

PSA 자동 정량화 체제 개발에 관한 연구

A study for the development of PSA automatic quantification system

김승환, 한상훈
한국원자력연구소
대전광역시 유성구 덕진동 150

요 약

한국원자력연구소에서는 중장기 과제의 하나로 PSA 통합 데이터베이스를 구축 중에 있다. PSA 통합 데이터베이스는 PSA 모델은 물론, PSA와 관련된 각종 정보들을 데이터베이스화 하여 통합 저장하여 PSA 정량화는 물론 검토 및 QA등에 사용하기 위한 데이터 집합체이다. 본 연구에서는 PSA 모델 데이터를 기초로 하여 각 초기사건 시나리오별로 PSA 정량화를 자동으로 수행시켜주기 위한 PSA 자동 정량화 시스템의 개발에 대한 연구에 대하여 논하였다. 즉 PSA 자동 정량화 시스템을 구축하기 위한 PSA 모델 DB의 구조 설계 및 인터페이스, 그리고 PSA 정량화 자동화 절차의 설계 및 구현에 대한 연구사항을 기술하였으며, 정량화 수행결과로 생성되는 최소 단절군 집합의 저장 및 활용에 대한 연구 상황을 기술하였다. 현재는 기초 시나리오에 대한 PSA 자동정량화 프로토타입이 개발되었고, 최소 단절군 저장시스템이 개발되었는데, 사용자의 다양한 시나리오를 수용할 수 있도록 시스템을 수정 중에 있다.

Abstract

KAERI is constructing the PSA Integrated database for UCN 3, 4 nuclear power plant. PSA Integrated DB includes PSA Model Database and PSA Information Database. PSA Model DB consists of PSA models and data which are used for PSA quantification. This paper describes the development of PSA automatic quantification system that calculates the core damage frequency according to the initiating event scenario. We describe the design of PSA model DB schema, the design and implementation of automatic quantification for PSA and the study for the storing of results of that quantification. Recently, we have developed the prototype of PSA automatic quantification system and we are developing more flexible system that applies the various user input and scenario changes.

1. 서론

PSA는 원자력 설비에서 발생 가능한 모든 사고에 대하여 사고 경위를 모델하고 그 발생 빈도를 추정함으로써 원전의 안전성을 정량적으로 평가하는 방법이다. PSA는 원전의 설계, 운전, 보수 등을 종합적으로 모델하기 때문에, 종합적으로 원전의 안전성을 평가하고 취약점을 도출하여, 그 대처방안을 체계적으로 마련할 수 있는 중요한 발전소 안전 확보 체계라

할 수 있다. 최근에 PSA 및 위험도 정보 활용 분야가 활성화 되면서 원전 현장에서도 PSA를 평가할 수 있는 기초적인 능력의 필요성이 증대하고 있다. 이러한 필요성에 맞추어, 원자력연구소에서는 중장기 연구 과제로서 위험도 정보 활용에 필수적인 국내의 PSA 종합 DB 체계를 구축하고 있다. PSA 종합 DB 체계의 구축이라 함은, 크게 PSA 정보를 저장하기 위하여 데이터베이스를 구축하는 것과 구축된 DB를 PSA 평가에 유용하게 사용할 수 있도록 검색기능 및 정량화 자동화 등의 각종 인터페이스를 제공하는 것이다. 이 중에는 PSA 정량화를 손쉽게 수행할 수 있도록 해주는 기능이 필수적이라 할 수 있다. 이는 PSA 정보 DB에 저장된 PSA 모델을 이용하여 각 사건 시나리오별로 정량화를 자동으로 수행할 수 있도록 도와주는 자동화 시스템이라 할 수 있다. 이러한 시스템을 통해 PSA 전문가의 PSA 분석 능력을 향상시키며, PSA 비전문가라 할지라도 간단한 PSA 정량화를 수행할 수 있도록 하여 국내 PSA 및 위험도 정보 활용분야를 활성화 할 것이다. 본 논문에서는 이를 위하여 PSA 자동 정량화를 위한 DB 모델의 설계 및 자동 정량화 절차 로직의 구현 그리고 계산결과의 저장을 위하여 연구한 내용에 대하여 서술하였다.

2. 본론

가. PSA 통합 DB 소개

PSA 통합 DB는 PSA 모델 및 결과 등의 PSA 관련 자료들을 정보화하여 데이터베이스에 저장하는 것으로, PSA 평가에 필요한 모든 입력 사항들을 제공하여 PSA 평가를 더욱 쉽게 수행할 수 있는 기반을 제공하여, 국내 위험도 정보 활용 및 PSA의 기반을 확대시키고, 결과의 신뢰성을 증진시키는데 기여하기 위하여 개발되는 PSA 정보화 모델이다.

