

스프레드시트 데이터를 이용한 시각적 FMEA 분석 시스템 구현

이재호[○] 최지수[○] 김상철[○] 낭종호

서강대학교 컴퓨터 공학과

kill1623@naver.com, mystic707@naver.com, smaslayer1@nate.com, jhnang@sogang.ac.kr

An Implementation of Spread-Sheet Data-Based Visual FMEA Analysis System

Jae-ho Lee[○] Ji-soo Choi[○] Sang-chul Kim[○] Jong-ho Nang
Department of Computer Science and Engineering, Sogang University

요 약

기계 부품(시스템 요소)의 고장이 기계(시스템) 전체에 미치는 영향을 예측하는 해석방법인 FMEA(Failure Mode and Effect Analysis) 분석은 기계뿐만 아니라 여러 목적을 가진 시스템 개발 과정에 사용된다. FMEA 분석 과정은 보통 표를 통해 명세하고 그 내용을 이해하기 쉽게 그래프로 시각화 하는데 수동적으로 작성하는 경우가 많아 효율적이지 못하다는 문제점이 있다.

본 논문에서는 이런 점을 보완하기 위해 FMEA 분석 과정에서 명세한 스프레드시트 표를 이용하여 자동으로 시각화된 그래프를 그려주는 시스템을 구현하여 FMEA 데이터의 분석 및 이해를 도울 수 있도록 하였다. 본 논문에서 구현한 시스템을 기반으로 FMEA 분석가가 그래프를 작성하는 시간을 단축시키고 편집이 간단하며 이해하기 쉬운 결과를 얻을 것으로 기대한다.

1. 서 론

다수의 기계 부품(시스템 요소)이 모여 큰 하나의 기계(시스템)을 이루는 경우 시스템이 오류를 일으켰을 때 원인을 찾기 위해 작동과 연관된 흐름을 따라 잘못된 시스템 요소(이하 컴포넌트)를 찾아야 한다. 이런 문제점을 지속적으로 예측하고 보완하기 위해 다양한 시스템 개발 분야에서 FMEA(Failure Mode and Effect Analysis)[1] 분석 방법을 사용한다. 현재 많은 FMEA 분석가들이 FMEA 분석 과정 중 도식화 과정에서 많은 시간을 소요하고 있다. 본 논문에서는 효율적인 FMEA 분석을 위해 시스템 내부의 각 컴포넌트들이 작용하는 내용을 스프레드시트 프로그램을 통해 상호작용표라는 새로운 양식을 정의하고, Open Source인 Graphviz[2]를 응용하여 만든 프로그램을 스프레드시트 프로그램에 Add-In하여 상호작용표를 시각화된 그래프로 생성하는 시스템을 구현하였다. <그림 1>은 본 시스템의 전체 구조도로서, 입력 데이터인 스프레드시트의 상호작용표로부터 시각화된 그래프 출력을 얻기 위해 다음과 같은 문제의 해결이 필요하다.

- 상호작용표의 정의
- 스프레드시트 데이터 추출 및 해당 데이터를 통해 편집 가능한 그래프의 구현
- Graphviz를 사용하여 FMEA 분석가의 스타일에 맞는 그래프 구현

본 논문에서는 시스템 구현을 위하여 위와 같은 문제를 해결하기 위한 노력을 기술하고, 실제 시뮬레이션을 통한 구현 결과를 소개한다. 이러한 시스템을 통하여 수동적인 방법을 통해 진행하였던 FMEA 분석 과정을 자동화하여 소요 시간을 줄이고 분석 및 편집이 용이한 데이터 양식을 제공함으로써 시스템 개발 과정의 효율을 제공할 것이라 기대한다.

2. 기존의 FMEA 분석 방법

2.1 기존 FMEA 분석가들의 방법

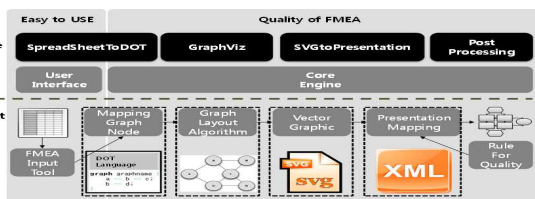
다양한 분야에서 다양한 시스템에 맞추어 FMEA 분석가들이 나름의 노하우를 통해 FMEA 분석을 진행하지만 보통 한 시스템의 컴포넌트들을 표로 구성하여 그것들 간의 작용 관계를 기술하고 분석가들만의 스타일에 맞추어 시각화된 그래프를 구성한다. 이 과정에서 많은 분석가들이 수작업으로 그래프를 작성하는데 1~3시간 정도의 시간을 소요하였다. 그리고 분석가마다 그래프를 작성하는 스타일이 다르기 때문에 컴포넌트를 이어주는 선들의 겹침(Crossing) 문제가 발생하거나 정형화 되지 않은 결과물로 인해 다른 전문가로 하여금 이해와 재사용에 어려움을 줄 수 있다.

