KAERI/TR-3311/2007

Bayesian Belief Networks를 이용한 원자로보호계통 안전소프트웨어 설계명세의 정량 평가

A Study on Quantitative Assessment of Design Specification of Reactor Protection System Software Using Bayesian Belief Networks

한 국 원 자 력 연 구 소

제 출 문

한국원자력연구소장 귀하

본 보고서를 2006 연도 “정지/저출력 및 디지털 계통의 위험도 평가기술 개발” 과제의 기술보고서로 제출합니다.

2007. 2.

부서명 : 종합안전평부

주 저 자 : 엄흥섭

공 저 자 : 박기용

강현국

권기춘

장승철

요 약 문

원자력발전소의 안전을 평가하는 중요한 수단 중의 하나인 확률론적 안전성 평가(Probabilistic Safety Assessment: PSA)에 사용하기 위하여 소프트웨어 신뢰도의 정량적인 정보에 대한 실용적인 요구가 최근에 생겨나고 있다. 그러나 기존의 소프트웨어 신뢰도 정량평가 방법들은 PSA가 요구하는 충분한 정보를 제공할 수 없기 때문에 현재는 디지털 시스템을 포함하는 PSA의 경우 소프트웨어 부분을 배제하거나 또는 임의의 값을 사용하고 있는 실정이다. 본 보고서에서는 최근 불확실성을 포함하는 시스템의 모델링에 많이 활용되고 있는 Bayesian Belief Networks 기법을 이용하여 규칙 기반의 정성적인 소프트웨어 평가 방법론을 Bayesian Belief Networks로 모델링하고 PSA가 요구하는 정보를 생산할 수 있는 사례 연구에 대하여 기술하였다.

제안된 BBN 모델은 안전 소프트웨어의 신뢰도에 관계된 정성적인 증거와 정량적인 증거 모두를 결합하여 정형적이고 정량적인 방법으로 결론을 추론할 수 있는 BBN의 특성을 활용하여 구축되었다. 그리고 사례 연구로서 원자로 보호 계통에 탑재될 안전 소프트웨어 설계명세서의 품질을 평가하는 데 적용하였는데, 전문가에 의해 수행된 확인 및 검증 결과들이 모델의 입력으로 사용되었다. 만들어진 BBN 모델의 결과와 분석 내용은 전문가의 정성적인 판단과 유사하게 나타났으며 구축된 모델과 분석 내용들은 추후에 원전 안전계통 디지털 시스템의 PSA 및 KNICS V&V 업무에 활용될 예정이다.

SUMMARY

Probabilistic Safety Assessment (PSA), which is one of the important methods in assessing the overall safety of nuclear power plant (NPP), requires quantitative reliability information of safety-critical software. But the conventional reliability assessment methods which were developed for non-safety-critical software could not provide enough information for PSA of NPP, therefore current PSA which includes safety-critical software usually ignores the software portion or uses arbitrary values for it. In order to solve this problem this report proposes a method that can produce quantitative reliability of safety-critical software for PSA by making use of Bayesian Belief Networks (BBN). BBN has generally been used to model the uncertain system in many research fields.

The proposed method was constructed by utilizing BBN that can combine the qualitative and the quantitative evidence relevant to the reliability of safety-critical software. The constructed BBN model can infer a conclusion in a formal and a quantitative way. A case study was also carried out with the proposed method to assess the quality of software design specification of safety-critical software that will be embedded in reactor protection system. The V&V results of the software were used as inputs to the BBN model. The calculation results of the BBN model showed that its conclusion is mostly equivalent to those of the V&V expert for a given input data set. The method and the results of the case study will be utilized in PSA of digital safety system in NPPs. The method also can support the V&V expert's decision making process in controlling further V&V activities in KNICS.

목 차

<표 차례>

<그림 차례>

제1 장. 서론

디지털 시스템이 원자력발전소의 안전 계통에 사용되고 이에 따라 원전의 확률론적 안전성 평가(Probabilistic Safety Assessment: PSA)를 수행하는데 필요한 안전 소프트웨어 정량적 신뢰도 정보에 대한 필요성이 대두되었다. 한편 기존의 소프트웨어 신뢰도 정량평가 방법들은 그 대상이 안전 소프트웨어가 아니고 비안전 소프트웨어를 대상으로 연구되었고 또 안전소프트웨어의 특성 때문에 규제 기관은 물론이고 학계와 산업계에서도 일반적으로 인정되는 안전 소프트웨어를 위한 신뢰도 정량 평가 방법은 현재까지는 없는 실정이다[1]. 이런 문제점을 해결하기 위한 연구가 최근 다양한 방법으로 행해지고 있는데 본 보고서에서는 불확실성을 포함하는 시스템의 모델링에 많이 활용되고 있고 최근에는 안전 소프트웨어의 안전성 평가에도 응용되고[2] 있는 Bayesian Belief Networks를 이용한 방안과 그 사례 연구에 대하여 기술하였다. 사례 연구의 대상은 원자로보호계통에 사용되는 안전 소프트웨어의 설계명세서이며 동 소프트웨어 명세서에 대한 원전계측제어시스템 개발사업단(KNICS)의 V&V 방법론을 Bayesian Belief Networks로 모델링하고 V&V 팀의 정성적 평가 결과를 정량화하여 모델의 입력으로 사용하였다.

제안된 소프트웨어 설계명세서 평가 모델은 안전 소프트웨어의 신뢰도에 관계된 정성적인 증거와 정량적인 증거 모두를 결합하여 정형적이고 정량적인 방법으로 결론을 추론할 수 있는 BBN의 특성을 활용하여 구축되었다. 제 2장에서는 BBN의 개요와 동 기술을 이용한 안전 소프트웨어의 신뢰도 정량평가 방안에 대하여 기술하였고, 제 3장에서는 KNICS 원자로보호계통 안전 소프트웨어 설계명세서의 V&V 개요와, 설계명세서 평가용 BBN 그래프와 노드 확률 테이블의 작성, 모델 입력 자료의 작성, 그리고 모델의 계산 결과와 분석 내용에 대하여 기술하였다.

제2 장. BBN을 이용한 소프트웨어 신뢰도 정량 평가

제1 절. BBN

Bayesian Belief Networks(BBN)는 대상 시스템의 관련된 변수들을 인과관계에 의해 모델링하고 변수들 간의 종속성 정도를 조건부 확률로 나타낸 다음 관찰된 여러 가지의 증거를 만들어진 BBN 모델에 입력한 후 베이스(Bayes) 확률 정리를 비롯한 확률 법칙을 적용하여 계산하고 정량적 결과를 이끌어 내는 방법론이다.

BBN은 그래프 상에서 원으로 표시되는 노드(Node)와 노드들 사이를 연결하는 연결선(arcs 또는 directed edges) 그리고 각 노드에 속한 확률 테이블(Node Probability Tables: NPT 또는 Conditional Probability Table: CPT)로 구성되어 있다. 노드는 모델에 포함된 변수들을 나타내며 노드 연결선은 노드간의 인과관계를 나타낸다. 각 노드는 무작위 변수로서 몇 개의 상태를 가지고 있으며(예: "Yes"와 "No"의 상태) 각 상태의 확률 값의 합은 1이 된다. 각 노드에 연결된 노드 확률 테이블은 노드간의 연결 강도를 결정하며 모 노드(parent node)의 각 상태에 대한 조건부 확률로 표현된다[3].

BBN 상의 노드들은 목표노드(target node), 관찰가능 노드(observable node) 그리고 중간 노드(intermediate node)로 구분할 수 있다.

○ 목표 노드는 모델에서 평가 목적에 해당하는 노드로서 "프로그램의 무결함" 등이 될 수 있다.

○ 관찰가능 노드는 직접 관찰 가능한 노드로서 "N 번 테스트 중 M번 실패" 또는 "ISO 9000 품질요건 만족" 등이 될 수 있다. 이들 관찰가능 노드들은 정량화 된 수치이거나 또는 측정 가능해야 하는데 이 측정은 판단에 의한 주관적 확률 값도 가능하다.

○ 중간 노드는 제한된 정보나 믿음(belief)을 나타내는 것으로 "개발 과정의 품질" 또는 "제작자의 명성" 등이 여기에 해당된다.

BBN의 모델링 순서는 다음과 같다

1 단계: 노드(변수) 확인 (in a target system)
2 단계: 그래프 작성
3 단계: 노드 확률 테이블(NPT) 작성
4 단계: 증거 확보(명세서 평가 결과, 시험 결과, V&V 결과 등)
5 단계: 계산 및 분석(추론)

제2절. BBN을 이용한 안전소프트웨어 신뢰도 정량평가 방안

안전 소프트웨어(SW)의 정량적 신뢰도는 디지털시스템의 PSA에서 필수적인 요소 중 하나이나 현재의 기술 수준에서는 PSA에서 요구되는 SW의 정량적 신뢰도 정보를 얻을 수 없는 상태이다. 이를 해결하기 위하여 동 분야에서 현재 가장 가능성이 높은 기술 중의 하나로 인정받고 있는 Bayesian Belief Net(BBN)을 사용하여 안전 SW의 정량적 신뢰도를 구하는 방법을 그림 1과 같이 구현하였다. 동 방법론의 기본은 현재 인허가 및 개발 단계에서 사용되고 있는 소프트웨어의 정성적인 신뢰도 평가 체제를 BBN으로 모델링하고, 기존의 SW 엔지니어링 척도에서 얻어진 정량적 결과 및 V&V와 같은 정성적 평가 결과를 정량화한 입력을 사용하여 최종적으로 동 소프트웨어의 정량적 신뢰도 정보를 획득하는 것이다.

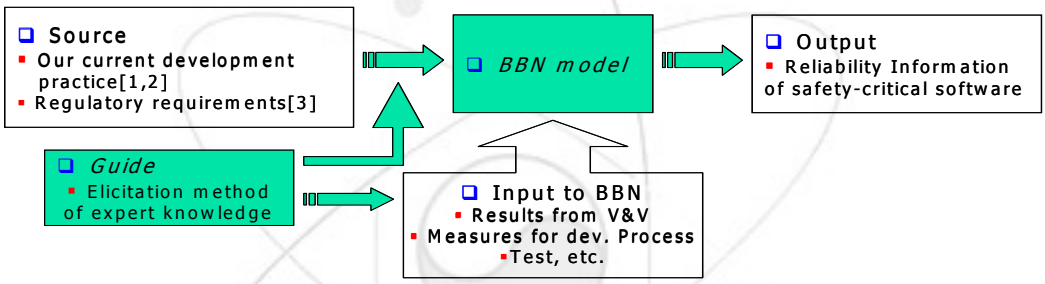

그림 1. BBN을 이용한 안전 SW 신뢰도 정량화 방법

이와 같이 개발된 BBN 방법론을 현재 원전계측제어시스템 개발사업단(KNICS)에서 개발 중인 “원자로보호계통 안전 필수 SW”의 개발 단계를 따라가면서 적용하여(그림 2 참조, 현재 요구명세 단계와 설계 단계까지 적용하였음) 동 방법론의 현장 적용 가능성과 그에 따른 문제점을 도출하였다.

그림 2. 원자로보호계통 안전 필수 SW 개발단계 별 BBN 방법론 적용 방안

제3 장. KNICS 원자로보호계통 SW 설계명세서 평가

제1 절. KNICS 소프트웨어 설계명세서의 확인 및 검증

KNICS의 원자로보호계통 소프트웨어 설계 검증은 KNICS에서 자체적으로 작성한 원자로보호계통 소프트웨어 설계명세 검증절차서[4]의 검증절차에 따라 수행된다. 소프트웨어 설계에 대한 확인 및 검증의 목적은 개념문서에서 정의된 원자로보호계통 소프트웨어의 기능, 성능 및 안전 요건에 부합되는 소프트웨어 설계명세들이 인허가 기준 및 기술적 관점에서 적합하게 작성되었는지를 확인하는 것인데 이는 (i) 소프트웨어 구조설계 검증, (ii) 상세설계 검증, 그리고 (iii) 상세 검증으로 이루어져 있다.

소프트웨어 구조설계 검증은 설계 단계의 첫 단계에서 작성된 소프트웨어의 구조설계 명세서에 대한 검토이며, 소프트웨어의 상세설계 검증은 소프트웨어 설계명세서의 완전성, 정확성, 일관성 등을 인허가 기준관점으로 검토하는 것이며, 상세 검증은 설계명세서에 대한 공학적 결정(engineering decision) 차원의 기술적인 검토를 하는 것이다. 각 단계의 검증은 소프트웨어 설계명세서의 특성 별로 구체적 질문 목록을 작성하여 평가하게 되는데 각 특성의 정의는 다음과 같다[4].

o정확도(accuracy) : 센서와 운전원의 입력에서 오류가 생기지 않는 정도, 근사치 또는 측정치에서 나타난 정확함의 정도, 그리고 작동기 출력에서 오류가 생기지 않는 정도이다.

o 기능성(functionality) : 소프트웨어에 의해서 수행되어야 할 동작이다. 기능이라 함은 일반적으로 원자로 운전에 영향을 미치는 입력 정보를 출력 정보로 변환하는 것이다. 입력은 센서류, 운전원, 다른 장비, 또는 다른 소프트웨어에서 받게 된다. 출력은 작동기, 운전원, 다른 장비 또는 다른 소프트웨어로 보내진다.

o 신뢰도(reliability) : 어떤 소프트웨어 시스템이나 기기가 고장 없이 동작하는 정도이다. 이 정의는 고장의 결말은 고려하지 않고 고장 발생만을 고려한 것이다.

o 강인성(robustness) : 어떤 소프트웨어 시스템이나 기기가 부정확한 입력 또는 응력의 환경조건을 받더라도 소정의 기능을 정확하게 발휘해 내는 능력이다. 이것은 그 명세서의 가정 사항과 어느 정도 다를지라도 정확하게 기능을 수행해 내는 능력도 포함한다.

o 안전성(safety) : 시스템의 안전성 고려사항에 직접 영향을 미치거나 또는 상호

연관되는 소프트웨어 시스템의 성질이나 특성이다. 이 SLCP에서 논의된 다른 특성은 소프트웨어-기반 안전계통의 전체 안전성에 미치는 중요한 기인자이지만, 그것은 일차적으로는 소프트웨어의 내적 동작에 관련된다. 그러나 안전성 특성은 소프트웨어가 시스템의 재해에 미치는 영향과 그러한 재해를 통제하기 위해 취해지는 수단에만 관련된다.

o 보안성(security) :

무단적이고, 불필요하고 불안전한 침투를 방지하기 위한 능력이다. 그러한 침투가 소프트웨어의 안전관련 기능에 영향을 줄 수 있는 한 보안은 안전성 현안이다.

o 타이밍(timing) : 사용 중인 계산시스템에 의해 부과된 그 타이밍 목적을 하드웨어 제약조건 내에서 달성해 내는 소프트웨어 시스템의 능력이다.

o 완전성(completeness) : 소프트웨어에서 요구되는 기능을 완전하게 구현해 내는 계획문서, 구현공정문서 및 설계 결과물의 속성이다. 소프트웨어가 수행해야 할 기능은 안전계통의 일반기능요건과 전체시스템의 설계에서 소프트웨어에 배정된 기능요건들로부터 비롯된다.

o 일관성(consistency) : 어떤 소프트웨어 시스템에 대한 여러 종류의 문서들과 기기 간에 서로 상반된 것이 없는 정도이다. 두 가지 관점의 일관성이 있다. 내적 일관성은 어떤 기기의 서로 다른 부분 내에서의 일관성으로서 예를 들면, 어떤 소프트웨어 설계는 만약 설계 구성요소들이 서로 상반되지 않는다면 내적으로 일관된 것이라고 말할 수 있다. 외적 일관성은 한 기기와 다른 기기 간의 일관성으로서 예를 들면, 소프트웨어 요건과 그 코드가 만약서로 간에 상반되지 않는다면 일관된 것으로 보아야 한다.

o 정확성(correctness) : 어떤 설계 결과물이 그 명세서, 설계, 그리고 구현에서 결함이 생기지 않을 정도이다. 정확성과 정확도 및 완전성과 같은 다른 특성 간에 서로 중첩하는 것이 바람직하다.

o 스타일(style) : 계획문서, 설계공정문서 및 설계 결과물의 형태와 구조이다. 문서 스타일은 어떤 문서의 구조와 형태를 말한다. 이것은 이해성(understandability), 판독성(readability), 그리고 수정성(modifiability)을 함축한 말이다. 프로그래밍 스타일은 소프트웨어의 프로그래밍 언어와 프로그래밍 그 자체의 특성을 말한다.

o 추적성(traceability) : 한 생명주기 제품의 각 요소가 어떤 선행 생명주기 제품의 하나 또는 그 이상의 요소들로 거슬러서 추적될 수 있고, 그리고 어떤 후행 생명주

기 제품의 하나 또는 그 이상의 요소로 추적될 수 있는 정도이다.

o 명확성(unambiguity) : 그 제품의 각 요소와, 모든 요소들이 서로 합쳐지더라도 한가지의 해석만을 갖는 정도이다.

o 확인성(verifiability) : 소프트웨어 계획문서, 설계공정문서 및 설계 결과물이 확인 기준의 수립과 그러한 기준이 만족되었는지를 결정하기 위한 분석, 검토, 혹은 시험의 수행을 쉽게 할 수 있도록 서술 또는 제공되는 정도이다.

제2절. BBN을 이용한 원자로보호계통 SW 설계명세 평가

전 단계의 연구에서는 원자로보호계통의 SW 요구명세서를 평가하는 BBN 모델을 구축하고 동 모델을 이용하여 정량화된 결과를 도출 및 분석하였고, 이번 연구에서는 요구명세 단계의 다음 단계인 설계 단계에서 작성된 SW 설계명세서를 평가하기 위하여 KNICS 설계문서 (원자로보호계통 소프트웨어 설계명세서[6] 및 동 명세서에 대한 V&V 절차서[4]와 보고서[5])를 대상으로 사례 연구를 수행하였다.

평가 작업의 첫 단계는 관련 문서(설계명세서 및 V&V절차서와 보고서)를 기본으로 BBN 모델을 구축하는 것이고 두 번째 단계는 모델에 필요한 입력 자료를 V&V 보고서로부터 추출하여 정량화하는 작업이며 마지막 단계는 이렇게 정량화된 입력 자료를 BBN 모델에 입력하여 관심의 대상이 되는 변수들의 값을 계산하고 이를 분석하는 것이다. 상세한 평가 내용은 다음과 같다.

1. 설계명세 평가용 BBN 모델

- 평가 모델은 ① 인허가에 활용될 수 있도록 기존 인허가 절차를 기반으로 한 모델, ② KNICS SW V&V 업무에 활용될 수 있는 내부 V&V 방법론을 기반으로 한 모델, ③ 위의 두 가지 모델에 공통으로 포함되는 구조설계 평가 모델 3가지로 나누어 구축하였고 최종적으로 이 세 가지 모델을 모두 합친 통합 모델을 구축하였다.

- 평가 모델을 구축하기 위해 KNICS 설계문서인 원자로보호계통 소프트웨어 설계명세서 및 동 명세서에 대한 V&V 절차서와 보고서로부터 변수를 도출하였다. 이렇게 도출된 각 서브 모델 별 특성과 변수의 수는 표 1과 같고 전체 통합 모델에는 약 300여개의 변수가 사용되었다.

표1. 설계명세 평가 모델 변수

모델\변수 형태	모델에 사용된 특성	변수 합계	목표 변수	중간 변수	입력 변수	비고
인허가 기반 모델	accuracy, reliability, robustness, safety, security, timing, completeness, consistency, correctness, style, traceability, verifiability	166	2	12	152	원자로보호계통 소프트웨어 설계명세 확인 및 검증 보고서에서 도출
V&V 기반 모델	traceability, correctness, consistency, completeness	65	2	16	47	원자로보호계통 소프트웨어 설계명세 확인 및 검증 보고서에서 도출
SW 구조 설계 평가 모델	reliability,safety,security,timing,completeness,consistency, style,traceability,verifiability	98	1	10	87	인허가 기반 모델과 V&V 기반 모델에 공통으로 포함됨
독립적 외부 변수	SRS_integrity, Dev_Process, V&V_Process, SDS_architecture	4	-	4	-	인허가 기반 모델과 V&V 기반 모델에 공통으로 포함됨

- 최상위 레벨 모델

그림 3에서 우측 상단에 있는 노드 “SDS_Design_Integrity"가 최종적으로 구하고자 하는 목표 변수이며 이를 구하기 위하여 구조설계명세 (SDS_Architecture) 평가, 내부 V&V 평가 (Detailed_Verification), 인허가 기준 평가 (SW_Design_V&V) 3개의 서브 그래프를 구축하였다.

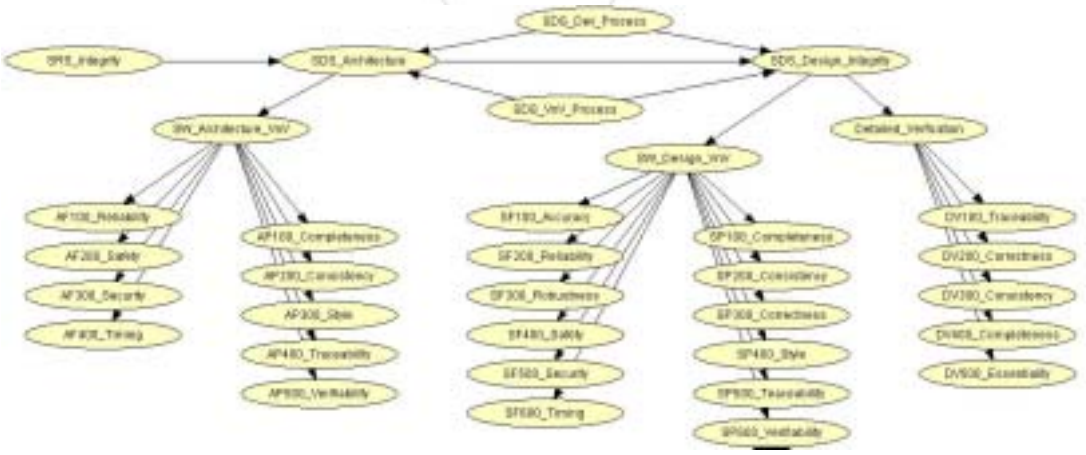

그림 3. 원자로보호계통 안전소프트웨어 신뢰도 평가용 BBN 그래프(최상위)

- 인허가 기반 모델

기존 인허가 절차와 기준을 기반으로 구축하였으며 정확성(accuracy)을 비롯하여 6개의 기능특성과 6개의 공정특성으로 구성되어 있고, 이들 특성에 속한 200여 개의 상세 체크리스트를 분석하여 변수를 도출하고 이를 토대로 BBN 그래프를 구축하였다. 각 변수 사이의 의존도(조건부 확률)는 KNICS V&V전문가가 일차로 작성하고 모델 전문가가 이를 검토하여 확정하였으며 구축된 모델의 BBN 그래프는 그림 4와 같다.

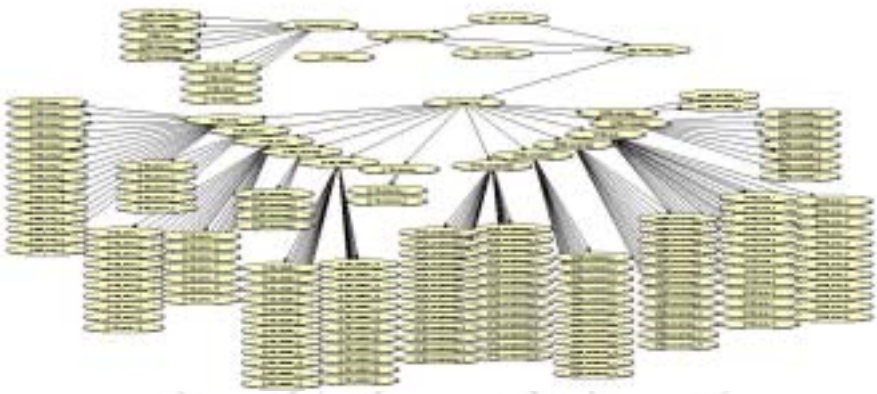

그림 4. 인허가 기반 BBN 모델 상세 그래프

- 내부 V&V 기반 모델

KNICS 내부 V&V 절차와 방법론을 기반으로 구축하였으며 추적성을 포함한 4개 특성과 여기에 속한 100여개의 상세 체크리스트를 분석하여 변수를 도출하고 BBN 그래프를 작성하였다. 각 변수 사이의 의존도 결정은 인허가 기반 모델과 동일한 방법으로 수행되었으며 구축된 모델의 BBN 그래프는 그림 5와 같다.

그림 5. 내부 V&V 기반 BBN 모델 상세 그래프

- SW 구조설계 평가 모델

SW 구조설계 평가 모델은 위에서 기술한 인허가 기반 모델과 내부 V&V절차 기반 모델에서 공통으로 사용하는 부분으로 확인성(completeness)을 포함한 9개 특성과 여기에 속한 상세 체크리스트를 분석하여 변수를 도출하고 그래프를 작성하였다. 각 변수 사이의 의존도 결정은 인허가 기반 모델과 동일한 방법으로 수행되었으며 구축된 모델의 BBN 그래프는 그림 6과 같다.

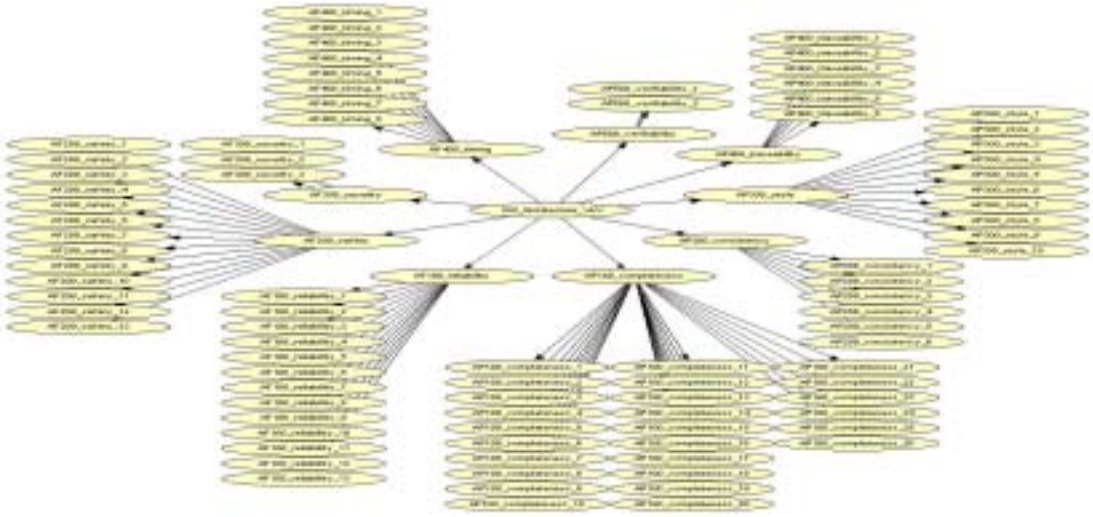

그림 6. SW 구조설계 평가 BBN 모델 상세 그래프

- 통합 모델

인허가 기반 모델, 내부 V&V 절차 기반 모델 그리고 구조설계 모델을 모두 통합한 전체 모델의 BBN 그래프는 그림 7과 같다. 그림 3(최상위 레벨 그래프)에서 나타난 특성별 노드에 각 특성에 속한 모든 세부 변수를 연결하여 작성되었으며 푸른색 점선으로 표시된 부분은 구조설계평가 모델, 붉은색 점선으로 표시된 부분은 내부 V&V기반 평가 모델, 그리고 녹색 점선으로 표시된 부분이 인허가 기반 평가 모델이다. 구조설계 평가 모델은 별도의 설계 문서(구조설계 명세서)를 참조하고 있으므로 문제가 없으나 인허가 기반 평가 모델과 내부 V&V 기반 모델은 동일한 설계 문서를 참조하고 있으며 또 평가에 사용되는 특성과 그 특성에 속한 세부 검증 내용이 이름은 다르나 실질적으로는 동일한 부분이 많다. 그래서 이 두 가지 모델을 통합하여 목표 변수의 값을 구하는 경우에는 모델 변수의 중복성 영향이 존재하므로 이를 고려할 필요가 있다.

