KAERI/TR-2035/2002

확률론적 안전성평가를 위한 BBN 기반의 소프트웨어 정량적 평가 방안: COTS Case Study

A Bayesian Belief Nets Based Quantitative Software Reliability Assessment for PSA : COTS Case Study

KAERI

한 국 원 자 력 연 구 소

제 출 문

한국원자력연구소장 귀하

본 보고서를 2001 연도 "차세대원자로 설계관련 요소기술 개발" 과제의 기술현황분석보고서로 제출합니다.

2002. 3. .

부서명 : 종합안전평가팀

주 저 자 : 엄홍섭
공 저 자 : 성태용
정환성
박진균
강현국

부서명 : 동력로 기술개발팀
이기영
박종균

요 약 문

현재 원전 안전 계통에 사용되는 소프트웨어의 신뢰도 평가는 규칙기반의 정성적 기준에 의하고 있으나 원자력발전소의 안전성 평가를 위한 중요한 수단으로 사용되고 있는 확률론적 안전성 평가(PSA)에 디지털 시스템을 포함시켜야 하는 현실적 요구를 충족시키기 위해서는 소프트웨어 신뢰도의 정량화가 요구된다. 그러나 현재 각 산업분야에서 사용되고 있는 소프트웨어의 정량적 신뢰도 평가 방법들은 원전의 안전계통에 사용되는 고 신뢰도 소프트웨어를 평가하기에 불충분하여 이러한 시스템의 정량적 신뢰도 분석에는 소프트웨어 부분을 제외시키거나 또는 임의로 특정한 값을 지정하여 사용하고 있는 실정인데 이와 같은 문제점을 해결하기 위해 여러 가지 연구가 진행되고 있다.

본 보고서에서는 규제기관이나 산업체 등에서 현재 채용하고 있는 소프트웨어의 정성적인 평가 방법을 Bayesian Belief Net을 이용하여 정형적으로 모델링하고 PSA에서 요구하는 정량화 된 결과를 구하는 방안에 대하여 논의하였으며 원자력연구소에서 연구중인 "원전 상용소프트웨어 인정 프로세스"를 제안된 방안을 사용하여 동 방안의 PSA 활용 가능성을 검토하였다.

SUMMARY

Current reliability assessments of safety critical software embedded in the digital systems in nuclear power plants are based on the rule-based qualitative assessment methods. Then recently practical needs require the quantitative features of software reliability for Probabilistic Safety Assessment (PSA) that is one of important methods being used in assessing the whole safety of nuclear power plant.

But conventional quantitative software reliability assessment methods are not enough to get the necessary results in assessing the safety critical software used in nuclear power plants. Thus current reliability assessment methods for these digital systems exclude the software part or use arbitrary values for the software reliability in the assessment. This reports discusses a Bayesian Belief Nets (BBN) based quantification method that models current qualitative software assessment in formal way and produces quantitative results required for PSA. Commercial Off-The-Shelf (COTS) software dedication process that KAERI developed was applied to the discussed BBN based method for evaluating the plausibility of the proposed method in PSA.

목 차

표 목차

그림 목차

제 1 장 서론

확률론적 안전성 평가(Probabilistic Safety Assessment : PSA)는 원전의 안전성을 종합적이며 정량적으로 평가하기 위한 중요한 안전성 평가 수단으로 신규 원자력발전소 건설 시 인허가 사항으로 제출이 요구되며 최근에는 미국을 중심으로 PSA 결과를 현재까지 사용되던 결정론적인 규제의 보완 수단으로 사용하고 있는데 국내에서도 이의 채택이 적극적으로 검토되고 있다. 그러나 기존 원전에 적용되어 왔던 PSA 방법론을 그대로 디지털 시스템에 적용하여 시스템의 안전성을 정량화 하는 데에는 아직까지 해결되지 못하고 있는 다음과 같은 몇 가지 문제점이 있다[1].

- o Modeling the multi-tasking of digital systems
- o Estimating software failure probability
- o Estimating the effect of software diversity and V&V efforts
- o Estimating the coverage of fault-tolerant features
- o Estimating the CCF probability in hardware
- o Modeling the interactions between hardware and software
- o Failure mode of digital systems
- o Environmental factors
- o Digital system induced initiating events including human errors

위의 문제점들 중 원전 안전 시스템에 사용되는 안전 소프트웨어의 신뢰도 평가는 고장의 원인이 설계 결함에 주로 기인하고 또 입력에 대해 비 선형적 출력을 가지는 소프트웨어의 특성으로 인하여 시험이나 신뢰도 성장모델과 같은 기존의 정량적 방법 단독으로는 불충분하다는 것이 현재의 정설이다[2][3][4]. 이에 따라 각종 국제 표준이나 각 국의 규제기관들은 소프트웨어의 신뢰도에 관계되는 모든 활동과 자료들을 종합적으로 판단하는 규칙 기반의 정성적 평가에 의존하고 있다[5][6][7].

본 보고서에서는 규칙이나 절차 기반의 정성적 평가 절차에 근거한 현행 소프트웨어의 정성적 신뢰도 평가 방법을 Bayesian Belief Net을 이용하여 정량적 결과를 얻을 수 있는 방안에 대하여 논의하였고 한국원자력연구소에서 연구 중인 상용소프트웨어의 인정 프로세스를 시험케이스로 적용하여 동 방안의 PSA 적

용 가능성을 검토하였다. 상용 소프트웨어는 기본적으로는 안전 디지털 시스템의 제작자가 직접 개발한 소프트웨어와 유사한 특성이 많지만 다른 점으로는 동 소프트웨어의 사용 이력을 구할 수 있다는 장점이 있는 반면 개발 과정의 평가가 어려운 경우가 대부분이고 또 각 개발 단계별 생산 문서를 평가 전문가가 쉽게 얻을 수 없다는 단점이 있다. 이러한 단점들 때문에 안전 소프트웨어 제작자가 개발한 소프트웨어의 평가 경우에 비해 보다 절차적이고 정성적인 평가에 의존하고 있는 실정이며 개발 팀의 명성이나 또는 회사의 소프트웨어 개발 품질에 대한 국제 인증 등이 하나의 중요한 요소로 작용한다[8].

제 2 장. 원전 상용 소프트웨어 인정 프로세스와 BBN 방법론

제 1 절 원전 상용 소프트웨어 인정 프로세스

BBN을 이용하여 문제를 해결할 때 일반적으로 전문가의 지식과 추론 과정을 추출하여 사용하거나 또는 표준이나 가이드라인의 절차나 기준을 따르게 된다. 본 보고서에서 BBN 모델을 적용한 대상은 상용 소프트웨어 인정(dedication) 방법 4가지 중 method 2인 공급자 조사 방법에 상용 소프트웨어 인정 프로세스에 관하여 NUREG/CR-6421의 기본 개념을 바탕으로 하고 EPRI/TR-106439 기준을 적용하여 절차적 관점에서 상용 등급 조사(commercial grade survey) 방법에 의한 상용 소프트웨어 인정 프로세스를 사용하였다. 모델을 위해 사용된 상용 소프트웨어 인정 프로세스는 9 단계로 되어 있으며 전체적 구성은 그림 2-1과 같이 되어있다[8].

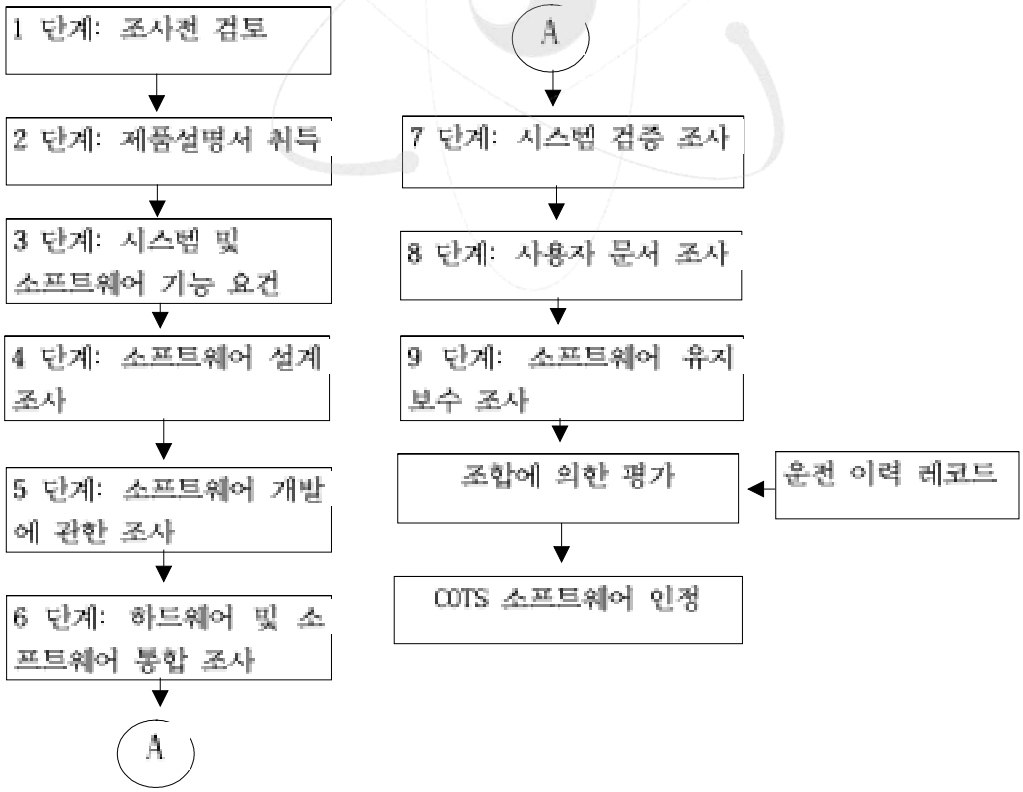


그림 2.1 상용 소프트웨어 인정 프로세스

상용 소프트웨어 인정 프로세스의 각 단계는 다시 세부적인 검토 항목이 정해져 있는데 그 주요 내용은 다음과 같다[8].

단계 1: 조사 전 검토 회의(pre-survey meeting)

정보공학 방법론에 따른 소프트웨어 엔지니어링 절차를 준수하여 소프트웨어를 개발하였는지 여부를 검토하며 ISO 9001 Part 3과 같은 관련 국제 표준이나 가이드라인의 준수 여부가 중요한 판단의 기준이 된다. 검토 항목은 다음과 같다.

- o 계획문서 검토(planning documentation review)
- o 생명주기 모델 및 단계별 활동사항 검토
- o ISO 9001 Part 3 또는 기타 국제 표준에서 규정한 항목들의 검토
  - 자체 자격(qualification) 여부
  - 외부 자격 여부

단계 2: 제품 설명서 검토

제조 회사가 제공할 수 있는 모든 제품 설명서를 검토하는 단계로 다음과 같은 검토 항목이 있다.

- o 새로운 기능의 추가 및 변경 정의
- o 개정 레벨, 최소한의 하드웨어 요건 및 개정 이전에 확인된 오류의 교정을 포함한 소프트웨어 개정 이력
- o 하드웨어 및 소프트웨어의 제품 설명자료 취득
- o 새로운 콤포넌트에 대한 소프트웨어 개발 프로세스와의 링크 여부

단계 3: 시스템 및 소프트웨어 기능 요건 조사

소프트웨어의 요구사양명세를 검토하는 단계로 상위 레벨의 시스템 명세로부터 소프트웨어 요구사양 명세까지 추적성 분석의 개념을 근간으로 다음과 같은 검토 항목이 있다.

- o 시스템 및 소프트웨어 기능요건 명세서 적절성
  - 기능 설명서 조사
  - 요구사항 명세서 조사
  - CASE를 비롯한 개발 도구 사용 여부

o 기능요건 명세 목록의 존재여부 및 적절성
o 제품에의 기능 요건 반영 여부
o 기타 요구사항의 존재 유무
o 소프트웨어 요구사양 명세서의 검토 수행 여부

단계 4: 소프트웨어 설계 조사

소프트웨어의 설계 요건이 적절하게 작성되었나를 단계로 만약 설계 요건을 취득할 수 없다면 어떤 설계 정보를 이용할 수 있는지 여부와 이들 설계 결과물에 대한 충분한 검토가 개발과정에서 이루어졌는지를 평가하는데 다음과 같은 검토 항목이 있다.

o 소프트웨어 설계 프로세스 검토
  - 예비 및 상세 설계의 검토 여부
  - 예비 및 상세 설계의 검토자 자격 요건
o 소프트웨어 설계, 구현, 통합, 테스팅 및 최종 사용에 대한 검토
o 범위, 정확도 및 갱신구간을 포함한 출력물 검토
o 초기와 요건 검토
o 컴퓨터 시스템에 탐지된 고장에 반응하는 프로그램 로직의 검토
o 오퍼레이터 인터페이스 검토
o 인서비스(in-service) 테스트 특성 및 진단의 검토
o 전반적인 컴퓨터 시스템 응답시간을 포함한 시간 요건 검토
o 하드웨어 성능과 일치하는 프로세싱 유휴시간 및 excess memory 검토
o 보안성 요건 검토

단계 5: 소프트웨어 개발에 관한 조사

소프트웨어 품질 보증 목표와 계획 그리고 계획의 준수 여부를 검토한다. 소프트웨어 개발과 동시에 적절한 검증이 소프트웨어 생명주기 단계별로 이루어졌는지를 평가해야 하는데 다음과 같은 검토 항목이 있다.

o 소프트웨어 품질 계획의 검토
  - 산업체 표준(ISO 9000-3) 요건의 준수 여부
o 소프트웨어 개발 문서들에 대한 검토
o 소프트웨어 개발 절차 검토

단계 6: 하드웨어 및 소프트웨어 통합 조사

하드웨어와 소프트웨어의 통합계획 여부, 통합 시험 절차 및 관련 승인기준 여부, 형상시험 여부, 통합 변경 제어 여부를 검토하는 단계로 다음과 같은 검토항목이 있다.

- o 하드웨어와 소프트웨어의 통합 계획 여부
- o 하드웨어 및 소프트웨어 인터페이스의 적합성을 입증하기 위한 통합시험절차 및 관련 승인 기준 존재 여부
- o 통합 컴퓨터 시스템에 대한 형상시험 수행 여부
- o 하드웨어 및 소프트웨어 통합 변경 제어에 대한 품질보증계획서 존재 여부
- o 비휘발성 메모리로 프로그래밍 된 프로세스 점검

단계 7: 시스템 검증 조사

시스템 검증에 대한 시험계획 여부, 테스트 절차서 존재 및 완결성 여부, 확인 과정에서 나타난 결과 검토, 시스템 테스트와 관련된 사항을 검토하며 다음과 같은 검토 항목이 있다.

- o 상세 유형시험 절차 검토
  - 절차서와 입력자료 및 기대치 결과의 적절성
  - 정적조건, 동적 조건하의 시험 여부
- o 시스템 검증 시험 검토
  - 개발부서와 시험자 사이의 조직관계
  - 형상 시험(configuration test)
  - 테스트 보고서의 이용여부와 적절성
  - 테스트 보고서의 평가 및 적절성
  - 테스트 과정에서의 오류 발견 적절성
  - 하드웨어 및 코드의 정확성 파급효과 고려 여부

단계 8: 사용자 문서 조사

회사로부터 취득 가능한 모든 매뉴얼에 대하여 요구사양 및 설계 요건 부합 여부를 평가하고 이들의 일관성, 명확성, 정확성 유지를 검토하는 단계로 다음과 같은 검토 항목이 있다.