2.2 FMEA 자동화 시스템을 통한 방법

FMEA 분석 과정에서 어떤 시스템을 분석하려 하는지 혹은 어떤 단계를 수행하는지에 따라 중점이 되는 개념이 다르다. 따라서 목적에 맞게 구현된 FMEA 자동화 시스템이 필요하며 산업 현장에서 이미 배포되어 사용되고 있는 프로그램들이 존재한다.

컴포넌트와 작용선의 자동배치를 통해 FMEA 분석가들의 그래프 생성 소요 시간을 줄이고 그래프 생성에 규칙을 적용하여 일부 정형화된 양식의 결과를 얻어 여러 분석가들이 공유가 가능한 시스템을 제안하였다. 만약 FMEA 분석가들이 결과물에 대해 수정을 요구할 경우 수작업으로 편집이 가능한 결과물을 얻을 수 있게 하였다.

[표 1]은 기존 FMEA 분석가들의 방법과 본 시스템을 이용한 분석 방법을 비교한 것으로서, 그래프 작성의 실시간성과 확장성



<그림 1> 시스템 구조도

본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합 원천기술개발사업(정보통신)의 일환으로 수행하였음. [10044615, 클라우드 기반 개방형 소셜 방송미디어 콘텐츠 융합 생성, 편집 및 재생을 위한 미디어 제작 및 전송 시스템 개발]

에서 유리하다고 볼 수 있다. 그리고 [표 2]는 타 FMEA 자동화 시스템들의 특징과 본 시스템을 비교한 것으로 그래프를 통한 데이터 시각화, 재사용 가능한 데이터 저장 방법과 자산화, 범용하고 있는 프로그램 수행환경이 가장 큰 장점이라 볼 수 있다.

[표 1] 기존 FMEA 분석 전문가들의 방법과 본 시스템을 이용한 분석 방법 비교

		기존 FMEA 분석가들의 방법	본 시스템을 이용한 방법
상호작용표 작성		스프레드시트에 작성 (FMEA 전문가가 참고하여 그래프로 작성)	스프레드시트에 작성 (시스템이 데이터를 추출하여 그래프로 작성)
그래프 작성	실시간성	1~3시간 소요	1초 이내
	확장성	수동적 편집 가능	수동적 편집 가능
	정형성	전문가들마다 다름	그래프 생성 옵션을 통해 비교적 비슷한 스타일의 출력결과 생성

[표 2] 타 FMEA 자동화 시스템과 본 시스템의 특징 비교

FMEA Tools	개발	FMEA 관계표	데이터 시각화	자산화 (데이터 저장)	상용화	수행 환경
Automatic FMEA [3]	Gerime Blankenship (現 Supplier Quality Engineering Leader at Hussmann)	생성 가능	불가	MS Excel	무료	MS Excel
FMEA TOOLS2 [3]	C R A N E ChemPharma & Energy 社	생성 가능	막대 차트	MS Excel	무료	MS Excel
Automating FMEA program with TechScan [4]	Naneva 社	생성 가능	그래프	자체 형식	유료	자체 형식
XFMEA [5]	ReliaSoft 社	생성 가능	막대 및 파이 차트	MS Excel, Word, PDF	유료	MS Excel, Word
IQ-FMEA [3]	APIS 社	생성 가능	계층적 트리 차트	서버 중앙 관리	유료	자체 형식
CIMOS FMEA [3]	MBFG GmbH & Co. KG 社	생성 가능	계층적 트리 차트	MS Excel, Word, PDF	유료	자체 형식
본 시스템	서강대 컴퓨터공학과 멀티미디어 연구실	생성 가능	그래프	MS Excel, PPT	-	MS Excel, PPT

3. FMEA 자동화 시스템

본 논문에서 구현한 시스템은 1장에서 언급한 여러 문제들의 해결을 필요로 한다. 본 장에서는 이러한 문제점들의 특성 및 해결 방안에 대하여 소개한다. <그림 2>는 시스템의 흐름도(flow chart)로 FMEA 분석가가 자동화 시스템을 통하여 결과물을 얻기까지의 과정을 볼 수 있다.

