉴이다. 해당 메뉴에서 사용자가 사용할 수 있는 메뉴, 기능을 조정할 수 있다.

4. 결론

본 논문에서 연구한 FRACAS S/W는 주 사용자를 설계 담당자와 시험 담당자로 설정하였으며, 프로젝트와 관련된 정보, 시험과 관련된 정보는 시험 전에 입력해야 한다. 시험 중, 시험 후에는 시험 담당자가 시험을 하며 발생한 시험 결과, 고장 및 고장조치 내용을 입력해야 하며, 설계 담당자는 시험 담당자가 입력한 내용을 토대로 시험을 제대로 수행하지 못하게 한 공통된 여러 고장을 하나의 문제로 등록하고, 문제의 근본원인을 분석한다. 문제를 해결하기 위한 시정조치와 시정조치 효과를 등록하고, 시정조치가 효과가 있다면 해당 문제의 재발을 방지하기 위한 절차를 등록한다. 본 논문에서 연구한 FRACAS S/W는 개발 후 FRACAS 국제 표준을 반영하여 정성적 검증을 수행하였다.

현재 개발한 FRACAS S/W는 시스템의 시험단계에서 고장을 발견하고 이를 해결하는 것을 지원하기 때문에 운영단계에서 고장을 발견하고 해결하는 것보다 비용, 시간 측면에서 효율적이며, 시험단계에서 발생한 데이터를 기록하고 관리함으로써 추후 새로운 시스템 개발 시 기록하고 관리한 내용을 활용하여 신뢰성 향상에 도움이 될 것으로 판단된다.

개발한 FRACAS S/W는 조회화면에서 사용자가 기록한 현황만 볼 수 있다. 사용자가 기록한 내용을 기간별, 고장 키워드 별로 조회할 수 있는 기능과 여러 프로젝트를 등록하였을 때 프로젝트간 현황 비교를 할 수 있는 기능이 추가된다면 추후 사용자가 개발하고 있는 대상에 대한 현황 및 고장 발생 특징 파악에 도움이 될 것으로 판단되기 때문에 추후 추가 개발 시 해당 내용에 대한 개선이 요구된다.

References

- [1] Department of Defense, "DOD GUIDE FOR ACHIEVING RELIABILITY, AVAILABILITY, AND MAINTAINABILITY", 2005, Chapter 4.
- [2] Department of Defense, "MIL-HDBK-338B: Electronic Reliability Design Handbook", 1998, Section 8.2.
- [3] M.Y. LEE, W.G. Kim, K.S Kim, "A Study on the Development of FRACAS-based Failure Analysis Workflow for Military weapon system", *Journal of Applied Reliability*, Vol.10, No.2, pp. 93-105, Jun. 2010.
- [4] Y.W. Hong, "Development of failure reporting analysis and corrective action system", *Journal of the Korean Data & Information Science Society*, Vol.21, No.1, pp. 109-119, Jan. 2010.
- [5] J.H. Jo, H.S Song, B.H Kim, "Development of TLCSM Based Integrated Architecture for Applying FRACAS to Defense Systems", *Journal of the Korea Academia-Industrial cooperation Society*, Vol.21, No.1, pp. 190-196, Jan. 2020.
DOI: <https://doi.org/10.5762/KAIS.2020.21.1.190>
- [6] J.E. Oh, C.Y. Kang, "Study on the Application of FRACAS System for Rolling Stock", *Proceedings of the KSR Conference*, The Korean Society For Railway, Korea, pp. 487-497, Nov. 2008.
- [7] G. Jang, C.W. Chung, D.H. Shim, "A Study on FRACAS-based Dependability Control Platform for Domestic Urban Railway Train", *Journal of Information Technology Service*, Vol.19, No.3, pp. 151-163, Jun. 2020.
DOI: <https://doi.org/10.9716/KITS.2020.19.3.151>
- [8] Nemosys, DCACAS Dashboard, Nemosys, c2022, Available From: <https://dcacas.nemosys.ai/main#> (accessed Sept. 2, 2022)

김 시 연(See Yeon Kim)

[정회원]



- 2019년 1월 ~ 현재 : 네모시스 주식회사 연구원

<관심분야>

철도, RAMS

김 종 윤(Jong Woon Kim)

[정회원]



- 1997년 2월 : 부산대학교 산업공학석사 (신뢰성공학)
- 2003년 2월 : 부산대학교 산업공학박사 (신뢰성공학)
- 2003년 4월 ~ 2005년 12월 : 현대로템 과장
- 2005년 12월 ~ 현재 : 한국철도기술연구원 선임연구원
- 2016년 2월 ~ 현재 : 네모시스 주식회사 대표

<관심분야>

철도, RAMS