- o 사용자 매뉴얼이 존재하고, 그것은 프로그래머가 소프트웨어를 개발하는데 필요한 충분한 정보를 포함하고 있는가?

- 사용자가 프로그램을 설치하고 운영하는데 필요한 충분한 정보를 제공하는 설치/운전 절차서가 있는가?
- 제품의 고장 진단에 필요한 충분한 정보를 담고 있는 유지보수 매뉴얼이 있는가?
- 오류 메시지의 목록과 그에 관련된 수정 및 권고 사항이 있는가?
- 사용자 문서에 제품의 이름과 버전과 같은 소프트웨어의 확인 항목들이 명시되어 있는가?
- 사용자 문서는 동 문서에 사용된 약어(심볼, 명령어 문법 등)를 포함하고 있는가?
- 사용자 문서는 사용자가 소프트웨어나 문서에 문제가 있을 경우 보고하는 방법과 절차를 명시하고 있는가?

단계 9: 소프트웨어 유지보수 조사

최초의 배포(original release)가 이루어진 이래 발생된 문제점들을 어떻게 해결했는가를 검토하고 관련 기록 정보를 평가한다. 또 제품에서 발견된 오류들이 어떻게 사용자들에게 통지되었는지 여부를 점검하고 이들 자료의 형상관리 여부를 검토하게 되는데 여기에는 다음과 같은 검토 항목이 있다.

- 변경 통지에 대한 감사 수행 여부
- 소프트웨어 선적 전 기능적 검사 및 물리적 감사 수행 여부
- 소프트웨어 소스코드와 실행코드에 대한 접근 및 형상제어를 적절히 유지하는지 여부

이상의 9단계 외에 추가로 상용 소프트웨어의 운전이력을 평가하여 이를 최종적인 인정 프로세스에 반영하게 되는데 이 단계에서 검토되는 항목은 다음과 같다.

- 충분한 운전 이력이 있는가?
- 운전 이력에 대한 기록 자료는 신뢰성이 있는가?
- 회사 내부와 외부에서 시스템 오류에 대한 공식적인 보고 과정이 있는가?
- 보고된 문제의 처리 상태를 추적할 수 있는 메커니즘이 있는가?
- 우선 순위, 스케줄링, 추적 등에 대한 문제를 해결하는 체계적인 접근방법이 존재하는가?
- 운영체제에 해결되지 않은 중요한 문제가 있는가?

o 고객 통지 과정이 존재하는가?
o 오류보고는 적절하게 추적되고 처리되는가?
o 해당 소프트웨어의 응용분야가 사용하고자 하는 분야(예: 원자력의 안전관련 응용분야)와 유사한가?

제 2 절 BBN 방법론

Bayesian Belief Net(BBN)은 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 적용하여 계산한 후 이를 분석하여 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[9].

BBN 상의 노드들은 목표노드(target node), 관찰가능 노드(observable node) 그리고 중간 노드(intermediate node)로 구분할 수 있다[10].

o 목표 노드는 모델에서 평가 목적에 해당하는 노드로서 "프로그램의 무결함" 등이 될 수 있다.
o 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.
o 중간 노드는 제한된 정보나 믿음(belief)을 나타내는 것으로 "개발 과정의 품질" 또는 "제작자의 명성" 등이 여기에 해당된다.

본 보고서에 사용된 BBN과 관련된 중요 용어와 개념은 다음과 같다.

o 실체(Entity)

BBN에서 목표 평가 대상과 관련된 사건 또는 실체들을 뜻한다. 소프트웨어 신뢰도 평가의 경우 이들 실체들은 다음과 같은 것들이 될 수 있다.

- 인간이나 기계에 의하여 수행되는 과정들
  (개발 과정, 시험 과정, 확인 및 검증과정 등)
- 각 과정에 입력이 되는 인적 물적 자원(개발부서, 시험 팀 등)
- 각 과정의 산출물(설계 문서, 시험 결과, 생산된 소프트웨어 등)

o 속성(Attribute)

실체(entity)의 특성을 가리킨다. 이런 특성들은 BBN에서 노드로 표현될 수 있다.

o 이디엄(Idiom)

(SERENE method의 고유 용어)

Safety argument를 구성하는 일반적이고 원소 적인 구성품(BBN 조각 그래프)으로 서로 다른 safety argument를 만들 때 중복적으로 발생하며 재사용 가능한 단위 BBN 그래프, 추론의 일반적 방법을 추상화한 것으로 볼 수 있다.

o 노드 확률 테이블(Node Probability Table: NPT)

BBN 상의 모든 노드는 NPT를 가지고 있다. 그 구성은 자신의 상태와 모 노드의 상태에 의하여 결정된다. A 노드가 모 노드 B, C를 가지고 있다면 A노드의 NPT는 노드 A, B, C의 모든 상태가 조합된 p(A|B, C)를 표현하게 된다. 즉 A 노드가 x개의 상태, B 노드가 y개의 상태 그리고 C 노드가 z개의 상태를 가지고 있다면 A 노드의 NPT는 x*y*z 개의 셀(cell)을 가지게 된다. 만약 A 노드가 모 노드가 없는 루트 노드인 경우에는 A 노드의 NPT는 자신의 각 상태 x개에 대한 사전 확률만을 가지게 된다.

o 안전성 논증(Safety argument)

(SERENE method 고유 용어)

안전성에 관련된 증거와 이들 관련된 증거들이 안전성 목표(논증, 주장)를 증명하는데 충분하다는 것을 보여주는 연결을 의미한다.

o 템플릿(Template)

동일 조직 또는 다른 조직에서 만들어진 safety argument 또는 safety sub-argument를 BBN으로 표현한 형태로서 재사용이 가능한 BBN 조각이다. 두 가지 형태의 재사용이 가능하며 하나는 그래프 형태의 재사용이고 다른 하나

는 NPT의 재사용이다. 템플릿은 특정 시스템의 형태나 추론의 형태에 적합하도록 일반화 될 수 있다. 가장 간단한 형태의 템플릿은 이디엄을 실증화 한 것이다.

o Root 노드(node)

BBN에서 모 노드가 없는 노드를 지칭한다. 이 노드에는 입력선(arc)이 없으며 따라서 이런 노드의 확률테이블에는 조건부 확률이 아니라 사전 확률 값을 지정한다.

o 증거(Evidence)

시스템 또는 그 시스템의 개발 과정에 관련된 여러 가지 사실들, 그리고 개발 도중 또는 개발 후에 측정되거나 평가된 사실들을 말하며 이들은 BBN에서 평가하고자 하는 목표 노드에 어떤 방식으로든 영향을 주는 요인들이 된다.

o SERENE Project

Safety and Risk Evaluation using bayesian Nets의 약자로, ESPRIT Framework IV project 22187로 수행된 과제이다. Center for Software Reliability(City University London)가 주관을 하고 Edf 외에 수 개 기관이 공동으로 수행하였다. 본 과제의 주목적은 BBN을 사용하여 Programmable Electronic Systems의 안전성을 추론하는 방법을 개발하는 것이었고 이에 따라 "SERENE method"라는 방법론과 SERENE method를 지원할 수 있는 기능을 추가한 기존 BBN 도구(HUGIN)의 기능 향상판(데모 버전)을 개발하였다[11].

제 3 장 상용소프트웨어 인정 프로세스의 BBN 구축

제 1 절 BBN 구축 절차

일반적인 BBN 구축 절차는 모델링 대상 시스템의 변수들을 확인하고 이들 확인된 변수들을 BBN의 노드로 사용하여 그래프를 작성한 후 각 노드들의 노드 확률 테이블을 작성하고 마지막으로 관찰된 값을 모델에 입력하여 관심 노드(변수)들의 계산된 결과를 분석하는 것이다. 각 단계별 구체적인 작업 내용은 다음과 같다.

(1) 단계-1: BBN 모델에 사용되는 변수들의 확인

모델을 구축하는데 필요한 모든 관련된 증거들을 확인/작성하는 단계이다. 이 증거들은 관련된 변수(실체: 과정, 제품, 자원 등)들과 그들의 속성 또는 특성에 대한 서술(description)이며, 추후 구축하려는 BBN 상에서 노드가 된다. 이 단계에서는 이들 실체와 속성들 간의 연결 관계는 고려하지 않아도 되며 단순히 나열하기만 하면 된다. 그리고 관련된 표준들이나 생명주기가 반드시 고려되어야 하는데 주요 수행 내용은 다음과 같다.

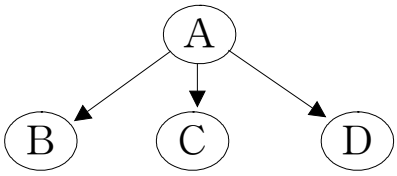
- o 목표 확인. (예측하려는 대상이 무엇인가)
- o 평가 대상에 관련되는 주요 실체(entity)들의 목록 작성
- o 시스템에 관련된 실체들의 주요 속성들을 결정
- o 관련된 속성들의 분류
- o 개발 생명주기를 검사하고 주요 과정을 명시화
- o 관련된 표준, 절차서, 지침서의 검토.

(2) 단계-2: 확인된 변수들을 사용하여 그래프 작성

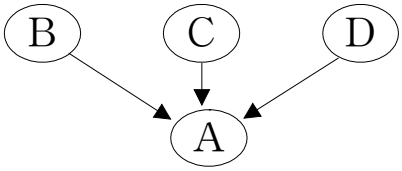
모델에 필요한 변수들이 모두 확인되면 다음 단계는 이들 변수들의 관계를 고려하여 그래프를 작성한다. BBN의 그래프에서 가능한 노드의 그래프 연결 형태는 다음의 3가지이다.

- o 직렬연결

ㅇ 분기연결



ㅇ 수렴연결



BBN 모델링에 사용될 노드(변수)들을 사용하여 목표 노드의 값을 구하기 위해서는 각 노드들의 연결 관계를 확인해야 한다. 이들 노드들의 연결관계는 인과관계적 결정(causal determination), 통계적 결정(statistical determination), 구조적 결정(structural determination) 등의 추론 형태를 가지는데 이들 추론 형태에 적합하도록 위에서 기술된 3가지 그래프 형태를 적용하여 전체 BBN 그래프를 작성한다.

(3) 단계-3: 각 변수들의 노드 확률 테이블 작성

BBN 그래프가 완성된 다음 각 노드의 노드 확률 테이블(NPT)을 정의하는 단계로서 먼저 노드의 상태를 정의한 다음 각 상태에 대한 확률 값을 정하게 된다. BBN 모델에서 가장 중요한 부분중의 하나이며 또 만들기 어려운 부분이 노드 확률 테이블이다. 대부분의 경우 노드 확률 테이블을 작성하기 위하여 분야 전문가에게 의존해야 되고 해당 전문가는 각 노드의 상태별 확률 값을 추출해야 한다. 노드의 상태별 확률 값을 작성하는 방법은:

(i) 실험치나 또는 통계적 자료와 같은 객관적 자료를 사용하여 작성하는 경우

(ii) 전문가의 주관적 판단에 의거하여 작성하는 경우의 2가지이다.

그러나 대부분의 분야 전문가들은 확률을 추산하거나 추론하는데 있어서 익숙하지 못하고 잘못을 저지르기 쉽다는 것이 일반적인 정설이고[12] 이런 점이 노드 확률 테이블의 작성을 어렵게 만들고 있다[11].

전문가의 확률 판단에 영향을 미치는 공통적 문제점들 특히 확률에 관련된 편

향과 오류는 아래와 같은 것들이 있으며 이러한 편향에 대해 충분히 주의를 기울이는 것이 전문가로부터 확률 값을 추출하거나 또는 추출된 확률을 조정하는데 중요하다[11].

o 확률 평가에 관련된 편향과 오류의 형태들
- 대표성(Representativeness)
- 확실성의 부인(Denial of certainty) 또는 제어(Control)
- 유효성(회상/상상의 용이성, Availability)
- 조정 및 고착(Adjustment and Anchoring)
- 결합 오류(Conjunction fallacy)
- 분산, 공분산, 상관성을 평가할 때의 어려움
- 보수적 경향(Conservatism)
- 과신(Overconfidence)
- 원인과 진단상의 추론에 관련된 오류들

(4) 단계-4: 해당 노드에 관찰된 값을 입력한 후 계산

BBN 그래프와 NPT가 모두 만들어지면 마지막 단계로 각 노드에 값을 입력한 후 목표 노드 및 관심 노드의 값을 계산하고 분석한다. 최근에 복잡한 계산을 신속하게 해 주고 또 모델의 작성과 분석을 용이하게 해주는 도구들이 개발되어 BBN의 실용적 분야 적용을 가능하게 만들었다[13][17].

BBN을 구축하는 세부적 절차는 대상 시스템의 특성과 확보 가능한 증거의 종류와 양에 따라 서로 다르고 또 현 시점이 BBN이 실제적 문제의 해결에 적용되기 시작한 초기 단계이므로 아직까지는 일반적으로 인정되는 절차가 없는 실정이다. 최근에 SERENE 과제에서 소프트웨어의 안전성 평가를 위한 BBN 모델을 구축하기 위해 최초로 체계적인 BBN 구축 절차를 개발하였는데 이 절차는 아래의 그림 3.1과 같다.

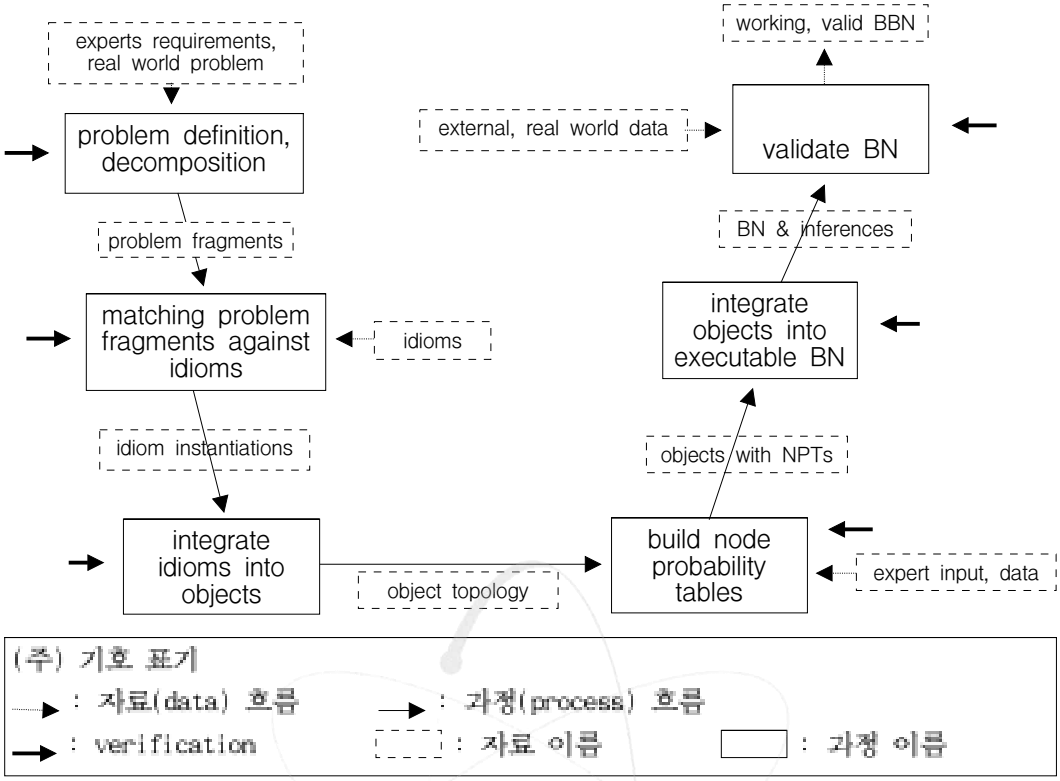


그림 3.1 SERENE Method에 의한 BBN 구축 세부 절차도

제 2 절 상용소프트웨어 인정 프로세스 BBN 작성

1 변수 확인

BBN을 이용하여 문제 해결을 위한 모델링을 할 때 일반적으로 가장 먼저 해야 할 작업은 모델 구축에 필요한 모든 관련된 변수들을 확인하는 것이다. 이 변수들은 최종 목표 변수에 관련된 변수(과정, 제품, 자원 등)들과 그들의 속성 또는 특성에 대한 서술이며 이것들이 추후 BBN 그래프를 작성할 때 노드로 사용된다.