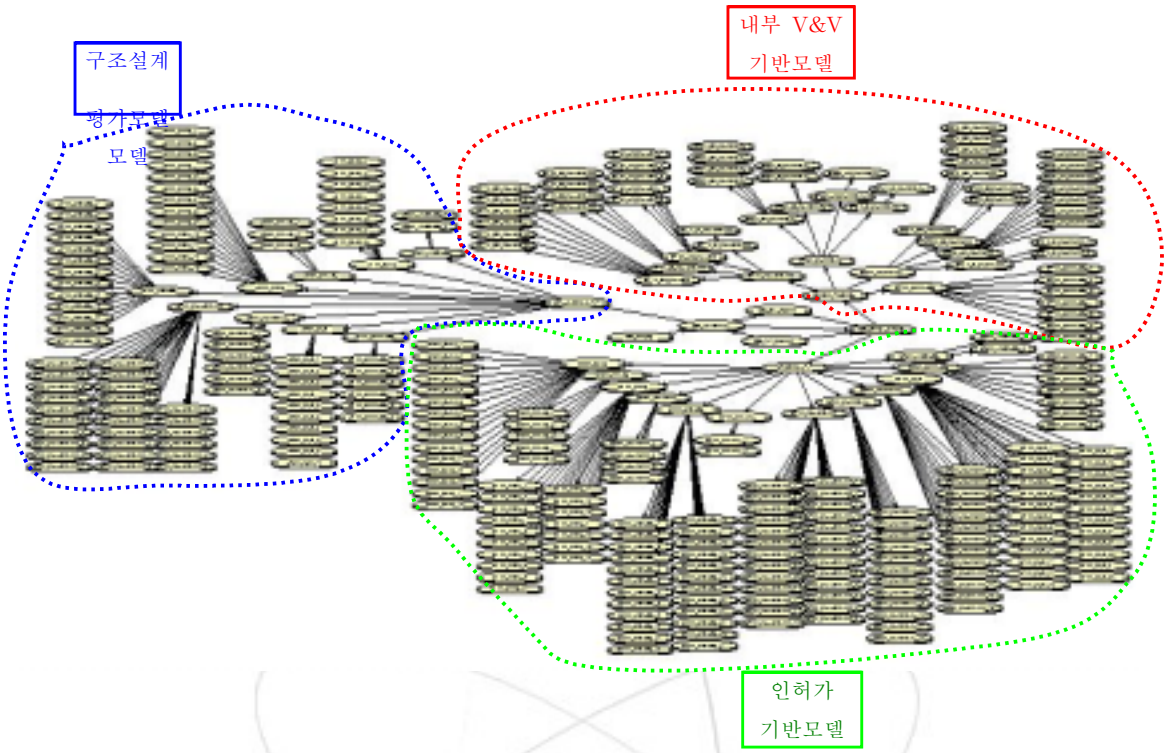

그림 7.원자로보호계통 소프트웨어 설계명세 평가용 BBN 그래프 (통합)

2. 노드확률테이블 작성과 노드 입력 값 (항목 평가치)

BBN 모델링의 목적은 관찰된 증거들에 근거하여 목표 노드의 값을 계산하는 것이고 따라서 모든 관찰 가능한 노드들의 값을 얻는 것이 필요하다. 물론 BBN에서 일부 또는 전체 관찰 가능 노드에 값을 입력하지 않고도 목표 노드의 값을 계산하는 것은 가능하지만 이럴 경우 목표 노드의 값은 완전하지 못하게 된다. 이들 관찰 가능 노드들에 입력할 값은 정성적인 형태나 또는 정량적인 형태로 얻어지는데 BBN에 사용되는 모든 값은 확률의 형식으로 표현되어야 하므로 정성적인 증거 값들은 정량 형태로 변환하는 것이 필요하다. 원자로보호계통 안전 소프트웨어 설계명세서를 평가하기 위한 본 BBN 모델에서는 모든 입력 자료를 KNICS V&V 보고서[5]의 평가 결과에 근거하여 보고서를 작성한 전문가와 모델 전문가가 semi-formal한 방법론과 기준 (KAERI/TR-2662/2004, 안전소프트웨어 신뢰도 정량평가 BBN을 위한 전문가 지식추출 지침)을 기준으로 작성하였으며 최종적으로 통합 모델에서 각 변수의 조건부 확률을 calibration하여 확정하였다. 그림 8은 정량화된 V&V 결과를 모델에 입력하여 계산하는 화면이다(사용 도구는 BBN 모델링 전용 Tool인 Hugin[7]).

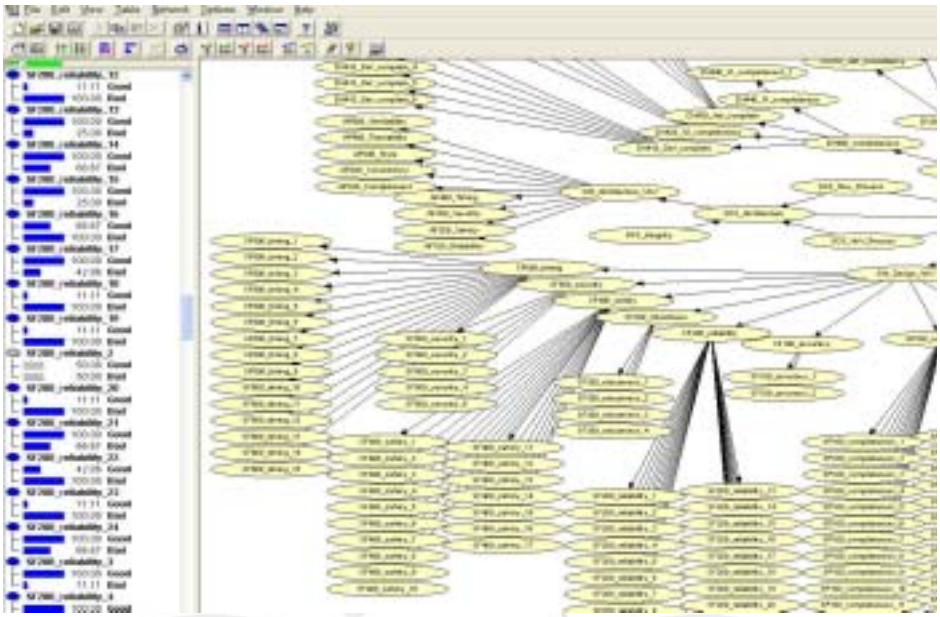

그림 8. BBN 모델 자료 입력 및 계산 화면(Hugin 도구)

3. 설계명세 평가 모델의 계산 및 결과 분석

통합 모델, 인허가 기반 모델 그리고 내부 V&V절차 기반 모델의 시나리오 별 중요 목표 변수(구조설계명세서와 상세설계명세서)의 평가 값은 그림 9 및 표 2와 같고, 중간 변수(각 서브 모델의 특성 값)의 계산 결과는 그림 10 및 표 3과 같다. 평가 시나리오 및 가정은 다음과 같이 설정하였다.

◇시나리오는 개발공정(SDS_Dev_Process), V&V공정(SDS_V&V_Process), SRS의 완전성이 미확인 된 경우(50%)와 완전한 상태(100%)인 경우의 두 가지로 설정하였음.

◇각 변수의 확률 값의 의미는 해당 변수의 요건 충족 상태로서 100%는 완벽한 상태, 0%는 요건이 전혀 충족되지 않은 상태임.

◇구조설계에 대한 부분은 본 연구가 종료되는 시점까지 평가가 이루어지지 않아 평가치가 없으므로 사전 확률을 50%로 설정하였고, 그림에 나타난 값은 모델의 계산에 따른 사후 확률임.

◇통합모델의 계산 결과는 위에서 언급되었던 변수의 중복성 영향은 고려하지 않은 상태임.

표 2. 시나리오 별 SDS 평가 주요 목표 변수의 계산 결과(단위: %)

주요 목표 변수	통합 모델		인허가기반 모델		내부 V&V기반 모델	
	개발/V&V 공정, SRS 50% 경우	개발/V&V 공정, SRS 100% 경우	개발/V&V 공정, SRS 50% 경우	개발/V&V 공정, SRS 100% 경우	개발/V&V 공정, SRS 50% 경우	개발/V&V 공정, SRS 100% 경우
SDS_Design_Integrity	99.37	99.91	95.00	99.29	89.22	98.38
SW_Design_V&V	99.99	99.99	99.99	99.99	-	-
Detailed_Verification	99.28	99.59	-	-	93.58	98.73

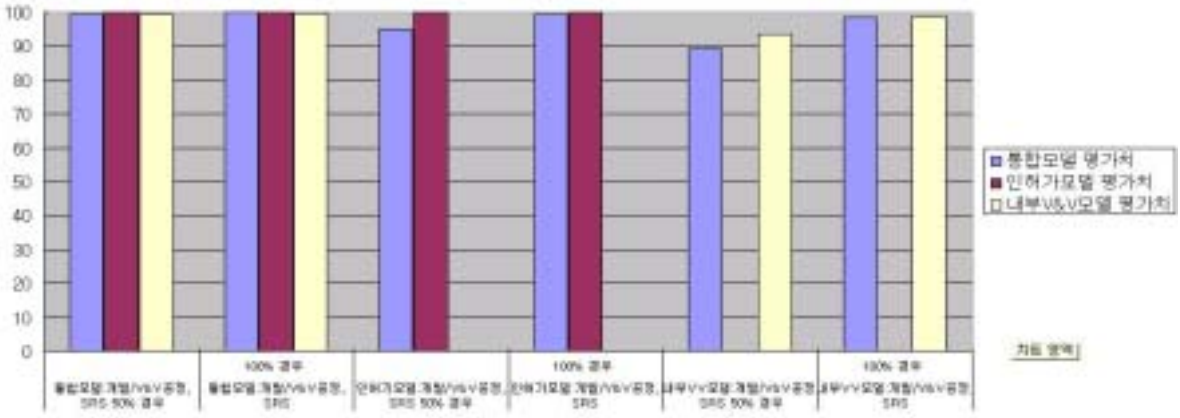

그림 9. 시나리오 별, 모델별 설계명세서 평가 값 (Y축 단위: %)

표 2에서 변수 'SDS_Design_Integrity'는 통합모델의 목표 변수이며 변수 'SW_Design_V&V'는 인허가기반 평가 모델의 목표 변수이고 변수 'Detailed_Verification'은 내부 V&V 모델의 목표 변수이다. 구조설계 평가 모델의 경우 본 보고서가 작성되는 시점까지 V&V가 수행되지 않은 상황이어서 실제 평가 값을 입력할 수 없었으며 따라서 모델의 계산에서는 여기에 속한 상세 변수들의 값들은 모두 기본 값(0.5)을 사용하였다. 계산 결과를 보면 통합 모델, 인허가기반 모델, 내부 V&V 모델 모두 유사하게 나타났는데, 이는 인허가기반 모델과 내부 V&V 모델이 비록 그 형태는 다르나 내용상으로는 유사한 특성을 평가하고 있기 때문이며 따라서 통합 모델의 결과도 별 차이가 없이 나타났다. 한편 구조설계 모델의 결과 값도 초기 입력 상태보다 훨씬 우수한 것으로 나타났는데 이것은 BBN의 변수 간 종속성 결과에 따른 것으로 보이며 이는 전문가의 일반적인 추론 기준과도 부합된다. 마찬가지 이유로 내부 V&V 기준 모델만 단독으로 사용한 경우 목표 변수의 값이 비교적 적게 나타났다. 표 3은 시나리오와 각 모델 별 주요 중간 변수의 계산 결과를 보여주고 있다.

표 3. 시나리오 별 주요 중간 변수의 계산 결과(단위 %)

주요 중간 변수	통합 모델		인허가기반 모델		내부 V&V기반 모델	
	개발/V&V 공정, SRS 50% 경우	개발/V&V 공정, SRS 100% 경우	개발/V&V 공정, SRS 50% 경우	개발/V&V 공정, SRS 100% 경우	개발/V&V 공정, SRS 50% 경우	개발/V&V 공정, SRS 100% 경우
DV100_traceability	40.81	40.91	-	-	38.92	40.62
DV200_correctness	99.79	99.83	-	-	99.04	99.72
DV300_consistency	97.78	97.83	-	-	96.88	97.69
DV400_completeness	99.89	99.89	-	-	99.89	99.89
SF100_accuracy	95.84	95.84	95.84	95.84	-	-
SF200_reliability	99.75	99.75	99.75	99.75	-	-
SF300_robustness	98.00	98.00	98.00	98.00	-	-
SF400_safety	99.82	99.82	99.82	99.82	-	-
SF500_security	11.53	11.53	11.53	11.53	-	-
SF600_timing	90.23	90.23	90.23	90.23	-	-
SP100_completeness	99.99	99.99	99.99	99.99	-	-
SP200_consistency	99.98	99.98	99.98	99.98	-	-
SP300_correctness	97.55	97.55	97.55	97.55	-	-
SP400_style	99.97	99.97	99.97	99.97	-	-
SP500_traceability	98.43	98.43	98.43	98.43	-	-
SP600_verifiability	52.30	52.30	52.30	52.30	-	-
SDS_Dev_Process	* 67.77	100	* 66.20		* 64.12	100
SDS_V&V_Process	* 67.77	100	* 66.20		* 64.12	100
SRS_Integrity	* 54.94	100	54.50		* 53.92	100
SDS_architecture	73.70	92.03	71.60		68.83	95.00

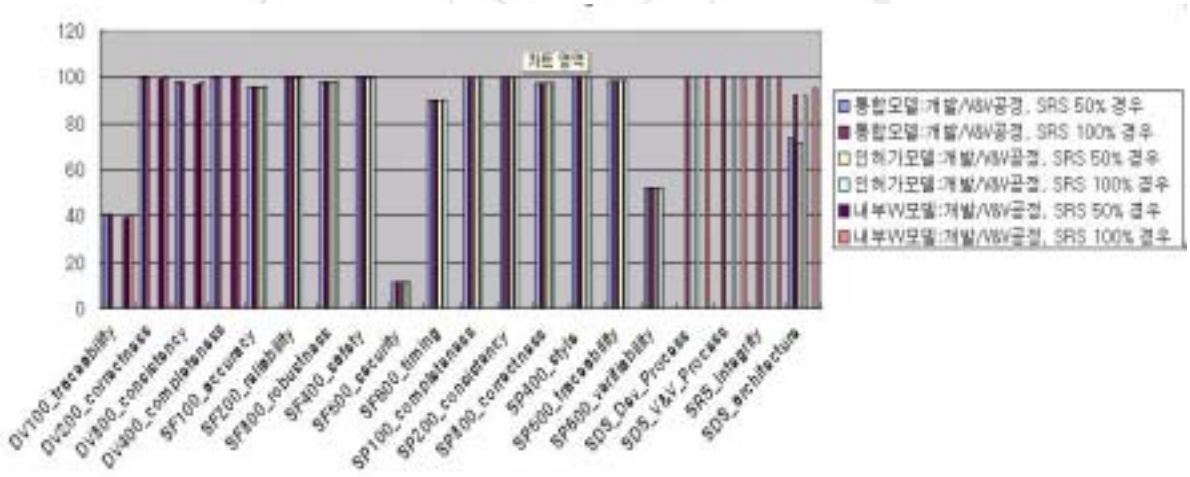

그림 10. 시나리오 별, 모델 별 해당 모델의 특성 평가 값 (Y축 단위: %)

제4 장. 요약 및 결론

전 단계 연구에서 Bayesian Belief Net 기술을 활용하여 원전 안전 소프트웨어의 신뢰도를 평가할 수 있는 방법론을 개발하였고 이번 연구에서는 이를 KNICS에서 현재 개발 중인 원전 보호계통 소프트웨어의 설계명세 평가에 적용하였다. 평가 모델은 ① 인허가에 활용될 수 있도록 기존 인허가 절차를 기반으로 한 모델, ② KNICS SW V&V 업무에 활용될 수 있는 내부 V&V 방법론을 기반으로 한 모델, ③ 위의 두 가지 모델에 공통으로 포함되는 구조설계 평가 모델 3가지로 나누어 구축하였고 최종적으로 이 세 가지 모델을 모두 합친 통합 모델을 구축하였다. 동 연구를 통하여 BBN 기술을 활용하여 안전 소프트웨어의 정량적 신뢰도 정보를 얻는 가능성을 입증하였고 또한 실제 문제에 적용 시 발생하는 어려운 점들을 확인하였다. 개발된 방법론은 일차적으로는 디지털 시스템의 PSA에서 요구하는 소프트웨어 신뢰도 정보를 제공할 수 있고, 부차적으로는 안전 소프트웨어의 개발 각 단계별로 V&V 활동의 의사결정 지원 도구로 사용될 수 있으며, 디지털 시스템의 인허가에서 필요로 하는 소프트웨어 신뢰도에 대한 근거 자료로 활용될 수 있다.

참고문헌

[1] R.W. Butler and G.B. Finelli, The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software, IEEE Transactions on Software Engineering, 19(1), 1993

[2] Neil, M., et al, "Applying Bayesian Belief Networks to System Dependability Assessments," Proceedings of Safety Critical Systems Club, February 1996.

[3] Jensen, F., An Introduction to Bayesian Belief Networks, Springer-Verlag, New York, NY, 1996.

[4] Gee-Yong, P., 원자로보호계통 소프트웨어설계명세 검증절차서, KNICS - RPS - SVP131, Rev.0.0, 2006.4.

[5] Gee-Yong, P., 원자로보호계통 비교논리프로세서 소프트웨어 설계명세 확인 및 검증 보고서, KNICS - RPS - SVR131-01, Rev. 00, 2006.

[6] Young-Ho, K., et al, 2001. 원자로보호계통 비교논리프로세서 소프트웨어 설계명세서, KNICS-RPS-SDS231-10, 두산중공업, 2005.

[7] Hugin, www.hugin.dk, Aalborg, Denmark

부록 A. 원자로 보호계통 SW(BP) 설계명세 평가를 위한 BBN의 변수 목록
부록 B. 원자로 보호계통 SW(BP) 설계명세 평가를 위한 BBN의 노드확률테이블
부록 C. 원자로 보호계통 SW(BP) 설계명세 평가를 위한 BBN 입력 값

부록A. 원자로 보호계통 SW(BP) 설계명세서 평가를 위한 BBN의 변수 목록

* Node Initial Code: 구조설계:A, 상세설계:S, 기능특성:F, 공정특성:P, 상세검증:DV

1. SDS 구조설계 검증

기능 특성/코드	공정 특성/코드
신뢰성(reliability)/AF1	완전성(completeness)/AP1
안전성(safety)/AF2	일관성(consistency)/AP2
보안성(security)/AF3	스타일(style)/AP3
타이밍(timing)/AF4	추적성(traceability)/AP4
	확인가능성(verifiability)/AP5

1.1 기능 특성 검증

◇ 신뢰성(reliability) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
신뢰성: AF100_reliability		
1	AF100_reliability_1	소프트웨어 구조에 의해 고장을 감지하였을 경우 취해지는 행위들을 명시하고 있는가?
2	AF100_reliability_2	프로그램 수행 동안 고장파급을 방지할 수 있는 기술이 사용되었는가?
3	AF100_reliability_3	컴퓨터 시스템 구조 설계 시 소프트웨어 공통모드 고장 가능성을 분석하고 고려하였는가?
4	AF100_reliability_4	소프트웨어 구조가 사고발생시 시스템 상태를 기록할 기능을 가지고 있는가?
5	AF100_reliability_5	신뢰도와 가용성 분석이 신뢰도 요구사항 만족을 보장할 수 있도록 충분한 보충자료를 포함하고 있는가?
6	AF100_reliability_6	소프트웨어 설계구조가 고장 감지 시 잘 정의된 출력을 생산할 능력을 포함하는가?
7	AF100_reliability_7	소프트웨어 구조가 소프트웨어와 하드웨어의 상태를 명시할 기능을 가지고 있는가?
8	AF100_reliability_8	하드웨어와 소프트웨어 오류를 처리할 표준 전략이 있는가?
9	AF100_reliability_9	소프트웨어 구조는 감지된 하드웨어, 소프트웨어 고장을 보고할 기능을 가지고 있는가?
10	AF100_reliability_10	소프트웨어 구조는 적절한 watch-dog 타이머를 가지고 있는가?
11	AF100_reliability_11	개별 하드웨어 단위의 고장이 컴퓨터 시스템 고장을 야기하지 않는 구조인가?
12	AF100_reliability_12	디지털 시스템이 운전원 오류를 감지하거나 감내할 것인가?
13	AF100_reliability_13	소프트웨어 구조가 부적절한 운전원 행위를 통보할 방법을 가지고 있는가?

◇ 안전성(safety) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
안전성: AF200_safety		
1	AF200_safety_1	제어 로직이 오류처리, 비정상 및 비상처리 요구사항을 정확히 구현하고 있는가?
2	AF200_safety_2	각 필수 안전구조 요소의 결함분석이 수행되었는가?
3	AF200_safety_3	소프트웨어 구조가 다른 안전 중요도를 가지는 하부 계통간의 상호작용을 엄격하게 제어하는가?
4	AF200_safety_4	소프트웨어 구조가 소프트웨어 상태를 안전하고 알려진 상태로 되돌리도록 하는 비정상 감지 검사기능을 가지고 있는가?
5	AF200_safety_5	소프트웨어 구조에 대하여 설계 제한분석을 수행하였는가?
6	AF200_safety_6	모든 안전-필수 소프트웨어의 소프트웨어 구조에 대한 소프트웨어 안전성분석이 수행되었는가?
7	AF200_safety_7	각 안전-필수 데이터 항목을 변경할 수 있는 구조적 요소가 파악되었는가?
8	AF200_safety_8	소프트웨어 구조의 제어 로직이 안전-필수 요구사항을 완전하고 정확하게 구현하고 있는가?
9	AF200_safety_9	안전에 영향을 줄 시간, 비용, 복잡도 또는 특별한 문제가 있다면, 완화계획을 통해 기록되고 있는가?
10	AF200_safety_10	소프트웨어 구조가 자원에 치명적인 요소를 식별할 수 있도록 분석되었는가?
11	AF200_safety_11	소프트웨어 계통이 그 환경에서 낮은 확률사고나 고장에 어떻게 반응할 것인가를 결정하기 위해 그 소프트웨어 구조가 분석되었는가?
12	AF200_safety_12	소프트웨어 구조설계가 소프트웨어에 할당된 모든 안전 요구사항과 부합함을 분석하였는가?
13	AF200_safety_13	모든 비-안전 구조요소가 소프트웨어 안전에 급작스럽게 영향을 주지 않음을 분석하였는가?
14	AF200_safety_14	안전에 중요한 기능이 잘 정의되고 엄격하게 통제되는 연계에 의하여 정상운전 및 부가기능과 분리되어 있는가?

◇ 보안성(security) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
보안성: AF300_security		
1	AF300_security_1	구조가 소프트웨어 요구사항 명세(SRS)에 명시된 보안 위협들을 정확하게 다루고 있는가?
2	AF300_security_2	구조가 SRS에 명시된 접근제한을 정확하게 다루고 있는가?
3	AF300_security_3	구조가 비-승인자의 침입을 감지할 능력을 가지고 있는가?

◇ 타이밍(timing) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
타이밍: AF400_timing		
1	AF400_timing_1	소프트웨어 구조 설계명세에서 스케줄링 및 프로세스간의 통신방식이 기술되어 있는가?
2	AF400_timing_2	소프트웨어 구조 설계명세가 모든 타이밍 제한사항, 각 제한사항에 대한 전략, 요구되는 여유도 및 여유도 측정방법 등에 대해 기술하고 있는가?
3	AF400_timing_3	각 구조 요소에 대한 최소 및 최대 수행시간이 명시되어 있는가?
4	AF400_timing_4	소프트웨어 구조가 응답시간을 만족하는지 분석되는가?
5	AF400_timing_5	운전이 정확한 순서로 수행되었음을 보장하도록 구조가 분석되었는가?
6	AF400_timing_6	컴퓨터 자원의 스케쥴링이 예측 가능하고 결정적으로 수행되는지 보장하기 위한 분석을 수행하는가?
7	AF400_timing_7	소프트웨어 구조 설계명세에 포함된 구조설계 요소에 대한 크기(Size)가 추정되는가?
8	AF400_timing_8	저장장치 크기 추정치가 운전, 유지보수 및 잠정적 개선을 위한 충분한 여유를 가지고 있는가?

1.2 공정 특성 검증

◇ 완전성(completeness) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
완전성: AP100_completeness		
1	AP100_completeness_1	소프트웨어 구조가 소프트웨어 설계명세, 안전성분석보고서, 요구사항명세서 등에 있는 모든 예상되는 상황과 조건을 고려하는가?
2	AP100_completeness_2	소프트웨어 구조설계 명세가 SRS에 명시된 설계 제한사항과 소프트웨어 요구사항을 만족하는 소프트웨어 구조를 묘사하고 있는가?
3	AP100_completeness_3	소프트웨어를 프로그램 단위들이 높은 내부 응집도를 갖도록 소프트웨어 구조가 설계되었는가?
4	AP100_completeness_4	소프트웨어를 프로그램 단위들 간에 낮은 연관도를 갖도록 소프트웨어 구조가 설계되었는가?
5	AP100_completeness_5	프로그램 요소들이 명료한 시스템 구조가 되도록 체계적으로 구성되어 있는가?
6	AP100_completeness_6	각 시스템 기능이 특정 소프트웨어 요소에서 수행되는가?
7	AP100_completeness_7	소프트웨어 구조가 소프트웨어 요소간의 논리적 연결을 보여 주는가?
8	AP100_completeness_8	소프트웨어 설계를 위해 일반적으로 인정된 소프트웨어공학 기법이 사용되었는가?
9	AP100_completeness_9	소프트웨어 구조가 소프트웨어 요소들 간의 데이터 흐름과 제어흐름을 보여 주는가?
10	AP100_completeness_10	운전환경이 구조설계에 반영되었는가?
11	AP100_completeness_11	소프트웨어 구조가 각 인터페이스의 요구되는 행동을 허용하는가?
12	AP100_completeness_12	소프트웨어 구조 설계가 SRS에 명시된 모든 운전모드를 고려하는가?