3.1 스프레드시트에서 상호작용표 작성 방법

그래프를 생성하기 위해서 스프레드시트에 상호작용표를 작성해야 한다. 그래프를 생성하기 위해 Add-In된 프로그램은 정해진 규칙에 따라 상호작용표의 데이터를 추출하고 새 프레젠테이션 파일에 파싱을 하여 그래프를 생성한다. 본 시스템은 크게 3가지 그룹의 컴포넌트들을 생성할 수 있으며 각각은 사각형, 육각형 그리고 타원으로 생성된다. 각 컴포넌트 그룹은 자기들만의 부분집합이 될 수 있으며 이를 서브그룹이라 표현한다. 표의 좌측과 상단은 컴포넌트 이름을 대칭적으로 작성하여 좌측이 주(Main)가 되고 상단이 대상(Target)이 되어 표 내부에 작성된 작용 요소들은 주에서 대상으로 방향성이 있는 화살표선(이하 작용선)이 된다. 그래프 출력 시 작용선 근처엔 작용 요소가 텍스트로 표시된다. <그림 3>에서 상호작용표의 형태를 볼 수 있다.

	A	B	C	D	E	F	G	H	I	J	K
1				GroupOfComp1				GroupOfComp2		GroupOfComp3	
2				Subgroup1							
3				Subgroup1	Comp1	Comp2	Comp3	Comp5	Comp6	Comp9	Comp10
4											
5	GroupOf Comp1	Subgroup1	Comp1			Pull					
6	Comp2								Push	Pull	
7	Comp3		Push							Pull	
8	GroupOf Comp2		Comp5								
9	Comp2		Comp6		Push	Push					
10	GroupOf Comp3		Comp9					Push			
11	Comp3		Comp10					Pull			

<그림 3> 상호작용표

3.2 스프레드시트 데이터를 DOT 언어로 전환

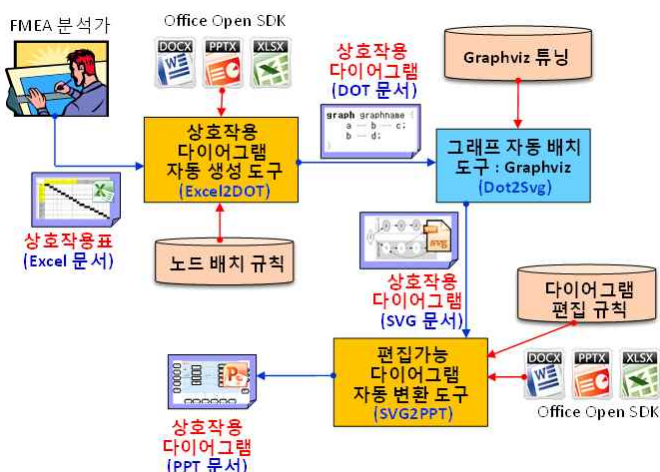
스프레드시트의 데이터를 Graphviz를 사용하여 그래프로 생성하기 위해 상호작용표에 있는 데이터를 Graphviz가 읽을 수 있는 DOT 언어로 전환하는 과정이 필요하다. DOT 언어는 크게 그래프 레이아웃 옵션, 컴포넌트 옵션 그리고 작용선 옵션으로 구성되며 사용자가 Add-In된 프로그램에서 설정한 옵션을 따라 각기 다르게 생성된다.

3.3 Graphviz를 통한 그래프 출력

DOT 언어로 전환된 텍스트 파일을 통해 Graphviz는 그래프를 생성한다. Graphviz는 JPG, PNG, BMP 등 다양한 이미지 파일 형식의 출력을 지원한다. 하지만 이미지 파일로 생성된 그래프는 추후 FMEA 분석가에게 편집 기능을 제공할 수 없다. 때문에 XML 형식으로 컴포넌트와 작용선의 정보가 담긴 SVG(Scalable Vector Graphic)[6]으로 출력한다.