본 보고서에서 논의된 BBN 모델에서는 제 2 장에서 기술된 상용 소프트웨어 인정 프로세스 9 단계와 "상용 소프트웨어의 운전 이력"을 기본 레벨의 변수들로 설정하였고 목표 노드는 "상용 소프트웨어 승인"으로 설정하였다. 그리고 기본 레벨의 변수 9개와 상용 소프트웨어의 운전이력 변수는 4개의 그룹으로 분류하였다. 상용 소프트웨어 인정 프로세스를 구성하는 9단계와 운전이력 변수는 프로세스 각 단계에서 검토되는 세부 질문사항들을 가지고 있는데 이들 세부 질

문사항들을 각 기본 레벨 변수의 하위 변수로 설정하였다. 이와 같이 설정된 변수의 구성은 다음과 같다.

o 목표 변수

"상용 소프트웨어 승인"

o 4개의 그룹으로 분류된 기본 레벨의 변수(9단계 프로세스와 운전이력)

표 3.1 COTS 인정 프로세스의 그룹 변수와 기본레벨 변수

목표 변수	그룹 변수	기본 레벨 변수
상용소프트웨어 승인	개발 과정	조사 전 검토 단계
		시스템, 소프트웨어 요건 검토 단계
		설계 검토 단계
		소프트웨어 개발 검토 단계
	소프트웨어 지원	제품설명서 검토 단계
		소프트웨어 유지보수 조사 단계
		사용자 문서 조사 단계
	통합 및 검증	하드웨어와 소프트웨어 통합 단계
		검증 검토 단계
	운전이력	운전이력 기록

o 기본 레벨 변수들의 하위 레벨 변수

각 기본레벨에 속하는 하위 레벨 변수들은 소프트웨어 인정 프로세스의 각 단계에서 검토되는 세부 질문사항들로 구성되어 있으며 부록 2에 수록되어 있다.

2. 그래프 작성

앞에서 기술된 변수확인 단계에서 모델링 대상이 되는 시스템의 모든 변수들이 확인되면 이들 변수들을 노드로 하여 BBN 그래프를 작성한다. 상용 소프트웨어 인정 프로세스의 9단계가 기본이 되는 기본레벨 변수들과 이들 기본 레벨 변수들을 그룹으로 만든 변수 그리고 운전이력 변수로 구성된 BBN 그래프는 다음 그림 3.2와 같다.

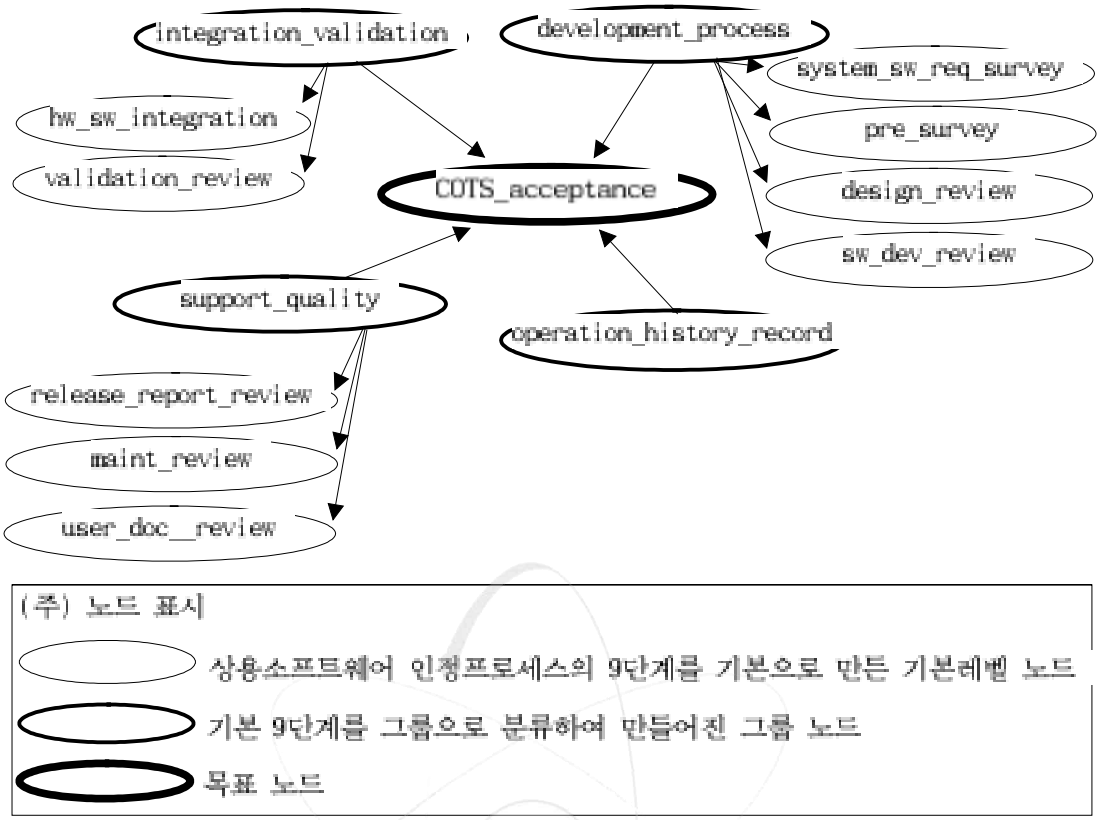


그림 3.2 상용 소프트웨어 평가 상위 레벨 BBN 그래프

그림 3.2 "상용소프트웨어 인정 프로세스 평가 상위 레벨 BBN그래프" 상의 각 기본 노드는 다시 각 단계에 포함되어 있는 세부 검토 항목을 자노드(child nodes)로 하여 구성되며 각 기본 노드별 그래프는 다음 그림 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12와 같다.