13	AP100_completeness_13	소프트웨어 계통의 설계와 실 세계 제한사항에 의해 부과된 어떠한 제약사항을 평가하기 위하여 제한분석이 수행되었는가?
14	AP100_completeness_14	데이터-관련 구조요소가 소프트웨어 요구사항과의 일치 여부를 결정하기 위한 자료분석이 수행되었는가?
15	AP100_completeness_15	소프트웨어 구성요소간의 인터페이스가 정확하게 설계되었는지를 결정하기 위한 인터페이스 분석이 수행되었는가?
16	AP100_completeness_16	특별한 어려움을 야기하는 모든 특정 구조 설계영역이 파악되고, 각각을 위한 완화 계획이 묘사되어 있는가?
17	AP100_completeness_17	소프트웨어 구조가 시스템과 개발노력에 부과된 제약사항 내에서 구현되었는가?
18	AP100_completeness_18	소프트웨어가 운전요원에 부담을 주는 것이 아니라 도움을 준다는 것을 보장할 수 있도록 소프트웨어 구조가 분석되고 검토되었는가?
19	AP100_completeness_19	모든 데이터 채널의 응답시간, 샘플링 빈도, 숫자, 크기, 데이터 율(data rate) 등이 소프트웨어 구조설계 명세에 정의되어 있는가?
20	AP100_completeness_20	운전원이나 보수요원이 쉽게 디지털 시스템 상태를 평가하고 문제영역을 찾아낼 수 있도록 적절한 인간공학 고려사항들이 구조설계에 포함되어 있는가?
21	AP100_completeness_21	구조 설계에 하드웨어 환경이 고려되었는가?
22	AP100_completeness_22	구조 설계가 각 소프트웨어 요소가 동작할 하드웨어 요소를 명시하고 있는가?
23	AP100_completeness_23	특정 하드웨어가 요구된다면 소프트웨어 구조설계 명세가 저장장치 용량, 명령어 집합, 속도, 입출력 레지스터 등을 포함하여 그 컴퓨터 내부의 각 컴퓨터를 파악하는가?
24	AP100_completeness_24	설계구조 명세에 주 하드웨어 요소가 나타나는가?
25	AP100_completeness_25	소프트웨어 구조는 하드웨어 요소들간의 물리적 연결을 보여 주는가?
26	AP100_completeness_26	하드웨어 구조가 존재하는가?

◇ 일관성(consistency) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
일관성: AP200_consistency		
1	AP200_consistency_1	소프트웨어 구조설계에서 사용된 하나 이상의 정형기법(formal method)은 이들 상호간의 일관성을 유지하고 있는가?
2	AP200_consistency_2	소프트웨어 구조설계 설명서(software architecture design description)의 표현 스타일과 상세화 수준이 일관성을 갖는가?
3	AP200_consistency_3	소프트웨어 구조도는 소프트웨어 구조설계 설명서 및 프로그램 작동환경과 일치하는가?
4	AP200_consistency_4	소프트웨어 구조도의 요소는 소프트웨어 요구사항 명세서와 일치하는가?
5	AP200_consistency_5	소프트웨어 구조도는 소프트웨어 구조도 요소(component)간의 상관관계의 정확성(correctness)을 입증할 수 있도록 분석되었는가?
6	AP200_consistency_6	소프트웨어 구조도는 하드웨어 구조와 일치하는가?

◇ 스타일(style) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
스타일: AP300_style		
1	AP300_style_1	소프트웨어 구조설계 설명서(software architecture design description)는 각 구조설계 요소의 기능(function)을 서술하고 있는가?
2	AP300_style_2	소프트웨어 구조설계 설명서는 구조설계 요소들간의 관계성(relationship)을 기술하고 있는가?
3	AP300_style_3	소프트웨어 구조설계 문서상에 설계구조 선택의 타당성을 설명하고 있는가?
4	AP300_style_4	소프트웨어 설계요소 중 처리(processing) 부분을 포함하고 있다면 소프트웨어 구조설계 설명서에 처리기능의 요소를 실행하는 방법(method)을 기술하고 있는가?
5	AP300_style_5	소프트웨어 구조설계 설명서에 각 구조설계 요소의 기능수행을 위하여 필요한 자원(resources)을 규정하고 있는가?
6	AP300_style_6	소프트웨어 설계구조 요소의 상호작용 설명에는 타이밍 트리거 사건들, 실행의 순서, 자료공유, 상호작용에 영향을 미치는 다른 요소 등을 포함하고 있는가?
7	AP300_style_7	소프트웨어 구조설계 설명서에 각 설계요소의 유형(type)을 확인할 수 있는가?
8	AP300_style_8	소프트웨어 구조설계 설명서에 적용한 표준(standard)을 확인할 수 있는가
9	AP300_style_9	만약 소프트웨어 설계요소가 데이터저장(data store), 메시지 또는 스크린 디스플레이라면 소프트웨어 구조설계 설명서에는 이들에 대한 일반적인 구조를 설명하고 있는가?
10	AP300_style_10	소프트웨어 구조설계의 각 요소들은 유일한 명칭을 가지고 있는가?

◇ 추적성(traceability) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
추적성: AP400_traceability		
1	AP400_traceability_1	소프트웨어 구조설계 요소가 소프트웨어 요구사항 명세서와 일치하는지 확인하는데 사용하는 기법인 특정 시험(specific tests) 또는 검증기준(validation criteria)을 이용하여 전향추적을 할 수 있는가?
2	AP400_traceability_2	소프트웨어 구조설계 설명서에는 소프트웨어 구조를 결정짓는데 적절히 활용될 수 있도록 소프트웨어 구조설계에 대한 참조 주목사항(notes)을 포함하고 있는가?
3	AP400_traceability_3	소프트웨어 구조설계 설명서에 소프트웨어 요구사항 명세서에 기술된 요건의 범위를 벗어나는 어떤 기능을 포함하고 있다는 정당한 근거가 있다면 요구사항의 범위를 벗어난 소프트웨어 구조설계 설명서에 대한 정당성 결과가 요구사항 명세서와 일치하는가?
4	AP400_traceability_4	소프트웨어 구조설계 설명서에는 소프트웨어 구조설계의 각 기능에 대해 확인할 수 있는 특정 식별자(ID)를 부여하고 있어서 구현 시 이들 식별자를 통해 추적할 수 있는가?

5	AP400_traceability_5	각 소프트웨어 구조설계 요소는 소프트웨어 요구사항 명세서의 특정 요소(specific elements)를 후향 추적(backward trace) 할 수 있도록 되어있는가?
6	AP400_traceability_6	소프트웨어 구조설계 상에서 소프트웨어 요구사항 명세서에 기입된 설계 제약조건 등을 점검할 수 있도록 되어 있는가?

◇ 확인가능성(verifiability) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
확인가능성: AP500_verifiability		
1	AP500_verifiability_1	소프트웨어 구조설계 사양은 각 소프트웨어 설계요소의 기능에 대해 시험할 수 있는 방법을 기술하고 있는가?
2	AP500_verifiability_2	시스템 및 하부시스템 시험절차서, 시험 사례 들을 개발하기 위하여 소프트웨어 구조도 분석을 하였는가?

2. SDS 상세설계 검증

기능 특성/코드	공정 특성/코드
정확도(Accuracy)/SF1	완전성(completeness)/SP1
신뢰성(reliability)/SF2	일관성(consistency)/SP2
강인성(Robustness)/SF3	정확성(Correctness)/SP3
안전성(safety)/SF4	스타일(style)/SP4
보안성(security)/SF5	추적성(traceability)/SP5
타이밍(timing)/SF6	검증확인성(verifiability)/SP6

2.1 기능 특성 검증

◇ 정확도(Accuracy) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
정확도: SF100_accuracy		
1	SF100_accuracy_1	소프트웨어 상세설계에서 사용한 모든 계산들은 계산에 대한 정확성 요구사항을 만족하는가?
2	SF100_accuracy_2	소프트웨어 상세 설계 시 입력변수 값을 검증하기 위한 방법을 기술하고 있는가?

◇ 신뢰성(reliability) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
신뢰성: SF200_reliability		
1	SF200_reliability_1	소프트웨어 설정(configuration)에서 초기화 검사(initialization checks)가 있는가?
2	SF200_reliability_2	데이터 일관성(consistency)을 보증하기 위해 해야 할 것이 무엇인가?
3	SF200_reliability_3	교착상태(deadlock)에 대한 잠재성이 존재하는지, 그리고 그와 같은 잠재성이 있다면 무슨 단계들이 교착상태를 방지하기 위해 취해져야 할 것인지를 결정하기 위해 분석이 수행되고 문서화되어야 하는가?
4	SF200_reliability_4	시간 종료검사(time-out checks-software watch-dogs)가 소프트웨어에 적용되고 있는가?
5	SF200_reliability_5	공통-모드 소프트웨어 고장(common-mode software failures)이 소프트웨어 상세 설계에서 분석되고 고려되는가?
6	SF200_reliability_6	고장 이벤트에서 시스템 상태를 기록하는 설비를 소프트웨어 상세설계에서 포함하고 있는가?
7	SF200_reliability_7	오류탐지 이벤트에서 취해지는 조치사항(actions)을 소프트웨어 상세설계에서 확인하고 있는가?
8	SF200_reliability_8	프로그램을 수행하고 있는 동안 고장 파급(failure propagation)을 방지하기 위한 기법들을 사용하고 있는가?
9	SF200_reliability_9	고장이 탐지되는 이벤트에서 정의가 잘 된 출력의 생성능력(capacity)을 소프트웨어 상세설계에서 포함하고 있는가?
10	SF200_reliability_10	완전하고 정확한 오류 회복 기법(error recovery techniques)을 소프트웨어 상세설계에서 명세하고 있는가?

11	SF200_reliability_11	소프트웨어 상태를 표시하는(displaying) 설비가 있는가?
12	SF200_reliability_12	신뢰도 및 가용도 분석이 신뢰도 요구사항을 만족하는지 보증하는 충분하게 지원 가능한 데이터(supportable data)를 포함하고 있는가?
13	SF200_reliability_13	적절한 운전원 조작 신호가 발생하는지를 소프트웨어 상세설계에서 포함하고 있는가?
14	SF200_reliability_14	소프트웨어 상세설계가 모든 운전원 입력에 대한 검증 검사를 포함하고 있는가?
15	SF200_reliability_15	통신 라인(communication lines)이 동작하고 있는지를 검사하고 있는가?
16	SF200_reliability_16	입출력 통신에 대한 무결성 검사(integrity checks)가 수행되고 있는가?
17	SF200_reliability_17	통신 메시지가 검증되는가?
18	SF200_reliability_18	절단 또는 쇼트(broken or shorted)된 입출력 통신 라인에 대한 검사가 이루어 지고 있는가?
19	SF200_reliability_19	누락(missing) 그리고/또는 지연 메시지(late messages)에 대한 검사가 이루어 지고 있는가?
20	SF200_reliability_20	하드웨어나 소프트웨어 결함에 기인하는 손상(damaged)이나 손실 데이터(lost data)를 소프트웨어 상세 설계에서 탐지하고 있는가?
21	SF200_reliability_21	소프트웨어나 하드웨어 결함을 다루기 위한 표준 전략이 있는가?
22	SF200_reliability_22	탐지된 모든 하드웨어와 소프트웨어 고장들이 보고되고 있는가?
23	SF200_reliability_23	충분한 기억장치 그리고/또는 통신시험 적용 범위(coverage)가 있는가?
24	SF200_reliability_24	하드웨어 결함을 탐지하기 위한 주기적 소프트웨어 검사가 이 하드웨어에서 이루어 지고 있는가?

◇ 강인성(robustness) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
강인성: AF300_robustness		
1	AF300_robustness_1	소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 입력의 발생에도 정확하게 동작할 설계인가?
2	AF300_robustness_2	소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 운전원 행위(operator behavior)의 발생에도 정확하게 동작할 설계인가?
3	AF300_robustness_3	소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 하드웨어 행위(hardware behavior)의 발생에도 정확하게 동작할 설계인가?
4	AF300_robustness_4	소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 소프트웨어 행위(software behavior)의 발생에도 정확하게 동작할 설계인가?

◇ 안전성(safety) 검증에 관한 질문 목록

노드명/노드확률테이블	질문

안전성: SF400_safety		
1	SF400_safety_1	소프트웨어 상세설계의 제어논리(control logic)가 안전성-필수 요건(safety-critical requirements))을 완전하고 정확하게 구현하는가?
2	SF400_safety_2	확인된 각 안전성-필수 데이터를 변경할 수 있는 상세설계 요소(detailed design elements)를 갖는가?
3	SF400_safety_3	소프트웨어 상세설계가 안전성이 다른 수준의 계통이나 하부계통들 사이의 상호작용(interactions)에 대하여 엄격한 제어를 제공하는가?
4	SF400_safety_4	확인된 안전성-필수 데이터가 형(types), 단위(units), 범위(ranges), 그리고 오류한계(error bounds)와 함께 있는가?
5	SF400_safety_5	소프트웨어 상세설계의 공식(equations)과 알고리즘(algorithms)이 안전성-필수 요건을 정확하게 구현하는가?
6	SF400_safety_6	안전성에 영향을 주는 특정 시간, 비용, 복잡도, 또는 다른 특정 문제점이 있다면, 완화 계획(mitigation plans)을 통해 문서화되고 제기되었는가?
7	SF400_safety_7	소프트웨어 상세설계가 자원-필수 단위나 하부계통(subsystem)을 확인하기 위해 분석되었는가?
8	SF400_safety_8	소프트웨어 상세설계가 소프트웨어가 안전하고 알려진 상태로 복귀하도록 하는 비정상 탐지검사(anomaly detection checks)를 포함하는가?
9	SF400_safety_9	안전성-관련 진단메시지가 다른 사용자들의 지식과 능력에 적합한 방식으로 정의되었는가?
10	SF400_safety_10	상세설계가 안전성-필수 소프트웨어와 비-안전성 소프트웨어를 격리하는가?
11	SF400_safety_11	소프트웨어 상세설계가 그 환경에서 어떻게 소프트웨어 시스템이 낮은 확률사건이나 고장에 반응하는지를 결정하기 위해 분석되었는가?
12	SF400_safety_12	모든 비-안전성 단위가 소프트웨어 안전성에 악영향을 주지 않는다는 적절한 확신을 제공하기 위해 분석되었는가?
13	SF400_safety_13	상세설계 요소들 사이의 인터페이스가 적절히 설계되고 안전위험(safety hazard)을 생성하지 않음을 결정하기 위해 수행되고 문서화된 설계 인터페이스 분석이 있었는가?
14	SF400_safety_14	데이터-관련 상세설계 요소들의 소프트웨어 요건과 일관되고 계통 안전성 요건을 위배하지 않는지를 결정하기 위해 수행되고 문서화된 설계데이터 분석이 있었는가?
15	SF400_safety_15	상세설계에서 설계 한계분석(design constraint analysis)이 수행되고 문서화 되었나?
16	SF400_safety_16	모든 안전성-필수 소프트웨어의 소프트웨어 상세설계에서 소프트웨어 안전성 분석이 수행되고 문서화 되었나?
17	SF400_safety_17	새로운 위험의 개입이 없음을 확증하는 분석이 수행되고 문서화 되었는가?
18	SF400_safety_18	소프트웨어에 할당된 모든 안전성 요건에 따른 소프트웨어 상세설계임을 확증하기 위해 분석되었는가?

◇ 보안성(security) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
보안성: SF500_security		
1	SF500_security_1	상세설계가 소프트웨어 요구명세에서 확인된 모든 보호위협(security threats)을 정확하게 제기하는가?
2	SF500_security_2	상세설계가 권한이 없는 사람(unauthorized personnel)에 의한 공격(intrusions)을 탐지하기 위한 능력을 포함하는가?
3	SF500_security_3	상세설계가 컴퓨터 바이러스를 탐지하는 능력을 포함하는가?
4	SF500_security_4	정확하게 설계된 소프트웨어 요구명세에 접근제한(access restrictions)이 명세되는가?
5	SF500_security_5	발전소 운전원에 의한 변경에 대하여 계통데이터나 코드가 안전하게 보호(secure) 되는가?

◇ 타이밍(timing) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
보안성: SF600_timing		
1	SF600_timing_1	소프트웨어 상세설계 명세가 모든 타이밍 자원 제약, 각각을 처리하는(handling) 전략, 요구된 여유도(margins), 그리고 이러한 여유도를 측정하는 방법의 명세를 포함하는가?
2	SF600_timing_2	상세 설계가 실행 타이밍이 결정적임을 보증하는가? 이는 인터럽트, 단일 프로세서에서의 다중 프로세싱, 동적 메모리 관리, 그리고 시건-기반 통신(event-driven communications) 중 어느 것이 사용되는지를 실증하기 위해 특별히 중요하다.
3	SF600_timing_3	인터럽트가 정확하게 처리되고 입출력 타이밍이 정확하며, 그리고 단위들이 정확하게 연계되는지 확증하기 위해 중요한 인터페이스(potential interface)와 타이밍 문제가 분석되었는가?
4	SF600_timing_4	소프트웨어 상세설계에서 수행된 타이밍분석이 클럭(clock), 표본율(sampling rate), 그리고 센서 반응시간의 변화(variations)를 계산하기 위해 절절한 여유도를 허용하는가?
5	SF600_timing_5	컴퓨터 자원의 스캐쥴링이 동적이라기 보다는 예측적이고 결정적임을 확증하기 위해 분석이 수행되었는가?
6	SF600_timing_6	자체-검사 활동(self-checking actions)이 타이밍 요건을 위배하지 않음을 확증하기 위해 분석이 수행되었는가?
7	SF600_timing_7	요구된 계통 반응 시간이 만족된다는 것을 상세설계가 보증하는가?
8	SF600_timing_8	소프트웨어 상세설계 명세는 사용된 스캐쥴링 구조(scheduling mechanisms)와 IPC (inter-process communication mechanisms)를 명세하는가?
9	SF600_timing_9	각 단위에 대한 타이밍이 실행(execution)을 위한 최대 및 최소 시간으로 명세되었는가?
10	SF600_timing_10	다른 프로그램과 동기된 다른 순차적인 프로그램으로부터 데이터를 받거나 또는 그 프로그램으로 데이터를 전송하는 순차적인 프로그램의 실행이 있는가?
11	SF600_timing_11	연산(operations)이 실행 속도와 독립적인 정확한 순서로 수행되도록 설계된 소프트웨어가 있는가?
12	SF600_timing_12	프로그램을 실행하기 위해 주어진 시간이나

		프로그램의 실행이 초기화되는 시간 중에서 독립적인 순차적인 프로그램에 의해 생성된 결과들이 있는가?
13	SF600_timing_13	인터럽트가 금지될 수 있는 시간(times)에 하드 상한(hard upper bounds) 범위 내에 존재하는가?
14	SF600_timing_14	소프트웨어 상세설계 명세가 모든 기억장치 자원 제약, 각각을 처리하는 전략, 요구된 여유도, 그리고 이러한 여유도를 측정하는 방법의 명세를 포함하는가?
15	SF600_timing_15	중앙처리장치(CPU), 기억장치, 그리고 입출력 채널을 위한 적절한 예산(budgets)이 확립되었나?

2.2 공정 특성 검증

◇ 완전성(Completeness) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
완전성: SP100_completeness		
1	SP100_completeness_1	각 단위 모듈이 오직 하나의 진입점을 가지는가?
2	SP100_completeness_2	각 단위 모듈이 오직 하나의 출구점을 가지는가? 고급언어를 사용한 프로그래밍 되는 하나의 단위에서 2개 이상의 "리턴" 문장을 사용하는 것은 이 질문의 의도를 벗어나는 것이 아니다.
3	SP100_completeness_3	명시된 모델, 알고리즘, 수치기법, 신호변환 및 자료처리 절차가 실용적으로 사용되는 범주 내의 것들인가?
4	SP100_completeness_4	소프트웨어 상세설계가 단위들 간의 규칙적이고 문서화된 연계를 사용하는가?
5	SP100_completeness_5	모든 내부 및 외부연계가 완전하게 정의되었는가?
6	SP100_completeness_6	상세설계가 연계요소들에 대한 문서화된 기술이나 알려진 특성과 일관성이 있는가?
7	SP100_completeness_7	발전소 운전원을 지원하도록 설계되었는가를 확증하기 위해 소프트웨어 상세설계가 분석되고 검토되었는가?
8	SP100_completeness_8	컴퓨터 시스템-인간 상호작용의 문법이 정의되는가?
9	SP100_completeness_9	센서 및 구동기 설계연계가 신호의 동적범위, 정확성, 타이밍, 인터럽트 및 분해능을 포함하는가?
10	SP100_completeness_10	소프트웨어 상세설계가 시스템 및 개발 노력에 주어지는 제약조건들 안에서 구현될 수 있는가?
11	SP100_completeness_11	모든 자료채널에 대한 개수, 크기, 자료속도, 샘플링 빈도 및 응답시간이 소프트웨어 상세설계 명세서에 정의되는가?
12	SP100_completeness_12	소프트웨어 상세설계가 단위 크기를 제한하는 요건을 명시하는가?
13	SP100_completeness_13	상세설계에 적절한 인간공학 요소들이 고려되어 운전원이나 유지보수자가 시스템의 상태를 평가하거나 문제가 있는 부분을 찾아내는데 용이하도록 하고 있는가?
14	SP100_completeness_14	소프트웨어 상세설계 명세서가 소프트웨어 요구사항 명세와 소프트웨어 구조설계 명세서에 기술된 시스템 요구사항 및 모든 제약조건들(기능, 성능, 안전성, 신뢰성 및 유지보수성 요구사항 포함)을 만족하는 소프트웨어 상세 설계를 기술하는가?

15	SP100_completeness_15	컴퓨터 시스템과-인간 상호작용에 대한 정형적 절차가 소프트웨어 설계명세서에 명시되는가?
16	SP100_completeness_1	특별한 어려움을 초래하는 특정 상세설계 분야(예를 들어, 공정 및 속도, 새로운 알고리즘, 보안, 새로운 하드웨어 및 언어)가 명시되고 그에 따른 완화책이 기술되는가?
17	SP100_completeness_16	실 세계의 제한사항이나 소프트웨어 구조설계에 따른 상세설계의 제약사항들을 평가하기 위한 설계 제약사항 분석이 수행되었는가?
18	SP100_completeness_17	자료-관련 상세설계 요소들이 소프트웨어 요구사항 및 구조와 일관성을 유지하는가를 결정하기 위한 설계자료 분석이 수행되었는가?
19	SP100_completeness_18	소프트웨어 상세설계 공식, 알고리즘 및 제어논리가 소프트웨어 요구사항들을 정확하게 구현하는가를 결정하기 위한 설계논리 분석이 수행되었는가?
20	SP100_completeness_19	소프트웨어 상세설계가 각 연계사항에 대하여 요구되는 프로그램의 행위를 구현하고 있는가?
21	SP100_completeness_20	소프트웨어 코드를 구현하는데 필요한 모든 프로그램 입력, 출력, 및 자료 요소들이 적절한 범위로 명시되고 기술되어 있는가?
22	SP100_completeness_21	소프트웨어 요구사항명세서의 각 요구사항들이 상세설계 명세서로 반영되어 있는가?
23	SP100_completeness_22	소프트웨어 상세설계에 하드웨어 환경(전기 접점, 전원 및 필터, 예비 전원, 프로세서의 선택, 입출력장치 속도, 소프트웨어 다중성, 자료 통신, 인간-기계 연계 및 하드웨어 형상 등)이 고려되었는가?
24	SP100_completeness_23	소프트웨어 상세설계에 운전환경이 고려되었는가?
25	SP100_completeness_24	소프트웨어 상세설계가 소프트웨어 요구사항명세서에 명시된 모든 운전모드를 고려하는가?
26	SP100_completeness_25	상세설계 요소들 사이의 연계가 정확하게 설계되었는가를 결정하기 위한 설계 연계분석이 수행되었는가?

◇ 일관성(Consistency) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
일관성: SP200_consistency		
1	SP200_consistency_1	표현 스타일이나 상세 정도가 소프트웨어 상세설계명세서 전반에 걸쳐 일관성이 있는가?
2	SP200_consistency_2	소프트웨어 상세설계가 하드웨어 및 소프트웨어 구조와 일관성이 있는가?
3	SP200_consistency_3	주변장치 접점들에 대한 표준연계가 있는가?
4	SP200_consistency_4	유사하거나 관련 있는 기능들에 대한 상세설계가 일관성이 있는가?
5	SP200_consistency_5	인간-기계 연계에 대한 표준연계가 있는가?
6	SP200_consistency_6	자료 전송에 대한 표준연계가 있는가?
7	SP200_consistency_7	설계에 명시된 모델, 알고리즘 및 수치기법이 수학적으로 상호 호환되는가?
8	SP200_consistency_8	설계에 명시된 모델, 알고리즘 및 수치기법이 적절한 표준 참고문서의 것과 일치하는가?. 표준 참고문서가 존재한다면 그것들은 모델, 알고리즘, 수치기법을 설계하는 사람들이 제공해야 한다.

9	SP200_consistency_9	각 소프트웨어 상세 설계요소가 프로그램이 작동하는 운전환경의 설명 및 특성들과 일관성이 있는가?
10	SP200_consistency_10	소프트웨어 상세설계가 소프트웨어 요구사항명세와 일관성이 있는가?
11	SP200_consistency_11	복수개의 정형적 상세설계 방법이 사용된다면 그들이 서로 일관성이 있는가?
12	SP200_consistency_12	소프트웨어 상세설계에 주어진 입력 및 출력명세(자료형태, 자료 크기, 자료 속도, 정확성, 오류범위 및 물리적 단위 등 포함)가 하드웨어 및 기성 소프트웨어에 의해 부여된 연계 요구사항과 일관성이 있는가?

◇ 정확성(Correctness) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
정확성: SP300_correctness		
1	SP300_correctness_1	모든 공식 및 알고리즘이 코딩이 시작될 수 있을 정도로 정확하게 또 충분히 상세한 수준까지 정의되어 있는가?
2	SP300_correctness_2	잠재적인 언더플로(underflow) 및 오버플로(overflow) 상태를 식별하기 위한 알고리즘 정확성 분석이 수행되었는가?
3	SP300_correctness_3	서브루틴이나 프로시저를 빠져 나왔을 때 상태가 점검되고, 오류 상태가 지시되었을 때 적절한 행위가 수행되는가?
4	SP300_correctness_4	잠재된 자료처리 문제(부정확한 자료 초기화, 저장 자료의 부정확한 평가, 부정확한 자료 측정 등)를 평가하기 위한 분석이 수행되었는가?
5	SP300_correctness_5	격리, 분할, 자료 별칭(data aliasing) 및 결함 억제 현안들을 둘러싼 자료 의존성에 대한 자료 구조를 평가하기 위한 분석이 수행되었는가?
6	SP300_correctness_6	상태변수의 병행성(concurrency) 및 일관성을 유지할 장치들이 소프트웨어 상세설계에 나타나는가?
7	SP300_correctness_7	알고리듬이 입력 및 시간변수의 전체 범위에 대하여 안정적인가를 확증하기 위한 알고리즘 분석이 수행되었는가?
8	SP300_correctness_8	각 단위의 입력에 대한 유효성(validity)이 점검되었는가?
9	SP300_correctness_9	인터럽트가 예상치 못한 방식으로 안전하고 중요한 자료값을 변경하지 않는다는 확실한 증거가 있는가?
10	SP300_correctness_10	안전에 중요한 자료 값이 예상치 못한 방식 혹은 예상치 못한 상세설계 요소에 의해 변경되지 않는다는 확실한 증거가 있는가?
11	SP300_correctness_11	공식, 알고리즘 및 제어논리가 잠재된 문제점들(논리오류, 누락된 경우나 단계, 중복논리, 극한조건의 무시, 불필요한 기능, 오해, 조건시험 누락, 잘못된 변수에 대한 점검 및 루프의 반복 등)에 대하여 평가되는가?
12	SP300_correctness_12	안전에 중요한 자료가 초기화되기 전에 사용되지 않는다는 확실한 증거가 있는가?
13	SP300_correctness_13	부동소수점 연산 및 재귀(recursion)가 사용되는 경우, 그 사용에 대한 적절한 정당화가 이루어져 있는가?