PSA 통합 DB는 그 목적과 성격에 따라서 PSA 모델 DB와 PSA 정보 DB의 두 가지 시스템으로 구분되어 질 수 있다. PSA 모델 DB는 PSA 정량화 수행에 직접적으로 필요한 PSA 모델과 데이터를 저장하는 시스템이며, PSA 정보 DB는 PSA 모델 이외의 각종 자료 즉 관련 PSA 보고서, 도큐먼트, 계산결과, 발전소 정보 등의 PSA 관련 정보들을 저장하여 놓은 시스템이다.

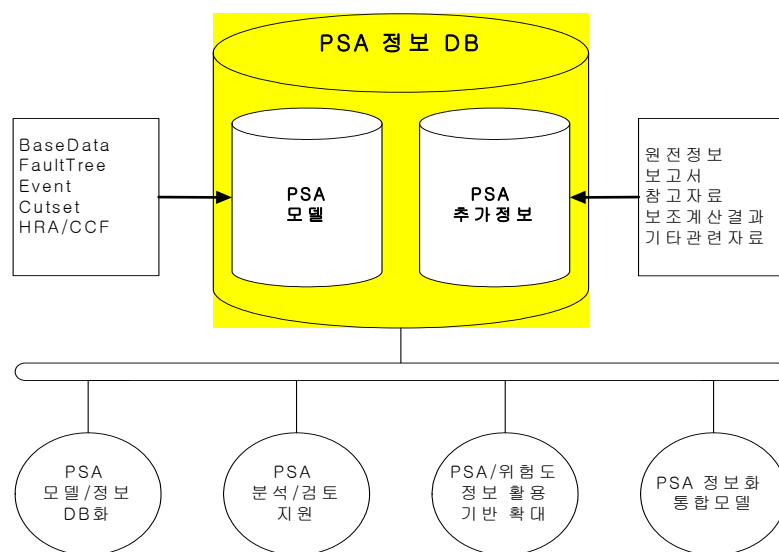


그림 1 PSA 정보 DB 구성도

앞의 그림은 PSA 정보 DB의 구성도이다.

그림에서 보는바와 같이 PSA 정보 DB는 PSA 평가에 필요한 모델과 데이터를 저장한 PSA 모델DB와 기타 PSA와 관련된 추가 자료들을 정보화한 PSA 추가정보 DB로 구분하여 구조를 설계하였으며 그 세부 DB별 상세사항은 다음과 같다.

- PSA 모델 DB

PSA 평가에 필요한 모델과 데이터를 저장하여, PSA 평가 및 민감도 분석에 사용하며 다음 분야에 사용된다.

- PSA모델링 : PSA정량화 수행에 필요한 기본사건, 고장수목, 사건수목 데이터 등의 정보를 제공한다.
- 정량화 모듈 : PSA 정량화 (CDF, LERF) 계산에 필요한 각종 자료를 제공하고 자동화에 필요한 기능을 제공한다.
- 보조분석 S/W
 - HRA분석 Tool : 인적오류분석 관련 데이터를 제공
 - CCF 분석 Tool : 공통원인고장 관련 데이터를 제공
 - PSA 모델 검색 S/W : PSA 모델과 데이터를 검색하기 위한 모듈을 제공

- PSA 정보 DB

PSA의 수행에 필요한 추가 자료들을 정보화하여, 다양하고 방대한 PSA 참고 자료들의 검색에 사용하며 다음 분야에 사용된다.

- 통합검색 S/W : PSA 통합 DB에 저장된 각종 자료를 검색할 수 있는 기능을 제공
- 민감도분석 모듈 : 민감도 분석의 수행에 필요한 각종 자료(PSA모델, 기기 정보등) 제공
- QA 절차 제공 : 품질 보증 절차 확립을 위하여 필요한 각종의 자료를 제공
- 원전기기신뢰도 DB연동 : 원전 기기 신뢰도 DB 와 유기적으로 연동되어 각 기기/기기종류, 기본 사건 등에 필요한 신뢰도 고장률 데이터를 제공

다음 그림은 PSA 종합 DB 체제 구축의 흐름도이다.

그림에서 보는바와 같이, PSA 정보 DB는 PSA 모델 데이터와 PSA 추가정보로 구분하여, 이를 기초로 PSA 정보 DB의 구조(스키마)를 작성하고, 이 DB 구조에 올린 3,4호기 PSA 모델과 추가정보를 DB화 입력하여 넣은 데이터베이스 시스템이다. 이렇게 구축된 DB를 이용하여 DB 기반의 PSA 정량화를 자동으로 수행할 수 있도록 시스템을 구축하는 것이 PSA 자동 정량화 방법개선이라 할 수 있다.