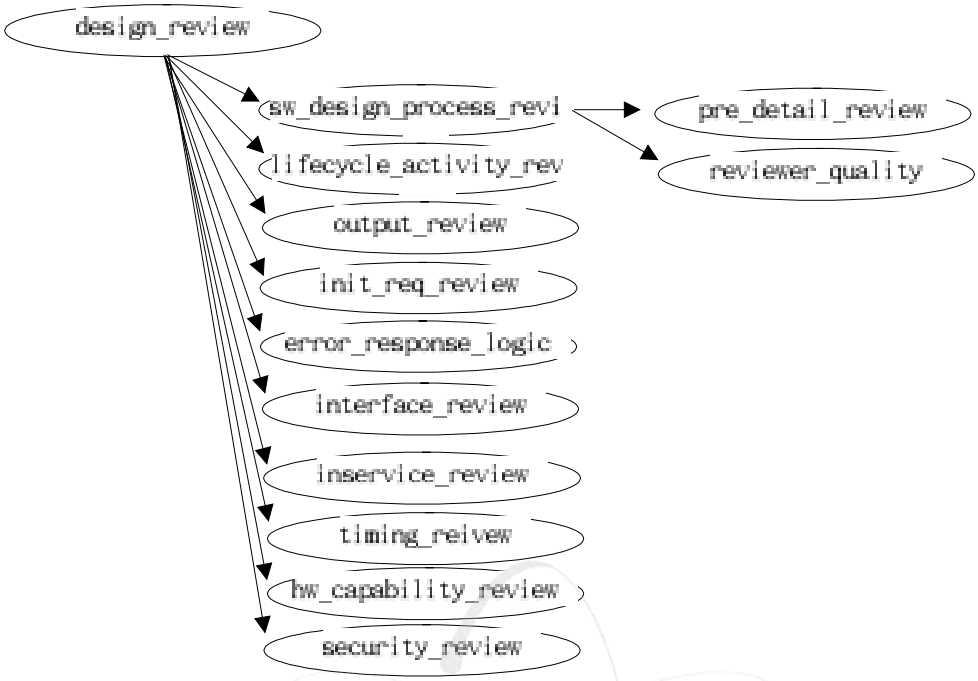


그림 3.3 design_review 노드의 하위 레벨 그래프

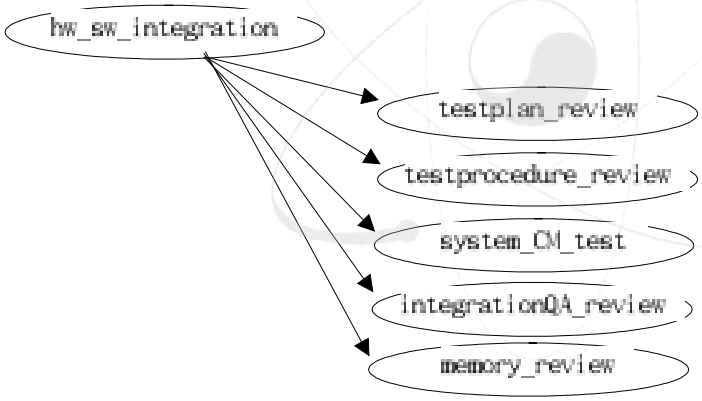


그림 3.4 hw_sw_integration 노드의 하위 레벨 그래프

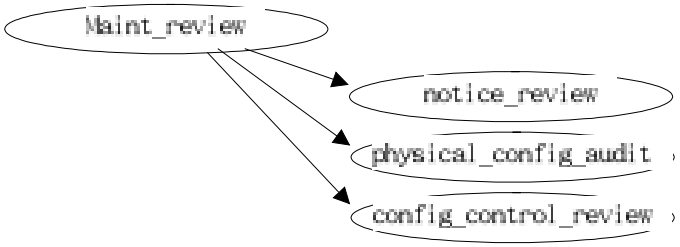


그림 3.5 maint_review 노드의 하위 레벨 그래프

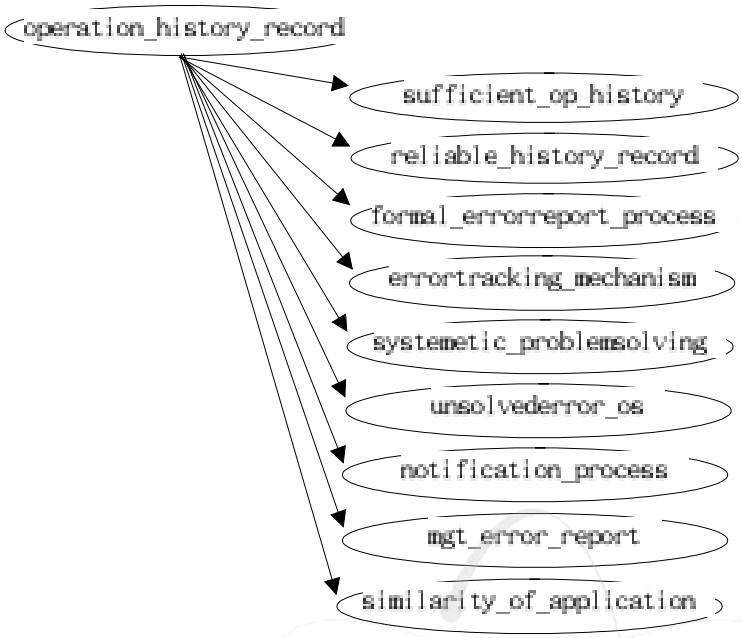


그림 3.6 operation_history_record 노드의 하위 레벨 그래프

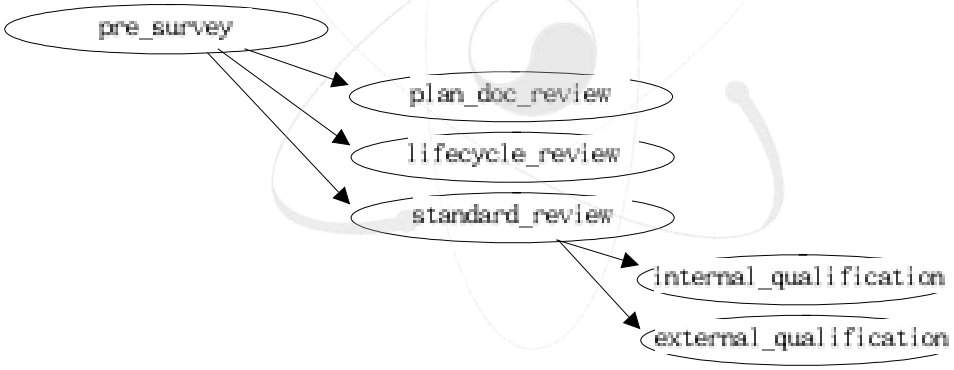


그림 3.7 pre_survey 노드의 하위 레벨 그래프

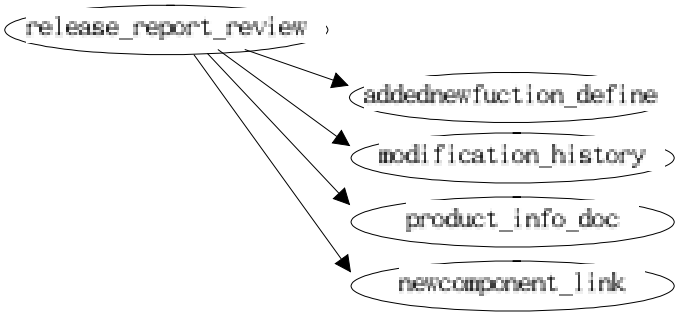


그림 3.8 release_report_review 노드의 하위 레벨 그래프

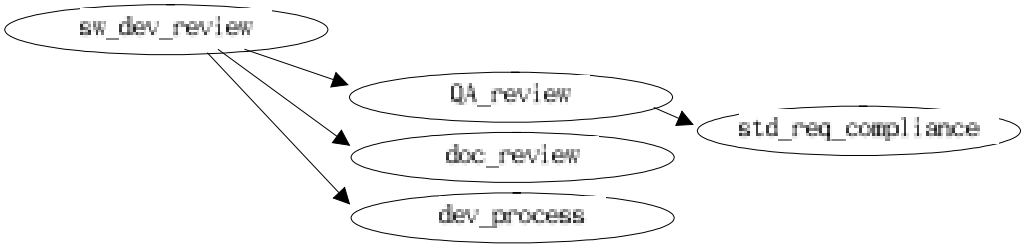


그림 3.9 sw_dev_review 노드의 하위 레벨 그래프

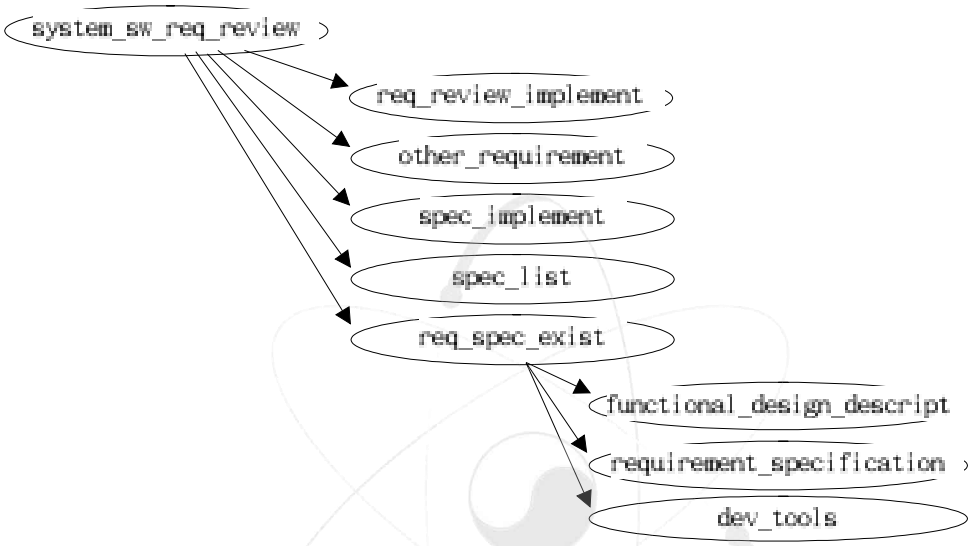


그림 3.10 system_sw_req_review 노드의 하위 레벨 그래프

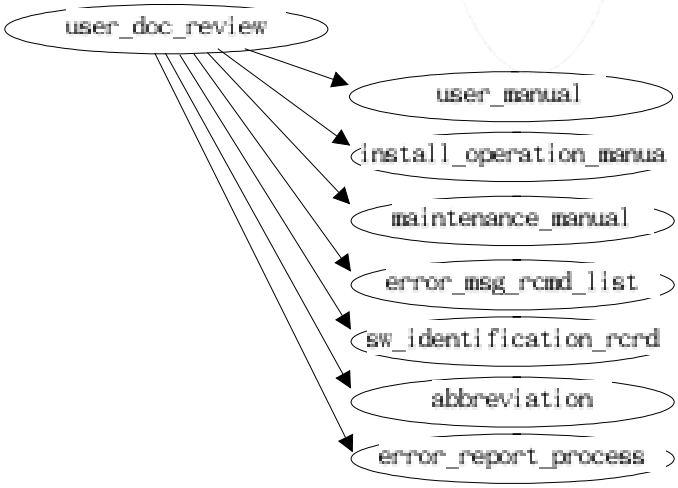


그림 3.11 user_doc_review 노드의 하위 레벨 그래프

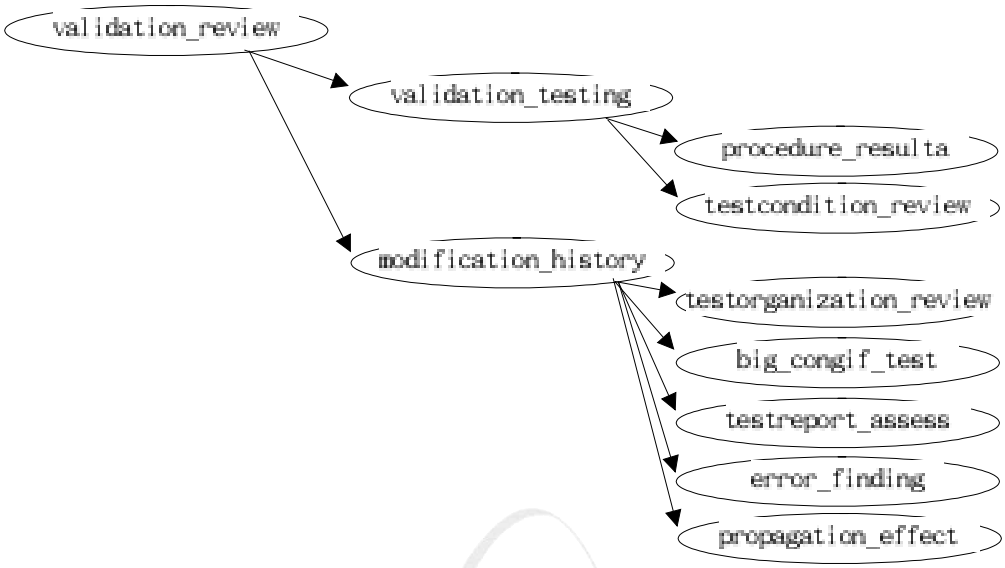


그림 3.12 validation_review 노드의 하위 레벨 그래프

위와 같은 방법으로 만들어진 상용소프트웨어 인정 프로세스의 전체 BBN 그래프는 부록 1의 그림과 같다.

3. BBN 노드들의 노드 확률 테이블 작성

BBN 그래프의 작성이 완료되면 다음 단계는 각 노드별로 노드 확률 테이블을 정의하는 것인데 이것은 (i)각 노드의 상태를 정의하는 것과 (ii)각 상태별 확률 값을 정의하는 것으로 나누어진다. 부록 1의 그림에서 나타난 전체 BBN 그래프 상의 각 노드들의 상태는 다음과 같이 정의되었다.

o 목표 노드 "COTS_ACCEPTANCE" :
[accept]와 [not accept] 두 개의 상태

o 중간 노드 전체 : [good]와 [bad] 두 개의 상태
상용 소프트웨어 인정 프로세스의 9단계에 해당되는 노드, 이들 9단계 노드들을 그룹으로 분류하여 만들어진 노드, 세부 검토 항목 중 자노드를 가진 노드들 그리고 운전이력 노드가 여기에 속한다.

o 관찰가능 노드 전체 : [Yes]와 [No] 두 개의 상태

상용 소프트웨어 인정 프로세스의 9단계와 운전이력 검토항목에 포함되어 있는 세부 검토 항목들이 여기에 속한다.

노드들의 각 상태가 정의되면 다음 단계는 노드 각 상태의 초기 확률 값 또는 조건부 확률 값을 설정해야 한다. 이들 확률 값은 객관적인 자료나 또는 전문가의 지식과 판단에서 추출되어야 하나 본 보고서에서는 모델의 대상이 된 상용 소프트웨어 인정 프로세스가 현재 연구 중에 있고 또 실제의 평가 케이스에 적용된 바가 없어 임의로 설정하였다. 설정된 노드의 상태별 확률 값은 다음과 같은 방법으로 작성하였으며 만들어진 전체 노드의 노드 확률 테이블은 부록 3에 수록되었다.

(1) 목표 노드의 노드 확률 테이블

본 BBN 모델에서 구하고자 하는 것은 상용 소프트웨어의 정성적인 평가 절차를 이용한 상용 소프트웨어의 정량화 된 신뢰도 값이다. 이 값은 시험(testing)과 같은 통계적 처리를 통해 구해질 수 없는 것이므로 목표 노드의 각 상태에 평가에 필요한 특정 신뢰도 값을 설정하여 구하는 것이 한 가지 방법이 될 수 있다. 본 BBN에서 목표 노드의 상태에 설정된 신뢰도 값은 다음과 같다.

목표노드 상태	설정된 신뢰도 값
accept	=<1x10-4 pfd
not accept	>1x10-4 pfd

이렇게 목표 노드 "COTS acceptance"에 설정된 두 개의 상태와 그 값이 의미하는 것은 상용 소프트웨어 인정 프로세스의 각 단계별로 조사를 통해 획득한 증거들을 관찰 가능 노드에 입력한 후 전체 네트(Net)의 계산을 수행했을 때 목표 노드의 각 상태가 어떤 확률 값을 가지고 있는지 추론하고 설정된 신뢰도 값의 사용 여부를 결정하기 위한 것이다. 즉 목표 노드가 [>1x10-4 pfd]인 상태가 될 확률과 [=<1x10-4 pfd]가 될 확률을 구해서 해당 상용 소프트웨어의 신뢰도 값으로 사용할 수 있는지 여부를 판단하는 목적으로 정의한 것이다.

여기에서 두 상태를 구분하는 기준 값 "1x10-4 pfd"은 BBN에서 직접적으로 구해지지 않는데 이것은 BBN이 시험(Testing)이나 신뢰도 성장 모델과 같은 통계적 기법을 사용하여 직접적인 고장 확률 값을 구하는 것이 곤란한 문제의 해결

에 적용되었기 때문이다. 따라서 이 값은:

(a) PSA를 신뢰도 배분을 위한 도구로 사용하여 전체 안전계통 디지털 시스템의 목표 신뢰도 값으로부터 역으로 계산하여 값을 구하거나,

(b) 시스템 요구 사항에서 결정된 수치를 사용하거나

(c) 전문가가 전체 시스템의 안전성 분석을 고려하여 필요하다고 판단한 값을 사용하는 방법을 택할 수 있다.

위와 유사한 방법을 사용한 경우로는:

(a)와 목적은 다르지만 방법상으로는 유사하게 확률론적 안전성 평가(PSA)로부터 목표 수치를 구한 방법의 예로는 PSA를 사용하여 원전 안전계통 소프트웨어의 시험횟수를 결정하는 방법에 대한 연구[14]가 있었고,

(b)의 예로는 Sizewell B 원전의 소프트웨어 기반 보호계통의 요구사항(1x10-3 pfd)이 있으며,

(c)의 예로는 Westinghouse AP600의 보호 및 안전계통 PSA 예(1.1x10-5 pfd)와 월성 원전 SDS2의 소프트웨어 고장수목 분석 예(1x10-4)가 있다.

목표노드 "COTS acceptance"는 4개의 모노드를 가지고 있고 이들 모노드들은 각기 2개씩의 상태를 가지고 있으므로 목표 노드에는 모노드의 각 상태 수와 자신의 상태 수 조합(2x2x2x2x2)에 해당하는 32개의 조건부 확률 값이 정해야 한다. 이 값들 역시 과거의 객관적 이력이나 또는 전문가의 지식과 판단으로부터 추출 되어야하나 본 BBN에서는 각 모노드의 중요성을 감안한 단순 가중치를 적용하여 각 상태별 확률 값을 작성하였다. 목표노드의 모노드들에 할당된 가중치는 다음과 같다.

모노드 이름	가중치
development_process	0.3
support_quality	0.2
integration_validation	0.2
operation_history_record	0.3

위와 같이 설정된 가중치를 사용하여 목표노드의 상태 확률 값을 만드는 예를 들어보면, 4개의 모노드 전체가 "good"의 경우 목표노드 "COTS acceptance"의

상태 "accept"의 조건부 확률 값은 1(0.3+0.3+0.2+0.2)이 되고 "not accept" 상태의 확률 값은 0이 된다. 그리고 모노드 development_process와 모노드 operation_history_record의 상태가 "good"이고 다른 두 개의 모노드는 "bad"일 경우에 목표노드의 상태 "accept"의 조건부 확률 값은 0.6(0.3+0.3)이고 상태 "not accept"의 확률 값은 0.4가 된다. 이와 같이 설정된 가중치를 사용하여 만들어진 목표노드의 전체 상태별 조건부 확률 값은 부록 3에 수록되어 있다.

(2) 중간 노드의 노드 확률 테이블

중간 노드의 상태 [good]와 [bad]는 직접적으로 측정 가능한 것이 아니고 제한된 정보나 믿음을 표현하는 것인데 자노드인 관찰가능 노드들의 상태 값에 의해서 결정된다. 본 BBN에서 중간 노드에는 상용 소프트웨어 인정 프로세스의 9단계에 해당하는 노드, 운전이력 노드, 9단계 노드를 그룹으로 분류하여 만든 노드 그리고 각 단계에 속하는 세부 검토항목 노드들 중 자노드를 가지고 있는 노드들이 있다. 각 중간 노드들의 노드 확률 테이블은 다음과 같은 방법으로 작성되었다.

o development_process 노드

모노드 4개의 중요성이 동일하다고 가정하여 모노드들의 가중치를 균등하게 설정한 후 목표노드의 상태 값 작성에서 설명된 것과 동일한 방법으로 노드 확률 테이블을 작성하였다. 모노드의 가중치는 다음과 같다.

그룹노드의 모노드 이름	가중치
pre_survey	0.25
sw_system_req_review	0.25
design_review	0.25
sw_dev_review	0.25

o integration_validation 노드

모노드 2개의 중요성이 동일하다고 가정하여 모노드들의 가중치를 균등하게 설정하여 노드 확률 테이블을 작성하였다.

그룹노드의 모노드 이름	가중치
validation_review	0.5
hw_sw_integration	0.5

o support_quality 노드

모노드의 중요성을 고려하여 다음과 같이 가중치를 설정하여 노드 확률 테이블을 작성하였다.

그룹노드의 모노드 이름	가중치
release_report_review	0.3
maint_review	0.3
user_doc_review	0.4

o root 노드

상용 소프트웨어의 인정 프로세스 9단계에 해당하는 노드들과 운전이력 노드는 모노드가 없는 root 노드들이다. 이들 root 노드는 초기 확률 값을 가지고 있어야 하는데 초기 상태는 노드에 영향을 미치는 어떠한 증거도 없는 상태이므로 각 상태의 값을 아래와 같이 모두 동일하게 설정하였다.

상태	초기 값
good	0.5
bad	0.5

o 자노드를 가진 세부 검토 항목 노드들의 노드 확률 테이블

그림 3.3에 있는 "sw_design_review" 노드와 같이 자노드를 가진 세부검토 항목 노드의 각 상태 값은 아래와 같이 설정하였다.

모노드(design_review)상태		good	bad
sw_design_review	good	0.9	0.1
	bad	0.1	0.9

상태 값 "good"와 "bad"에 사용된 "0.9" "0.1"은 실제의 평가 케이스라면 전문가의 지식 또는 과거의 이력으로부터 추출해야 하나 여기서는 전문가의 정성적 판단을 백분율로 등급화(ranking)하여 정량화 하는데 사용되고 있는 Sherman Kent의 등급 척도를 이용하여 설정하였다[16].

표 3.2 Sheman Kent의 등급 척도

Order of Likelihood	Synonyms	Chances in 10	Percent
Nearly Certain	Virtually certain		99
	We are convinced	**9**	**90**
	Highly probable		
	Highly likely	8	80
Probable	Likely		
	We believe		
	We estimate	7	
	Chances are good		
	It is probable	6	60
Even Chance	Chances slightly better than even		
	Chances about even	5	
	Chances slightly less than even	4	40
Improbable	Probably not		
	Unlikely	3	
	We believe not	2	20
Nearly Impossible	**Almost impossible**		
	Only a slight chance	**1**	
	Highly doubtful		**10**

(3) 관찰가능 노드의 노드 확률 테이블

관찰 가능 노드의 상태 [Yes]와 [No]는 모두 측정 가능한 변수들이다. 이들 변수들은 정량적인 값으로 구해질 수도 있고 전문가의 판단에 의한 주관적 확률 값으로 측정될 수도 있다. 관찰 가능 노드들은 구체적인 형태로 측정 가능하고 모 노드와의 상관관계를 보다 정확하게 알 수 있으므로 중간 노드에 비해 보다 높은 정밀도의 확률을 부여할 수 있다. 전체 관찰 가능 노드들의 노드 확률은 해당 노드(인정 프로세스상의 세부 검토 항목)와 모노드의 상관관계가 어느 정도 명확한 가에 따라 다음과 같이 설정되었다.

o 상관관계가 가장 명확한 경우의 관찰가능노드 NPT

모노드 상태		good	bad
관찰가능 노드	yes	0.99	0.01
	no	0.01	0.99

o 상관관계가 명확한 경우의 관찰가능노드 NPT

모노드 상태		good	bad
관찰가능 노드	yes	0.9	0.1
	no	0.1	0.9

o 상관관계가 어느 정도 명확한 경우의 관찰가능노드 NPT

모노드 상태		good	bad
관찰가능 노드	yes	0.8	0.2
	no	0.2	0.8

o 상관관계가 어느 정도 추상적인 경우의 관찰가능노드 NPT

모노드 상태		good	bad
관찰가능 노드	yes	0.7	0.3
	no	0.3	0.7

여기에서 논의된 BBN 모델의 관찰가능 노드들은 거의 다 정성적인 것들이어서 [yes]/[no]와 같은 상태로 표현되었지만 정량적인 값(예를 들어 확률 등)을 구할 수 있는 관찰가능 노드들로 이루어진 BBN에서는 그 정량적인 값들을 사용하여 목표노드로부터 구체적인 고장확률과 같은 point value를 구할 수 있다.