◇ 스타일(Style) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
스타일: SP400_style		
1	SP400_style_1	소프트웨어 설계명세서(SDS)는 감시계통 성능(supervisory system performance) 및 신뢰도(reliability)에 대한 정량적인 데이터들에 대해 문서화하고 있는가?
2	SP400_style_2	모든 감시기능들에 대해 확고한 정의가 있고 연계사항(interfaces)들은 체계를 잘 갖추고 있는가?
3	SP400_style_3	전역변수(global variables)의 사용이 금지되거나 또는 세밀하게 입증되도록 되어 있는가?
4	SP400_style_4	소프트웨어 상세 설계명세는 타임 슬라이싱(time-slicing)의 사용을 제한하고 있는가?
5	SP400_style_5	소프트웨어 상세 설계명세는 메모리 스와핑(memory swapping) 사용을 제한하고 있는가?
6	SP400_style_6	필요한 모든 지원 소프트웨어들에 대해서 충분히 기술하고 있는가?
7	SP400_style_7	소프트웨어 상세 설계명세에서 프로그래밍언어 변수들에 대해서 명료한 정의 및 형(typing)이 있는가?
8	SP400_style_8	소프트웨어 상세설계에서는 안전성-필수 기능(safety-critical functions)들의 수행시간 동안 소프트웨어 인터럽트(interrupts) 취급을 제한하도록 되어 있는가?
9	SP400_style_9	사용될 구축 언어들은 소프트웨어 상세설계 사양에서 식별되도록 되어 있는가?
10	SP400_style_10	소프트웨어 상세설계에서 특별하게 개발된 장치 드라이버(device drivers)를 사용하는데 대해서 논리적 근거를 제공하고 있는가?
11	SP400_style_11	어떤 공유메모리 영역(shared memory locations)이 일치가 안된 상태에서 동시에 변경되지 않도록 상세설계에서 보장하고 있는가?
12	SP400_style_12	인터럽트를 피하는 최대 가능한 루프 수를 소프트웨어 상세설계에서 기술하고 있는가?
13	SP400_style_13	데이터 저장공간(store), 메시지 또는 스크린 디스플레이(screen display)가 소프트웨어 상세 설계요소일 때 소프트웨어 상세설계 명세에서 이 요소의 구조에 대해서 기술하고 있는가?
14	SP400_style_14	소프트웨어 상세 설계명세는 간단한 폼에 대한 (즉, 순서(sequence), 사례 선택, 반복, 추상화(abstractions) (예를 들어 절차, 기능 그리고/또는 서브루틴)) 제어구조를 제한하고 있는가?
15	SP400_style_15	소프트웨어 상세설계 요소가 어떤 절차를 수행하면서 요소가 기능을 수행하는 방법에 대해서 소프트웨어 상세 설계명세에서 기술하고 있는가?
16	SP400_style_16	소프트웨어 상세설계 명세에서 소프트웨어 상세설계의 일환으로 기술되는 구축 제약사항(implementation constraints)들에 대해서 모두 열거하고 있는가?
17	SP400_style_17	소프트웨어 상세설계 명세에서 상세설계 의사결정에 대한 논리적 근거에 대해서 문서화하고 있는가?
18	SP400_style_18	소프트웨어 상세설계 명세에서 필요로 하는 프로그래밍 언어에 대한 표준들에 대해서 식별하고 있는가?

19	SP400_style_19	각 상세설계 요소는 특정한 이름을 갖도록 하고 있는가?
20	SP400_style_20	소프트웨어 상세설계 명세는 각 상세설계 요소의 형태(예를 들면, 체계, 하부체계, 유니트, 데이터베이스, 파일, 데이터 구조, 스크린 디스플레이, 메시지, 프로그램 또는 절차)에 대해서 식별하고 있는가?
21	SP400_style_21	소프트웨어 상세설계 명세는 각 상세설계 요소의 기능(즉, 무엇을 하는가)에 대해서 기술하고 있는가?
22	SP400_style_22	소프트웨어 상세설계 명세는 상세설계 요소들 간의 관계에 대해서 기술하고 있는가?
23	SP400_style_23	상세설계 요소간의 상호작용 기술은 타이밍, 사건유발(triggering events), 수행순서, 데이터 공유, 그리고 상호작용에 영향을 미치는 다른 작용요인을 포함하고 있는가?
24	SP400_style_24	소프트웨어 상세설계 명세에서 각 상세설계 요소가 그 기능을 수행하기 위한 자원들을 명기하고 있는가?
25	SP400_style_25	소프트웨어 상세설계 명세는 적용 가능한 표준에 준수해서 작성되도록 되어 있는가?

◇ 추적성(Traceability) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
추적성: SP500_traceability		
1	SP500_traceability_1	각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?
2	SP500_traceability_2	각 상세설계 요소가 소프트웨어 요구사항 명세(Software Requirements Specification)내 특정 요소들로 역방향 추적기록이 가능한가?
3	SP500_traceability_3	각 상세설계 요소에 대해서 특정 코드 요소로의 전방향 추적기록이 가능한가?
4	SP500_traceability_4	요구사항에서 열거된 상세설계 제약사항들이 상세설계 내에서 따르도록 되어 있는가?
5	SP500_traceability_5	소프트웨어 상세설계 명세는 소프트웨어 상세설계 내의 각 기능을 특정하게 식별하도록 되어 있어 구축 시 특정하게 참조 될 수 있도록 되어 있는가?
6	SP500_traceability_6	SRS에서 기술된 요구사항 범위가 아닌 기능들을 포함하는데 대해 정당성이 입증되어 있는가? 그리고 결과적인 설계 상세내용이 SRS에서 의도한 바와 일치하는가?
7	SP500_traceability_7	소프트웨어 상세설계 사양이 소프트웨어 상세설계와 관련된 상세설계 의사결정을 문서화한 설계노트(design notes)를 참조하도록 되어 있는가?

◇ 확인성(verifiability) 검증에 관한 질문 목록

노드명/노드확률테이블		질문
확인성: AP600_verifiability		
1	AP600_verifiability_1	소프트웨어 상세설계 명세에서는 시험 가능한 방법으로 각 소프트웨어 요소의 기능들을 기술하고

		있는가?
2	AP600_verifiability_2	소프트웨어 상세설계 분석에서 컴포넌트 시험 절차 및 시험 사례들을 발하는데 사용되어 왔는가?

3. SDS 상세 검증

3.1 설계명세 추적성 상세 검증

노드명/노드확률테이블		질문
추적성: DV100_traceability		
1	DV100_traceability_1	소프트웨어 구조설계 설명서에는 소프트웨어 구조설계의 각 기능을 유일하게 확인할 수 있는 식별자(ID)를 가지고 있어서 구현(implementation)시 이들 식별자를 가지고 추적할 수 있는가?
2	DV100_traceability_2	소프트웨어 상세설계 명세는 소프트웨어 상세설계 내의 각 기능을 특정하게 식별하도록 되어 있어 구축 시 특정하게 참조 될 수 있도록 되어 있는가?
3	DV100_traceability_3	소프트웨어 구조설계 요소가 소프트웨어 요구사항 명세서와 일치하는지 확인하는데 사용하는 기법인 특별시험(specific tests) 또는 검증기준(validation criteria)을 이용하여 전향추적을 할 수 있는가?
4	DV100_traceability_4	각 소프트웨어 구조설계 요소는 소프트웨어 요구사항 명세서의 특별한 요소(specific elements)를 후향 추적(backward trace) 할 수 있는가?
5	DV100_traceability_5	각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?
6	DV100_traceability_6	각 상세설계 요소가 소프트웨어 요구사항 명세(Software Requirements Specification)내 특정 요소들로 역방향 추적 기록이 가능한가?
7	DV100_traceability_7	소프트웨어 구조설계 설명서에 소프트웨어 요구사항 명세서에 기술된 요건의 범위를 벗어나는 어떤 기능을 포함하고 있다는 정당한 근거가 있다면 요건의 범위를 벗어난 소프트웨어 구조설계 설명서의 정당성 결과가 요구사항명세서와 일치하는가?

3.2 설계 정확성 상세 검증

◇ 소프트웨어 설계 정의의 정확성

노드명/노드확률테이블		질문
정확성: DV210_Def_correctness		
1	DV210_Def_correctness_1	각 운전모드에서 실행되어야 하는 소프트웨어의 기능 및 행위가 정확하게 기술되어 있는가?
2	DV210_Def_correctness_2	알고리즘 (논리 및 공식)이 정확한가?
3	DV210_Def_correctness_3	각 기능들이 결정론적인가? 각 기능들을 비결정론적으로 만들 수 있는 조건들 (오류 복구 등)이 정의되어 있는가?
4	DV210_Def_correctness_4	기능이 단일고장요건을 만족하는가?
5	DV210_Def_correctness_5	안전계통 소프트웨어에 대한 임의의 변경 및 제어를 차단하는 수단이 있는가?

◇ 소프트웨어 입/출력 정의의 정확성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
정확성: DV220_IO_correctness		
1	DV220_IO_correctness_1	각 기능들이 그 기능에 요구되는 입력 및 출력을 정확하게 명시하고 있는가?
2	DV220_IO_correctness_2	FBD 및 상세모듈에 모든 입력 및 출력의 출발점이 맞게 기술되어 있는가?

◇ 소프트웨어 행위 명세의 정확성

노드명/노드확률테이블		질문
정확성: DV230_Act_correctness		
1	DV230_Act_correctness_1	각 기능이 어떻게 시작되는가 (초기화 조건)를 정확하게 기술하고 있는가?
2	DV230_Act_correctness_2	각 기능적 요구사항이 그 기능을 수행하는데 요구되는 작업 순서, 행위 및 사건들을 정확하게 명시하고 있는가?
3	DV230_Act_correctness_3	각 기능적 요구사항이 종결 조건이나 기능의 종료 시 시스템의 상태에 대하여 정확하게 명시하고 있는가?
4	DV230_Act_correctness_4	Function들 사이에 순환적 의존관계가 존재하지 않는가?
5	DV230_Act_correctness_5	각 기능의 정상적인 완료를 방해하는 조건이나 입력이 없는가?
6	DV230_Act_correctness_6	각 기능이 임의의 설계기준사고시에도 요구되는 기능을 완전히 수행할 수 있는가?
7	DV230_Act_correctness_7	비안전계통의 임의의 단일고장이 안전계통의 일부분에 영향을 주더라도 남은 안전계통 부분들이 적절한 안전기능을 수행할 수 있는가?

◇ 소프트웨어 인터페이스기술의 정확성

노드명/노드확률테이블		질문
정확성: DV240_IF_correctness		
1	DV240_IF_correctness_1	타 채널 및 프로세서와의 신호 연계가 안전기능을 방해하지 않는가?
2	DV240_IF_correctness_2	비안전계통과의 연계가 안전계통에 요구되는 기능을 방해하지 않는가?

3.3 설계 일관성 상세 검증

◇ 소프트웨어 설계 정의의 일관성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
추적성: DV310_Def_consistency		
1	DV310_Def_consistency_1	각각의 상세설계가 소프트웨어의 요구사항들과 일관되는가?
2	DV310_Def_consistency_2	한 기능에 대한 명세 및 비슷한 기능들에 대한

		명세가 서로 일관성이 있는가?
3	DV310_Def_consistency_3	내부적 일관성 측면에서, Block Diagram으로 표현된 기능과 알고리즘으로 표현된 기능 사이에 일관성이 있는가?
4	DV310_Def_consistency_4	정의된 용어 (또는 변수)를 사용하는 기능이 그것들의 정의와 모순되지 않는가?

◇ 소프트웨어 입출력 설계의 일관성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
추적성: DV320_IO_consistency		
1	DV320_IO_consistency_1	입력, 계산 및 출력 자료들에 요구되는 정확성이 상호 호환성이 있는가?
2	DV320_IO_consistency_2	입력 및 출력의 명세가 요구사항에 나오는 설명에 맞게 기술되어 있는가?

◇ 소프트웨어 설계명세의 일관성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
추적성: DV330_Spec_consistency		
1	DV330_Spec_consistency_1	SDS에 명시된 모델, 알고리즘, 및 계산식들이 적용 가능한 표준 및 참고문헌과 일치성이 있는가?

◇ 소프트웨어 인터페이스 설계기술의 일관성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
추적성: DV340_consistency		
1	DV340_consistency_1	SDS에 있는 입력 및 출력이 하드웨어 및 기성 소프트웨어 등에 의해 부여되는 연계 요구사항 들과 일관되는가?

3.4. 소프트웨어 설계 완전성 검증

◇ 소프트웨어 설계(기능?) 정의의 완전성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
완전성: DV410_Def_completeness		
1	DV410_Def_completeness_1	소프트웨어가 수행해야 하는 모든 운전모드가 기술되어 있으며, 각 운전모드에서 실행되어야 하는 소프트웨어의 기능 및 행위가 모두 기술되어 있는가?
2	DV410_Def_completeness_2	각 운전모드 간의 전환을 일으키는 조건들이 명시되어 있는가?
3	DV410_Def_completeness_3	SRS에 기술된 모든 기능적 요구사항들이 SDS에 함수로 기술되었는가?
4	DV410_Def_completeness_4	각각의 입력이 최소한 하나의 FBD에 의해 사용되었는가?

5	DV410_Def_completeness_5	각 출력을 정의하는 FBD가 유일하게 있는가?
6	DV410_Def_completeness_6	Function의 performance criteria (error performance 포함) 가 모두 정의되었는가?

◇ 소프트웨어 입출력 정의의 완전성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
완전성: DV420_IO_completeness		
1	DV420_IO_completeness_1	입력 및 출력이 완전히 정의되어 있는가?
2	DV420_IO_completeness_2	Block Diagram 및 Function Block에 모든 입력 및 출력이 기술되었는가?
3	DV420_IO_completeness_3	입력변수가 갖추어야 할 특성이 모두 정의되었는가?
4	DV420_IO_completeness_4	출력변수가 갖추어야 할 특성이 모두 정의되었는가?

◇ 소프트웨어 행위명세 완전성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
완전성: DV430_Act_completeness		
1	DV430_Act_completeness_1	오류 사항들이 수정조치 사항과 함께 기술되어 있는가?
2	DV430_Act_completeness_2	SDS가 비정상적인 입력에 대한 소프트웨어의 행위를 명시하고 있는가?
3	DV430_Act_completeness_3	각 상세설계가 그 기능을 수행하는데 요구되는 작업 순서, 행위, 사건 및 타이밍을 명시하고 있는가?
4	DV430_Act_completeness_4	고장-안전 차원에서 컴퓨터 시스템에 요구되는 모든 행위가 완전하게 기술되어 있는가?
5	DV430_Act_completeness_5	SDS가 소프트웨어가 수행하지 말아야 할 것들도 언급하고 있는가?

◇ 소프트웨어 인터페이스 기술의 완전성 검증에 관한 질문 목록

노드명/노드확률테이블		질문
완전성: DV440_IF_completeness		
1	DV440_IF_completeness_1	다음과 같은 각 프로세서간의 연계가 모두 정의되었는가? - 공정계측입력 (PI Inputs) - 노외핵계측 (ENF Instrument) - 노심보호연산기 (CPC) - 자동시험 및 연계 프로세서 (ATIP) - 캐비닛 운전원 모듈 (COM) - 동시논리프로세서 (CP)

부록B. 원자로 보호계통 SW(BP) 설계명세 평가를 위한 BBN의 노드확률테이블

B-1. Root node NPTs(Unconditional pdfs)

Node name \ State	Good	Poor
SDS_Dev_Process	0.5	0.5

Node name \ State	Good	Poor
SDS_VnV_Process	0.5	0.5

Node name \ State	Good	Poor
SRS_Integrity	0.5	0.5

B-2. NPTs of High level network

SDS_Dev_Process		HighQuality			
SDS_VnV_Process		HighQuality		LowQuality	
SRS_Integrity		Good	Poor	Good	Poor
SDS_Architecture	Good	0.9	0.7	0.7	0.3
	Poor	0.1	0.3	0.3	0.7

SDS_Dev_Process		LowQuality			
SDS_VnV_Process		HighQuality		LowQuality	
SRS_Integrity		Good	Poor	Good	Poor
SDS_Architecture	Good	0.7	0.3	0.3	0.1
	Poor	0.3	0.7	0.7	0.9

SDS_Dev_Process		HighQuality			
SDS_VnV_Process		HighQuality		LowQuality	
SDS_Architecture		Good	Poor	Good	Poor
SDS_Design_Integrity	Good	0.9	0.7	0.7	0.3
	Poor	0.1	0.3	0.3	0.7

SDS_Dev_Process		LowQuality			
SDS_VnV_Process		HighQuality		LowQuality	
SDS_Architecture		Good	Poor	Good	Poor
SDS_Design_Integrity	Good	0.7	0.3	0.3	0.1
	Poor	0.3	0.7	0.7	0.9

o SW_Design_VnV sub-graph

SDS_Design_Integrity		Good	Poor
Detailed_Verification	good	0.95	0.05
	bad	0.05	0.95

SDS_Design_Integrity		Good	Poor
SW_Design_VnV	good	0.95	0.05
	bad	0.05	0.95

SDS_Architecture		Good	Poor
SW_Architecture_VnV	good	0.95	0.05
	bad	0.05	0.95

SW_Design_VnV		Good	Poor
SP100_completeness	good	0.99	0.01
	bad	0.01	0.99

SW_Design_VnV		Good	Poor
SP200_consistency	good	0.84	0.16
	bad	0.16	0.84

SW_Design_VnV		Good	Poor
SP300_correctness	good	0.84	0.16
	bad	0.16	0.84

SW_Design_VnV		Good	Poor
SP400_style	good	0.99	0.01
	bad	0.01	0.99

SW_Design_VnV		Good	Poor
SP500_traceability	good	0.74	0.26
	bad	0.26	0.74

SW_Design_VnV		Good	Poor
SP600_verifiability	good	0.64	0.36
	bad	0.36	0.64

SW_Design_VnV		Good	Poor
SF100_accuracy	good	0.64	0.36
	bad	0.36	0.64

SW_Design_VnV		Good	Poor
SF200_reliabiltiy	good	0.99	0.01
	bad	0.01	0.99

SW_Design_VnV		Good	Poor
SF300_robustness	good	0.68	0.32
	bad	0.32	0.68

SW_Design_VnV		Good	Poor
SF400_safety	good	0.94	0.06
	bad	0.06	0.94

SW_Design_VnV		Good	Poor
SF500_security	good	0.7	0.3
	bad	0.3	0.7

SW_Design_VnV		Good	Poor
SF600_timing	good	0.9	0.1
	bad	0.1	0.9

o Detailed_Verification sub-graph

Detailed_Verification		Good	Poor
DV100_traceability	good	0.74	0.26
	bad	0.26	0.74

Detailed_Verification		Good	Poor
DV200_correctness	good	0.92	0.08
	bad	0.08	0.92

Detailed_Verification		Good	Poor
DV300_consistency	good	0.74	0.26
	bad	0.26	0.74

Detailed_Verification		Good	Poor
DV400_completeness	good	0.92	0.08
	bad	0.08	0.92

DV200_correctness		Good	Poor
DV210_Def_correctness	good	0.9	0.1
	bad	0.1	0.9

DV200_correctness		Good	Poor
DV220_IO_correctness	good	0.72	0.28
	bad	0.28	0.72

DV200_correctness		Good	Poor
DV230_Act_correctness	good	0.99	0.01
	bad	0.01	0.99

DV200_correctness		Good	Poor
DV240_IF_correctness	good	0.72	0.28
	bad	0.28	0.72

DV300_consistency		Good	Poor
DV310_Def_consistency	good	0.99	0.01
	bad	0.01	0.99

DV300_consistency		Good	Poor
DV320_IO_consistency	good	0.8	0.2
	bad	0.2	0.8

DV300_consistency		Good	Poor
DV330_Act_consistency	good	0.7	0.3
	bad	0.3	0.7

DV300_consistency		Good	Poor
DV340_IF_consistency	good	0.7	0.3
	bad	0.3	0.7

DV400_completeness		Good	Poor
DV410_Def_completeness	good	0.99	0.01
	bad	0.01	0.99

DV400_completeness		Good	Poor
DV420_IO_completeness	good	0.88	0.12
	bad	0.12	0.88

DV400_completeness		Good	Poor
DV430_Act_completeness	good	0.5	0.5
	bad	0.5	0.5

DV400_completeness		Good	Poor
DV440_IF_completeness	good	0.66	0.34
	bad	0.34	0.66

o SW_Architecture_VnV sub-graph

SW_Architecture_VnV		Good	Poor
AF100_reliability	good	0.86	0.14
	bad	0.14	0.86

SW_Architecture_VnV		Good	Poor
AF200_safety	good	0.88	0.12
	bad	0.12	0.88

SW_Architecture_VnV		Good	Poor
AF300_security	good	0.66	0.34
	bad	0.34	0.66

SW_Architecture_VnV		Good	Poor
AF400_timing	good	0.76	0.24
	bad	0.24	0.76

SW_Architecture_VnV		Good	Poor
AP100_completeness	good	0.99	0.01
	bad	0.01	0.99

SW_Architecture_VnV		Good	Poor
AP200_consistency	good	0.72	0.28
	bad	0.28	0.72

SW_Architecture_VnV		Good	Poor
AP300_style	good	0.8	0.2
	bad	0.2	0.8

SW_Architecture_VnV		Good	Poor
AP400_traceability	good	0.72	0.28
	bad	0.28	0.72

SW_Architecture_VnV		Good	Poor
AP500_verifiability	good	0.64	0.36
	bad	0.36	0.64

B-3. NPTs of low level networks

1. SDS 구조설계 검증

기능 특성/코드	공정 특성/코드
신뢰성(reliability)/AF100	완전성(completeness)/AP100
안전성(safety)/AF200	일관성(consistency)/AP200
보안성(security)/AF300	스타일(style)/AP300
타이밍(timing)/AF400	추적성(traceability)/AP400
	확인가능성(verifiability)/AP500

1.1 기능 특성 검증

◇ 신뢰성(reliability) 검증에 관한 노드확률테이블

AF100_reliability		good	bad
AF101:소프트웨어 구조에 의해 고장을 감지하였을 경우 취해지는 행위들을 명시하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF102:프로그램 수행 동안 고장파급을 방지할 수 있는 기술이 사용되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF103:컴퓨터 시스템 구조 설계 시 소프트웨어 공통모드 고장 가능성을 분석하고 고려하였는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF104:소프트웨어 구조가 사고발생시 시스템 상태를 기록할 기능을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF105:신뢰도와 가용성 분석이 신뢰도 요구사항 만족을 보장할 수 있도록 충분한 보충자료를 포함하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF106:소프트웨어 설계구조가 고장 감지 시 잘 정의된 출력을 생산할 능력을 포함하는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF107:소프트웨어 구조가 소프트웨어와 하드웨어의 상태를 명시할 기능을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF108:하드웨어와 소프트웨어 오류를 처리할 표준 전략이 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF109:소프트웨어 구조는 감지된 하드웨어, 소프트웨어 고장을 보고할 기능을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF110:소프트웨어 구조는 적절한 watch-dog 타이머를 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF111:개별 하드웨어 단위의 고장이 컴퓨터 시스템 고장을 야기하지 않는 구조인가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF112:디지털 시스템이 운전원 오류를 감지하거나 감내할 것인가?	yes	0.90	0.10
	no	0.10	0.90

AF100_reliability		good	bad
AF113:소프트웨어 구조가 부적절한 운전원 행위를 통보할 방법을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 안전성(safety) 검증에 관한 노드확률테이블

AF200_safety		good	bad
AF201:제어 로직이 오류처리, 비정상 및 비상처리 요구사항을 정확히 구현하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF202:각 필수 안전구조 요소의 결함분석이 수행되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF203:소프트웨어 구조가 다른 안전 중요도를 가지는 하부 계통간의 상호작용을 엄격하게 제어하는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF204:소프트웨어 구조가 소프트웨어 상태를 안전하고 알려진 상태로 되돌리도록 하는 비정상 감지 검사기능을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF205:소프트웨어 구조에 대하여 설계 제한분석을 수행하였는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF206:모든 안전-필수 소프트웨어의 소프트웨어 구조에 대한 소프트웨어 안전성분석이 수행되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF207:각 안전-필수 데이터 항목을 변경할 수 있는 구조적 요소가 파악되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF208:소프트웨어 구조의 제어 로직이 안전-필수 요구사항을 완전하고 정확하게 구현하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF209:안전에 영향을 줄 시간, 비용, 복잡도 또는 특별한 문제가 있다면, 완화계획을 통해 기록되고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF210:소프트웨어 구조가 자원에 치명적인 요소를 식별할 수 있도록 분석되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF211:소프트웨어 계통이 그 환경에서 낮은 확률사고나 고장에 어떻게 반응할 것인가를 결정하기 위해 그 소프트웨어 구조가 분석되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF212:소프트웨어 구조설계가 소프트웨어에 할당된 모든 안전 요구사항과 부합함을 분석하였는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF213:모든 비-안전 구조요소가 소프트웨어 안전에 급작스럽게 영향을 주지 않음을 분석하였는가?	yes	0.90	0.10
	no	0.10	0.90

AF200_safety		good	bad
AF214:안전에 중요한 기능이 잘 정의되고 엄격하게 통제되는 연계에 의하여 정상운전 및 부가기능과 분리되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 보안성(security) 검증에 관한 노드확률테이블

AF300_security		good	bad
AF301:구조가 소프트웨어 요구사항 명세(SRS)에 명시된 보안 위협들을 정확하게 다루고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF300_security		good	bad
AF302:구조가 SRS에 명시된 접근제한을 정확하게 다루고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF300_security		good	bad
AF303:구조가 비-승인자의 침입을 감지할 능력을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 타이밍(timing) 검증에 관한 노드확률테이블

AF400_timing		good	bad
AF401:소프트웨어 구조 설계명세에서 스케줄링 및 프로세스간의 통신방식이 기술되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF402:소프트웨어 구조 설계명세가 모든 타이밍 제한사항, 각 제한사항에 대한 전략, 요구되는 여유도 및 여유도 측정방법 등에 대해 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF403:각 구조 요소에 대한 최소 및 최대 수행시간이 명시되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF404:소프트웨어 구조가 응답시간을 만족하는지 분석되는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF405:운전이 정확한 순서로 수행되었음을 보장하도록 구조가 분석되었는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF406:컴퓨터 자원의 스케쥴링이 예측 가능하고 결정적으로 수행되는지 보장하기 위한 분석을 수행하는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF407:소프트웨어 구조 설계명세에 포함된 구조설계 요소에 대한 크기(Size)가 추정되는가?	yes	0.90	0.10
	no	0.10	0.90

AF400_timing		good	bad
AF408:저장장치 크기 추정치가 운전, 유지보수 및 잠정적 개선을 위한 충분한 여유를 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