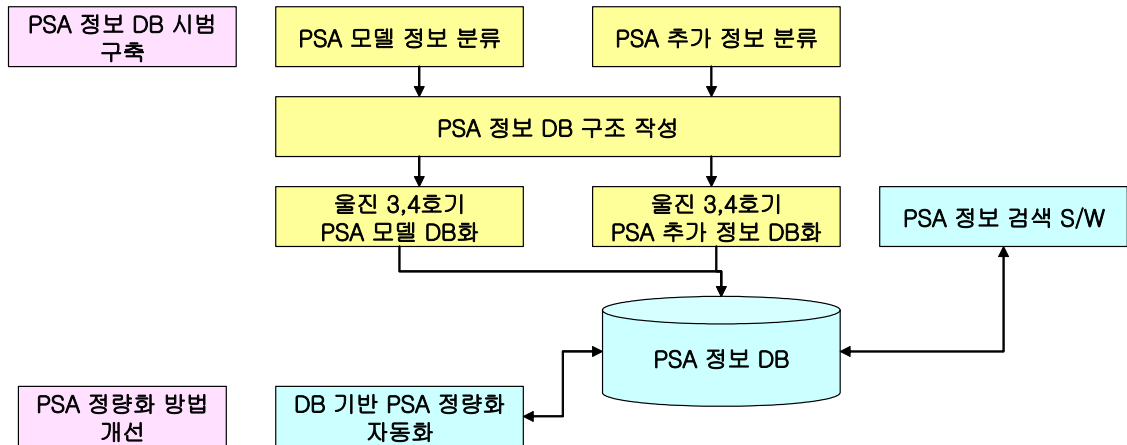


그림 2 PSA 종합 DB 체제 구축

나. PSA 자동 정량화 시스템 설계

PSA 자동 정량화 시스템은 PSA 정보 DB에 저장된 PSA 모델을 이용하여 각 사건 시나리오별로 정량화를 자동으로 수행할 수 있도록 해주는 DB 기반의 PSA 자동 정량화 지원 시스템이다. 이 시스템은 PSA 모델 DB에 저장된 사건수목(Event Tree) 및 고장수목(Fault Tree)등, 정량화에 필요한 각종 데이터를 기초로 하여 PSA 정량화 계산을 위한 입력 파일을 생성시켜주고, 이를 이용하여 PSA정량화를 수행시켜주며, 그 계산 결과인 최소 단절군(Cutsets)을 저장하여 주는 시스템이다.

그림에서 보는바와 같이 PSA 모델 DB는 사건수목 및 고장수목 등의 자료로부터 구축되는데, 자동 정량화 모듈은 PSA 모델 DB에 저장된 각각의 시나리오, 사건수목, 분기정보 및 변환조건 등의 정보를 이용하여, PSA 입력 파일을 생성시킨다.

이러한 PSA 입력 파일은 정량화 엔진인 KIRAP, FORTE 등에 입력되어 정량화를 수행하게 되는데, 이 결과로 생성되는 최소 단절군 집합(Cutsets)등은 PSA 모델에 저장되어지며, 저장된 Cutset은 Cutset Browser를 통하여 검색되어 질수 있다.

따라서 PSA 자동 정량화 시스템은 다음과 같은 사항을 만족 시키도록 구현되어야 한다.

- PSA 수행에 필요한 모든 데이터를 DB화 구축한 후, 이를 쉽게 검색할 수 있도록 한다.
- KIRAP, FORTE등의 각종 정량화 엔진을 수용할 수 있도록 표준화된 인터페이스를 제공.
- 사용자의 다양한 시나리오 변경에 수월하게 적응하여 민감도 분석을 손쉽게 수행할 수 있도록 한다.
- 자동화를 통하여 민감도 분석 시간을 대폭 절감시킬 수 있도록 한다.
- 기타 사용자의 요구사항을 만족시키도록 구축되어야 한다.

다. PSA 자동 정량화 시스템 구현

PSA 자동 정량화 시스템은 다음과 같은 주요 3가지 단계에 의하여 구성될 수 있다.

- DB 구조설계
- 정량화 로직 개발
- 최소 단절군 저장 및 처리

본 연구에서 수행한 각 항목별 상세 구성 내용은 다음과 같다.