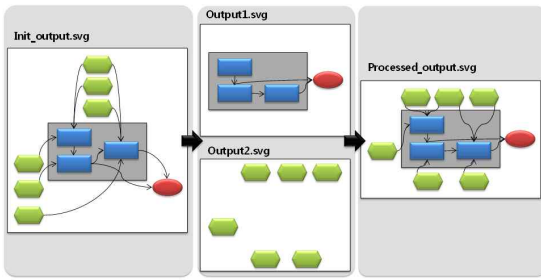
3.4 SVG Post_Processing

Graphviz가 가진 rank라는 기능을 사용하게 되면 컴포넌트들을 세로 열에 맞추거나 가로 행에 맞출 수 있는데 특성상 동시에 두 조건을 만족시킬 수 없기 때문에 세로 열에 맞출 경우 컴포넌트들의 가로 행 정렬이 맞지 않을 수 있다. rank를 세로로 기준할 경우 지나치게 컴포넌트들이 같은 열에 배치되어 전체적으로 그래프가 길어지거나 가로로 늘어지는 결과가 생긴다. 이런 문제를 해결하기 위해 첫 번째 생성된 SVG 파일을 분석하여 컴포넌트들의 좌표 값을 확인 후 지나치게 세로 혹은 가로로 늘어난 컴포넌트들을 분리하여 output1.svg와 output2.svg 2개의 SVG 파일로 나누어 재 정렬한다. 이후 프레젠테이션 파일을 생성하는 과정에서 두 SVG 파일의 scale을 조정하고 하나의 좌표

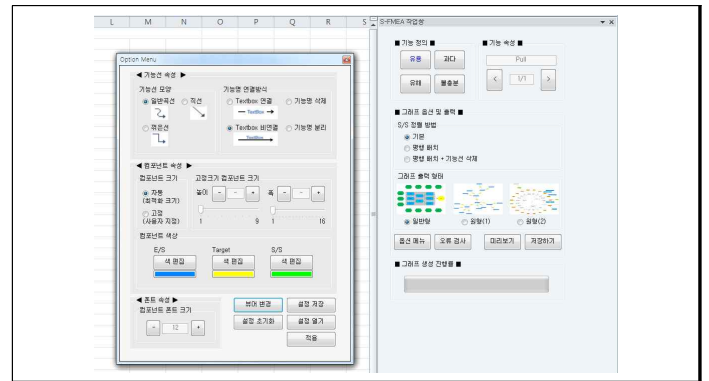


<그림 2> 시스템의 흐름도

에 매칭하여 단일 그래프 파일로 생성하게 된다. <그림 4>에서 SVG에 대한 Post-Processing 과정을 확인할 수 있다.



<그림 4> SVG Post-Processing



(b) FMEA 자동화 시스템 UI

3.5 SVG 파일을 통해 프레젠테이션 데이터 생성

XML 형식으로 되어 있는 SVG 파일 정보를 파싱하여 새 프레젠테이션 파일에 도형들을 생성한다. 본 시스템에서 사용한 프레젠테이션 프로그램은 Microsoft社의 PowerPoint를 사용하였고 최종 결과로 PPTX 파일을 생성하였다. PPTX 파일 역시 XML 형식으로 되어있는 PowerPoint 파일 확장자 중 하나로 XML 텍스트 편집을 통해 완성된 그래프 파일을 생성할 수 있다.

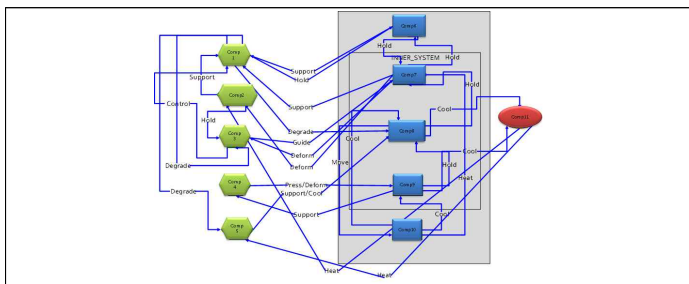
4. 시스템 구현 및 분석

본 논문에서는 2장에서 기술한 시스템의 여러 이슈 및 특성들을 고려하여 <그림 5>과 같이 시스템을 구현하였다. 산업 현장에서 사용하는 FMEA 분석 자료를 준비하고 FMEA 분석가가 직접 수동적으로 작성한 시간과 FMEA 자동화 시스템을 사용하여 작성 후 전문가의 스타일에 맞게 일부 편집한 시간을 비교하였다.