1.2 공정 특성 검증

◇ 완전성(completeness) 검증에 관한 노드확률테이블

AP100_completeness		good	bad
AP101:소프트웨어 구조가 소프트웨어 설계명세, 안전성분석보고서, 요구사항명세서 등에 있는 모든 예상되는 상황과 조건을 고려하는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP102:소프트웨어 구조설계 명세가 SRS에 명시된 설계 제한사항과 소프트웨어 요구사항을 만족하는 소프트웨어 구조를 묘사하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP103:소프트웨어를 프로그램 단위들이 높은 내부 응집도를 갖도록 소프트웨어 구조가 설계되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP104:소프트웨어를 프로그램 단위들 간에 낮은 연관도를 갖도록 소프트웨어 구조가 설계되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP105:프로그램 요소들이 명료한 시스템 구조가 되도록 체계적으로 구성되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP106:각 시스템 기능이 특정 소프트웨어 요소에서 수행되는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP107:소프트웨어 구조가 소프트웨어 요소간의 논리적 연결을 보여 주는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP108:소프트웨어 설계를 위해 일반적으로 인정된 소프트웨어공학 기법이 사용되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP109:소프트웨어 구조가 소프트웨어 요소들 간의 데이터 흐름과 제어흐름을 보여 주는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP110:운전환경이 구조설계에 반영되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP111:소프트웨어 구조가 각 인터페이스의 요구되는 행동을 허용하는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP112:소프트웨어 구조 설계가 SRS에 명시된 모든 운전모드를 고려하는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP113:소프트웨어 계통의 설계와 실 세계 제한사항에 의해 부과된 어떠한 제약사항을 평가하기 위하여 제한분석이 수행되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP114:데이터-관련 구조요소가 소프트웨어 요구사항과의 일치 여부를 결정하기 위한 자료분석이 수행되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP115:소프트웨어 구성요소간의 인터페이스가 정확하게 설계되었는지를 결정하기 위한 인터페이스 분석이 수행되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP116:특별한 어려움을 야기하는 모든 특정 구조 설계영역이 파악되고, 각각을 위한 완화 계획이 묘사되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP117:소프트웨어 구조가 시스템과 개발노력에 부과된 제약사항 내에서 구현되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP118:소프트웨어가 운전요원에 부담을 주는 것이 아니라 도움을 준다는 것을 보장할 수 있도록 소프트웨어 구조가 분석되고 검토되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP119:모든 데이터 채널의 응답시간, 샘플링 빈도, 숫자, 크기, 데이터 율(data rate) 등이 소프트웨어 구조설계 명세에 정의되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP120:운전원이나 보수요원이 쉽게 디지털 시스템 상태를 평가하고 문제영역을 찾아낼 수 있도록 적절한 인간공학 고려사항들이 구조설계에 포함되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP121:구조 설계에 하드웨어 환경이 고려되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP122:구조 설계가 각 소프트웨어 요소가 동작할 하드웨어 요소를 명시하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP123:특정 하드웨어가 요구된다면 소프트웨어 구조설계 명세가 저장장치 용량, 명령어 집합, 속도, 입출력 레지스터 등을 포함하여 그 컴퓨터 내부의 각 컴퓨터를 파악하는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP124:설계구조 명세에 주 하드웨어 요소가 나타나는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP125:소프트웨어 구조는 하드웨어 요소들간의 물리적 연결을 보여주는가?	yes	0.90	0.10
	no	0.10	0.90

AP100_completeness		good	bad
AP126:하드웨어 구조가 존재하는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 일관성(consistency) 검증에 관한 노드확률테이블

AP200_consistency		good	bad
AP201:소프트웨어 구조설계에서 사용된 하나 이상의 정형기법(formal method)은 이들 상호간의 일관성을 유지하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP200_consistency		good	bad
AP202:소프트웨어 구조설계 설명서(software architecture design description)의 표현 스타일과 상세화 수준이 일관성을 갖는가?	yes	0.90	0.10
	no	0.10	0.90

AP200_consistency		good	bad
AP203:소프트웨어 구조도는 소프트웨어 구조설계 설명서 및 프로그램 작동환경과 일치하는가?	yes	0.90	0.10
	no	0.10	0.90

AP200_consistency		good	bad
AP204:소프트웨어 구조도의 요소는 소프트웨어 요구사항 명세서와 일치하는가?	yes	0.90	0.10
	no	0.10	0.90

AP200_consistency		good	bad
AP205:소프트웨어 구조도는 소프트웨어 구조도 요소(component)간의 상관관계의 정확성 (correctness)을 입증할 수 있도록 분석되었는가?	yes	0.90	0.10
	no	0.10	0.90

AP200_consistency		good	bad
AP206:소프트웨어 구조도는 하드웨어 구조와 일치하는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 스타일(style) 검증에 관한 노드확률테이블

AP300_style		good	bad
AP301:소프트웨어 구조설계 설명서(software architecture design description)는 각 구조설계 요소의 기능(function)을 서술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP302:소프트웨어 구조설계 설명서는 구조설계 요소들간의 관계성(relationship)을 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP303:소프트웨어 구조설계 문서상에 설계구조 선택의 타당성을 설명하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP304:소프트웨어 설계요소 중 처리(processing) 부분을 포함하고 있다면 소프트웨어 구조설계 설명서에 처리기능의 요소를 실행하는 방법(method)을 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP305:소프트웨어 구조설계 설명서에 각 구조설계 요소의 기능수행을 위하여 필요한 자원(resources)을 규정하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP306:소프트웨어 설계구조 요소의 상호작용 설명에는 타이밍 트리거 사건들, 실행의 순서, 자료공유, 상호작용에 영향을 미치는 다른 요소 등을 포함하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP307:소프트웨어 구조설계 설명서에 각 설계요소의 유형(type)을 확인할 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP308:소프트웨어 구조설계 설명서에 적용한 표준(standard)을 확인할 수 있는가	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP309:만약 소프트웨어 설계요소가 데이터저장(data store), 메시지 또는 스크린 디스플레이라면 소프트웨어 구조설계 설명서에는 이들에 대한 일반적인 구조를 설명하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP300_style		good	bad
AP310:소프트웨어 구조설계의 각 요소들은 유일한 명칭을 가지고 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 추적성(traceability) 검증에 관한 노드확률테이블

AP400_traceability		good	bad
AP401:소프트웨어 구조설계 요소가 소프트웨어 요구사항 명세서와 일치하는지 확인하는데 사용하는 기법인 특정 시험(specific tests) 또는 검증기준(validation criteria)을 이용하여 전향추적을 할 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP400_traceability		good	bad
AP402:소프트웨어 구조설계 설명서에는 소프트웨어 구조를 결정짓는데 적절히 활용될 수 있도록 소프트웨어 구조설계에 대한 참조 주목사항(notes)을 포함하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP400_traceability		good	bad
AP403:소프트웨어 구조설계 설명서에 소프트웨어 요구사항 명세서에 기술된 요건의 범위를 벗어나는 어떤 기능을 포함하고 있다는 정당한 근거가 있다면 요구사항의 범위를 벗어난 소프트웨어 구조설계 설명서에 대한 정당성 결과가 요구사항 명세서와 일치하는가?	yes	0.90	0.10
	no	0.10	0.90

AP400_traceability		good	bad
AP404:소프트웨어 구조설계 설명서에는 소프트웨어 구조설계의 각 기능에 대해 확인할 수 있는 특정 식별자(ID)를 부여하고 있어서 구현 시 이들 식별자를 통해 추적할 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP400_traceability		good	bad
AP405:각 소프트웨어 구조설계 요소는 소프트웨어 요구사항 명세서의 특정 요소(specific elements)를 후향 추적(backward trace) 할 수 있도록 되어있는가?	yes	0.90	0.10
	no	0.10	0.90

AP400_traceability		good	bad
AP406:소프트웨어 구조설계 상에서 소프트웨어 요구사항 명세서에 기입된 설계 제약조건 등을 점검할 수 있도록 되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 확인가능성(verifiability) 검증에 관한 노드확률테이블

AP500_verifiability		good	bad
AP501:소프트웨어 구조설계 사양은 각 소프트웨어 설계요소의 기능에 대해 시험할 수 있는 방법을 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

AP500_verifiability		good	bad
AP502:시스템 및 하부시스템 시험절차서, 시험 사례 들을 개발하기 위하여 소프트웨어 구조도 분석을 하였는가?	yes	0.90	0.10
	no	0.10	0.90

2. SDS 상세설계 검증

기능 특성/코드	공정 특성/코드
정확도(Accuracy)/SF1	완전성(completeness)/SP1
신뢰성(reliability)/SF2	일관성(consistency)/SP2
강인성(Robustness)/SF3	정확성(Correctness)/SP3
안전성(safety)/SF4	스타일(style)/SP4
보안성(security)/SF5	추적성(traceability)/SP5
타이밍(timing)/SF6	확인가능성(verifiability)/SP6

2.1 기능 특성 검증

◇ 정확도(Accuracy) 검증에 관한 노드확률테이블

SF100_accuracy		good	bad
SF100_accuracy_1:소프트웨어 상세설계에서 사용한 모든 계산들은 계산에 대한 정확성 요구사항을 만족하는가?	yes	0.80	0.20
	no	0.20	0.80

SF100_accuracy		good	bad
SF100_accuracy_2:소프트웨어 상세 설계 시 입력변수 값을 검증하기 위한 방법을 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 신뢰성(reliability) 검증에 관한 노드확률테이블

SF200_reliability		good	bad
SF200_reliability_1:소프트웨어 설정(configuration)에서 초기화 검사(initialization checks)가 있는가?	yes	0.90	0.10
	no	0.10	0.90

SF200_reliability		good	bad
SF200_reliability_2:데이터 일관성(consistency)을 보증하기 위해 해야 할 것이 무엇인가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_3:교착상태(deadlock)에 대한 잠재성이 존재하는지, 그리고 그와 같은 잠재성이 있다면 무슨 단계들이 교착상태를 방지하기 위해 취해져야 할 것인지를 결정하기 위해 분석이 수행되고 문서화되어야 하는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_4:시간 종료검사(time-out checks-software watch-dogs)가 소프트웨어에 적용되고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SF200_reliability		good	bad
SF200_reliability_5:공통-모드 소프트웨어 고장(common-mode software failures)이 소프트웨어 상세 설계에서 분석되고 고려되는가?	yes	0.90	0.10
	no	0.10	0.90

SF200_reliability		good	bad
SF200_reliability_6:고장 이벤트에서 시스템 상태를 기록하는 설비를 소프트웨어 상세설계에서 포함하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SF200_reliability		good	bad
SF200_reliability_7:오류탐지 이벤트에서 취해지는 조치사항(actions)을 소프트웨어 상세설계에서 확인하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_8:프로그램을 수행하고 있는 동안 고장 파급(failure propagation)을 방지하기 위한 기법들을 사용하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SF200_reliability		good	bad
SF200_reliability_9:고장이 탐지되는 이벤트에서 정의가 잘 된 출력의 생성능력(capacity)을 소프트웨어 상세설계에서 포함하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SF200_reliability		good	bad
SF200_reliability_10:완전하고 정확한 오류 회복 기법(error recovery techniques)을 소프트웨어 상세설계에서 명세하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_11:소프트웨어 상태를 표시하는(displaying) 설비가 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_12:신뢰도 및 가용도 분석이 신뢰도 요구사항을 만족하는지 보증하는 충분하게 지원 가능한 데이터(supportable data)를 포함하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_13:적절한 운전원 조작 신호가 발생하는지를 소프트웨어 상세설계에서 포함하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_14:소프트웨어 상세설계가 모든 운전원 입력에 대한 검증 검사를 포함하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SF200_reliability		good	bad
SF200_reliability_15:통신 라인(communication lines)이 동작하고 있는지를 검사하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_16:입출력 통신에 대한 무결성 검사(integrity checks)가 수행되고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_17:통신 메시지가 검증되는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_18:절단 또는 쇼트(broken or shorted)된 입출력 통신 라인에 대한 검사가 이루어 지고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_19:누락(missing) 그리고/또는 지연 메시지(late messages)에 대한 검사가 이루어 지고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_20:하드웨어나 소프트웨어 결함에 기인하는 손상(damaged)이나 손실 데이터(lost data)를 소프트웨어 상세 설계에서 탐지하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF200_reliability		good	bad
SF200_reliability_21:소프트웨어나 하드웨어 결함을 다루기 위한 표준 전략이 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_22:탐지된 모든 하드웨어와 소프트웨어 고장들이 보고되고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_23:충분한 기억장치 그리고/또는 통신시험 적용범위(coverage)가 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF200_reliability		good	bad
SF200_reliability_24:하드웨어 결함을 탐지하기 위한 주기적 소프트웨어 검사가 이 하드웨어에서 이루어 지고 있는가?	yes	0.70	0.30
	no	0.30	0.70

◇ 강인성(robustness) 검증에 관한 노드확률테이블

SF300_robustness		good	bad
SF300_robustness_1:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 입력의 발생에도 정확하게 동작할 설계인가?	yes	0.90	0.10
	no	0.10	0.90

SF300_robustness		good	bad
SF300_robustness_2:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 운전원 행위(operator behavior)의 발생에도 정확하게 동작할 설계인가?	yes	0.90	0.10
	no	0.10	0.90

SF300_robustness		good	bad
SF300_robustness_3:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 하드웨어 행위(hardware behavior)의 발생에도 정확하게 동작할 설계인가?	yes	0.80	0.20
	no	0.20	0.80

SF300_robustness		good	bad
SF300_robustness_4:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 소프트웨어 행위(software behavior)의 발생에도 정확하게 동작할 설계인가?	yes	0.85	0.15
	no	0.15	0.85

◇ 안전성(safety) 검증에 관한 노드확률테이블

SF400_safety		good	bad
SF400_safety_1:소프트웨어 상세설계의 제어논리(control logic)가 안전성-필수 요건(safety-critical requirements))을 완전하고 정확하게 구현하는가?	yes	0.90	0.10
	no	0.10	0.90

SF400_safety		good	bad
SF400_safety_2:확인된 각 안전성-필수 데이터를 변경할 수 있는 상세설계 요소(detailed design elements)를 갖는가?	yes	0.80	0.20
	no	0.20	0.80

SF400_safety		good	bad
SF400_safety_3:소프트웨어 상세설계가 안전성이 다른 수준의 계통이나 하부계통들 사이의 상호작용(interactions)에 대하여 엄격한 제어를 제공하는가?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_4:확인된 안전성-필수 데이터가 형(types), 단위(units), 범위(ranges), 그리고 오류한계(error bounds)와 함께 있는가?	yes	0.80	0.20
	no	0.20	0.80

SF400_safety		good	bad
SF400_safety_5:소프트웨어 상세설계의 공식(equations)과 알고리즘(algorithms)이 안전성-필수 요건을 정확하게 구현하는가?	yes	0.90	0.10
	no	0.10	0.90

SF400_safety		good	bad
SF400_safety_6:안전성에 영향을 주는 특정 시간, 비용, 복잡도, 또는 다른 특정 문제점이 있다면, 완화 계획(mitigation plans)을 통해 문서화되고 제기되었는가?	yes	0.80	0.20
	no	0.20	0.80

SF400_safety		good	bad
SF400_safety_7:소프트웨어 상세설계가 자원-필수 단위나 하부계통(subsystem)을 확인하기 위해 분석되었는가?	yes	0.70	0.30
	no	0.30	0.70

SF400_safety		good	bad
SF400_safety_8:소프트웨어 상세설계가 소프트웨어가 안전하고 알려진 상태로 복귀하도록 하는 비정상 탐지검사(anomaly detection checks)를 포함하는가?	yes	0.90	0.10
	no	0.10	0.90

SF400_safety		good	bad
SF400_safety_9:안전성-관련 진단메시지가 다른 사용자들의 지식과 능력에 적합한 방식으로 정의되었는가?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_10:소프트웨어 상세설계가 그 환경에서 어떻게 소프트웨어 시스템이 낮은 확률사건이나 고장에 반응하는지를 결정하기 위해 분석되었는가?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_11:모든 비-안전성 단위가 소프트웨어 안전성에 악영향을 주지 않는다는 적절한 확신을 제공하기 위해 분석되었는가?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_12:상세설계 요소들 사이의 인터페이스가 적절히 설계되고 안전위험(safety hazard)을 생성하지 않음을 결정하기 위해 수행되고 문서화된 설계 인터페이스 분석이 있었는가?	yes	0.70	0.30
	no	0.30	0.70

SF400_safety		good	bad
SF400_safety_13:데이터-관련 상세설계 요소들의 소프트웨어 요건과 일관되고 계통 안전성 요건을 위배하지 않는지를 결정하기 위해 수행되고 문서화된 설계데이터 분석이 있었는가?	yes	0.80	0.20
	no	0.20	0.80

SF400_safety		good	bad
SF400_safety_14:상세설계에서 설계 한계분석(design constraint analysis)이 수행되고 문서화 되었나?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_15:모든 안전성-필수 소프트웨어의 소프트웨어 상세설계에서 소프트웨어 안전성 분석이 수행되고 문서화 되었나?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_16:모든 안전성-필수 소프트웨어의 소프트웨어 상세설계에서 소프트웨어 안전성 분석이 수행되고 문서화 되었나?	yes	0.60	0.40
	no	0.40	0.60

SF400_safety		good	bad
SF400_safety_17:소프트웨어에 할당된 모든 안전성 요건에 따른 소프트웨어 상세설계임을 확증하기 위해 분석되었는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 보안성(security) 검증에 관한 노드확률테이블

SF500_security		good	bad
SF500_security_1:상세설계가 소프트웨어 요구명세에서 확인된 모든 보호위협(security threats)을 정확하게 제기하는가?	yes	0.90	0.10
	no	0.10	0.90

SF500_security		good	bad
SF500_security_2:상세설계가 권한이 없는 사람(unauthorized personnel)에 의한 공격(intrusions)을 탐지하기 위한 능력을 포함하는가?	yes	0.60	0.40
	no	0.40	0.60

SF500_security		good	bad
SF500_security_3:상세설계가 컴퓨터 바이러스를 탐지하는 능력을 포함하는가?	yes	0.60	0.40
	no	0.40	0.60

SF500_security		good	bad
SF500_security_4:정확하게 설계된 소프트웨어 요구명세에 접근제한(access restrictions)이 명세되는가?	yes	0.90	0.10
	no	0.10	0.90

SF500_security		good	bad
SF500_security_5:발전소 운전원에 의한 변경에 대하여 계통데이터나 코드가 안전하게 보호(secure) 되는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 타이밍(timing) 검증에 관한 노드확률테이블

SF600_timing		good	bad
SF600_timing_1:소프트웨어 상세설계 명세가 모든 타이밍 자원 제약, 각각을 처리하는(handling) 전략, 요구된 여유도(margins), 그리고 이러한 여유도를 측정하는 방법의 명세를 포함하는가?	yes	0.80	0.20
	no	0.20	0.80

SF600_timing		good	bad
SF600_timing_2:상세 설계가 실행 타이밍이 결정적임을 보증하는가? 이는 인터럽트, 단일 프로세서에서의 다중 프로세싱, 동적 메모리 관리, 그리고 시건-기반 통신(event-driven communications) 중 어느 것이 사용되는지를 실증하기 위해 특별히 중요하다.	yes	0.80	0.20
	no	0.20	0.80

SF600_timing		good	bad
SF600_timing_3:인터럽트가 정확하게 처리되고 입출력 타이밍이 정확하며, 그리고 단위들이 정확하게 연계되는지 확증하기 위해 중요한 인터페이스(potential interface)와 타이밍 문제가 분석되었는가?	yes	0.80	0.20
	no	0.20	0.80

SF600_timing		good	bad
SF600_timing_4:소프트웨어 상세설계에서 수행된 타이밍분석이 클럭(clock), 표본율(sampling rate), 그리고 센서 반응시간의 변화(variations)를 계산하기 위해 절절한 여유도를 허용하는가?	yes	0.70	0.30
	no	0.30	0.70

SF600_timing		good	bad
SF600_timing_5:컴퓨터 자원의 스캐쥴링이 동적이라기 보다는 예측적이고 결정적임을 확증하기 위해 분석이 수행되었는가?	yes	0.60	0.40
	no	0.40	0.60

SF600_timing		good	bad
SF600_timing_6:자체-검사 활동(self-checking actions)이 타이밍 요건을 위배하지 않음을 확증하기 위해 분석이 수행되었는가?	yes	0.70	0.30
	no	0.30	0.70

SF600_timing		good	bad
SF600_timing_7:요구된 계통 반응 시간이 만족된다는 것을 상세설계가 보증하는가?	yes	0.80	0.20
	no	0.20	0.80

SF600_timing		good	bad
SF600_timing_8:소프트웨어 상세설계 명세는 사용된 스캐쥴링 구조(scheduling mechanisms)와 IPC (inter-process communication mechanisms)를 명세하는가?	yes	0.60	0.40
	no	0.40	0.60

SF600_timing		good	bad
SF600_timing_9:각 단위에 대한 타이밍이 실행(execution)을 위한 최대 및 최소 시간으로 명세되었는가?	yes	0.80	0.20
	no	0.20	0.80

SF600_timing		good	bad
SF600_timing_10:다른 프로그램과 동기된 다른 순차적인 프로그램으로부터 데이터를 받거나 또는 그 프로그램으로 데이터를 전송하는 순차적인 프로그램의 실행이 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF600_timing		good	bad
SF600_timing_11:연산(operations)이 실행 속도와 독립적인 정확한 순서로 수행되도록 설계된 소프트웨어가 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF600_timing		good	bad
SF600_timing_12:프로그램을 실행하기 위해 주어진 시간이나 프로그램의 실행이 초기화되는 시간 중에서 독립적인 순차적인 프로그램에 의해 생성된 결과들이 있는가?	yes	0.60	0.40
	no	0.40	0.60

SF600_timing		good	bad
SF600_timing_13:인터럽트가 금지될 수 있는 시간(times)에 하드 상한(hard upper bounds) 범위 내에 존재하는가?	yes	0.60	0.40
	no	0.40	0.60

SF600_timing		good	bad
SF600_timing_14:소프트웨어 상세설계 명세가 모든 기억장치 자원 제약, 각각을 처리하는 전략, 요구된 여유도, 그리고 이러한 여유도를 측정하는 방법의 명세를 포함하는가?	yes	0.90	0.10
	no	0.10	0.90

SF600_timing		good	bad
SF600_timing_15:중앙처리장치(CPU), 기억장치, 그리고 입출력 채널을 위한 적절한 예산(budgets)이 확립되었나?	yes	0.60	0.40
	no	0.40	0.60

2.2 공정 특성 검증

◇ 완전성(Completeness) 검증에 관한 노드확률테이블

SP100_completeness		good	bad
SP100_completeness_1:각 단위 모듈이 오직 하나의 진입점을 가지는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_2:각 단위 모듈이 오직 하나의 출구점을 가지는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_3:명시된 모델, 알고리즘, 수치기법, 신호변환 및 자료처리 절차가 실용적으로 사용되는 범주 내의 것들인가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_4:소프트웨어 상세설계가 단위들 간의 규칙적이고 문서화된 연계를 사용하는가?	yes	0.90	0.10
	no	0.10	0.90

SP100_completeness		good	bad
SP100_completeness_5:모든 내부 및 외부연계가 완전하게 정의되었는가?	yes	0.90	0.10
	no	0.10	0.90

SP100_completeness		good	bad
SP100_completeness_6:상세설계가 연계요소들에 대한 문서화된 기술이나 알려진 특성과 일관성이 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_7:발전소 운전원을 지원하도록 설계되었는가를 확증하기 위해 소프트웨어 상세설계가 분석되고 검토되었는가?	yes	0.60	0.40
	no	0.40	0.60

SP100_completeness		good	bad
SP100_completeness_8:컴퓨터 시스템-인간 상호작용의 문법이 정의되는가?	yes	0.60	0.40
	no	0.40	0.60

SP100_completeness		good	bad
SP100_completeness_9:센서 및 구동기 설계연계가 신호의 동적범위, 정확성, 타이밍, 인터럽트 및 분해능을 포함하는가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_10:소프트웨어 상세설계가 시스템 및 개발 노력에 주어지는 제약조건들 안에서 구현될 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP100_completeness		good	bad
SP100_completeness_11:모든 자료채널에 대한 개수, 크기, 자료속도, 샘플링 빈도 및 응답시간이 소프트웨어 상세설계 명세서에 정의되는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_12:소프트웨어 상세설계가 단위 크기를 제한하는 요건을 명시하는가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_13:상세설계에 적절한 인간공학 요소들이 고려되어 운전원이나 유지보수자가 시스템의 상태를 평가하거나 문제가 있는 부분을 찾아내는데 용이하도록 하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP100_completeness		good	bad
SP100_completeness_14:소프트웨어 상세설계 명세서가 소프트웨어 요구사항 명세와 소프트웨어 구조설계 명세서에 기술된 시스템 요구사항 및 모든 제약조건들을 만족하는 소프트웨어 상세 설계를 기술하는가?	yes	0.90	0.10
	no	0.10	0.90

SP100_completeness		good	bad
SP100_completeness_15:컴퓨터 시스템과-인간 상호작용에 대한 정형적 절차가 소프트웨어 설계명세서에 명시되는가?	yes	0.60	0.40
	no	0.40	0.60

SP100_completeness		good	bad
SP100_completeness_16:특별한 어려움을 초래하는 특정 상세설계 분야(예를 들어, 공정 및 속도, 새로운 알고리즘, 보안, 새로운 하드웨어 및 언어)가 명시되고 그에 따른 완화책이 기술되는가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_17:실세계의 제한사항이나 소프트웨어 구조설계에 따른 상세설계의 제약사항들을 평가하기 위한 설계 제약사항 분석이 수행되었는가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_18:자료-관련 상세설계 요소들이 소프트웨어 요구사항 및 구조와 일관성을 유지하는가를 결정하기 위한 설계자료 분석이 수행되었는가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_19:소프트웨어 상세설계 공식, 알고리즘 및 제어논리가 소프트웨어 요구사항들을 정확하게 구현하는가를 결정하기 위한 설계논리 분석이 수행되었는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_20:소프트웨어 상세설계가 각 연계사항에 대하여 요구되는 프로그램의 행위를 구현하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_21:소프트웨어 코드를 구현하는데 필요한 모든 프로그램 입력, 출력, 및 자료 요소들이 적절한 범위로 명시되고 기술되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP100_completeness		good	bad
SP100_completeness_22:소프트웨어 요구사항명세서의 각 요구사항들이 상세설계 명세서로 반영되어 있는가?	yes	0.95	0.05
	no	0.05	0.95

SP100_completeness		good	bad
SP100_completeness_23:소프트웨어 상세설계에 하드웨어 환경(전기 접점, 전원 및 필터, 예비 전원, 프로세서의 선택, 입출력장치 속도 등)이 고려되었는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_24:소프트웨어 상세설계에 운전환경이 고려되었는가?	yes	0.80	0.20
	no	0.20	0.80

SP100_completeness		good	bad
SP100_completeness_25:소프트웨어 상세설계가 소프트웨어 요구사항 명세서에 명시된 모든 운전모드를 고려하는가?	yes	0.70	0.30
	no	0.30	0.70

SP100_completeness		good	bad
SP100_completeness_26:상세설계 요소들 사이의 연계가 정확하게 설계되었는가를 결정하기 위한 설계 연계분석이 수행되었는가?	yes	0.80	0.20
	no	0.20	0.80