1) DB 구조 설계

이 단계는 PSA 정량화에 필요한 각종 정보들을 DB화 하기위한 단계인데, 관련 테이블로는 먼저 사건수목 시나리오를 관리하기 위한 EventTree 와 관련된 테이블, 그리고 고장수목을 관리하기 위한 FaultTree 관련 테이블 등이 있다. 따라서 DB 구조설계 단계에서는 이들 테이블들의 구조를 확립하였으며, 각 테이블별 상세 설계 내용은 다음과 같다.

- EventTree

EventTree는 사건수목을 저장하기 위한 테이블로서 각 사건별로 사건명, 사건명세, 고장수목 등의 정보를 저장하고 또한 사건경위 즉 사건 수목 등의 시퀀스 정보를 저장하여야 하므로 여러 개의 테이블로 분리하여 구성하였으며 다음과 같은 항목들로 구성하였다.

ET : 사건수목의 정보를 보관

- ID
- IE
- IESeq
- Seq
- Heading
- Branch
- SuccessOrFail

ID	IE	IESeq	Seq	Heading	Branch	SuccessOrFail
1	ITRSM	ITRSM-04	0	ITRSM	ITRSM	FAIL
2	ITRSM	ITRSM-04	1	IT	GRITF	Success
3	ITRSM	ITRSM-04	2	FW	FMV	Success
4	ITRSM	ITRSM-04	3	SFO	SFA	Success
5	ITRSM	ITRSM-04	4	SDC	SDCV	FAIL
6	ITRSM	ITRSM-04	5	HGHF	FJN	FAIL
7	ITRSM	ITRSM-04	6	BD	SDGL	Success
8	ITRSM	ITRSM-04	7	HFI	HFIHM	Success
9	ITRSM	ITRSM-04	8	HFF	HFFHM	Success
10	ITRSM	ITRSM-04	9	CSF	CSRV	FAIL
11	ITRSM	ITRSM-05	0	ITRSM	ITRSM	FAIL
12	ITRSM	ITRSM-05	1	IT	GRITF	Success
13	ITRSM	ITRSM-05	2	FW	FMV	Success
14	ITRSM	ITRSM-05	3	SFO	SFA	Success
15	ITRSM	ITRSM-05	4	SDC	SDCV	FAIL
16	ITRSM	ITRSM-05	5	HGHF	FJN	FAIL

ETBranch : 사건수목에 포함된 분기정보를 보관

- ID
- IE
- Case
- No
- 계통
- Heading
- Branch

ID	IE	Case	No	계통	Heading	Branch	FFlag	SuccessOrFail	Condition	Remarks
1	ITRSM	Normal	0	IT	IT	IT				
2	ITRSM	Normal	1	FW	fw	fw	yes			after-rtbc
3	ITRSM	Normal	2	SFO	sfo	sfo	yes			after-rtbc
4	ITRSM	Normal	3	SDC	sdc	sdc	yes			after-rtbc
5	ITRSM	Normal	4	HGHF	hghf	hghf	yes			after-rtbc
6	ITRSM	Normal	5	HGHF	hghf	hghf	yes			after-rtbc
7	ITRSM	Normal	6	SDC	sdc	sdc	no			
8	ITRSM	Normal	7	BD	bd	bd	no			
9	ITRSM	Normal	8	GO	go	go	no			
10	ITRSM	Normal	9	HFI	hfi	hfi	no			
11	ITRSM	Normal	10	HFF	hff	hff	no			
12	ITRSM	Normal	11	CSF	csf	csf	no			
13	ITRSM	Normal	12	SDC	sdc	sdc	no			

- FTgate
- hasSupportLogic
- Condition
- Remarks

ETGroup : 각 유사 경로를 갖는 EventTree 그룹정보

- ID
- IE
- Group
- Seq
- IESeq

ID	IE	Group	Seq	IESeq
1	ITR01V	1	1	ITR01V-01
2	ITR01V	1	2	ITR01V-02
3	ITR01V	1	3	ITR01V-03
4	ITR01V	1	4	ITR01V-04
5	ITR01V	2	1	ITR01V-11
6	ITR01V	2	2	ITR01V-12
7	ITR01V	2	3	ITR01V-13
8	ITR01V	2	4	ITR01V-14
9	ITR01V	3	1	ITR01V-17
10	ITR01V	3	2	ITR01V-18
11	ITR01V	3	3	ITR01V-19
12	ITR01V	3	4	ITR01V-20
13	ITR01V	4	1	ITR01V-23
14	ITR01V	4	2	ITR01V-24
15	ITR01V	4	3	ITR01V-25
16	ITR01V	4	4	ITR01V-26