예를 들면, PSA에서 사용되는 고장수목 분석법(FTA)과 같은 방식으로 BBN을 모델링 할 수 있는데 이렇게 하면 FTA에 사용된 기본사건의 값들을 BBN의 관찰가능 노드에 입력해서 FTA의 정점 사건의 값과 동일한 결과를 BBN의 목표 노드로부터 얻을 수 있다.

제 3 절 상용소프트웨어 인정 프로세스를 응용한 BBN

제 2 절에서와 같은 방법으로 만들어진 BBN을 이용하여 상용 소프트웨어의 신뢰도 값을 정량적으로 구하는 방법 외에, 이렇게 만들어진 BBN을 응용하여 아래와 같은 방식으로 상용 소프트웨어의 신뢰도 값을 정량적으로 계산하거나 또는 응용하는 것도 가능하다.

o 방법-A: 소프트웨어 개발회사에서 제시하는 신뢰도 값이 있을 경우 또는 상용 소프트웨어의 운전 이력으로부터 동 소프트웨어의 신뢰도 값을 추출할 수 있는 경우에 추출된 상용 소프트웨어 신뢰도 값의 확신도

(confidence)를 계산하는 용도로 인정 프로세스 BBN을 활용하는 방법.

o 방법-B. 상용 소프트웨어 인정 프로세스 BBN의 목표 노드를 제2절의 경우와 같이 하나의 특정한 신뢰도 값(두 개의 노드 상태)이 되는 확률을 구하는 방식이 아니라 몇 개의 구간 신뢰도 값(수 개의 노드 상태)으로 설정하여 인정 프로세스의 평가 결과에 따라 이들 구간 값 중 하나를 동 소프트웨어의 신뢰도 값으로 선택하는 방법.

1. 방법-A의 BBN

o 변수(노드)와 그래프

상용 소프트웨어 인정 프로세스 9단계를 기초로 한 기본 레벨 노드 9개와 그 하위 노드 그리고 그래프는 제 2절에서 작성된 기본 BBN과 동일하다. 또한 이들 9개 기본 레벨 노드를 그룹으로 분류하여 만든 3개의 그룹 노드 "development_process", "integration_validation", "support_quality"도 기본 BBN과 동일하다.

새로 만들어지는 노드(변수)는 "COTS_quality"로 이 변수는 상용 소프트웨어 인정 프로세스에 있는 모든 단계에서 검토되는 항목들의 평가 결과가 조합된 것으로 소프트웨어 개발 회사에서 제시하거나 또는 해당 소프트웨어의 운전이력에서 추출된 동 소프트웨어의 신뢰도 값을 어느 정도 믿을 수 있는가에 영향을 미치는 하나의 요소가 된다.

"operation_record_quality" 노드는 해당 소프트웨어의 (제시되거나 추출된) 신뢰도 수치에 대한 확신에 영향을 미치는 또 하나의 변수로서 기본 BBN 상의 "operation_history_record"와 동일하다.

최종적으로 만들어진 상위 레벨 BBN 그래프는 다음 그림과 같다.

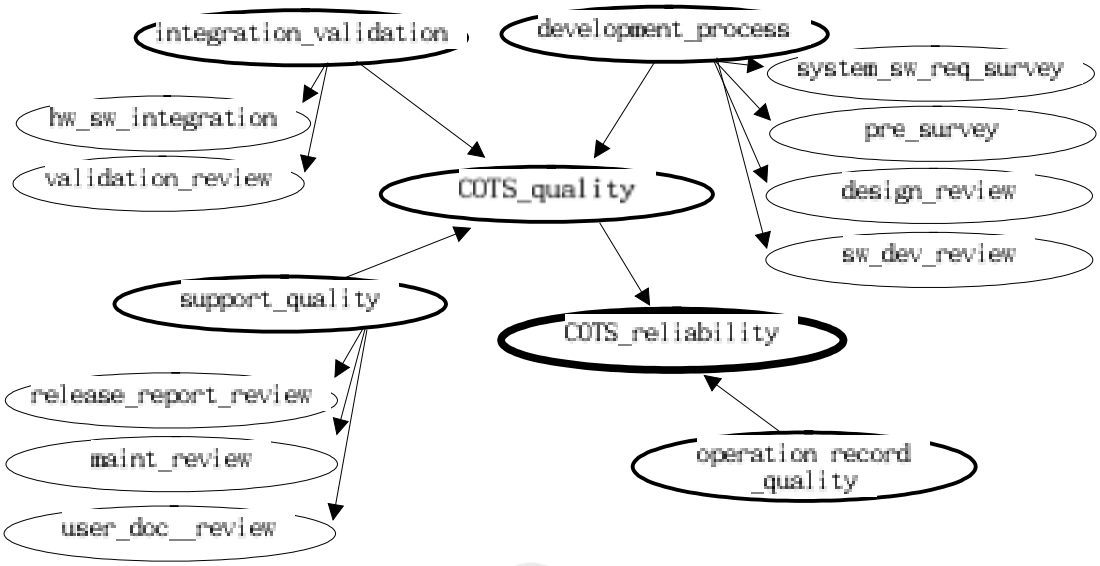


그림3-13. 운전이력으로부터 COTS 신뢰도 값을 구할 수 있을 경우의 BBN (방법-1 BBN)

o 노드 확률 테이블

목표 노드의 상태 값은 운전이력으로부터 추출된 해당 상용 소프트웨어의 신뢰도 값이거나 또는 소프트웨어 개발 회사에서 제시한 신뢰도 값이다. 여기서는 제 2절에서와 같이 만들어진 상용 소프트웨어의 인정 프로세스의 BBN을 이 신뢰도 값에 대한 믿음의 정도를 구하는 데 사용할 목적이므로 목표 노드의 모 노드는 "상용 소프트웨어 인정 프로세스의 평가 결과"와 "운전 이력 및 오류 관리의 품질" 두 가지로 설정하였으며 작성된 목표노드 "COTS_reliability"의 노드확률 테이블은 아래와 같다.

◇ 목표노드 "COTS_reliability"의 노드확률테이블

operation_record_correctness		good			bad		
COTS_quality		good	aver.	bad	good	aver.	bad
COTS_reliability	<=0.0001	0.9	0.7	0.6	0.4	0.3	0.1
	>0.0001	0.1	0.3	0.4	0.6	0.7	0.2

새로 만들어진 "COTS_quality" 노드는 운전 이력에 관련된 부분을 제외하고 상용 소프트웨어 인정 프로세스를 기본으로 만들어진 BBN의 최종 결과가 조합되는 노드로서 그룹 노드인 "development_process", "integration_validation", "support_quality"를 모노드로 가지며 설정된 노드확률테이블은 아래와 같다.

◇ COTS_quality 노드의 노드확률테이블

development_process		good				bad			
support_quality		good		bad		good		bad	
integration_validation		good	bad	good	bad	good	bad	good	bad
COTS_quality	good	0.8	0.5	0.5	0.1	0.5	0.1	0.1	0.0
	average	0.2	0.4	0.4	0.4	0.4	0.4	0.4	0.2
	bad	0.0	0.1	0.1	0.5	0.1	0.5	0.5	0.8

운전이력 및 오류관리를 의미하는 변수 "operation_record_quality" 노드는 소프트웨어 개발 회사에서 제시한 소프트웨어 신뢰도 값 또는 기록으로부터 추출된 해당 소프트웨어의 신뢰도 값이 있는 경우에 그 중요성이 커지므로 목표 노드의 모 노드로 설정되었다. "operation_record_quality" 노드는 모 노드가 없는 root 노드이며 각 상태의 값은 이 노드의 자노드들에 관찰된 값을 입력한 후 계산을 하면 변동하게 된다. 작성된 노드확률테이블은 아래와 같다.

◇ operation_record_quality 노드의 노드확률테이블(초기 상태)

상태	값
good	0.5
bad	0.5

2. 방법-2의 BBN

o 변수(노드)와 그래프

목표 변수(노드)를 제외한 모든 변수와 그래프 그리고 노드확률테이블은 기본 BBN과 동일하다. 그러나 여기서는 상용 소프트웨어 인정 프로세스 각 단계에 있는 세부 검토 항목들의 평가 결과를 제 2절에서 작성된 기본 BBN의 경우와 같이 특정 신뢰도 값이 될 확률을 구하는 것이 아니라 해당 소프트웨어가 어떤 신뢰도 값(구간 값)을 가질 것인가를 구하는 것이 목적이므로 목표 노드의 노드확률테이블은 아래와 같은 형태로 만들어진다.

◇ 방법-2에서 목표 노드의 노드확률테이블

operation_history_record		good							
development_process		good				bad			
support_quality		good		bad		good		bad	
integration_validation		good	bad	good	bad	good	bad	good	bad
COTS_ acceptance	<0,0,00001]	0,7	0,1	0,1	0,0	0,1	0,0	0,0	0,0
	<0,00001,0,0001]	0,25	0,45	0,45	0,2	0,45	0,2	0,2	0,0
	<0,0001,0,001]	0,04	0,35	0,35	0,25	0,35	0,25	0,25	0,1
	<0,001,0,01]	0,01	0,1	0,1	0,4	0,1	0,4	0,4	0,2
	<0,01,0,1]	0,0	0,0	0,0	0,15	0,0	0,15	0,15	0,6
	<0,1,1]	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1

operation_history_record		bad							
development_process		good				bad			
support_quality		good		bad		good		bad	
integration_validation		good	bad	good	bad	good	bad	good	bad
COTS_ acceptance	<0,0,00001]	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	<0,00001,0,0001]	0,45	0,2	0,2	0,0	0,2	0,0	0,0	0,0
	<0,0001,0,001]	0,35	0,25	0,25	0,1	0,25	0,1	0,1	0,0
	<0,001,0,01]	0,1	0,4	0,4	0,2	0,4	0,2	0,2	0,0
	<0,01,0,1]	0,0	0,15	0,15	0,6	0,15	0,6	0,6	0,1
	<0,1,1]	0,0	0,0	0,0	0,1	0,0	0,1	0,1	0,9

본 보고서에서 기술된 상용 소프트웨어 인정 프로세스 뿐 아니라 여타의 소프트웨어 개발 방법이나 품질 등과 같은 정성적인 내용의 평가 결과와 그 소프트웨어의 정량적인 신뢰도 값(여기서는 목표노드의 상태(구간 값) "<0.1,1]" .. "<0.0, 0.00001]")의 상관관계를 파악하는 것은 어렵다. 그 이유를 보면, 지금까지는 특정한 개발 과정(또는 방법론)을 사용하여 만들어진 소프트웨어가 실제 운전 상황에서 운영되었을 때의 경험적 정량적 신뢰도 값이 발표된 적이 드물고 또 실제 운전 상황으로부터 나온 특정 소프트웨어어의 신뢰도 값이 있더라도 소프트웨어의 운전 상황이 달라지면 그 값을 그대로 채용하는데 불확실성이 내포되기 때문이다. 그러나 적절한 소프트웨어 개발 방법론을 준수하여 소프트웨어를 개발하면 그 소프트웨어는 각 산업이나 응용분야에서 요구하는 신뢰도 요구사항을 만족할 수 있다는 것이 소프트웨어 신뢰도에 대한 일반적 인식이고 원자력 산업과 항공산업을 비롯한 각 산업 분야의 국제 표준이나 규제 기관들도 기본적으로는 이런 방침을 취하고 있다. 이 점은 소프트웨어의 정성적인 평가 결과(예를 들면 개발 방법, 확인 및 검증, 품질 등에 대한 평가 결과)와 그 소프트웨어의 정량적 신뢰도 값의 상관관계를 정량적으로 명확하게 알기는 힘들지만

관계가 밀접하다는 것을 암시적으로 인정하고 있다고 볼 수 있다. 따라서 소프트웨어의 정성적인 평가 결과를 어떤 형태로든 정량적인 신뢰도 값과 연결시켜 이를 분석하는 것은 의미가 있으며 방법-2에서와 같은 형태로 만들어진 노드확률테이블과 그 계산 결과는 다른 평가 방법들과 더불어 의사결정의 근거가 되는 하나의 자료로 유용하다고 보여진다.

제 4 장 원전 상용소프트웨어 BBN을 이용한 계산

BBN 그래프를 완성하고 노드 확률 테이블의 정의가 완료되면 각 노드에 관찰된 값을 입력하여 구하고자 하는 목표 노드의 값을 계산하게 된다. BBN 모델의 일차적인 목적은 관찰 및 획득된 증거에 근거한 목표 노드의 값을 구하는 것이지만 BBN이 가진 특성과 확률 계산을 자동으로 해 주는 HUGIN[13]과 같은 BBN 도구를 이용하면 What if 분석이나 여러 가지 시나리오 분석이 가능하므로 이를 활용하여 평가 뿐 아니라 소프트웨어 생명 주기의 모든 단계에서 최적화 문제를 비롯한 여러 가지 문제의 해결에 사용될 수 있다[15].

본 장에서는 제 3 장에서와 같이 만들어진 상용소프트웨어 인정 프로세스 BBN에 여러 가지 입력(증거) 조합을 가정한 시나리오를 만들어 계산한 결과다. 본 장의 계산 결과에서 나타난 목표 노드의 값은 실제적인 증거에 근거하여 계산된 것이 아니고 또 노드 확률 테이블의 확률을 정의하는데 전문가의 지식이 반영되지 않았기 때문에 계산된 결과 값에 대해 실질적인 의미를 부여할 수는 없다. 그러나 여러 경우에 대한 시나리오 분석의 결과 만들어진 BBN모델은 전문가의 판단과정을 유사하게 모의하고 있고 또 시나리오 별 계산 결과도 일반적으로 예상되는 결과와 유사하게 나타난 것을 알 수 있다.

제 1 절. 시나리오

제 3장에서와 같이 만들어진 "상용 소프트웨어 인정 프로세스 BBN" "방법-1 BBN" "방법-2 BBN" 3 가지의 BBN에 대하여 다음과 같은 시나리오를 작성하고 시나리오별 입력 조건을 사용하여 그 결과를 계산하였다.

o 상용 소프트웨어 인정 프로세스 BBN(기본 BBN)의 시나리오

A. 초기 상태: 어떠한 관찰 값도 입력되지 않은 초기 상태의 BBN
B. 최적 경우: 모든 측정 가능 노드의 "yes" 상태를 100%로 설정
C. 최악 경우: 모든 측정 가능 노드의 "no" 상태를 100%로 설정
D. 측정 가능 노드에 관찰된 값을 사용한 경우
   전체 관찰가능 노드 61개의 10%에 해당하는 6개까지의 노드에 부정적 값이

관찰되어 이들 노드의 "no" 상태를 100%로 설정.

D-1. 1개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

D-2. 2개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

D-3. 3개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

D-4. 4 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

D-5. 5 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

D-6. 6 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

위의 시나리오 별로 목표노드와 목표노드의 모노드인 "development_process" "operation_history_record" "support_quality" "support_quality" 노드의 값을 계산하였다.

o 방법-1 BBN의 시나리오

방법-1 BBN에서는 소프트웨어 개발 회사에서 제시되거나 또는 운전이력으로부터 추출된 해당 상용 소프트웨어 신뢰도 수치의 이력 관리에 대한 평가(operation_record_quality 노드) 결과가 상용 소프트웨어 인정 프로세스 상의 전체 평가 결과(COTS_quality)와 대등한 중요성을 가지는 것으로 설정되었다.