[표 3] 수동적 작성 방법과 자동화 시스템을 이용한 작성 방법의 소요 시간

실험 데이터	총 컴포넌트 수	총 작용선 수	수동적 작성 소요 시간	자동화 시스템 작성 소요 시간
1	20	29	약 70분	약 15분
2	11	30	약 60분	약 12분
3	11	27	약 60분	약 12분
4	13	27	약 60분	약 7분
5	13	27	약 60분	약 8분

수동적 작성한 시간과 FMEA 자동화 시스템을 사용한 시간에 대해 정확한 검증을 위해 실제 FMEA 분석가를 초빙하여 실험을 하였다. FMEA 분석가는 각 실험 데이터 1~5번에 해당하는 상호작용표를 토대로 먼저 기존에 그래프를 그리는 방법대로 수동적 작성을 실시하였다. 그 다음 FMEA 자동화 시스템의 사용방법을 분석가에게 지도하고 분석가가 스스로 실험 데이터를 이용하여 그래프 파일을 생성하고 자신의 스타일에 맞게 편집할 수 있는 시간을 측정하였다. 그 결과, 수동적인 방법으로 약 70분 정도가 소요되는 데이터에 대해서 약 15분 정도로 소요시간을 줄일 수 있었다. 그리고 약 60분 정도가 소요되는 데이터 2~5에 대해서 평균 약 9.75분으로 소요시간을 줄일 수 있었다. FMEA 자동화 시스템을 활용하여 전체적으로 수동적 작성 시간보다 약 80%의 소요 시간을 줄일 수 있었다.



(a) 자동화 시스템을 사용한 출력 결과

<그림 5> FMEA 자동화 시스템 출력 결과와 UI

5. 결 론

본 논문에서는 기존 FMEA 분석 과정에서 전문가들이 수동적으로 진행되는 과정을 가급적 자동화하고 효율성을 높이기 위해 기존에 사용하고 있는 스프레드시트 프로그램을 사용하여 입력 데이터가 되는 상호작용표를 작성하고 최적화된 그래프를 그려 주는 Open Source인 Graphviz를 사용하여 그래프 형태를 생성, 최종적으로 편집이 가능한 확장성 있는 프레젠테이션 파일로 결과물을 출력하는 시스템을 제안하였다. 그리고 기존 방법과 비교하여 FMEA 자동화 시스템이 문제가 되는 이슈를 효과적으로 해결할 수 있는지 시뮬레이션을 통해 확인 및 분석하였다.

본 논문의 시스템을 통해 FMEA 분석가들은 상호작용표를 사용하여 그래프를 생성하는 시간을 단축시킬 수 있으며 편집이 용이한 결과물을 얻어 추후 재사용 및 분석이 효율적일 것이다. 그리고 어느 정도 정형화된 양식의 범위에서 출력물을 얻을 수 있기 때문에 타 분석가가 작성한 그래프를 이해하고 편집하는데도 용이하며 분석가들의 스타일에 맞출 수 있는 일정 범위내의 다양한 옵션을 지원하고 있어 FMEA 분석을 보다 유용하게 할 수 있을 것이다. 차후 1차원 FMEA 그래프가 가진 한계점을 극복하기 위해 다차원, 계층적 구조를 갖는 그래프에 대해서 연구를 계속 진행할 계획이다.

6. 참고 문헌

[1] Goddard, P.L., "Software FMEA Techniques", Proceedings of Annual Reliability and Maintainability Symposium, pp. 118-123, 2000
 [2] Ellson, J., Gansner, E., Koutsofios, L., North, S. and Woodhull, G., "Graphviz - Open Source Graph Drawing Tools", Graph Drawing, pp. 483-484, 2002.
 [3] Google, "automatic fmea", <http://www.fmeainfocentre.com/tools.htm>, (2014.06.12)
 [4] Google, "techscan naneva", <http://nanevacorp.com/category/papers/>, (2014.06.12)
 [5] Google, "fmea tool", <http://www.reliasoft.com/xfmea/features1.htm>, (2014.06.12)
 [6] Ferraiolo, J., Jun, J. and Jackson, D., "Scalable Vector Graphics(SVG) 1.1 Specification", World Wide Web Consortium, technical report, 2003.