• FaultTree

고장수목의 구조는 전형적인 계층 트리 구조로서 각 노드(이벤트)가 트리를 구성하는 하위 노드의 정보를 가지고 있어야 한다. 따라서 각 고장 수목에 대한 정보를 저장하기 위해서, 각 노드의 정보를 담은 ftEventList 테이블과 각 노드에서 분기된 하위 노드에 대한 분기 정보를 저장하기 위한 ftTreeStructure 의 두 개의 테이블로 구성하였으며, 이 테이블들은 기본사건 (Basic Event) 테이블과 조인되어 해당정보를 제공하도록 구성하였다.

ftEventList

- idx : 인덱스키
- Name : 이름
- nChild : Child 이벤트의 갯수
- Type : 게이트종류
- Desc : Event Description
- ptParent : pointer to parent

ftTreeStructure

- idx : Parent Index
- seq : Child 일련번호 , 즉 자신이 부모 이벤트의 몇째 Child인가를 나타냄.
- idxChild : Child Index

ftTreeStructure는 세로 나열형 1칼럼 방식으로 구성하였는데, 이는 한 이벤트의 자식 (Child)이벤트를 테이블 형식의 칼럼 형식 즉 세로 방향으로 표현하는 것이다, 가로로 표현

하면 구현은 수월하나, 자식 노드의 개수를 한정시킬 수밖에 없어서, 한 노드당 차일드 이벤트가 많은 경우의 고장수목을 표현하기엔 한계가 있다.

따라서 ftTreeStructure 의 구조에서 보는바와 같이 1칼럼 방식의 세로 나열 형으로 구현하였는데, 이렇게 함으로써, 차일드 노드 이벤트의 개수에 상관없이 무한정 트리 구조를 확장할 수 있게 되었다.

다음 그림은 고장수목을 표현하기 위한 테이블 ftEventList 와 ftTreeStructure 테이블의 구성 예이다. 그림의 예를 보면, index(1)인 "GCDF-UCN"은 4개의 하부노드를 가지고 있으며, 각 노드들은 index { 2, 3, 2459, 3323 }으로 구성되어 있고, 각각 자신의 하부노드들의 정보를 가진 트리형태로 구성되어 있음을 알 수 있다.

idx	name	nChild	Type	Desc
0	GCDF-U3	6	+	Ulchin Unit 3 Core Damage Model (Unit 3)
1	GCDF-UCN	4	+	Ulchin Units 3&4 Core Damage Model for Risk Monitor
2	GCDF-UCN4	2	*	Ulchin Unit 4 Core Damage Model (Unit 4)
3	GCDF-UCN3	2	*	Ulchin Unit 3 Core Damage Model (Unit 3)
4	FLAG-U4	0	H	Flag for Ulchin Unit 4
5	GCDF-U4	0	?	Ulchin Unit 4 Core Damage Model (Unit 4)
6	FLAG-U3	0	H	Flag for UCN #3
7	G-U3-LOCAS	5	+	LOCA Sequences (Unit 3)
8	G-U3-LLOCA	2	*	Large LOCA Occur (Unit 3: Assumed Loop 1A Cold leg break)
9	G-U3-LL-SQ2	1	=	Large LOCA CD Sequence #2 (U3: ill*csrm)
10	G-U3-LL-SQ3	1	=	Large LOCA CD Sequence #3 (U3: ill*hp12n)
11	G-U3-LL-SQ4	1	=	Large LOCA CD Sequence #4 (U3: ill*hp13n)
12	G-U3-LL-SQ5	1	=	Large LOCA CD Sequence #5 (U3: ill*pi13n)
13	G-U3-LL-SQ6	1	=	Large LOCA CD Sequence #6 (U3: ill*sit23n)
14	G-U3-MLOCA	2	*	Medium LOCA Occur (Unit 3: Assumed Loop 1A Cold leg break)

idx	seq	idxChild
0	0	1869
0	1	1908
0	2	7
0	3	1932
0	4	1973
0	5	2240
1	0	3
1	1	2
1	2	2459
1	3	3323
2	0	4
2	1	5
3	0	6
3	1	0
7	0	8
7	1	14
7	2	1846
7	3	1847
7	4	1962