그러나 operation_record_quality 노드는 COTS_quality 노드에 비해 훨씬 적은 수의 세부 평가 항목을 가지고 있으므로 operation_record_quality 노드의 자노드에 해당하는 세부 평가 항목들은 인정 프로세스 상의 세부 평가 항목들보다 목표 노드에 더 큰 영향을 준다. 이런 이유로 상용 소프트웨어 인정 프로세스 상의 전체 평가 결과(COTS_quality)는 3가지 상태 "good, average, bad"에 대해서만 시나리오를 설정하고 "COTS_quality" 노드 3가지 상태별로 operation_record_quality의 자노드 입력 조건들을 아래와 같이 설정하였다.

A. COTS_quality 노드가 "good"인 경우 & operation_record_quality 자노드의 일부(1개부터 5개까지)에 부정적 관찰 증거가 입력된 경우

B. COTS_quality 노드가 "average"인 경우 & operation_record_quality 자노드의 일부(1개부터 5개까지)에 부정적 관찰 증거가 입력된 경우

C. COTS_quality 노드가 "bad"인 경우 & operation_record_quality 자노드의 일

부(1개부터 5개까지)에 부정적 관찰 증거가 입력된 경우

o 방법-2 BBN의 시나리오

방법-2 BBN의 계산 시나리오는 상용 소프트웨어 인정 프로세스 BBN(기본 BBN)의 시나리오와 동일하다.

제 2 절 계산 결과

1. 상용소프트웨어 인정 프로세스 BBN(기본 BBN)

o 초기 상태 계산 결과

BBN의 초기 상태는 어떠한 관찰 값도 입력되지 않은 상태이며 따라서 목표 노드의 계산 결과는 아래의 표4.1과 같이 노드의 각 상태가 동일한 확률을 가지는 가장 불확실한 상황(0.5:0.5)을 보여준다.

표 4.1 목표 노드의 초기 상태 계산 결과

상태	COTS_acceptance
good	0.5
bad	0.5

상위 레벨들의 초기 상태 계산 결과도 목표 노드의 계산 결과와 마찬가지로 노드 각 상태의 확률 값이 동일하며 따라서 가장 불확실한 상황이다.

표 4.2 상위 레벨 노드들의 초기 상태 계산 결과

상태	development_process	operation_history_record	support_quality	integration_validation
good	0.5	0.5	0.5	0.5
bad	0.5	0.5	0.5	0.5

o Best case와 Worst case 계산 결과

◇ 최적 경우(best case): 모든 측정 가능 노드의 "yes" 상태를 100%로 설정

모든 측정 가능 노드의 입력 값을 "yes=1" 상태로 한 경우이며 상용 소프트웨어 인정 프로세스 각 단계의 검토 항목들을 검토한 결과 모두 만족스러운 평가 결과가 나온 경우이다. 이 경우 목표 노드의 계산 결과는 아래의 표4.3과 같다.

표 4.3 최적 경우 목표 노드 "COTS_acceptance"의 계산 결과

상태	COTS_acceptance
good	0.9966
bad	0.0034

모든 측정 가능 노드의 입력 값을 "no=1" 상태로 한 경우의 상위 레벨 노드들의 계산 결과는 아래의 표 4.4와 같다.

표 4.4 최적 경우 상위 레벨 노드들의 계산 결과

상태	development_process	operation_history_record	support_quality	integration_validation
good	0.9941	1.0000	0.9983	0.9938
bad	0.0059	0.0000	0.0017	0.0062

◇ 최악 경우(worst case): 모든 측정 가능 노드의 "no" 상태를 100%로 설정

모든 측정 가능 노드의 입력 값을 "no=1" 상태로 한 경우이며 상용 소프트웨어 인정 프로세스 각 단계의 검토 항목들을 검토한 결과 모두 만족스럽지 못한 평가 결과가 나온 경우이다. 이 경우 목표 노드의 계산 결과는 아래의 표4.5와 같다.

표 4.5 최악 경우 목표 노드 "COTS_acceptance"의 계산 결과

상태	CPTS_acceptance
good	0.0034
bad	0.9966

모든 측정 가능 노드의 입력 값을 "no=1" 상태로 한 경우의 상위 레벨 노드들의 계산 결과는 아래의 표 4.6과 같다.

표 4.6 최악 경우 상위 레벨 노드들의 계산 결과

상태	development_process	operation_history_record	support_quality	integration_validation
good	0.0059	0.0000	0.0017	0.0062
bad	0.9941	1.0000	0.9983	0.9938

o What if 시나리오 계산 결과

전체 관찰가능 노드 61개의 약 10%에 해당하는 6개까지의 노드에 부정적 값이 관찰되었다고 가정하고, 또 부정적 값이 입력되는 노드를 하나의 기본레벨(상용 소프트웨어 인정 프로세스의 어느 한 단계) 노드에 한정시킬 경우와 6개의 기본 노드에 각기 나누어 할 경우 두 가지를 설정하여 계산한 결과이다.

D-1: 1개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-2: 2개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-3: 3개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-4: 4 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-5: 5 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-6: 6 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

표 4.7 하나의 기본 레벨(user_doc_review)에 시나리오 D-1, D-2, D-3, D-4, D-5, D-6을 적용한 경우 목표노드 "COTS_acceptance" 계산 결과

D 시나리오	COTS_acceptance	
	good	bad
D-1	0.9966	0.0034
D-2	0.9962	0.0038
D-3	0.9944	0.0056
D-4	0.9407	0.0593
D-5	0.9225	0.0775
D-6	0.9178	0.0822

표 4.8 각기 다른 기본 레벨 노드에 D-1, D-2, D-3, D-4, D-5, D-6을 한 개씩 적용한 경우의 목표노드 "COTS_acceptance" 계산 결과

D 시나리오	COTS_acceptance	
	good	bad
D-1	0.9960	0.0040
D-2	0.9958	0.0042
D-3	0.9958	0.0042
D-4	0.9931	0.0069
D-5	0.9587	0.0413
D-6	0.9521	0.0479

2. 방법-1 BBN

"COTS_quality" 노드의 세 가지 상태 "good" "average" "bad" 각각에 대하여 "operation_record_quality" 자노드들의 입력 조건 D-1..D-6 경우를 계산한 결과는 표4.9 표4.10 표4.11과 같다.

표 4.9 "COTS_quality" 노드의 상태가 "good"인 경우의 목표 노드 계산 결과
(COTS_quality 노드의 상태 입력 값:good=0.8, average=0.2, bad=0.0)

D 시나리오	COTS_reliability	
	accept	reject
D-1	0.8880	0.1120
D-2	0.8874	0.1126
D-3	0.8418	0.1582
D-4	0.5802	0.4198
D-5	0.4434	0.5566
D-6	0.3947	0.6053

표4.10 "COTS_quality" 노드의 상태가 "average"인 경우의 목표 노드 계산 결과
(COTS_quality 노드의 상태 입력 값:average=0.8, good=0.1, bad=0.1)

D 시나리오	COTS_reliability	
	accept	reject
D-1	0.7657	0.2343
D-2	0.7652	0.2348
D-3	0.7252	0.2748
D-4	0.4960	0.5040
D-5	0.3761	0.6239
D-6	0.3334	0.6666

표 4.11 "COTS_quality" 노드의 상태가 "bad"인 경우의 목표 노드 계산 결과
(COTS_quality 노드의 상태 입력 값:bad=0.8, average=0.2, good=0.0)

D 시나리오	COTS_reliability	
	accept	reject
D-1	0.6998	0.3002
D-2	0.6993	0.3007
D-3	0.6624	0.3376
D-4	0.4505	0.5495
D-5	0.3397	0.6603
D-6	0.3002	0.6998

3. 방법-2 BBN

o 초기 상태 계산 결과

어떠한 관찰 값도 입력되지 않은 초기 상태의 목표 노드 계산 결과는 아래의 표 4.12와 같다.

표 4.12 목표 노드의 초기 상태 계산 결과

목표 노드		COTS_acceptance 값
COTS_ acceptance 상태	<0,0.00001]	0.0688
	<0.00001,0.0001]	0.2031
	<0.0001,0.001]	0.2087
	<0.001,0.01]	0.2256
	<0.01,0.1]	0.2125
	<0.1,1]	0.0812

목표 노드의 모노드인 상위 레벨 노드들의 초기 상태 계산 결과는 기본 BBN의 계산 결과와 동일하다.

o Best case와 Worst case 계산 결과

◇ 최적 경우(best case): 모든 측정 가능 노드의 "yes" 상태를 100%로 설정

모든 측정 가능 노드의 평가 결과가 만족스러운 상태일 경우 목표 노드의 계산 결과는 아래의 표 4.13과 같다.

표 4.13 최적 경우 목표 노드 "COTS_acceptance"의 계산 결과

목표 노드		COTS_acceptance 값
COTS_ acceptance 상태	<0, 0, 0000 1]	0, 6921
	<0, 00001, 0, 0001]	0, 2526
	<0, 0001, 0, 001]	0, 0441
	<0, 001, 0, 01]	0, 0112
	<0, 01, 0, 1]	0, 0000
	<0, 1, 1]	0, 0000

목표 노드의 모노드인 상위 레벨 노드들의 최적 경우 계산 결과는 기본 BBN의 계산 결과와 동일하다.

◇ 최악 경우(worst case): 모든 측정 가능 노드의 "no" 상태를 100%로 설정

모든 측정 가능 노드의 평가 결과가 만족스럽지 못할 상태일 경우 목표 노드의 계산 결과는 아래의 표 4.13과 같다.

표 4.14 최악 경우 목표 노드 "COTS_acceptance"의 계산 결과

목표 노드		COTS_acceptance 값
COTS_ acceptance 상태	<0, 0, 0000 1]	0, 0000
	<0, 00001, 0, 0001]	0, 0000
	<0, 0001, 0, 001]	0, 0014
	<0, 001, 0, 01]	0, 0028
	<0, 01, 0, 1]	0, 1069
	<0, 1, 1]	0, 8890

목표 노드의 모노드인 상위 레벨 노드들의 최악 경우 계산 결과는 기본 BBN과 동일하다.

o What if 시나리오 계산 결과

전체 관찰가능 노드 61개의 약 10%에 해당하는 6개까지의 노드에 부정적 값이 관찰되었다고 가정하여 이를 적용하며, 동시에 부정적 값이 입력되는 노드를 하나의 기본레벨 노드에 한정시킬 경우와 6개의 기본 노드에 각기 나누어 할 경우.

D-1: 1개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-2: 2개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-3: 3개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-4: 4 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-5: 5 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우
D-6: 6 개의 측정 가능 노드에만 부정적 값(no 상태가 100%)이 사용된 경우

하나의 기본 레벨에 시나리오 D-1, D-2, D-3, D-4, D-5, D-6을 적용한 경우 목표노드 "COTS_acceptance" 계산 결과는 아래의 표4.15와 같다.

표 4.15 하나의 기본 레벨에 적용한 경우 목표노드 "COTS_acceptance" 계산 결과

목표 노드: COTS_acceptance		시나리오 D의 경우 목표노드 값					
		D-1	D-2	D-3	D-4	D-5	D-6
상태	<0, 0, 00001]	0.6919	0.6908	0.6854	0.5257	0.4718	0.4578
	<0, 00001, 0, 0001]	0.2527	0.2530	0.2548	0.3072	0.3249	0.3294
	<0, 0001, 0, 001]	0.0442	0.0447	0.0475	0.1295	0.1573	0.1644
	<0, 001, 0, 01]	0.0112	0.0114	0.0122	0.0371	0.0455	0.0476
	<0, 01, 0, 1]	0.0000	0.0000	0.0000	0.0005	0.0007	0.0007
	<0, 1, 1]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

각기 다른 기본 레벨에 시나리오 D-1, D-2, D-3, D-4, D-5, D-6을 적용한 경우 목표노드 "COTS_acceptance" 계산 결과는 아래의 표4.15와 같다.

표 4.16 각기 다른 기본 레벨에 적용한 경우의 목표노드 "COTS_acceptance" 계산 결과

목표 노드: COTS_acceptance		시나리오 D의 경우 목표노드 값					
		D-1	D-2	D-3	D-4	D-5	D-6
상태	<0, 0, 00001]	0, 6919	0, 6915	0, 6915	0, 6837	0, 5820	0, 5720
	<0, 00001, 0, 0001]	0, 2527	0, 2528	0, 2528	0, 2554	0, 2883	0, 2905
	<0, 0001, 0, 001]	0, 0442	0, 0444	0, 0444	0, 0484	0, 1004	0, 1050
	<0, 001, 0, 01]	0, 0112	0, 0113	0, 0113	0, 0125	0, 0288	0, 0314
	<0, 01, 0, 1]	0, 0000	0, 0000	0, 0000	0, 0000	0, 0006	0, 0012
	<0, 1, 1]	0, 0000	0, 0000	0, 0000	0, 0000	0, 0000	0, 0000

제 3 절 BBN 작성 및 계산에 대한 논의

◇ 일반적으로 기본 레벨에 속한 자노드의 수(본 case study의 경우에는 상용 소프트웨어 인정 프로세스 상의 상세 검토 항목의 수)가 목표노드의 계산 결과에 영향을 미치는 것으로 나타났다. 그 원인은 상용소프트웨어 인정 프로세스 각 단계의 중요성을 거의 동일한 것으로 가정하여 노드 확률 테이블을 작성한 것에 주로 기인한다. 그러나 어떤 특정 단계에 속한 정성적인 평가 항목 수와 그 단계의 중요성이 항상 비례 관계가 있는 것은 아니다.

따라서 정성적인 평가 절차를 BBN으로 모델링 할 때는 그 중요성의 정도에 따라 각 기본 레벨의 가중치를 적절하게 정하는 것과 또 세부 검토 항목들을 BBN의 변수로 변환할 때 변수의 개수를 적절하게 조정하는 것이 중요한 것으로 보여진다.

◇ 부정적 관찰 결과를 한 개의 기본 레벨 노드에 적용한 결과와 여러 개의 기본 레벨 노드에 분산하여 적용한 계산 결과에서 나타난 사항은 다음과 같다.

- o 부정적 관찰 결과를 각 기본 레벨 노드에 분산하여 적용할 경우에는 목표 노드의 부정적 상태 값(not accept) 값에 큰 변화가 없다.
- o 그러나 부정적 관찰 결과를 하나의 기본 레벨 노드에 적용할 경우에는 목표 노드의 부정적 상태 값(not accept) 값에 상대적으로 큰 변화가 생긴다. 특히 부정적 값의 입력 대상이 되는 기본 레벨의 전체 검토 항목 수의 절반을 초과하는 수의 노드에 부정적 관찰 결과를 입력했을 때는 목표 노드의 부정적 상태(not accept) 값이 급격하게 높아진다.

본 case study의 시나리오 같이 실제의 상황에서도 특정 단계에 집중적으로

부정적 평가 결과들이 나올 경우와 각 단계에 분산되어 부정적 평가 결과들이 나올 수 있다. 제 3장에서와 같은 방식으로 만들어진 BBN에서는 부정적 평가 결과가 나온 세부 검토 항목의 개수가 같더라도 그 항목들이 하나의 기본 단계에 집중되어 있는가 아니면 각기 다른 단계에 분산되어 있는가에 따라 최종 평가(목표 변수의 상태 값)가 달라진다. 따라서 평가 대상이 되는 시스템(상용 소프트웨어)의 특성을 고려하여 사전에 각 노드의 조건부 확률(노드 사이의 관계 강도)을 조정하는 것이 필요하다고 보여진다.