◇ 일관성(Consistency) 검증에 관한 노드확률테이블

SP200_consistency		good	bad
SP200_consistency_1:표현 스타일이나 상세 정도가 소프트웨어 상세설계명세서 전반에 걸쳐 일관성이 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP200_consistency		good	bad
SP200_consistency_2:소프트웨어 상세설계가 하드웨어 및 소프트웨어 구조와 일관성이 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP200_consistency		good	bad
SP200_consistency_3:주변장치 접점들에 대한 표준연계가 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP200_consistency		good	bad
SP200_consistency_4:유사하거나 관련 있는 기능들에 대한 상세설계가 일관성이 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP200_consistency		good	bad
SP200_consistency_5:인간-기계 연계에 대한 표준연계가 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP200_consistency		good	bad
SP200_consistency_6:자료 전송에 대한 표준연계가 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP200_consistency		good	bad
SP200_consistency_7:설계에 명시된 모델, 알고리즘 및 수치기법이 수학적으로 상호 호환되는가?	yes	0.90	0.10
	no	0.10	0.90

SP200_consistency		good	bad
SP200_consistency_8:설계에 명시된 모델, 알고리즘 및 수치기법이 적절한 표준 참고문서의 것과 일치하는가?. 표준 참고문서가 존재한다면 그것들은 모델, 알고리즘, 수치기법을 설계하는 사람들이 제공해야 한다.	yes	0.80	0.20
	no	0.20	0.80

SP200_consistency		good	bad
SP200_consistency_9:각 소프트웨어 상세 설계요소가 프로그램이 작동하는 운전환경의 설명 및 특성들과 일관성이 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP200_consistency		good	bad
SP200_consistency_10:소프트웨어 상세설계가 소프트웨어 요구사항 명세와 일관성이 있는가?	yes	0.95	0.05
	no	0.05	0.95

SP200_consistency		good	bad
SP200_consistency_11:복수개의 정형적 상세설계 방법이 사용된다면 그들이 서로 일관성이 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP200_consistency		good	bad
SP200_consistency_12:소프트웨어 상세설계에 주어진 입력 및 출력 명세(자료형태, 자료 크기, 자료 속도, 정확성, 오류범위 및 물리적 단위 등 포함)가 하드웨어 및 기성 소프트웨어에 의해 부여된 연계 요구사항과 일관성이 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 정확성(Correctness) 검증에 관한 노드확률테이블

SP300_correctness		good	bad
SP300_correctness_1:모든 공식 및 알고리즘이 코딩이 시작될 수 있을 정도로 정확하게 또 충분히 상세한 수준까지 정의되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP300_correctness		good	bad
SP300_correctness_2:잠재적인 언더플로(underflow) 및 오버플로(overflow) 상태를 식별하기 위한 알고리즘 정확성 분석이 수행되었는가?	yes	0.60	0.40
	no	0.40	0.60

SP300_correctness		good	bad
SP300_correctness_3:서브루틴이나 프로시저를 빠져 나왔을 때 상태가 점검되고, 오류 상태가 지시되었을 때 적절한 행위가 수행되는가?	yes	0.60	0.40
	no	0.40	0.60

SP300_correctness		good	bad
SP300_correctness_4:잠재된 자료처리 문제(부정확한 자료 초기화, 저장 자료의 부정확한 평가, 부정확한 자료 측정 등)를 평가하기 위한 분석이 수행되었는가?	yes	0.80	0.20
	no	0.20	0.80

SP300_correctness		good	bad
SP300_correctness_5:격리, 분할, 자료 별칭(data aliasing) 및 결함 억제 현안들을 둘러싼 자료 의존성에 대한 자료 구조를 평가하기 위한 분석이 수행되었는가?	yes	0.70	0.30
	no	0.30	0.70

SP300_correctness		good	bad
SP300_correctness_6:상태변수의 병행성(concurrency) 및 일관성을 유지할 장치들이 소프트웨어 상세설계에 나타나는가?	yes	0.80	0.20
	no	0.20	0.80

SP300_correctness		good	bad
SP300_correctness_7:알고리듬이 입력 및 시간변수의 전체 범위에 대하여 안정적인가를 확증하기 위한 알고리즘 분석이 수행되었는가?	yes	0.70	0.30
	no	0.30	0.70

SP300_correctness		good	bad
SP300_correctness_8:각 단위의 입력에 대한 유효성(validity)이 점검되었는가?	yes	0.90	0.10
	no	0.10	0.90

SP300_correctness		good	bad
SP300_correctness_9:인터럽트가 예상치 못한 방식으로 안전하고 중요한 자료값을 변경하지 않는다는 확실한 증거가 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP300_correctness		good	bad
SP300_correctness_10:안전에 중요한 자료 값이 예상치 못한 방식 혹은 예상치 못한 상세설계 요소에 의해 변경되지 않는다는 확실한 증거가 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP300_correctness		good	bad
SP300_correctness_11:공식, 알고리즘 및 제어논리가 잠재된 문제점들(논리오류, 누락된 경우나 단계, 중복논리, 극한조건의 무시, 불필요한 기능, 오해, 조건시험 누락, 잘못된 변수에 대한 점검 및 루프의 반복 등)에 대하여 평가되는가?	yes	0.90	0.10
	no	0.10	0.90

SP300_correctness		good	bad
SP300_correctness_12:안전에 중요한 자료가 초기화되기 전에 사용되지 않는다는 확실한 증거가 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP300_correctness		good	bad
SP300_correctness_13:부동소수점 연산 및 재귀(recursion)가 사용되는 경우, 그 사용에 대한 적절한 정당화가 이루어져 있는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 스타일(Style) 검증에 관한 노드확률테이블

SP400_style		good	bad
SP400_style_1:소프트웨어 설계명세서(SDS)는 감시계통 성능(supervisory system performance) 및 신뢰도(reliability)에 대한 정량적인 데이터들에 대해 문서화하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_2:모든 감시기능들에 대해 확고한 정의가 있고 연계사항 (interfaces)은 체계를 잘 갖추고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_3:전역변수(global variables)의 사용이 금지되거나 또는 세밀하게 입증되도록 되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_4:소프트웨어 상세 설계명세는 타임 슬라이싱(time-slicing)의 사용을 제한하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_5:소프트웨어 상세 설계명세는 메모리 스와핑(memory swapping) 사용을 제한하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_6:필요한 모든 지원 소프트웨어들에 대해서 충분히 기술하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP400_style		good	bad
SP400_style_7:소프트웨어 상세 설계명세에서 프로그래밍언어 변수들에 대해서 명료한 정의 및 형(typing)이 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_8:소프트웨어 상세설계에서는 안전성-필수 기능(safety-critical functions)들의 수행시간 동안 소프트웨어 인터럽트(interrupts) 취급을 제한하도록 되어 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_9:사용될 구축 언어들은 소프트웨어 상세설계 사양에서 식별되도록 되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_10:소프트웨어 상세설계에서 특별하게 개발된 장치 드라이버(device drivers)를 사용하는데 대해서 논리적 근거를 제공하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_11:어떤 공유메모리 영역(shared memory locations)이 일치가 안된 상태에서 동시에 변경되지 않도록 상세설계에서 보장하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SP400_style		good	bad
SP400_style_12:인터럽트를 피하는 최대 가능한 루프 수를 소프트웨어 상세설계에서 기술하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_13:데이터 저장공간(store), 메시지 또는 스크린 디스플레이(screen display)가 소프트웨어 상세 설계요소일 때 소프트웨어 상세설계 명세에서 이 요소의 구조에 대해서 기술하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SP400_style		good	bad
SP400_style_14:소프트웨어 상세 설계명세는 간단한 폼에 대한 (즉, 순서(sequence), 사례 선택, 반복, 추상화(abstractions) (예를 들어 절차, 기능 그리고/또는 서브루틴)) 제어구조를 제한하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_15:소프트웨어 상세설계 요소가 어떤 절차를 수행하면서 요소가 기능을 수행하는 방법에 대해서 소프트웨어 상세 설계명세에서 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_16:소프트웨어 상세설계 명세에서 소프트웨어 상세설계의 일환으로 기술되는 구축 제약사항(implementation constraints)들에 대해서 모두 열거하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_17:소프트웨어 상세설계 명세에서 상세설계 의사결정에 대한 논리적 근거에 대해서 문서화하고 있는가?	yes	0.60	0.40
	no	0.40	0.60

SP400_style		good	bad
SP400_style_18:소프트웨어 상세설계 명세에서 필요로 하는 프로그래밍 언어에 대한 표준들에 대해서 식별하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SP400_style		good	bad
SP400_style_19:각 상세설계 요소는 특정한 이름을 갖도록 하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_20:소프트웨어 상세설계 명세는 각 상세설계 요소의 형태(예를 들면, 체계, 하부체계, 유니트, 데이터베이스, 파일, 데이터 구조, 스크린 디스플레이, 메시지, 프로그램 또는 절차)에 대해서 식별하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

SP400_style		good	bad
SP400_style_21:소프트웨어 상세설계 명세는 각 상세설계 요소의 기능(즉, 무엇을 하는가)에 대해서 기술하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SP400_style		good	bad
SP400_style_22:소프트웨어 상세설계 명세는 상세설계 요소들 간의 관계에 대해서 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_23:상세설계 요소간의 상호작용 기술은 타이밍, 사건유발(triggering events), 수행순서, 데이터 공유, 그리고 상호작용에 영향을 미치는 다른 작용요인을 포함하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_24:소프트웨어 상세설계 명세에서 각 상세설계 요소가 그 기능을 수행하기 위한 자원들을 명기하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP400_style		good	bad
SP400_style_25:소프트웨어 상세설계 명세는 적용 가능한 표준에 준수해서 작성되도록 되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 추적성(Traceability) 검증에 관한 노드확률테이블

SP500_traceability		good	bad
SP500_traceability_1:각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?	yes	0.90	0.10
	no	0.10	0.90

SP500_traceability		good	bad
SP500_traceability_2:각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?	yes	0.90	0.10
	no	0.10	0.90

SP500_traceability		good	bad
SP500_traceability_3:각 상세설계 요소에 대해서 특정 코드 요소로의 전방향 추적기록이 가능한가?	yes	0.80	0.20
	no	0.20	0.80

SP500_traceability		good	bad
SP500_traceability_4:요구사항에서 열거된 상세설계 제약사항들이 상세설계 내에서 따르도록 되어 있는가?	yes	0.70	0.30
	no	0.30	0.70

SP500_traceability		good	bad
SP500_traceability_5:소프트웨어 상세설계 명세는 소프트웨어 상세설계 내의 각 기능을 특정하게 식별하도록 되어 있어 구축 시 특정하게 참조 될 수 있도록 되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

SP500_traceability		good	bad
SP500_traceability_6:SRS에서 기술된 요구사항 범위가 아닌 기능들을 포함하는데 대해 정당성이 입증되어 있는가? 그리고 결과적인 설계 상세내용이 SRS에서 의도한 바와 일치하는가?	yes	0.80	0.20
	no	0.20	0.80

SP500_traceability		good	bad
SP500_traceability_7:소프트웨어 상세설계 사양이 소프트웨어 상세설계와 관련된 상세설계 의사결정을 문서화한 설계노트(design notes)를 참조하도록 되어 있는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 확인성(verifiability) 검증에 관한 노드확률테이블

SP600_verifiability		good	bad
SP600_verifiability_1:소프트웨어 상세설계 명세에서는 시험 가능한 방법으로 각 소프트웨어 요소의 기능들을 기술하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

SP600_verifiability		good	bad
SP600_verifiability_2:소프트웨어 상세설계 분석에서 컴포넌트 시험 절차 및 시험 사례들을 발하는데 사용되어 왔는가?	yes	0.90	0.10
	no	0.10	0.90

3. SDS 상세 검증

3.1 설계 추적성 상세 검증

DV100_traceability		good	bad
DV100_traceability_1:소프트웨어 구조설계 설명서에는 소프트웨어 구조설계의 각 기능을 유일하게 확인할 수 있는 식별자(ID)를 가지고 있어서 구현(implementation)시 이들 식별자를 가지고 추적할 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV100_traceability		good	bad
DV100_traceability_2:소프트웨어 상세설계 명세는 소프트웨어 상세설계 내의 각 기능을 특정하게 식별하도록 되어 있어 구축 시 특정하게 참조 될 수 있도록 되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV100_traceability		good	bad
DV100_traceability_3:소프트웨어 구조설계 요소가 소프트웨어 요구사항 명세서와 일치하는지 확인하는데 사용하는 기법인 특별시험(specific tests) 또는 검증기준(validation criteria)을 이용하여 전향추적을 할 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV100_traceability		good	bad
DV100_traceability_4:각 소프트웨어 구조설계 요소는 소프트웨어 요구사항 명세서의 특별한 요소(specific elements)를 후향 추적(backward trace) 할 수 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV100_traceability		good	bad
DV100_traceability_5:각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?	yes	0.98	0.20
	no	0.20	0.80

DV100_traceability		good	bad
DV100_traceability_6:각 상세설계 요소가 소프트웨어 요구사항 명세(Software Requirements Specification)내 특정 요소들로 역방향 추적 기록이 가능한가?	yes	0.90	0.10
	no	0.10	0.90

DV100_traceability		good	bad
DV100_traceability_7:소프트웨어 구조설계 설명서에 소프트웨어 요구사항 명세서에 기술된 요건의 범위를 벗어나는 어떤 기능을 포함하고 있다는 정당한 근거가 있다면 요건의 범위를 벗어난 소프트웨어 구조설계 설명서의 정당성 결과가 요구사항명세서와 일치하는가?	yes	0.90	0.10
	no	0.10	0.90

3.2 설계 정확성 상세 검증

◇ 소프트웨어 설계 정의의 정확성

DV210_Def_correctness		good	bad
DV210_Def_correctness_1:각 운전모드에서 실행되어야 하는 소프트웨어의 기능 및 행위가 정확하게 기술되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV210_Def_correctness		good	bad
DV210_Def_correctness_2:알고리즘 (논리 및 공식)이 정확한가?	yes	0.90	0.10
	no	0.10	0.90

DV210_Def_correctness		good	bad
DV210_Def_correctness_3:각 기능들이 결정론적인가? 각 기능들을 비결정론적으로 만들 수 있는 조건들 (오류 복구 등)이 정의되어 있는가?	yes	0.70	0.30
	no	0.30	0.70

DV210_Def_correctness		good	bad
DV210_Def_correctness_4:기능이 단일고장요건을 만족하는가?	yes	0.80	0.20
	no	0.20	0.80

DV210_Def_correctness		good	bad
DV210_Def_correctness_5:안전계통 소프트웨어에 대한 임의의 변경 및 제어를 차단하는 수단이 있는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 소프트웨어 입/출력 정의의 정확성 검증에 관한 노드확률테이블

DV220_IO_correctness		good	bad
DV220_IO_correctness_1:각 기능들이 그 기능에 요구되는 입력 및 출력을 정확하게 명시하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV220_IO_correctness		good	bad
DV220_IO_correctness_2:FBD 및 상세모듈에 모든 입력 및 출력의 출발점이 맞게 기술되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

◇ 소프트웨어 행위 명세의 정확성

DV230_Act_correctness		good	bad
DV230_Act_correctness_1:각 기능이 어떻게 시작되는가 (초기화 조건)를 정확하게 기술하고 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV230_Act_correctness		good	bad
DV230_Act_correctness_2:각 기능적 요구사항이 그 기능을 수행하는데 요구되는 작업 순서, 행위 및 사건들을 정확하게 명시하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV230_Act_correctness		good	bad
DV230_Act_correctness_3:각 기능적 요구사항이 종결 조건이나 기능의 종료 시 시스템의 상태에 대하여 정확하게 명시하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV230_Act_correctness		good	bad
DV230_Act_correctness_4:Function들 사이에 순환적 의존관계가 존재하지 않는가?	yes	0.70	0.30
	no	0.30	0.70

DV230_Act_correctness		good	bad
DV230_Act_correctness_5:각 기능의 정상적인 완료를 방해하는 조건이나 입력이 없는가?	yes	0.70	0.30
	no	0.30	0.70

DV230_Act_correctness		good	bad
DV230_Act_correctness_6:각 기능이 임의의 설계기준사고시에도 요구되는 기능을 완전히 수행할 수 있는가?	yes	0.60	0.40
	no	0.40	0.60

DV230_Act_correctness		good	bad
DV230_Act_correctness_7:비안전계통의 임의의 단일고장이 안전계통의 일부분에 영향을 주더라도 남은 안전계통 부분들이 적절한 안전기능을 수행할 수 있는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 소프트웨어 인터페이스 기술의 정확성

DV240_IF_correctness		good	bad
DV240_IF_correctness_1:타 채널 및 프로세서와의 신호 연계가 안전기능을 방해하지 않는가?	yes	0.60	0.40
	no	0.40	0.60

DV240_IF_correctness		good	bad
DV240_IF_correctness_2:비안전계통과의 연계가 안전계통에 요구되는 기능을 방해하지 않는가?	yes	0.60	0.40
	no	0.40	0.60

3.3 설계 일관성 상세 검증

◇ 소프트웨어 설계 정의의 일관성 검증에 관한 노드확률테이블

DV310_Def_consistency		good	bad
DV310_Def_consistency_1:각각의 상세설계가 소프트웨어의 요구사항들과 일관되는가?	yes	0.90	0.10
	no	0.10	0.90

DV310_Def_consistency		good	bad
DV310_Def_consistency_2:한 기능에 대한 명세 및 비슷한 기능들에 대한 명세가 서로 일관성이 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV310_Def_consistency		good	bad
DV310_Def_consistency_3:내부적 일관성 측면에서, Block Diagram으로 표현된 기능과 알고리즘으로 표현된 기능 사이에 일관성이 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV310_Def_consistency		good	bad
DV310_Def_consistency_4:정의된 용어 (또는 변수)를 사용하는 기능이 그것들의 정의와 모순되지 않는가?	yes	0.80	0.20
	no	0.20	0.80

◇ 소프트웨어 입출력 설계의 일관성 검증에 관한 노드확률테이블

DV320_IO_consistency		good	bad
DV320_IO_consistency_1:입력, 계산 및 출력 자료들에 요구되는 정확성이 상호 호환성이 있는가?	yes	0.60	0.40
	no	0.40	0.60

DV320_IO_consistency		good	bad
DV320_IO_consistency_2:입력 및 출력의 명세가 요구사항에 나오는 설명에 맞게 기술되어 있는가?	yes	0.70	0.30
	no	0.30	0.70

◇ 소프트웨어 설계명세의 일관성 검증에 관한 노드확률테이블

DV330_Spec_consistency		good	bad
DV330_Spec_consistency_1:SDS에 명시된 모델, 알고리즘, 및 계산식들이 적용 가능한 표준 및 참고문헌과 일치성이 있는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 소프트웨어 인터페이스 설계기술의 일관성 검증에 관한 노드확률테이블

DV340_IF_consistency		good	bad
DV340_IF_consistency_1:SDS에 있는 입력 및 출력이 하드웨어 및 기성 소프트웨어 등에 의해 부여되는 연계 요구사항 들과 일관되는가?	yes	0.60	0.40
	no	0.40	0.60

3.4. 소프트웨어 설계 완전성 검증

◇ 소프트웨어 설계(기능?) 정의의 완전성 검증에 관한 노드확률테이블

DV410_Def_completeness		good	bad
DV410_Def_completeness_1:소프트웨어가 수행해야 하는 모든 운전모드가 기술되어 있으며, 각 운전모드에서 실행되어야 하는 소프트웨어의 기능 및 행위가 모두 기술되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV410_completeness		good	bad
DV410_Def_completeness_2:각 운전모드 간의 전환을 일으키는 조건들이 명시되어 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV410_completeness		good	bad
DV410_Def_completeness_3:SRS에 기술된 모든 기능적 요구사항들이 SDS에 함수로 기술되었는가?	yes	0.90	0.10
	no	0.10	0.90

DV410_completeness		good	bad
DV410_Def_completeness_4:각각의 입력이 최소한 하나의 FBD에 의해 사용되었는가?	yes	0.60	0.40
	no	0.40	0.60

DV410_completeness		good	bad
DV410_Def_completeness_5:각 출력을 정의하는 FBD가 유일하게 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV410_completeness		good	bad
DV410_Def_completeness_6:Function의 performance criteria (error performance 포함) 가 모두 정의되었는가?	yes	0.60	0.40
	no	0.40	0.60

◇ 소프트웨어 입출력 정의의 완전성 검증에 관한 노드확률테이블

DV420_IO_completeness		good	bad
DV420_IO_completeness_1:입력 및 출력이 완전히 정의되어 있는가?	yes	0.90	0.10
	no	0.10	0.90

DV420_IO_completeness		good	bad
DV420_IO_completeness_2:Block Diagram 및 Function Block에 모든 입력 및 출력이 기술되었는가?	yes	0.80	0.20
	no	0.20	0.80

DV420_IO_completeness		good	bad
DV420_IO_completeness_3:입력변수가 갖추어야 할 특성이 모두 정의되었는가?	yes	0.80	0.20
	no	0.20	0.80

DV420_IO_completeness		good	bad
DV420_IO_completeness_4:출력변수가 갖추어야 할 특성이 모두 정의되었는가?	yes	0.80	0.20
	no	0.20	0.80

◇ 소프트웨어 행위명세 완전성 검증에 관한 노드확률테이블

DV430_Act_completeness		good	bad
DV430_Act_completeness_1:오류 사항들이 수정조치 사항과 함께 기술되어 있는가?	yes	0.70	0.30
	no	0.30	0.70

DV430_Act_completeness		good	bad
DV430_Act_completeness-2:SDS가 비정상적인 입력에 대한 소프트웨어의 행위를 명시하고 있는가?	yes	0.80	0.20
	no	0.20	0.80

DV430_Act_completeness		good	bad
DV430_Act_completeness_3:각 상세설계가 그 기능을 수행하는데 요구되는 작업 순서, 행위, 사건 및 타이밍을 명시하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

DV430_Act_completeness		good	bad
DV430_Act_completeness_4:고장-안전 차원에서 컴퓨터 시스템에 요구되는 모든 행위가 완전하게 기술되어 있는가?	yes	0.60	0.40
	no	0.40	0.60

DV430_Act_completeness		good	bad
DV430_Act_completeness_5:SDS가 소프트웨어가 수행하지 말아야 할 것들도 언급하고 있는가?	yes	0.70	0.30
	no	0.30	0.70

◇ 소프트웨어 인터페이스 기술의 완전성 검증에 관한 노드확률테이블

DV440_IF_completeness		good	bad
DV440_IF_completeness_1:다음과 같은 각 프로세서간의 연계가 모두 정의되었는가? - 공정계측입력 (PI Inputs) - 노외핵계측 (ENF Instrument) - 노심보호연산기 (CPC) - 자동시험 및 연계 프로세서 (ATIP) - 캐비닛 운전원 모듈 (COM) - 동시논리프로세서 (CP)	yes	0.60	0.40
	no	0.40	0.60

부록C. 원자로 보호계통 SW(BP) 설계명세 평가를 위한 BBN 입력 값

1. SDS 구조설계 검증

기능 특성/코드	공정 특성/코드
신뢰성(reliability)/AF1	완전성(completeness)/AP1
안전성(safety)/AF2	일관성(consistency)/AP2
보안성(security)/AF3	스타일(style)/AP3
타이밍(timing)/AF4	추적성(traceability)/AP4
	확인가능성(verifiability)/AP5

1.1 기능 특성 검증

◇ 신뢰성(reliability) 검증에 관한 평가 값

신뢰성: AF100_reliability		yes	no
1	AF101:소프트웨어 구조에 의해 고장을 감지하였을 경우 취해지는 행위들을 명시하고 있는가?	0.5	0.5
2	AF102:프로그램 수행 동안 고장파급을 방지할 수 있는 기술이 사용되었는가?	0.5	0.5
3	AF103:컴퓨터 시스템 구조 설계 시 소프트웨어 공통모드 고장 가능성을 분석하고 고려하였는가?	0.5	0.5
4	AF104:소프트웨어 구조가 사고발생시 시스템 상태를 기록할 기능을 가지고 있는가?	0.5	0.5
5	AF105:신뢰도와 가용성 분석이 신뢰도 요구사항 만족을 보장할 수 있도록 충분한 보충자료를 포함하고 있는가?	0.5	0.5
6	AF106:소프트웨어 설계구조가 고장 감지 시 잘 정의된 출력을 생산할 능력을 포함하는가?	0.5	0.5
7	AF107:소프트웨어 구조가 소프트웨어와 하드웨어의 상태를 명시할 기능을 가지고 있는가?	0.5	0.5
8	AF108:하드웨어와 소프트웨어 오류를 처리할 표준 전략이 있는가?	0.5	0.5
9	AF109:소프트웨어 구조는 감지된 하드웨어, 소프트웨어 고장을 보고할 기능을 가지고 있는가?	0.5	0.5
10	AF110:소프트웨어 구조는 적절한 watch-dog 타이머를 가지고 있는가?	0.5	0.5
11	AF111:개별 하드웨어 단위의 고장이 컴퓨터 시스템 고장을 야기하지 않는 구조인가?	0.5	0.5
12	AF112:디지털 시스템이 운전원 오류를 감지하거나 감내할 것인가?	0.5	0.5
13	AF113:소프트웨어 구조가 부적절한 운전원 행위를 통보할 방법을 가지고 있는가?	0.5	0.5

◇ 안전성(safety) 검증에 관한 평가 값

안전성: AF200_safety		yes	no
1	AF201:제어 로직이 오류처리, 비정상 및 비상처리 요구사항을 정확히 구현	0.5	0.5

	하고 있는가?		
2	AF202:각 필수 안전구조 요소의 결함분석이 수행되었는가?	0.5	0.5
3	AF203:소프트웨어 구조가 다른 안전 중요도를 가지는 하부 계통간의 상호작용을 엄격하게 제어하는가?	0.5	0.5
4	AF204:소프트웨어 구조가 소프트웨어 상태를 안전하고 알려진 상태로 되돌리도록 하는 비정상 감지 검사기능을 가지고 있는가?	0.5	0.5
5	AF205:소프트웨어 구조에 대하여 설계 제한분석을 수행하였는가?	0.5	0.5
6	AF206:모든 안전-필수 소프트웨어의 소프트웨어 구조에 대한 소프트웨어 안전성분석이 수행되었는가?	0.5	0.5
7	AF207:각 안전-필수 데이터 항목을 변경할 수 있는 구조적 요소가 파악되었는가?	0.5	0.5
8	AF208:소프트웨어 구조의 제어 로직이 안전-필수 요구사항을 완전하고 정확하게 구현하고 있는가?	0.5	0.5
9	AF209:안전에 영향을 줄 시간, 비용, 복잡도 또는 특별한 문제가 있다면, 완화계획을 통해 기록되고 있는가?	0.5	0.5
10	AF210:소프트웨어 구조가 자원에 치명적인 요소를 식별할 수 있도록 분석되었는가?	0.5	0.5
11	AF211:소프트웨어 계통이 그 환경에서 낮은 확률사고나 고장에 어떻게 반응할 것인가를 결정하기 위해 그 소프트웨어 구조가 분석되었는가?	0.5	0.5
12	AF212:소프트웨어 구조설계가 소프트웨어에 할당된 모든 안전 요구사항과 부합함을 분석하였는가?	0.5	0.5
13	AF213:모든 비-안전 구조요소가 소프트웨어 안전에 급작스럽게 영향을 주지 않음을 분석하였는가?	0.5	0.5
14	AF214:안전에 중요한 기능이 잘 정의되고 엄격하게 통제되는 연계에 의하여 정상운전 및 부가기능과 분리되어 있는가?	0.5	0.5

◇ 보안성(security) 검증에 관한 평가 값

보안성: AF300_security		yes	no
1	AF301:구조가 소프트웨어 요구사항 명세(SRS)에 명시된 보안 위협들을 정확하게 다루고 있는가?	0.5	0.5
2	AF302:구조가 SRS에 명시된 접근제한을 정확하게 다루고 있는가?	0.5	0.5
3	AF303:구조가 비-승인자의 침입을 감지할 능력을 가지고 있는가?	0.5	0.5