그림 6 고장수목 관련 테이블 구성예

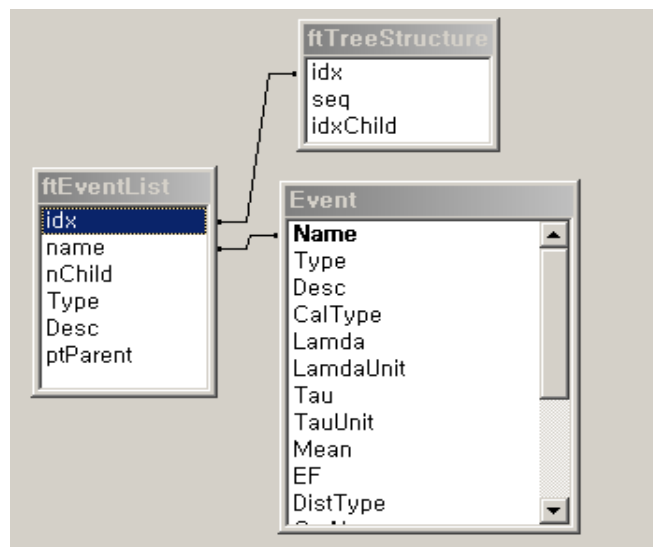


그림 7 고장수목 테이블 관계도

(2) 정량화 로직 개발

PSA 정량화를 자동으로 수행하기 위한 자동 정량화 로직은 다음 항목들을 기준으로 개발하였다.

- 1) FT/Data 읽기 : 고장수목 및 기본사건 데이터를 프로그램에서 읽어 들인다.
- 2) Branch Logic 개발 : 사건 수목의 브랜치별 처리 로직을 계산한다.
- 3) Sequence 계산 : 각 sequence 별로 노심손상빈도를 계산한다.
 - 먼저 노심 손상을 야기하는 경로의 별로 계산한다.
 - 성공항목(Heading)의 경우를 제외시킨다.
- 4) 정량화 결과의 저장 : 정량화 수행결과로 생성되는 최소단절군 집합을 저장한다.

따라서 PSA 자동 정량화 로직은 위에 언급한 항목의 순으로 계산되도록 개발하였는데, 다음 그림은 PSA 자동 정량화 수행 논리를 그림으로 나타낸 것이다.

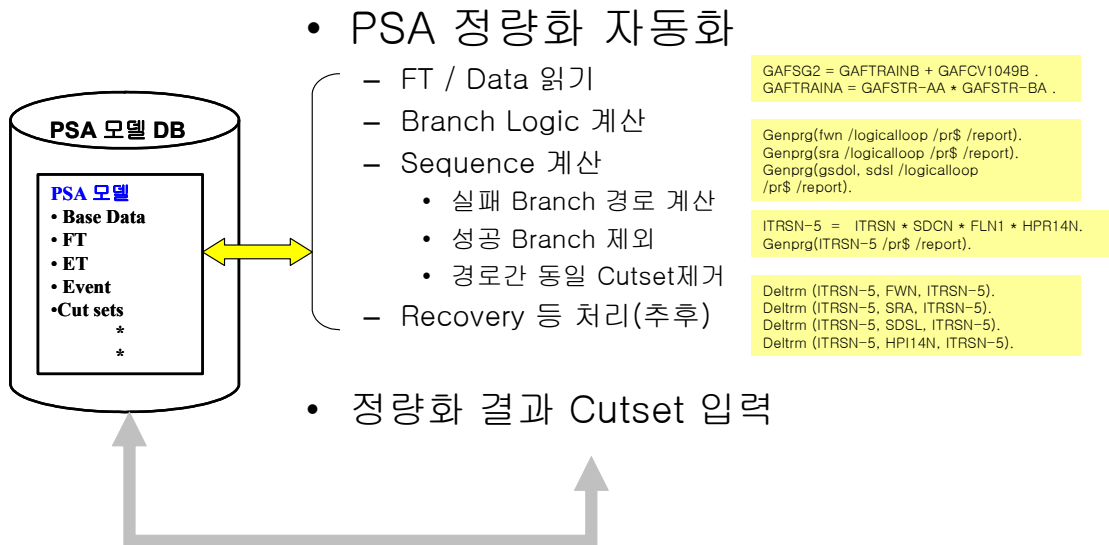


그림 8 PSA 자동 정량화 수행

(3) 최소 단절군 저장 및 처리

정량화 수행의 결과인 최소 단절군(Cutsets)을 저장하기 위한 로직을 개발하였는데, 최소단절군의 저장 방법은 다음과 같이 나누어 고려할 수 있다.

- 데이터베이스에 저장
- 파일시스템으로 저장

먼저 데이터베이스에 저장하는 경우는 최소 단절군 집합을 데이터베이스 테이블로 구성해서 저장하는 방식이고, 파일 시스템으로 저장하는 경우는 Raw파일 형태로 저장하는 방식이다. 이 두 가지 방식이 모두 장단점을 가지고 있는데, 데이터베이스 방식은 구현과 데이터의 재활용이 수월한 반면, 수행 속도가 느리고, 파일방식은 cutset 의 저장 및 읽는 시간이 빠르지만, 재활용이 수월하지 않은 단점이 있다.