◇ 각 기본 레벨에 속한 세부 검토 항목의 일부(1개)에 부정적 결과를 적용한 경우에는 목표 노드의 긍정적 상태 "accept" 값이 best case와 차이가 없는 것으로 나타났다. 이것은 실제로 정성적 평가 절차와 기준을 사용하여 전문가가 최종 판단을 내리는 것과 유사하게 보여진다.

◇ 하나의 모노드에 속한 여러 개의 자노드들 중 어느 하나의 특정 자노드가 다른 자노드들보다 모노드와의 관계가 강할 경우(예를 들면, 특정 노드의 NPT는 {0.99, 0.01}이고 다른 노드들의 NPT {0.8,02})에는 그 특정 자노드에 입력되는 값이 다른 자노드들의 입력 값에 비해 월등하게 모노드의 상태 값에 영향을 미치는데 이것은 BBN의 계산 특성에 기인한다. 따라서 한 모노드에 속한 여러 개의 자노드에 대한 NPT를 작성할 경우에는 정성적인 판단과 유사한 결과가 나올 때까지 각 자노드들에 대하여 몇 번의 NPT 수치(조건부 확률 값) 조정을 거치는 것이 필요한 것으로 보여진다.

◇ 제 2장에서 기술된 상용 소프트웨어 인정 프로세스 상의 세부 검토 항목들은 정성적인 평가에서는 모두 사용된다. 그러나 이들 중에는 평가 대상이 되는 상용 소프트웨어의 신뢰도에는 직접적으로 관련이 없는 항목들도 일부 있다(예: security 항목 등). 본 case study의 BBN에서는 이들 항목에 낮은 조건부 확률을 부여하여 포함시켰으나 실제적인 평가를 위한 BBN에서는 직접적으로 관련이 없는 평가 항목들은 제외시키고 정성적 평가 절차에는 없지만 신뢰도와 관련이 있다고 보여지는 항목들은 도출해서 추가하는 것이 필요하다고 보여진다.

제 5 장 결론 및 추후 연구내용

원자력발전소의 안전성 평가를 위한 중요한 수단으로 사용되고 있는 확률론적 안전성 평가에 안전 계통 디지털 시스템을 포함시켜야 되는 현실적인 요구가 있고 이를 위해서는 안전 소프트웨어의 정량적인 신뢰도 평가가 필요하다. 그러나 시험이나 신뢰도 성장 모델과 같은 기존의 정량적 소프트웨어 신뢰도 평가 방법 단독으로는 원전 안전 계통 디지털 시스템에 사용되는 소프트웨어의 신뢰도를 구하기 어려운 것이 현재의 기술 수준이며 따라서 원자력분야를 비롯한 타 산업 분야의 안전성/신뢰도 관련 표준들이나 각 국의 규제 기관들은 규칙기반의 정성적 평가 방식을 따르고 있다.

본 보고서에서는 안전 소프트웨어의 특성으로 인하여 만족할 만한 새로운 정량적 신뢰도 평가 방법이 가까운 장래에 나오기 어렵지만 디지털 시스템의 확률론적 안전성 평가와 같은 현실적인 필요성은 당장 대두되고 있는 현재의 상황에서, 하나의 대안으로서 기존에 채택되고 있는 소프트웨어의 정성적인 평가 방법을 Bayesian Belief Net 방법론을 이용하여 정형적으로 모델링하고 PSA에서 요구하는 정량화 된 결과를 구하는 방안을 원전 상용소프트웨어 인정 프로세스를 시험케이스로 적용하여 논의하였다.

당초 정성적 평가 절차를 기반으로 하여 BBN을 구축한 후 정량적 평가 결과를 구하는 방법을 시도한 이유는 다음과 같은 유용한 점들이 있기 때문이었다.

- o 신뢰도 평가에 관련된 다양한 증거들(과정, 제품 정보, 경험적 자료, 전문가의 판단, 불완전한 정보 등)을 일관된 평가체제 안에서 정형적으로 결합하여 정량적 결론(확률)을 추론할 수 있고
- o 모델의 직관적 그래프 형태가 평가에 관련된 복잡한 연관 관계와 감추어진 가정들을 명시적으로 나타내어 결론이 도출되는 과정에 대한 투명도와 감사도(auditability)를 높일 수 있으며
- o 불확실하거나 애매한 증거들이 필연적으로 포함되는 신뢰도 평가에 있어서 "what if" 분석을 가능하게 해주므로 의사결정에 있어 효과적인 도구로 사용될 수 있다.

본 보고서에서 논의된 BBN의 변수와 그래프는 기존의 상용 소프트웨어 인정

프로세스를 기반으로 하여 작성되었기 때문에 어느 정도 그 타당성이 인정되나 각 변수(노드)들의 노드 확률 테이블은 임의로 작성되었기 때문에 BBN 계산 결과에 대해 실용성을 부여하기는 힘들다. 그러나 기존의 정성적 평가 절차를 사용하여 BBN을 구축하고 시나리오를 만들어 결과를 계산하고 분석해 봄으로서 정성적 평가 절차를 정량화 시키는데 발생할 수 있는 여러 가지 점들을 발견할 수 있었고 또 어려운 문제점들이 어떤 것인지를 도출할 수 있었으며 위에서 기술된 BBN의 유용성들을 확인할 수 있었다.

BBN을 사용하여 정성적인 소프트웨어 평가 절차로부터 정량적인 평가 결과를 추론하기 위해 앞으로 연구되어야 할 것으로 보여지는 중요한 사항들은:

(1) 만들어진 모델이 기존 규칙 기반의 정성적 평가 시스템을 적절하게 반영했는가에 대한 검증 문제와

(2) BBN을 적용하는 대상들은 정량화 된 값을 구하기 어려운 불확실성이 많이 포함된다는 특성으로 인해 변수(노드) 간의 연결 강도를 나타내는 노드 확률 테이블을 전문가의 판단으로부터 확률 형태로 추출하여 정의하는데 따르는 문제점이다.

이런 문제점들은 BBN 기반 방법론의 공통적인 문제점으로 원자력분야를 비롯해서 항공분야나 군수분야에서 디지털 시스템의 안전성 평가의 일부로 이에 대한 연구가 현재 진행 중에 있다[15].

참고문헌

[1] T. Sung, H.G. Kang, "Intermediate Probabilistic Safety Assessment Approach for Safety Critical Digital Systems," Proceeding of ICON9, Nice, France, 2001.

[2] B. Littlewood and L. Strigini, Validation of Ultrahigh Dependability for Software-Based Systems, Communication of the ACM, 36(11), 1993

[3] 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

[4] N.E. Fenton and M. Neil, A Critique of Software Defect Prediction Models, 25(5) IEEE Transactions on Software Engineering, 1999

[5] NUREG-0800, "Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants," (SRP). The updated SRP Chapter7

[6] 차세대 원자로 안전규제 요건, 한국원자력안전기술원, 1999

[7] 경수로형 원전 안전심사지침, 한국원자력안전기술원, 1998

[8] 김장열 외, 공급자 조사 방법에 의한 원전 상용소프트웨어 인정 프로세스, 한국원자력학회 추계학술발표회, 2000.

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

[10] G. Dahll et al, "The Use of Bayesian Belief Bets in Safety Assessment of Software Based Systems," Int. J. General Systems, Vol. 29(2), pp 205-229, 2000.

[11] SERENE, ESPRIT Project 22187, SERENE Method Manual, http://www.dcs.qmw.ac.uk/~norman/SERENE_Help/start.htm, 1999

[12] R.M. Cooke, Experts in Uncertainty, Opinion and Subjective Probability in Science, Oxford University Press, 1991

[13] HUGIN Expert A/S., http://www.hugin.dk

[14] 강현국 외, PSA를 이용한 원전 안전계통 소프트웨어 시험횟수 결정, 한국원자력학회 추계학술발표회, 2000.

[15] 엄흥섭 외, 원전 안전 소프트웨어의 정량적 신뢰도 평가를 위한 Bayesian Belief Nets 기술 분석, KAERI/AR-594/2001, 한국원자력연구소, 2001

[16] M. Myer and J. Booker, Eliciting ad Analyzing Expert Judgement, A Practical Guide, Knowledge Based Systems Vol.5, Academic Press, 1991

[17] G. Dahll, The use of Bayesian Belief Nets is Safety Assessment of Software based Systems, HWP-527, Halden Project, 1998

부록 1. 상용 소프트웨어 인정 프로세스 BBN과 응용 BBN 전체 그래프

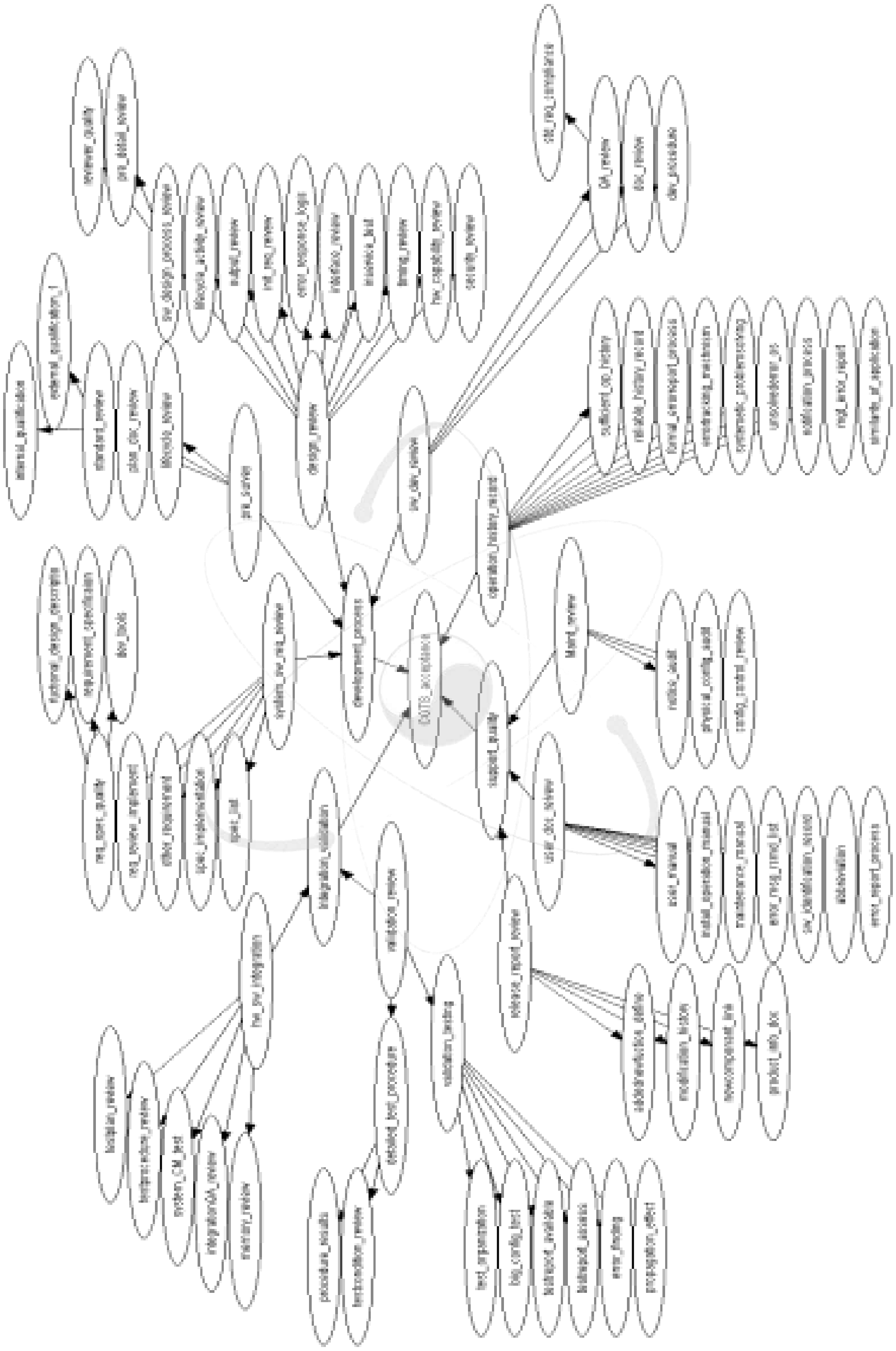


그림 A-1. 상용 소프트웨어 인정 프로세서 전체 BBN 그래프

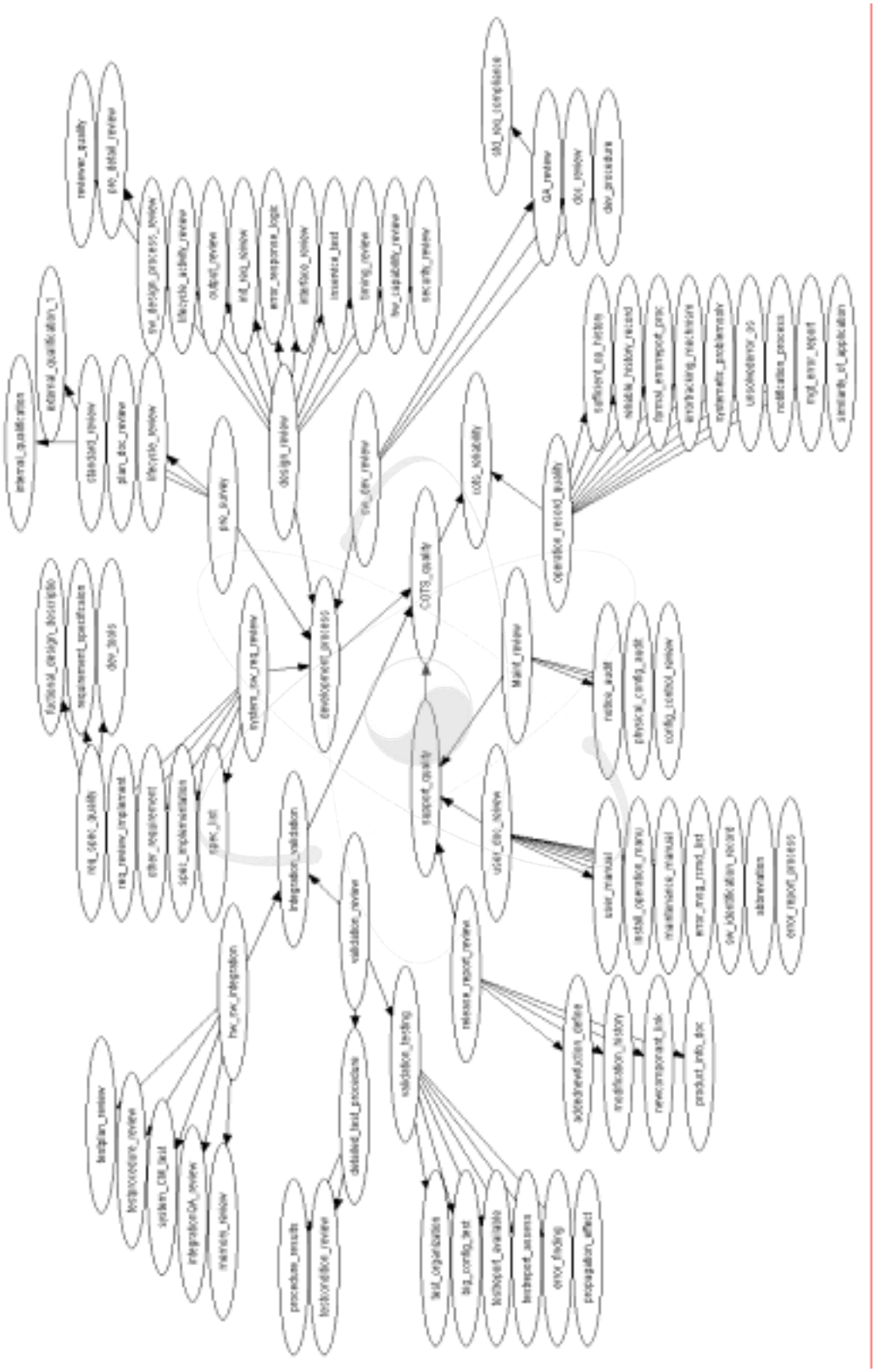


그림 A-2. 방법-1 전체 BBN 그래프

부록 2. 상용소프트웨어 인정프로세스의 질문 목록

질문 목록 1 : 조사 전 검토

노드 이름 : pre_survey

상세 노드 및 질문 목록

노드 이름	노드 설명
plan_doc_review	계획 문서(planning documentation)의 존재 여부 및 적절성
lifecycle_review	생명주기 모델 및 단계별 활동 사항 검토
standard_review	ISO 9001 Part 3과 같은 국제 표준 준수여부 조사
o internal_qualification	o 자체 qualification 여부
o external_qualification	o 외부조직에 의한 requalification 여부