◇ 타이밍(timing) 검증에 관한 평가 값

타이밍: AF400_timing		yes	no
1	AF401:소프트웨어 구조 설계명세에서 스케줄링 및 프로세스간의 통신방식이 기술되어 있는가?	0.5	0.5
2	AF402:소프트웨어 구조 설계명세가 모든 타이밍 제한사항, 각 제한사항에 대한 전략, 요구되는 여유도 및 여유도 측정방법 등에 대해 기술하고 있는가?	0.5	0.5
3	AF403:각 구조 요소에 대한 최소 및 최대 수행시간이 명시되어 있는가?	0.5	0.5
4	AF404:소프트웨어 구조가 응답시간을 만족하는지 분석되는가?	0.5	0.5
5	AF405:운전이 정확한 순서로 수행되었음을 보장하도록 구조가 분석되었는가?	0.5	0.5

6	AF406:컴퓨터 자원의 스케쥴링이 예측 가능하고 결정적으로 수행되는지 보장하기 위한 분석을 수행하는가?	0.5	0.5
7	AF407:소프트웨어 구조 설계명세에 포함된 구조설계 요소에 대한 크기(Size)가 추정되는가?	0.5	0.5
8	AF408:저장장치 크기 추정치가 운전, 유지보수 및 잠정적 개선을 위한 충분한 여유를 가지고 있는가?	0.5	0.5

1.2 공정 특성 검증

◇ 완전성(completeness) 검증에 관한 평가 값

완전성: AP100_completeness		yes	no
1	AP101:소프트웨어 구조가 소프트웨어 설계명세, 안전성분석보고서, 요구사항명세서 등에 있는 모든 예상되는 상황과 조건을 고려하는가?	0.5	0.5
2	AP102:소프트웨어 구조설계 명세가 SRS에 명시된 설계 제한사항과 소프트웨어 요구사항을 만족하는 소프트웨어 구조를 묘사하고 있는가?	0.5	0.5
3	AP103:소프트웨어를 프로그램 단위들이 높은 내부 응집도를 갖도록 소프트웨어 구조가 설계되었는가?	0.5	0.5
4	AP104:소프트웨어를 프로그램 단위들 간에 낮은 연관도를 갖도록 소프트웨어 구조가 설계되었는가?	0.5	0.5
5	AP105:프로그램 요소들이 명료한 시스템 구조가 되도록 체계적으로 구성되어 있는가?	0.5	0.5
6	AP106:각 시스템 기능이 특정 소프트웨어 요소에서 수행되는가?	0.5	0.5
7	AP107:소프트웨어 구조가 소프트웨어 요소간의 논리적 연결을 보여 주는가?	0.5	0.5
8	AP108:소프트웨어 설계를 위해 일반적으로 인정된 소프트웨어공학 기법이 사용되었는가?	0.5	0.5
9	AP109:소프트웨어 구조가 소프트웨어 요소들 간의 데이터 흐름과 제어흐름을 보여 주는가?	0.5	0.5
10	AP110:운전환경이 구조설계에 반영되었는가?	0.5	0.5
11	AP111:소프트웨어 구조가 각 인터페이스의 요구되는 행동을 허용하는가?	0.5	0.5
12	AP112:소프트웨어 구조 설계가 SRS에 명시된 모든 운전모드를 고려하는가?	0.5	0.5
13	AP113:소프트웨어 계통의 설계와 실 세계 제한사항에 의해 부과된 어떠한 제약사항을 평가하기 위하여 제한분석이 수행되었는가?	0.5	0.5
14	AP114:데이터-관련 구조요소가 소프트웨어 요구사항과의 일치 여부를 결정하기 위한 자료분석이 수행되었는가?	0.5	0.5
15	AP115:소프트웨어 구성요소간의 인터페이스가 정확하게 설계되었는지를 결정하기 위한 인터페이스 분석이 수행되었는가?	0.5	0.5
16	AP116:특별한 어려움을 야기하는 모든 특정 구조 설계영역이 파악되고, 각각을 위한 완화 계획이 묘사되어 있는가?	0.5	0.5
17	AP117:소프트웨어 구조가 시스템과 개발노력에 부과된 제약사항 내에서 구현되었는가?	0.5	0.5
18	AP118:소프트웨어가 운전요원에 부담을 주는 것이 아니라 도움을 준다는 것을 보장할 수 있도록 소프트웨어 구조가 분석되고 검토되었는가?	0.5	0.5
19	AP119:모든 데이터 채널의 응답시간, 샘플링 빈도, 숫자, 크기, 데이터율(data rate) 등이 소프트웨어 구조설계 명세에 정의되어 있는가?	0.5	0.5

20	AP120:운전원이나 보수요원이 쉽게 디지털 시스템 상태를 평가하고 문제영역을 찾아낼 수 있도록 적절한 인간공학 고려사항들이 구조설계에 포함되어 있는가?	0.5	0.5
21	AP121:구조 설계에 하드웨어 환경이 고려되었는가?	0.5	0.5
22	AP122:구조 설계가 각 소프트웨어 요소가 동작할 하드웨어 요소를 명시하고 있는가?	0.5	0.5
23	AP123:특정 하드웨어가 요구된다면 소프트웨어 구조설계 명세가 저장장치 용량, 명령어 집합, 속도, 입출력 레지스터 등을 포함하여 그 컴퓨터 내부의 각 컴퓨터를 파악하는가?	0.5	0.5
24	AP124:설계구조 명세에 주 하드웨어 요소가 나타나는가?	0.5	0.5
25	AP125:소프트웨어 구조는 하드웨어 요소들간의 물리적 연결을 보여 주는가?	0.5	0.5
26	AP126:하드웨어 구조가 존재하는가?	0.5	0.5

◇ 일관성(consistency) 검증에 관한 평가 값

일관성: AP200_consistency		yes	no
1	AP201:소프트웨어 구조설계에서 사용된 하나 이상의 정형기법(formal method)은 이들 상호간의 일관성을 유지하고 있는가?	0.5	0.5
2	AP202:소프트웨어 구조설계 설명서(software architecture design description)의 표현 스타일과 상세화 수준이 일관성을 갖는가?	0.5	0.5
3	AP203:소프트웨어 구조도는 소프트웨어 구조설계 설명서 및 프로그램 작동환경과 일치하는가?	0.5	0.5
4	AP204:소프트웨어 구조도의 요소는 소프트웨어 요구사항 명세서와 일치하는가?	0.5	0.5
5	AP205:소프트웨어 구조도는 소프트웨어 구조도 요소 (component)간의 상관관계의 정확성 (correctness)을 입증할 수 있도록 분석되었는가?	0.5	0.5
6	AP206:소프트웨어 구조도는 하드웨어 구조와 일치하는가?	0.5	0.5

◇ 스타일(style) 검증에 관한 평가 값

스타일: AP300_style		yes	no
1	AP301:소프트웨어 구조설계 설명서(software architecture design description)는 각 구조설계 요소의 기능(function)을 서술하고 있는가?	0.5	0.5
2	AP302:소프트웨어 구조설계 설명서는 구조설계 요소들간의 관계성(relationship)을 기술하고 있는가?	0.5	0.5
3	AP303:소프트웨어 구조설계 문서상에 설계구조 선택의 타당성을 설명하고 있는가?	0.5	0.5
4	AP304:소프트웨어 설계요소 중 처리(processing) 부분을 포함하고 있다면 소프트웨어 구조설계 설명서에 처리기능의 요소를 실행하는 방법(method)을 기술하고 있는가?	0.5	0.5
5	AP305:소프트웨어 구조설계 설명서에 각 구조설계 요소의 기능수행을 위하여 필요한 자원(resources)을 규정하고 있는가?	0.5	0.5
6	AP306:소프트웨어 설계구조 요소의 상호작용 설명에는 타이밍 트리거 사건들, 실행의 순서, 자료공유, 상호작용에 영향을 미치는 다른 요소 등을 포함하고 있는가?	0.5	0.5

7	AP307:소프트웨어 구조설계 설명서에 각 설계요소의 유형(type)을 확인할 수 있는가?	0.5	0.5
8	AP308:소프트웨어 구조설계 설명서에 적용한 표준(standard)을 확인할 수 있는가	0.5	0.5
9	AP309:만약 소프트웨어 설계요소가 데이터저장(data store), 메시지 또는 스크린 디스플레이라면 소프트웨어 구조설계 설명서에는 이들에 대한 일반적인 구조를 설명하고 있는가?	0.5	0.5
10	AP310:소프트웨어 구조설계의 각 요소들은 유일한 명칭을 가지고 있는가?	0.5	0.5

◇ 추적성(traceability) 검증에 관한 평가 값

추적성: AP400_traceability		yes	no
1	AP401:소프트웨어 구조설계 요소가 소프트웨어 요구사항 명세서와 일치하는지 확인하는데 사용하는 기법인 특정 시험(specific tests) 또는 검증기준(validation criteria)을 이용하여 전향추적을 할 수 있는가?	0.5	0.5
2	AP402:소프트웨어 구조설계 설명서에는 소프트웨어 구조를 결정짓는데 적절히 활용될 수 있도록 소프트웨어 구조설계에 대한 참조 주목사항(notes)을 포함하고 있는가?	0.5	0.5
3	AP403:소프트웨어 구조설계 설명서에 소프트웨어 요구사항 명세서에 기술된 요건의 범위를 벗어나는 어떤 기능을 포함하고 있다는 정당한 근거가 있다면 요구사항의 범위를 벗어난 소프트웨어 구조설계 설명서에 대한 정당성 결과가 요구사항 명세서와 일치하는가?	0.5	0.5
4	AP404:소프트웨어 구조설계 설명서에는 소프트웨어 구조설계의 각 기능에 대해 확인할 수 있는 특정 식별자(ID)를 부여하고 있어서 구현 시 이들 식별자를 통해 추적할 수 있는가?	0.5	0.5
5	AP405:각 소프트웨어 구조설계 요소는 소프트웨어 요구사항 명세서의 특정 요소(specific elements)를 후향 추적(backward trace) 할 수 있도록 되어있는가?	0.5	0.5
6	AP406:소프트웨어 구조설계 상에서 소프트웨어 요구사항 명세서에 기입된 설계 제약조건 등을 점검할 수 있도록 되어 있는가?	0.5	0.5

◇ 확인가능성(verifiability) 검증에 관한 평가 값

확인가능성: AP500_verifiability		yes	no
1	AP501:소프트웨어 구조설계 사양은 각 소프트웨어 설계요소의 기능에 대해 시험할 수 있는 방법을 기술하고 있는가?	0.5	0.5
2	AP502:시스템 및 하부시스템 시험절차서, 시험 사례 들을 개발하기 위하여 소프트웨어 구조도 분석을 하였는가?	0.5	0.5

2. SDS 상세설계 검증

기능 특성/코드	공정 특성/코드
정확도(Accuracy)/SF1	완전성(completeness)/SP1
신뢰성(reliability)/SF2	일관성(consistency)/SP2
강인성(Robustness)/SF3	정확성(Correctness)/SP3
안전성(safety)/SF4	스타일(style)/SP4
보안성(security)/SF5	추적성(traceability)/SP5
타이밍(timing)/SF6	확인가능성(verifiability)/SP6

2.1 기능 특성 검증

◇ 정확도(Accuracy) 검증에 관한 평가 값

정확도: SF100_accuracy		yes	no
1	SF100_accuracy_1:소프트웨어 상세설계에서 사용한 모든 계산들은 계산에 대한 정확성 요구사항을 만족하는가?	0.9	0.1
3	SF100_accuracy_2:소프트웨어 상세 설계 시 입력변수 값을 검증하기 위한 방법을 기술하고 있는가?	0.9	0.1

◇ 신뢰성(reliability) 검증에 관한 평가 값

신뢰성: SF200_reliability		yes	no
1	SF200_reliability_1:소프트웨어 설정(configuration)에서 초기화 검사(initialization checks)가 있는가?	0.8	0.2
2	SF200_reliability_2:데이터 일관성(consistency)을 보증하기 위해 해야 할 것이 무엇인가?	??	
3	SF200_reliability_3:교착상태(deadlock)에 대한 잠재성이 존재하는지, 그리고 그와 같은 잠재성이 있다면 무슨 단계들이 교착상태를 방지하기 위해 취해져야 할 것인지를 결정하기 위해 분석이 수행되고 문서화되어야 하는가?	0.9	0.1
4	SF200_reliability_4:시간 종료검사(time-out checks-software watch-dogs)가 소프트웨어에 적용되고 있는가?	0.9	0.1
5	SF200_reliability_5:공통-모드 소프트웨어 고장(common-mode software failures)이 소프트웨어 상세 설계에서 분석되고 고려되는가?	0.4	0.6
6	SF200_reliability_6:고장 이벤트에서 시스템 상태를 기록하는 설비를 소프트웨어 상세설계에서 포함하고 있는가?	0.1	0.9
7	SF200_reliability_7:오류탐지 이벤트에서 취해지는 조치사항(actions)을 소프트웨어 상세설계에서 확인하고 있는가?	0.9	0.1
8	SF200_reliability_8:프로그램을 수행하고 있는 동안 고장 파급(failure propagation)을 방지하기 위한 기법들을 사용하고 있는가?	0.5	0.5
9	SF200_reliability_9:고장이 탐지되는 이벤트에서 정의가 잘 된 출력의 생성능력(capacity)을 소프트웨어 상세설계에서 포함하고 있는가?	0.7	0.3
10	SF200_reliability_10:완전하고 정확한 오류 회복 기법(error recovery techniques)을 소프트웨어 상세설계에서 명세하고 있는가?	0.3	0.7
11	SF200_reliability_11:소프트웨어 상태를 표시하는(displaying) 설비가 있는가?	0.9	0.1

12	SF200_reliability_12:신뢰도 및 가용도 분석이 신뢰도 요구사항을 만족하는지 보증하는 충분하게 지원 가능한 데이터(supportable data)를 포함하고 있는가?	0.1	0.9
13	SF200_reliability_13:적절한 운전원 조작 신호가 발생하는지를 소프트웨어 상세설계에서 포함하고 있는가?	0.8	0.2
14	SF200_reliability_14:소프트웨어 상세설계가 모든 운전원 입력에 대한 검증 검사를 포함하고 있는가?	0.6	0.4
15	SF200_reliability_15:통신 라인(communication lines)이 동작하고 있는지를 검사하고 있는가?	0.8	0.2
16	SF200_reliability_16:입출력 통신에 대한 무결성 검사(integrity checks)가 수행되고 있는가?	0.4	0.6
17	SF200_reliability_17:통신 메시지가 검증되는가?	0.7	0.3
18	SF200_reliability_18:절단 또는 쇼트(broken or shorted)된 입출력 통신 라인에 대한 검사가 이루어지고 있는가?	0.1	0.9
19	SF200_reliability_19:누락(missing) 그리고/또는 지연 메시지(late messages)에 대한 검사가 이루어지고 있는가?	0.1	0.9
20	SF200_reliability_20:하드웨어나 소프트웨어 결함에 기인하는 손상(damaged)이나 손실 데이터(lost data)를 소프트웨어 상세 설계에서 탐지하고 있는가?	0.1	0.9
21	SF200_reliability_21:소프트웨어나 하드웨어 결함을 다루기 위한 표준 전략이 있는가?	0.6	0.4
22	SF200_reliability_22:탐지된 모든 하드웨어와 소프트웨어 고장들이 보고되고 있는가?	0.3	0.7
23	SF200_reliability_23:충분한 기억장치 그리고/또는 통신시험 적용 범위(coverage)가 있는가?	0.1	0.9
24	SF200_reliability_24:하드웨어 결함을 탐지하기 위한 주기적 소프트웨어 검사가 이 하드웨어에서 이루어지고 있는가?	0.6	0.4

◇ 강인성(robustness) 검증에 관한 평가 값

강인성: AF300_robustness		yes	no
1	AF300_robustness_1:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 입력의 발생에도 정확하게 동작할 설계인가?	0.5	0.5
2	AF300_robustness_2:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 운전원 행위(operator behavior)의 발생에도 정확하게 동작할 설계인가?	0.9	0.1
3	AF300_robustness_3:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 하드웨어 행위(hardware behavior)의 발생에도 정확하게 동작할 설계인가?	0.9	0.1
4	AF300_robustness_4:소프트웨어가 예기치 않은, 정확하지 않은, 비정상적, 그리고 적절치 않은 소프트웨어 행위(software behavior)의 발생에도 정확하게 동작할 설계인가?	0.7	0.3

◇ 안전성(safety) 검증에 관한 평가 값

안전성: SF400_safety		yes	no
1	SF400_safety_1:소프트웨어 상세설계의 제어논리(control logic)가	0.9	0.1

	안전성-필수 요건(safety-critical requirements))을 완전하고 정확하게 구현하는가?		
2	SF400_safety_2:확인된 각 안전성-필수 데이터를 변경할 수 있는 상세설계 요소(detailed design elements)를 갖는가?	0.9	0.1
3	SF400_safety_3:소프트웨어 상세설계가 안전성이 다른 수준의 계통이나 하부계통들 사이의 상호작용(interactions)에 대하여 엄격한 제어를 제공하는가?	0.9	0.1
4	SF400_safety_4:확인된 안전성-필수 데이터가 형(types), 단위(units), 범위(ranges), 그리고 오류한계(error bounds)와 함께 있는가?	0.8	0.2
5	SF400_safety_5:소프트웨어 상세설계의 공식(equations)과 알고리즘(algorithms)이 안전성-필수 요건을 정확하게 구현하는가?	0.7	0.3
6	SF400_safety_6:안전성에 영향을 주는 특정 시간, 비용, 복잡도, 또는 다른 특정 문제점이 있다면, 완화 계획(mitigation plans)을 통해 문서화되고 제기되었는가?	0.9	0.1
7	SF400_safety_7:소프트웨어 상세설계가 자원-필수 단위나 하부계통(subsystem)을 확인하기 위해 분석되었는가?	0.6	0.4
8	SF400_safety_8:소프트웨어 상세설계가 소프트웨어가 안전하고 알려진 상태로 복귀하도록 하는 비정상 탐지검사(anomaly detection checks)를 포함하는가?	0.6	0.4
9	SF400_safety_9:안전성-관련 진단메시지가 다른 사용자들의 지식과 능력에 적합한 방식으로 정의되었는가?	0.5	0.5
10	SF400_safety_10:소프트웨어 상세설계가 그 환경에서 어떻게 소프트웨어 시스템이 낮은 확률사건이나 고장에 반응하는지를 결정하기 위해 분석되었는가?	0.2	0.8
11	SF400_safety_11:모든 비-안전성 단위가 소프트웨어 안전성에 악 영향을 주지 않는다는 적절한 확신을 제공하기 위해 분석되었는가?	0.2	0.8
12	SF400_safety_12:상세설계 요소들 사이의 인터페이스가 적절히 설계되고 안전위험(safety hazard)을 생성하지 않음을 결정하기 위해 수행되고 문서화된 설계 인터페이스 분석이 있었는가?	0.3	0.7
13	SF400_safety_13:데이터-관련 상세설계 요소들의 소프트웨어 요건과 일관되고 계통 안전성 요건을 위배하지 않는지를 결정하기 위해 수행되고 문서화된 설계데이터 분석이 있었는가?	0.4	0.6
14	SF400_safety_14:상세설계에서 설계 한계분석(design constraint analysis)이 수행되고 문서화 되었나?	0.2	0.8
15	SF400_safety_15:모든 안전성-필수 소프트웨어의 소프트웨어 상세설계에서 소프트웨어 안전성 분석이 수행되고 문서화 되었나?	0.1	0.9
16	SF400_safety_16:모든 안전성-필수 소프트웨어의 소프트웨어 상세설계에서 소프트웨어 안전성 분석이 수행되고 문서화 되었나?	0.1	0.9
17	SF400_safety_17:소프트웨어에 할당된 모든 안전성 요건에 따른 소프트웨어 상세설계임을 확증하기 위해 분석되었는가?	0.1	0.9

◇ 보안성(security) 검증에 관한 평가 값

보안성: SF500_security		yes	no
1	SF500_security_1:상세설계가 소프트웨어 요구명세에서 확인된 모든 보호위협(security threats)을 정확하게 제기하는가?	0.2	0.8

2	SF500_security_2:상세설계가 권한이 없는 사람(unauthorized personnel)에 의한 공격(intrusions)을 탐지하기 위한 능력을 포함하는가?	0.1	0.9
3	SF500_security_3:상세설계가 컴퓨터 바이러스를 탐지하는 능력을 포함하는가?	0.1	0.9
4	SF500_security_4:정확하게 설계된 소프트웨어 요구명세에 접근제한(access restrictions)이 명세되는가?	0.1	0.9
5	SF500_security_5:발전소 운전원에 의한 변경에 대하여 계통데이터나 코드가 안전하게 보호(secure) 되는가?	0.6	0.4

◇ 타이밍(timing) 검증에 관한 평가 값

타이밍: SF600_timing		yes	no
1	SF600_timing_1:소프트웨어 상세설계 명세가 모든 타이밍 자원 제약, 각각을 처리하는(handling) 전략, 요구된 여유도(margins), 그리고 이러한 여유도를 측정하는 방법의 명세를 포함하는가?	0.5	0.5
2	SF600_timing_2:상세 설계가 실행 타이밍이 결정적임을 보증하는가?	0.8	0.2
3	SF600_timing_3:인터럽트가 정확하게 처리되고 입출력 타이밍이 정확하며, 그리고 단위들이 정확하게 연계되는지 확증하기 위해 중요한 인터페이스(potential interface)와 타이밍 문제가 분석되었는가?	0.4	0.6
4	SF600_timing_4:소프트웨어 상세설계에서 수행된 타이밍분석이 클럭(clock), 표본율(sampling rate), 그리고 센서 반응시간의 변화(variations)를 계산하기 위해 절절한 여유도를 허용하는가?	0.8	0.2
5	SF600_timing_5:컴퓨터 자원의 스캐쥴링이 동적이라기 보다는 예측적이고 결정적임을 확증하기 위해 분석이 수행되었는가?	0.1	0.9
6	SF600_timing_6:자체-검사 활동(self-checking actions)이 타이밍 요건을 위배하지 않음을 확증하기 위해 분석이 수행되었는가?	0.5	0.5
7	SF600_timing_7:요구된 계통 반응 시간이 만족된다는 것을 상세설계가 보증하는가?	0.3	0.7
8	SF600_timing_8:소프트웨어 상세설계 명세는 사용된 스캐쥴링 구조(scheduling mechanisms)와 IPC (inter-process communication mechanisms)를 명세하는가?	0.8	0.2
9	SF600_timing_9:각 단위에 대한 타이밍이 실행(execution)을 위한 최대 및 최소 시간으로 명세되었는가?	0.5	0.5
10	SF600_timing_10:다른 프로그램과 동기된 다른 순차적인 프로그램으로부터 데이터를 받거나 또는 그 프로그램으로 데이터를 전송하는 순차적인 프로그램의 실행이 있는가?	0.7	0.3
11	SF600_timing_11:연산(operations)이 실행 속도와 독립적인 정확한 순서로 수행되도록 설계된 소프트웨어가 있는가?	0.6	0.4
12	SF600_timing_12:프로그램을 실행하기 위해 주어진 시간이나 프로그램의 실행이 초기화되는 시간 중에서 독립적인 순차적인 프로그램에 의해 생성된 결과들이 있는가?	0.8	0.2
13	SF600_timing_13:인터럽트가 금지될 수 있는 시간(times)에 하드 상한(hard upper bounds) 범위 내에 존재하는가?	0.1	0.9
14	SF600_timing_14:소프트웨어 상세설계 명세가 모든 기억장치 자원 제약, 각각을 처리하는 전략, 요구된 여유도, 그리고 이러한 여유도를 측정하는 방법의 명세를 포함하는가?	0.4	0.6
15	SF600_timing_15:중앙처리장치(CPU), 기억장치, 그리고 입출력 채널을 위한 적절한 예산(budgets)이 확립되었나?	0.1	0.9

2.2 공정 특성 검증

◇ 완전성(Completeness) 검증에 관한 평가 값

완전성: SP100_completeness		yes	no
1	SP100_completeness_1:각 단위 모듈이 오직 하나의 진입점을 가지는가?	0.8	0.2
2	SP100_completeness_2:각 단위 모듈이 오직 하나의 출구점을 가지는가?	0.8	0.2
3	SP100_completeness_3:명시된 모델, 알고리즘, 수치기법, 신호변환 및 자료처리 절차가 실용적으로 사용되는 범주 내의 것들인가?	0.8	0.2
4	SP100_completeness_4:소프트웨어 상세설계가 단위들 간의 규칙적이고 문서화된 연계를 사용하는가?	0.8	0.2
5	SP100_completeness_5:모든 내부 및 외부연계가 완전하게 정의되었는가?	0.5	0.5
6	SP100_completeness_6:상세설계가 연계요소들에 대한 문서화된 기술이나 알려진 특성과 일관성이 있는가?	0.7	0.3
7	SP100_completeness_7:발전소 운전원을 지원하도록 설계되었는가를 확증하기 위해 소프트웨어 상세설계가 분석되고 검토되었는가?	0.1	0.9
8	SP100_completeness_8:컴퓨터 시스템-인간 상호작용의 문법이 정의되는가?	0.1	0.9
9	SP100_completeness_9:센서 및 구동기 설계연계가 신호의 동적범위, 정확성, 타이밍, 인터럽트 및 분해능을 포함하는가?	0.7	0.3
10	SP100_completeness_10:소프트웨어 상세설계가 시스템 및 개발 노력에 주어지는 제약조건들 안에서 구현될 수 있는가?	0.6	0.4
11	SP100_completeness_11:모든 자료채널에 대한 개수, 크기, 자료속도, 샘플링 빈도 및 응답시간이 소프트웨어 상세설계 명세서에 정의되는가?	0.9	0.1
12	SP100_completeness_12:소프트웨어 상세설계가 단위 크기를 제한하는 요건을 명시하는가?	0.5	0.5
13	SP100_completeness_13:상세설계에 적절한 인간공학 요소들이 고려되어 운전원이나 유지보수자가 시스템의 상태를 평가하거나 문제가 있는 부분을 찾아내는데 용이하도록 하고 있는가?	0.1	0.9
14	SP100_completeness_14:소프트웨어 상세설계 명세서가 소프트웨어 요구사항 명세와 소프트웨어 구조설계 명세서에 기술된 시스템 요구사항 및 모든 제약조건들을 만족하는 소프트웨어 상세 설계를 기술하는가?	0.8	0.2
15	SP100_completeness_15:컴퓨터 시스템과-인간 상호작용에 대한 정형적 절차가 소프트웨어 설계명세서에 명시되는가?	0.1	0.9
16	SP100_completeness_16:특별한 어려움을 초래하는 특정 상세설계 분야(예를 들어, 공정 및 속도, 새로운 알고리즘, 보안, 새로운 하드웨어 및 언어)가 명시되고 그에 따른 완화책이 기술되는가?	0.7	0.3
17	SP100_completeness_17:실세계의 제한사항이나 소프트웨어 구조설계에 따른 상세설계의 제약사항들을 평가하기 위한 설계 제약사항 분석이 수행되었는가?	0.2	0.8
18	SP100_completeness_18:자료-관련 상세설계 요소들이 소프트웨어 요구사항 및 구조와 일관성을 유지하는가를 결정하기 위한 설계자료 분석이 수행되었는가?	0.1	0.9
19	SP100_completeness_19:소프트웨어 상세설계 공식, 알고리즘 및 제어논리가 소프트웨어 요구사항들을 정확하게 구현하는가를 결정하기 위한 설계논리 분석이 수행되었는가?	0.4	0.6
20	SP100_completeness_20:소프트웨어 상세설계가 각 연계사항에 대하여 요구되는 프로그램의 행위를 구현하고 있는가?	0.7	0.3
21	SP100_completeness_21:소프트웨어 코드를 구현하는데 필요한 모든 프로그램 입력, 출력, 및 자료 요소들이 적절한 범위로 명시되고 기술되어 있는가?	0.9	0.1
22	SP100_completeness_22:소프트웨어 요구사항명세서의 각 요구사항들이 상세설계 명세서로 반영되어 있는가?	0.9	0.1
23	SP100_completeness_23:소프트웨어 상세설계에 하드웨어 환경(전기 접점, 전원 및 필터, 예비 전원, 프로세서의 선택, 입출력장치 속도 등)이 고려되었는가?	0.5	0.5
24	SP100_completeness_24:소프트웨어 상세설계에 운전환경이	0.6	0.4