본 연구에서는 위의 두 가지 방법을 모두 구현하였는데, 이는 사용자로 하여금 원하는 방식을 선택하여 사용할 수 있도록 하기 위함이다.

본 논문에서는 데이터베이스에 저장하는 방식에 대하여 테이블 구성 방법 및 기술하였다.

- CutSets

PSA 정량화 결과로 생성되는 최소 단절군 집합(CutSets)을 DB화하여 저장하기 위한 테이블이다.

CutSet은 크게 Data 블록과 CutSet 블록의 두 가지 정보로 구성되는데, Data 블록은 Cutset에 포함된 각 이벤트와 고장율 정보를 저장한 테이블이고, CutSet 블록은 각 최소 단절군 집합의 정보를 정형화하여 저장한 테이블이다.

CutSet 블록을 정형화하기 위해서 다음과 같이 두 가지 방법을 고려할 수 있는데, 첫째 방식은 세로 나열형 1칼럼 방식이고 둘째는 가로 나열형 테이블방식이다.

1 칼럼 방식은 각 Cutset의 정보를 한 칼럼에 세로로 표현하는 방식이고, 테이블 방식은 Cutset을 표현하기 위하여 테이블 행렬 방식으로 표현한 것이다.

즉, 1 칼럼 방식의 구조는 Event, Index, Prob 형식으로 구성된 세로 나열식이고, 테이블 방식은 Cutset을 Block, BE1, BE2, BE3, ... BEn 의 형식으로 가로 나열 식으로 구현한 것이다. 각 테이블의 특성은 다음과 같다.

울진 3,4호기 RM 모델기준

구분		세로 나열형(1칼럼 방식)	가로 나열형(Table 방식)
테이블구성형식		Block명, Event, Index, Prob	Block명, Event1, Event2, ... EventN
장점		하나의 컷셋을 구성하는 이벤트의 개수에 제한이 없다.	구현이 수월하다.
단점		DB 로의 적재 시간이 오래 걸린다.	컷셋을 구성하는 이벤트의 개수에 제한이 있다.
레코드수		컷셋내 포함된 이벤트 총수만큼 레코드가 생성됨.	컷셋 수만큼의 레코드 생성됨
소요시간	메모리 → DB	아주 오래걸림	오래 걸림
	DB → 메모리	조금 걸림	조금 걸림

표 1 최소단절군 테이블 구성 방식 비교

다음은 Cutset 데이터베이스의 테이블 구조이다.

csBase

- ID
- DataBlock
- nData
- CutSetBlock
- nCutset

csDataBlock

- IDX
- DataBlock
- Event
- Prob

csCutsetBlockCol : 세로 나열형 Cutset 구성

- ROW
- CutSetBlock : CutsetBlock 명
- Prob
- Col :Cutset Block 어레이의 세로 인덱스값
- BE
- RowIDX

csCutsetBlockTable : 테이블형식 가로 나열형 Cutset 구성

- ID
- CutSetBlock : CutsetBlock 명
- Prob
- nBE : 현 Cutset 레코드를 구성하고 있는 이벤트수
- BE1 : 1st Event
- BE2 : 2nd Event
- BE3 : 3rd Event
- *
- *
- BE_n : nth Event

이렇게 Cutset 블록을 두가지 테이블로 구성한 이유는 세로 나열형의 경우 확장성은 좋으나, 수행시간이 오래 걸리고, 가로 나열형(테이블형)의 경우 수행시간은 빠르나, 확장성이 결여되므로, 사용자가 정량화시 Cutset의 규모에 따라 알맞은 테이블 형식을 선택할 수 있도록 하기위해서이다.

다음그림은 세로나열식과 가로나열식의 Cutset블럭 구성예이다.

CutSetBlock	Row	Col	BE	Prob
GCDF-UCN	0	0	%U3-IE39	1.400e000
GCDF-UCN	1	0	%U3-IE40	4.700e-001
GCDF-UCN	2	0	%U3-IE19	4.400e-001
GCDF-UCN	3	0	%U3-IE1	2.800e-001
GCDF-UCN	4	0	PC-FRPCS	1.000e-001
GCDF-UCN	4	1	%U3-IE15	1.500e000
GCDF-UCN	5	0	PC-FRPCS	1.000e-001
GCDF-UCN	5	1	%U3-IE33	1.200e000
GCDF-UCN	6	0	%U3-IE28	9.000e-002
GCDF-UCN	7	0	%U3-IE38	8.000e-002
GCDF-UCN	8	0	%U3-LOOP	6.150e-002
GCDF-UCN	8	1	XF-LOOP-ENV1	1.000e000
GCDF-UCN	8	2	XF-LOOP-ENV2	1.000e000
GCDF-UCN	8	3	XF-LOOP-ENV3	1.000e000