	고려되었는가?		
25	SP100_completeness_25:소프트웨어 상세설계가 소프트웨어 요구사항명세서에 명시된 모든 운전모드를 고려하는가?	0.8	0.2
26	SP100_completeness_26:상세설계 요소들 사이의 연계가 정확하게 설계되었는가를 결정하기 위한 설계 연계분석이 수행되었는가?	0.4	0.6

◇ 일관성(Consistency) 검증에 관한 평가 값

일관성: SP200_consistency		yes	no
1	SP200_consistency_1:표현 스타일이나 상세 정도가 소프트웨어 상세설계명세서 전반에 걸쳐 일관성이 있는가?	0.8	0.2
2	SP200_consistency_2:소프트웨어 상세설계가 하드웨어 및 소프트웨어 구조와 일관성이 있는가?	0.9	0.1
3	SP200_consistency_3:주변장치 접점들에 대한 표준연계가 있는가?	0.6	0.4
4	SP200_consistency_4:유사하거나 관련 있는 기능들에 대한 상세설계가 일관성이 있는가?	0.8	0.2
5	SP200_consistency_5:인간-기계 연계에 대한 표준연계가 있는가?	0.1	0.9
6	SP200_consistency_6:자료 전송에 대한 표준연계가 있는가?	0.8	0.2
7	SP200_consistency_7:설계에 명시된 모델, 알고리즘 및 수치기법이 수학적으로 상호 호환되는가?	0.8	0.2
8	SP200_consistency_8:설계에 명시된 모델, 알고리즘 및 수치기법이 적절한 표준 참고문서의 것과 일치하는가?. 표준 참고문서가 존재한다면 그것들은 모델, 알고리즘, 수치기법을 설계하는 사람들이 제공해야 한다.	0.4	0.6
9	SP200_consistency_9:각 소프트웨어 상세 설계요소가 프로그램이 작동하는 운전환경의 설명 및 특성들과 일관성이 있는가?	0.9	0.1
10	SP200_consistency_10:소프트웨어 상세설계가 소프트웨어 요구사항명세와 일관성이 있는가?	0.5	0.5
11	SP200_consistency_11:복수개의 정형적 상세설계 방법이 사용된다면 그들이 서로 일관성이 있는가?	0.6	0.4
12	SP200_consistency_12:소프트웨어 상세설계에 주어진 입력 및 출력명세(자료형태, 자료 크기, 자료 속도, 정확성, 오류범위 및 물리적 단위 등 포함)가 하드웨어 및 기성 소프트웨어에 의해 부여된 연계 요구사항과 일관성이 있는가?	0.6	0.4

◇ 정확성(Correctness) 검증에 관한 평가 값

정확성: SP300_correctness		yes	no
1	SP300_correctness_1:모든 공식 및 알고리즘이 코딩이 시작될 수 있을 정도로 정확하게 또 충분히 상세한 수준까지 정의되어 있는가?	0.7	0.3
2	SP300_correctness_2:잠재적인 언더플로(underflow) 및 오버플로(overflow) 상태를 식별하기 위한 알고리즘 정확성 분석이 수행되었는가?	0.8	0.2
3	SP300_correctness_3:서브루틴이나 프로시저를 빠져 나왔을 때 상태가 점검되고, 오류 상태가 지시되었을 때 적절한 행위가 수행되는가?	0.5	0.5
4	SP300_correctness_4:잠재된 자료처리 문제(부정확한 자료 초기화, 저장 자료의 부정확한 평가, 부정확한 자료 측정 등)를 평가하기 위한 분석이 수행되었는가?	0.2	0.8

5	SP300_correctness_5:격리, 분할, 자료 별칭(data aliasing) 및 결함 억제 현안들을 둘러싼 자료 의존성에 대한 자료 구조를 평가하기 위한 분석이 수행되었는가?	0.1	0.9
6	SP300_correctness_6:상태변수의 병행성(concurrency) 및 일관성을 유지할 장치들이 소프트웨어 상세설계에 나타나는가?	0.6	0.4
7	SP300_correctness_7:알고리듬이 입력 및 시간변수의 전체 범위에 대하여 안정적인가를 확증하기 위한 알고리즘 분석이 수행되었는가?	0.2	0.8
8	SP300_correctness_8:각 단위의 입력에 대한 유효성(validity)이 점검되었는가?	0.9	0.1
9	SP300_correctness_9:인터럽트가 예상치 못한 방식으로 안전하고 중요한 자료값을 변경하지 않는다는 확실한 증거가 있는가?	0.1	0.9
10	SP300_correctness_10:안전에 중요한 자료 값이 예상치 못한 방식 혹은 예상치 못한 상세설계 요소에 의해 변경되지 않는다는 확실한 증거가 있는가?	0.8	0.2
11	SP300_correctness_11:공식, 알고리즘 및 제어논리가 잠재된 문제점들(논리오류, 누락된 경우나 단계, 중복논리, 극한조건의 무시, 불필요한 기능, 오해, 조건시험 누락, 잘못된 변수에 대한 점검 및 루프의 반복 등)에 대하여 평가되는가?	0.5	0.5
12	SP300_correctness_12:안전에 중요한 자료가 초기화되기 전에 사용되지 않는다는 확실한 증거가 있는가?	0.9	0.1
13	SP300_correctness_13:부동소수점 연산 및 재귀(recursion)가 사용되는 경우, 그 사용에 대한 적절한 정당화가 이루어져 있는가?	0.1	0.9

◇ 스타일(Style) 검증에 관한 평가 값

스타일: SP400_style		yes	no
1	SP400_style_1:소프트웨어 설계명세서(SDS)는 감시계통 성능(supervisory system performance) 및 신뢰도(reliability)에 대한 정량적인 데이터들에 대해 문서화하고 있는가?	0.1	0.9
2	SP400_style_2:모든 감시기능들에 대해 확고한 정의가 있고 연계사항(interfaces)은 체계를 잘 갖추고 있는가?	0.1	0.9
3	SP400_style_3:전역변수(global variables)의 사용이 금지되거나 또는 세밀하게 입증되도록 되어 있는가?	0.3	0.7
4	SP400_style_4:소프트웨어 상세 설계명세는 타임 슬라이싱(time-slicing)의 사용을 제한하고 있는가?	0.1	0.9
5	SP400_style_5:소프트웨어 상세 설계명세는 메모리 스와핑(memory swapping) 사용을 제한하고 있는가?	0.1	0.9
6	SP400_style_6:필요한 모든 지원 소프트웨어들에 대해서 충분히 기술하고 있는가?	0.8	0.2
7	SP400_style_7:소프트웨어 상세 설계명세에서 프로그래밍언어 변수들에 대해서 명료한 정의 및 형(typing)이 있는가?	0.6	0.4
8	SP400_style_8:소프트웨어 상세설계에서는 안전성-필수 기능(safety-critical functions)들의 수행시간 동안 소프트웨어 인터럽트(interrupts) 취급을 제한하도록 되어 있는가?	0.1	0.9
9	SP400_style_9:사용될 구축 언어들은 소프트웨어 상세설계 사양에서 식별되도록 되어 있는가?	0.8	0.2
10	SP400_style_10:소프트웨어 상세설계에서 특별하게 개발된 장치 드라이버(device drivers)를 사용하는데 대해서 논리적 근거를 제공하고 있는가?	0.1	0.9

11	SP400_style_11:어떤 공유메모리 영역(shared memory locations)이 일치가 안된 상태에서 동시에 변경되지 않도록 상세설계에서 보장하고 있는가?	0.4	0.6
12	SP400_style_12:인터럽트를 피하는 최대 가능한 루프 수를 소프트웨어 상세설계에서 기술하고 있는가?	0.1	0.9
13	SP400_style_13:데이터 저장공간(store), 메시지 또는 스크린 디스플레이(screen display)가 소프트웨어 상세 설계요소일 때 소프트웨어 상세설계 명세에서 이 요소의 구조에 대해서 기술하고 있는가?	0.3	0.7
14	SP400_style_14:소프트웨어 상세 설계명세는 간단한 폼에 대한 (즉, 순서(sequence), 사례 선택, 반복, 추상화(abstractions) (예를 들어 절차, 기능 그리고/또는 서브루틴)) 제어구조를 제한하고 있는가?	0.8	0.2
15	SP400_style_15:소프트웨어 상세설계 요소가 어떤 절차를 수행하면서 요소가 기능을 수행하는 방법에 대해서 소프트웨어 상세 설계명세에서 기술하고 있는가?	0.8	0.2
16	SP400_style_16:소프트웨어 상세설계 명세에서 소프트웨어 상세설계의 일환으로 기술되는 구축 제약사항(implementation constraints)들에 대해서 모두 열거하고 있는가?	0.5	0.5
17	SP400_style_17:소프트웨어 상세설계 명세에서 상세설계 의사결정에 대한 논리적 근거에 대해서 문서화하고 있는가?	0.4	0.6
18	SP400_style_18:소프트웨어 상세설계 명세에서 필요로 하는 프로그래밍 언어에 대한 표준들에 대해서 식별하고 있는가?	0.5	0.5
19	SP400_style_19:각 상세설계 요소는 특정한 이름을 갖도록 하고 있는가?	0.8	0.2
20	SP400_style_20:소프트웨어 상세설계 명세는 각 상세설계 요소의 형태(예를 들면, 체계, 하부체계, 유니트, 데이터베이스, 파일, 데이터 구조, 스크린 디스플레이, 메시지, 프로그램 또는 절차)에 대해서 식별하고 있는가?	0.8	0.2
21	SP400_style_21:소프트웨어 상세설계 명세는 각 상세설계 요소의 기능(즉, 무엇을 하는가)에 대해서 기술하고 있는가?	0.6	0.4
22	SP400_style_22:소프트웨어 상세설계 명세는 상세설계 요소들 간의 관계에 대해서 기술하고 있는가?	0.2	0.4
23	SP400_style_23:상세설계 요소간의 상호작용 기술은 타이밍, 사건유발(triggering events), 수행순서, 데이터 공유, 그리고 상호작용에 영향을 미치는 다른 작용요인을 포함하고 있는가?	0.7	0.3
24	SP400_style_24:소프트웨어 상세설계 명세에서 각 상세설계 요소가 그 기능을 수행하기 위한 자원들을 명기하고 있는가?	0.6	0.4
25	SP400_style_25:소프트웨어 상세설계 명세는 적용 가능한 표준에 준수해서 작성되도록 되어 있는가?	0.8	0.2

◇ 추적성(Traceability) 검증에 관한 평가 값

추적성: SP500_traceability		yes	no
1	SP500_traceability_1:각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?	0.8	0.2
2	SP500_traceability_2:각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?	0.6	0.4
3	SP500_traceability_3:각 상세설계 요소에 대해서 특정 코드 요소로의	0.8	0.2

	전방향 추적기록이 가능한가?		
4	SP500_traceability_4:요구사항에서 열거된 상세설계 제약사항들이 상세설계 내에서 따르도록 되어 있는가?	0.8	0.2
5	SP500_traceability_5:소프트웨어 상세설계 명세는 소프트웨어 상세설계 내의 각 기능을 특정하게 식별하도록 되어 있어 구축 시 특정하게 참조 될 수 있도록 되어 있는가?	0.8	0.2
6	SP500_traceability_6:SRS에서 기술된 요구사항 범위가 아닌 기능들을 포함하는데 대해 정당성이 입증되어 있는가? 그리고 결과적인 설계 상세내용이 SRS에서 의도한 바와 일치하는가?	0.4	0.6
7	SP500_traceability_7:소프트웨어 상세설계 사양이 소프트웨어 상세설계와 관련된 상세설계 의사결정을 문서화한 설계노트(design notes)를 참조하도록 되어 있는가?	0.1	0.9

◇ 확인성(verifiability) 검증에 관한 평가 값

확인성: AP600_verifiability		yes	no
1	AP600_verifiability_1:소프트웨어 상세설계 명세에서는 시험 가능한 방법으로 각 소프트웨어 요소의 기능들을 기술하고 있는가?	0.4	0.6
2	AP600_verifiability_2:소프트웨어 상세설계 분석에서 컴포넌트 시험 절차 및 시험 사례들을 발하는데 사용되어 왔는가?	0.4	0.6

3. SDS 상세 검증

3.1 설계 추적성 상세 검증

추적성: DV100_traceability		yes	no
1	DV100_traceability_1:소프트웨어 구조설계 설명서에는 소프트웨어 구조설계의 각 기능을 유일하게 확인할 수 있는 식별자(ID)를 가지고 있어서 구현(implementation)시 이들 식별자를 가지고 추적할 수 있는가?	0.5	0.5
2	DV100_traceability_2:소프트웨어 상세설계 명세는 소프트웨어 상세설계 내의 각 기능을 특정하게 식별하도록 되어 있어 구축 시 특정하게 참조 될 수 있도록 되어 있는가?	0.4	0.6
3	DV100_traceability_3:소프트웨어 구조설계 요소가 소프트웨어 요구사항 명세서와 일치하는지 확인하는데 사용하는 기법인 특별시험(specific tests) 또는 검증기준(validation criteria)을 이용하여 전향추적을 할 수 있는가?	0.2	0.8
4	DV100_traceability_4:각 소프트웨어 구조설계 요소는 소프트웨어 요구사항 명세서의 특별한 요소(specific elements)를 후향 추적(backward trace) 할 수 있는가?	0.4	0.6
5	DV100_traceability_5:각 상세설계 요소는 특정시험 또는 상세설계 요소가 만족되었는가에 대한 요구사항을 확증하는데 이용될 검증 기준으로 전방향 추적기록이 가능한가?	0.8	0.2
6	DV100_traceability_6:각 상세설계 요소가 소프트웨어 요구사항 명세(Software Requirements Specification)내 특정 요소들로 역방향 추적 기록이 가능한가?	0.8	0.2
7	DV100_traceability_7:소프트웨어 구조설계 설명서에 소프트웨어 요구사항 명세서에 기술된 요건의 범위를 벗어나는 어떤 기능을 포함하고 있다는 정당한 근거가 있다면 요건의 범위를 벗어난 소프트웨어 구조설계 설명서의 정당성 결과가 요구사항명세서와 일치하는가?	0.1	0.9

3.2 설계 정확성 상세 검증

◇ 소프트웨어 설계 정의의 정확성

정확성: DV210_Def_correctness		yes	no
1	DV210_Def_correctness_1:각 운전모드에서 실행되어야 하는 소프트웨어의 기능 및 행위가 정확하게 기술되어 있는가?	0.6	0.4
2	DV210_Def_correctness_2:알고리즘 (논리 및 공식)이 정확한가?	0.5	0.5
3	DV210_Def_correctness_3:각 기능들이 결정론적인가? 각 기능들을 비결정론적으로 만들 수 있는 조건들 (오류 복구 등)이 정의되어 있는가?	0.7	0.3
4	DV210_Def_correctness_4:기능이 단일고장요건을 만족하는가?	0.7	0.3
5	DV210_Def_correctness_5:안전계통 소프트웨어에 대한 임의의 변경 및 제어를 차단하는 수단이 있는가?	0.7	0.3

◇ 소프트웨어 입/출력 정의의 정확성 검증에 관한 평가 값

정확성: DV220_IO_correctness		yes	no

1	DV220_IO_correctness_1:각 기능들이 그 기능에 요구되는 입력 및 출력을 정확하게 명시하고 있는가?	0.6	0.4
2	DV220_IO_correctness_2:FBD 및 상세모듈에 모든 입력 및 출력의 출발점이 맞게 기술되어 있는가?	0.5	0.6

◇ 소프트웨어 행위 명세의 정확성

정확성: DV230_Act_correctness		yes	no
1	DV230_Act_correctness_1:각 기능이 어떻게 시작되는가 (초기화 조건)를 정확하게 기술하고 있는가?	0.7	0.3
2	DV230_Act_correctness_2:각 기능적 요구사항이 그 기능을 수행하는데 요구되는 작업 순서, 행위 및 사건들을 정확하게 명시하고 있는가?	0.8	0.2
3	DV230_Act_correctness_3:각 기능적 요구사항이 종결 조건이나 기능의 종료 시 시스템의 상태에 대하여 정확하게 명시하고 있는가?	0.6	0.4
4	DV230_Act_correctness_4:Function들 사이에 순환적 의존관계가 존재하지 않는가?	0.8	0.2
5	DV230_Act_correctness_5:각 기능의 정상적인 완료를 방해하는 조건이나 입력이 없는가?	0.8	0.2
6	DV230_Act_correctness_6:각 기능이 임의의 설계기준사고시에도 요구되는 기능을 완전히 수행할 수 있는가?	0.8	0.2
7	DV230_Act_correctness_7:비안전계통의 임의의 단일고장이 안전계통의 일부분에 영향을 주더라도 남은 안전계통 부분들이 적절한 안전기능을 수행할 수 있는가?	0.8	0.2

◇ 소프트웨어 인터페이스 기술의 정확성

정확성: DV240_IF_correctness		yes	no
1	DV240_IF_correctness_1:타 채널 및 프로세서와의 신호 연계가 안전기능을 방해하지 않는가?	0.9	0.1
2	DV240_IF_correctness_2:비안전계통과의 연계가 안전계통에 요구되는 기능을 방해하지 않는가?	0.9	0.1

3.3 설계 일관성 상세 검증

◇ 소프트웨어 설계 정의의 일관성 검증에 관한 평가 값

설계정의일관성: DV310_Def_consistency		yes	no
1	DV310_Def_consistency_1:각각의 상세설계가 소프트웨어의 요구사항들과 일관되는가?	0.7	0.3
2	DV310_Def_consistency_2:한 기능에 대한 명세 및 비슷한 기능들에 대한 명세가 서로 일관성이 있는가?	0.8	0.2
3	DV310_Def_consistency_3:내부적 일관성 측면에서, Block Diagram으로 표현된 기능과 알고리즘으로 표현된 기능 사이에 일관성이 있는가?	0.7	0.3
4	DV310_Def_consistency_4:정의된 용어 (또는 변수)를 사용하는 기능이 그것들의 정의와 모순되지 않는가?	0.6	0.4

◇ 소프트웨어 입출력 설계의 일관성 검증에 관한 평가 값

입출력일관성: DV320_IO_consistency		yes	no
1	DV320_IO_consistency_1:입력, 계산 및 출력 자료들에 요구되는 정확성이 상호 호환성이 있는가?	0.8	0.2
2	DV320_IO_consistency_2:입력 및 출력의 명세가 요구사항에 나오는 설명에 맞게 기술되어 있는가?	0.8	0.2

◇ 소프트웨어 설계명세의 일관성 검증에 관한 평가 값

설계명세일관성: DV330_Spec_consistency		yes	no
1	DV330_Spec_consistency_1:SDS에 명시된 모델, 알고리즘, 및 계산식들이 적용 가능한 표준 및 참고문헌과 일치성이 있는가?	0.8	0.2

◇ 소프트웨어 인터페이스 설계기술의 일관성 검증에 관한 평가 값

추적성: DV340_IF_consistency		yes	no
1	DV340_IF_consistency_1:SDS에 있는 입력 및 출력이 하드웨어 및 기성 소프트웨어 등에 의해 부여되는 연계 요구사항 들과 일관되는가?	0.8	0.2

3.4. 소프트웨어 설계 완전성 검증

◇ 소프트웨어 설계(기능?) 정의의 완전성 검증에 관한 평가 값

완전성: DV410_Def_completeness		yes	no
1	DV410_Def_completeness_1:소프트웨어가 수행해야 하는 모든 운전모드가 기술되어 있으며, 각 운전모드에서 실행되어야 하는 소프트웨어의 기능 및 행위가 모두 기술되어 있는가?	0.4	0.6
2	DV410_Def_completeness_2:각 운전모드 간의 전환을 일으키는 조건들이 명시되어 있는가?	0.7	0.3
3	DV410_Def_completeness_3:SRS에 기술된 모든 기능적 요구사항들이 SDS에 함수로 기술되었는가?	0.9	0.1
4	DV410_Def_completeness_4:각각의 입력이 최소한 하나의 FBD에 의해 사용되었는가?	0.8	0.2
5	DV410_Def_completeness_5:각 출력을 정의하는 FBD가 유일하게 있는가?	0.8	0.2
6	DV410_Def_completeness_6:Function의 performance criteria (error performance 포함) 가 모두 정의되었는가?	0.4	0.6

◇ 소프트웨어 입출력 정의의 완전성 검증에 관한 평가 값

완전성: DV420_IO_completeness		yes	no
1	DV420_IO_completeness_1:입력 및 출력이 완전히 정의되어 있는가?	0.6	0.4

2	DV420_IO_completeness_2:Block Diagram 및 Function Block에 모든 입력 및 출력이 기술되었는가?	0.8	0.2
3	DV420_IO_completeness_3:입력변수가 갖추어야 할 특성이 모두 정의되었는가?	0.7	0.3
4	DV420_IO_completeness_4:출력변수가 갖추어야 할 특성이 모두 정의되었는가?	0.7	0.3

◇ 소프트웨어 행위명세 완전성 검증에 관한 평가 값

완전성: DV430_Act_completeness		yes	no
1	DV430_Act_completeness_1:오류 사항들이 수정조치 사항과 함께 기술되어 있는가?	0.4	0.6
2	DV430_Act_completeness_2:SDS가 비정상적인 입력에 대한 소프트웨어의 행위를 명시하고 있는가?	0.4	0.6
3	DV430_Act_completeness_3:각 상세설계가 그 기능을 수행하는데 요구되는 작업 순서, 행위, 사건 및 타이밍을 명시하고 있는가?	0.8	0.2
4	DV430_Act_completeness_4:고장-안전 차원에서 컴퓨터 시스템에 요구되는 모든 행위가 완전하게 기술되어 있는가?	0.7	0.3
5	DV430_Act_completeness_5:SDS가 소프트웨어가 수행하지 말아야 할 것들도 언급하고 있는가?	0.7	0.3

◇ 소프트웨어 인터페이스 기술의 완전성 검증에 관한 평가 값

완전성: DV440_IF_completeness		yes	no
1	DV440_IF_completeness_1:다음과 같은 각 프로세서간의 연계가 모두 정의되었는가? - 공정계측입력 (PI Inputs) - 노외핵계측 (ENF Instrument) - 노심보호연산기 (CPC) - 자동시험 및 연계 프로세서 (ATIP) - 캐비닛 운전원 모듈 (COM) - 동시논리프로세서 (CP)	0.9	0.1

서 지 정 보 양 식

수행기관보고서번호	위탁기관보고서번호	표준보고서번호	INIS 주제코드
KAERI/TR-3311/2007			

제목 / 부제	Bayesian Belief Networks를 이용한 원자로보호계통 안전소프트웨어 설계명세의 정량 평가
연구책임자 및 부서명 (주저자)	엄흥섭 (종합안전평가부)
연 구 자 및 부 서 명	강현국 (종합안전평가부), 장승철(종합안전평부), 박기용(KNICS), 권기춘(KNICS)

출 판 지	대전	발행기관	KAERI	발행년	2006. 2.
페 이 지	97 p.	도 표	있음(○), 없음( )	크 기	21×29.7cm
참고사항					
비밀여부	공개(○), 대외비( ), __ 급비밀		보고서종류	기술보고서	
연구위탁기관			계약 번호		

초록

본 보고서는 불확실성을 포함하는 시스템의 모델링에 많이 활용되고 있는 Bayesian Belief Networks 기법을 이용하여 규칙 기반의 정성적인 소프트웨어 평가 방법론을 Bayesian Belief Networks로 모델링하여 PSA가 요구하는 소프트웨어의 정량적 신뢰도 정보를 생산할 수 있는 방안과 이를 활용한 사례 연구에 대하여 기술하였다. 제안된 BBN 모델은 안전 소프트웨어의 신뢰도에 관계된 정성적인 증거와 정량적인 증거 모두를 결합하여 정형적이고 정량적인 방법으로 결론을 추론할 수 있는 BBN의 특성을 활용하여 구축되었다. 그리고 사례 연구로서 원자로 보호 계통에 탑재될 안전 소프트웨어 설계명세서의 품질을 평가하는 데 적용하였고, 전문가에 의해 수행된 확인 및 검증 결과들이 모델의 입력으로 사용되었다. 만들어진 BBN 모델의 결과와 분석 내용은 전문가의 정성적인 판단과 유사하게 나타났으며 구축된 모델과 분석 내용들은 추후에 원전 안전계통 디지털 시스템의 PSA 및 KNICS V&V 업무에 활용될 예정이다.

주제명 키워드 (10단어내외)	안전 소프트웨어, 신뢰도, Bayesian Belief Nets, BBN, V&V

BIBLIOGRAPHIC INFORMATION SHEET

Performing Org. Report No.	Sponsoring Org. Report No.	Standard Report No.	INIS Subject Code
KAERI/TR-3311/2007			

Title / Subtitle	A Study on Quantitative Assessment of Design Specification of Reactor Protection System Software Using Bayesian Belief Networks
Project Manager and Department	H.S. Eom (Integrated Safety Assessment Division)
Researcher and Department	H.G. Kang (ISA), S. C. Chang(ISA), G. Y, Park(KNICS), K.C. Kwon(KNICS)

Publication Place	Daejon	Publisher	KAERI	Publication Date	2006. 2.
Page	97 p.	Ill. & Tab.	Yes(○), No ( )	Size	21× 29.7cm
Note					

Classified	Open(○),Restricted( ),-___ Class Document	Report Type	Technical Report
Sponsoring Org.		Contract No.	

Abstract(15-20 Lines)

This report propose a method that can produce quantitative reliability of safety-critical software for PSA by making use of Bayesian Belief Networks (BBN). BBN has generally been used to model the uncertain system in many research fields. The proposed method was constructed by utilizing BBN that can combine the qualitative and the quantitative evidence relevant to the reliability of safety-critical software, and then can infer a conclusion in a formal and a quantitative way. A case study was also carried out with the proposed method to assess the quality of software design specification of safety-critical software that will be embedded in reactor protection system. The V&V results of the software were used as inputs for the BBN model. The calculation results of the BBN model showed that its conclusion is mostly equivalent to those of the V&V expert for a given input data set. The method and the results of the case study will be utilized in PSA of NPP. The method also can support the V&V expert's decision making process in controlling further V&V activities.

Subject Keywords (About 10 words)	Safety critical software, Bayesian Belief Nets, BBN, V&V