그림 9 CutSet 블록(세로나열식-1컬럼)

ID	CutSetBlock	Prob	nBE	BE0	BE1	BE2	BE3		BE15
0	GCDF-UCN	1.400e000	1	%U3-IE39					
1	GCDF-UCN	4.700e-001	1	%U3-IE40					
2	GCDF-UCN	4.400e-001	1	%U3-IE19					
3	GCDF-UCN	2.800e-001	1	%U3-IE1					
4	GCDF-UCN	1.500e-001	2	PC-FRPCS	%U3-IE15				
5	GCDF-UCN	1.200e-001	2	PC-FRPCS	%U3-IE33				
6	GCDF-UCN	9.000e-002	1	%U3-IE28					
7	GCDF-UCN	8.000e-002	1	%U3-IE38					
8	GCDF-UCN	6.150e-002	4	%U3-LOOP	XF-LOOP-ENV1	XF-LOOP-ENV2	XF-LOOP-ENV3	*****	
9	GCDF-UCN	5.000e-002	1	%U3-IE5					
10	GCDF-UCN	4.600e-002	2	PC-FRPCS	%U3-IE34				
11	GCDF-UCN	3.000e-002	1	%U3-IE26					
12	GCDF-UCN	3.000e-002	1	%U3-IE6					
13	GCDF-UCN	3.000e-002	1	%U3-IE8					
14	GCDF-UCN	3.000e-002	1	%U3-IE36					
15	GCDF-UCN	3.000e-002	1	%U3-IE11					

그림 10 CutSet 블록(가로나열식-테이블)

3. 결론

PSA 정보 데이터베이스는 PSA 모델 및 각종 관련 정보들을 통합하여 데이터베이스에 저장함으로써, PSA 평가에 필요한 모든 입력 사항들을 제공하고 PSA 평가를 효율적으로 수행할 수 있도록 하는 기반을 제공하여 국내 위험도 정보 활용 및 PSA 기반 확대에 이바지하기 위한 시스템이다.

PSA 정보 데이터베이스는 크게 PSA 모델에 대한 정보를 저장하고 있는 PSA 모델 데이터베이스와 PSA 모델 이외의 기타 관련 정보를 저장하고 있는 PSA 추가 정보 데이터베이스로 구분되어 질 수 있다.

본 논문에서는 PSA 모델 데이터베이스에 저장된 PSA 모델을 이용하여 PSA 정량화를 자동으로 수행하여주기 위한 PSA 자동 정량화 시스템의 설계 및 구축에 관한 연구를 기술하였다.

먼저 PSA 자동 정량화를 위하여 PSA 모델 DB를 설계 및 구축 하였으며, PSA 모델 DB에 저장된 사건수목 및 고장 수목 등을 기초로 정량화를 수행하는 정량화 자동수행 로직을 개발하였다. 또한 정량화 계산 결과인 최소 단절군 집합을 저장하기 위한 로직을 개발하였다. 현재 개발된 정량화 로직들은 국내 정량화 엔진인 KIRAP과 FORTE 등과 손쉽게 연동될 수 있도록 개방된 인터페이스 구조를 가지고 있다.

그러나 완전한 PSA 자동 정량화 시스템의 구축을 위해서는 사건수목 에디터 및 고장수목 에디터 등으로부터 직접 사용자 시나리오 변경사항을 반영하여 정량화 로직에 전달될 수 있는 인터페이스의 구현이 필요하여, 현재 이러한 추가 인터페이스 연구가 수행 중에 있다.

향후 추가하여야할 연구 사항으로, 본 논문에서는 언급하지 않았지만, 기기 신뢰도 데이터베이스 등과 같이 PSA 통합 데이터베이스와 유기적으로 연동되어야할 시스템과의 효율적인 연계 체계 확립도 중요한 고려 사항인데, 이러한 타 시스템과의 인터페이스 즉 효율적인 자료 전달을 위한 향상된 데이터 전달 기법에 대한 연구가 필요할 것으로 생각된다.

참고문헌

1. 김승환 등, “PSA 통합 데이터베이스 구축에 관한 연구”, ‘02 추계 학술 발표회, 2002
2. 한상훈 등, “PSA Workstation KIRAP 개발” , KAERI/TR-847/97, 1997.
3. Jaejoo Ha et al., Development of an Integrated Reliabilities Analysis Code Package (KIRAP-Win) Based on the Object-Oriented Programming Technique,