질문 목록 2 : 제품설명서 검토

노드 이름 : release_report_review

상세 노드 및 질문 목록

노드 이름	노드 설명
addednewfunction_define	새로운 기능의 추가 및 변경 정의
modification_history	개정 레벨, 최소한의 하드웨어 요건 및 개정 이전에 확인된 오류의 교정을 포함한 소프트웨어 개정 이력
product_info_doc	하드웨어 및 소프트웨어의 제품 설명자료 취득
newcomponant_link	새로운 콤포넌트에 대한 소프트웨어 개발 프로세스와의 링크 여부

질문 목록 3 : 시스템 및 소프트웨어 기능 요건 조사

노드 이름 : system_sw_req_review

상세 노드 및 질문 목록

노드 이름	노드 설명
req_spec_quality o fuctional_design_description o requirement_specification o dev_tools	시스템 및 소프트웨어 기능요건 명세서 적절성 o 기능 설명서 조사 o 요구사항 명세서 조사 o CASE를 비롯한 개발 도구 사용 여부
spec_list	기능요건 명세 목록의 존재여부 및 적절성
spec_implementation	제품에의 기능 요건 반영 여부
other_requirements	기타 요구사항의 존재 유무
req_review_implement	소프트웨어 요구사양 명세서의 검토 수행 여부

질문 목록 4 : 소프트웨어 설계 조사

노드 이름 : design_review

상세 노드 및 질문 목록

노드 이름	노드 설명
sw_design_process_review o pre_detail_review o reviewer_quality	소프트웨어 설계 프로세스 검토 o 예비 및 상세 설계의 검토 여부 o 예비 및 상세 설계의 검토자 자격 요건
lifecycle_activity_review	소프트웨어 설계, 구현, 통합, 테스팅 및 최종 사용에 대한 검토
output_review	범위, 정확도 및 갱신구간을 포함한 출력물 검토
init_req_review	초기와 요건 검토
error_response_logic	컴퓨터 시스템에 탑지된 고장에 반응하는 프로그램 로직의 검토
interface_review	오퍼레이터 인터페이스 검토
inservice_review	인서비스(in-service) 테스트 특성 및 진단의 검토
timing_review	전반적인 컴퓨터 시스템 응답시간을 포함한 시간 요건 검토
hw_capability_review	하드웨어 성능과 일치하는 프로세싱 유휴시간 및 excess memory 검토
security_review	보안성 요건 검토

질문 목록 5 : 소프트웨어 개발에 대한 조사

노드 이름 : sw_dev_review

상세 노드 및 질문 목록

노드 이름	노드 설명
QA_review	소프트웨어 품질 계획의 검토
o std_req_compliance	o 산업체 표준(ISO 9000-3) 요건의 준수 여부
document_review	소프트웨어 개발 문서들에 대한 검토
dev_procedure	소프트웨어 개발 절차 검토

질문 목록 6 : 하드웨어 및 소프트웨어 통합 조사

노드 이름 : hw_sw_integration

상세 노드 및 질문 목록

노드 이름	노드 설명
testplan_review	하드웨어와 소프트웨어의 통합 계획 여부
testprocedure_review	하드웨어 및 소프트웨어 인터페이스의 적합성을 입증하기 위한 통합시험절차 및 관련 승인 기준 존재 여부
system_CM_test	통합 컴퓨터 시스템에 대한 형상시험 수행 여부
integrationQA_review	하드웨어 및 소프트웨어 통합 변경 제어에 대한 품질 보증계획서 존재 여부
memory_review	비휘발성 메모리로 프로그래밍 된 프로세스 점검

질문 목록 7 : 시스템 검증 조사

노드 이름 : validation_review

상세 노드 및 질문 목록

노드 이름	노드 설명
detailed_testprocedure	상세 유형시험 절차 검토
o procedure_results	o 절차서와 입력자료 및 기대치 결과의 적절성
o testcondition_review	o 정적조건, 동적 조건하의 시험 여부
validation_review	시스템 검증 시험 검토
o test_organization	o 개발팀과 시험자 사이의 조직관계
o big_conf_test	o big configuration test
o testreport_available	o 테스트 보고서의 이용여부와 적절성
o testreport_assess	o 테스트 보고서의 평가 및 적절성
o error_finding	o 테스트 과정에서의 오류 발견 적절성
o propagation_effect	o 하드웨어 및 코드의 정확성 파급효과 고려 여부

질문 목록 8 : 사용자 문서 조사

노드 이름 : user_doc_review

상세 노드 및 질문 목록

노드 이름	노드 설명
user_manual	사용자 매뉴얼이 존재하고, 그것은 프로그래머가 소프트웨어를 개발하는데 필요한 충분한 정보를 포함하고 있는가?
install_op_manual	사용자가 프로그램을 설치하고 운영하는데 필요한 충분한 정보를 제공하는 설치/운전 절차서가 있는가?
maintenance_manual	제품의 고장 진단에 필요한 충분한 정보를 담고 있는 유지보수 매뉴얼이 있는가?
error_msg_rcmd_list	오류 메시지의 목록과 그에 관련된 수정 및 권고 사항이 있는가?
sw_identification_record	사용자 문서에 제품의 이름과 버전과 같은 소프트웨어의 확인 항목들이 명시되어 있는가?
abbreviation	사용자 문서는 동 문서에 사용된 약어(심볼, 명령어 문법 등)를 포함하고 있는가?
error_report_process	사용자 문서는 사용자가 소프트웨어나 문서에 문제가 있을 경우 보고하는 방법과 절차를 명시하고 있는가?

질문 목록 9 : 소프트웨어 유지보수 조사

노드 이름 : maint_review

상세 노드 및 질문 목록

노드 이름	노드 설명
notice_audit	변경 통지에 대한 감사 수행 여부
physical_config_audit	소프트웨어 선적 전 기능적 검사 및 물리적 감사 수행 여부
config_control_review	소프트웨어 소스코드와 실행코드에 대한 접근 및 형상 제어를 적절히 유지하는지 여부

질문 목록 10 : 운전 이력 및 오류 관리(Supplier/Item Performance Record)

노드 이름 : operation_history_record

상세 노드 및 질문 목록

노드 이름	노드 설명
sufficient_op_history	충분한 운전 이력이 있는가?
reliable_history_record	운전 이력에 대한 기록 자료는 신뢰성이 있는가?
formal_errorreport_process	회사 내부와 외부에서 시스템 오류에 대한 공식적인 보고 과정이 있는가?
errortracking_mechanism	보고된 문제의 처리 상태를 추적할 수 있는 메커니즘이 있는가?
systemetic_problemsolving	우선 순위, 스케줄링, 추적 등에 대한 문제를 해결하는 체계적인 접근방법이 존재하는가?
unsolvederror_os	운영체제에 해결되지 않은 중요한 문제가 있는가?
notification_process	고객 통지 과정이 존재하는가?
mgt_error_report	오류보고는 적절하게 추적되고 처리되는가?
similarity_of_application	해당 소프트웨어의 응용분야가 사용하고자 하는 분야(예: 원자력의 안전관련 응용분야)와 유사한가?

부록 3. 상용소프트웨어 인정 프로세스 BBN의 노드 확률 테이블

1. 상위 레벨 네트워크

o COTS_acceptance 노드의 npt

operation_history_record		good							
development_process		good				bad			
support_quality		good		bad		good		bad	
integration_validation		good	bad	good	bad	good	bad	good	bad
COTS_acceptance	accept	1	0.8	0.8	0.6	0.7	0.5	0.5	0.3
	no accept	0	0.2	0.2	0.4	0.3	0.5	0.5	0.7

operation_history_record		bad							
development_process		good				bad			
support_quality		good		bad		good		bad	
integration_validation		good	bad	good	bad	good	bad	good	bad
COTS_acceptance	accept	0.7	0.5	0.5	0.3	0.4	0.2	0.2	0
	no accept	0.3	0.5	0.5	0.7	0.6	0.8	0.9	1

o development_process 노드의 npt

pre_survey		good							
sw_system_req_review		good				bad			
design_review		good		bad		good		bad	
sw_dev_review		good	bad	good	bad	good	bad	good	bad
development_process	good	1	0.75	0.75	0.5	0.75	0.5	0.5	0.25
	bad	0	0.25	0.25	0.5	0.25	0.5	0.5	0.75

pre_survey		bad							
sw_system_req_review		good				bad			
design_review		good		bad		good		bad	
sw_dev_review		good	bad	good	bad	good	bad	good	bad
development_process	good	0.75	0.5	0.5	0.25	0.5	0.25	0.25	0
	bad	0.25	0.5	0.5	0.75	0.5	0.75	0.75	1

o integration_validation 노드의 npt

validation_review		good		bad	
hw_sw_integration		good	bad	good	bad
integration_ validation	good	1	0.5	0.5	0
	bad	0	0.5	0.5	1

o support_quality 노드의 npt

release_report_review		good				bad			
maint_review		good		bad		good		bad	
user_doc_review		good	bad	good	bad	good	bad	good	bad
support_quality	good	1	0.6	0.7	0.3	0.7	0.3	0.4	0
	bad	0	0.4	0.3	0.7	0.3	0.7	0.6	1

o operation_history_record 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o pre_survey 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o system_sw_req_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o design_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o sw_dev_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o maint_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o user_doc_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o release_report_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o hw_sw_integration 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

o validation_review 노드의 npt와 pdfs

상태	값
good	0.5
bad	0.5

2. 하위 레벨 네트워크

2.1 pre_survey 노드의 하위 레벨 네트워크

pre_survey		good	bad
plan_doc_review	yes	0.7	0.3
	no	0.3	0.7

pre_survey		good	bad
lifecycle_review	yes	0.7	0.3
	no	0.3	0.7

pre_survey		good	bad
standard_review	good	0.9	0.1
	bad	0.1	0.9

standard_review		good	bad
internal_qualification	yes	0.9	0.1
	no	0.1	0.9

standard_review		good	bad
external_qualification	yes	0.9	0.1
	no	0.1	0.9

2.2 system_sw_req_review 노드의 하위 레벨 네트워크

system_sw_req_review		good	bad
spec_list	yes	0.8	0.8
	no	0.2	0.2

system_sw_req_review		good	bad
spec_implementation	yes	0.9	0.1
	no	0.1	0.9

system_sw_req_review		good	bad
other_requirements	yes	0.7	0.3
	no	0.3	0.7

system_sw_req_review		good	bad
req_review_implementation	yes	0.9	0.1
	no	0.1	0.9

system_sw_req_review		good	bad
req_spec_quality	good	0.9	0.1
	bad	0.1	0.9

req_spec_quality		good	bad
functional_design_description	yes	0.9	0.1
	no	0.1	0.9

req_spec_quality		good	bad
requirement_specification	yes	0.9	0.1
	no	0.1	0.9

req_spec_quality		good	bad
dev_tools	yes	0.7	0.3
	no	0.3	0.7

2.3 design_review 노드의 하위 레벨 네트워크

design_review		good	bad
sw_design_process_review	good	0.9	0.1
	bad	0.1	0.9

sw_design_process_review		good	bad
pre_detail_review	yes	0.8	0.2
	no	0.2	0.8

sw_design_process_review		good	bad
reviewer_quality	yes	0.8	0.2
	no	0.2	0.8

design_review		good	bad
lifecycle_activity_review	yes	0.7	0.3
	no	0.3	0.7

design_review		good	bad
output_review	yes	0.9	0.1
	no	0.1	0.9

design_review		good	bad
init_req_review	yes	0.7	0.3
	no	0.3	0.7

design_review		good	bad
error_response_logic	yes	0.8	0.2
	no	0.2	0.8

design_review		good	bad
interface_review	yes	0.8	0.2
	no	0.2	0.8

design_review		good	bad
inservice_review	yes	0.8	0.2
	no	0.2	0.8

design_review		good	bad
timing_review	yes	0.99	0.01
	no	0.01	0.99

design_review		good	bad
hw_capability_review	yes	0.99	0.01
	no	0.01	0.99

design_review		good	bad
security_review	yes	0.7	0.3
	no	0.3	0.7

2.4 sw_dev_review 노드의 하위 레벨 네트워크

sw_dev_review		good	bad
QA_review	good	0.9	0.1
	bad	0.1	0.9

QA_review		good	bad
std_req_compliance	yes	0.99	0.01
	no	0.01	0.99

sw_dev_review		good	bad
doc_review	yes	0.9	0.1
	no	0.1	0.9

sw_dev_review		good	bad
dev_procedure	yes	0.9	0.1
	no	0.1	0.9

2.5 maint_review 노드의 하위 레벨 네트워크

maint_review		good	bad
notice_audit	yes	0.8	0.2
	no	0.2	0.8

maint_review		good	bad
physical_config_audit	yes	0.9	0.1
	no	0.1	0.9

maint_review		good	bad
config_control_review	yes	0.9	0.1
	no	0.1	0.9

2.6 user_doc_review 노드의 하위 레벨 네트워크

user_doc_review		good	bad
user_manual	yes	0.7	0.7
	no	0.3	0.3

user_doc_review		good	bad
install_op_manual	yes	0.7	0.3
	no	0.3	0.7

user_doc_review		good	bad
maintenance_manual	yes	0.9	0.1
	no	0.1	0.9

user_doc_review		good	bad
error_msg_rcmd_list	yes	0.7	0.3
	no	0.3	0.7

user_doc_review		good	bad
sw_identification_record	yes	0.7	0.3
	no	0.3	0.7

user_doc_review		good	bad
abbreviation	yes	0.7	0.3
	no	0.3	0.7

user_doc_review		good	bad
error_report_process	yes	0.9	0.1
	no	0.1	0.9

2.7 release_report_review 노드의 하위 레벨 네트워크

release_report_review		good	bad
addednewfunction_define	yes	0.7	0.3
	no	0.3	0.7

release_report_review		good	bad
modification_history	yes	0.9	0.1
	no	0.1	0.9

release_report_review		good	bad
product_info_doc	yes	0.7	0.3
	no	0.3	0.7

release_report_review		good	bad
newcomponent_link	yes	0.9	0.1
	no	0.1	0.9

2.8 hw_sw_integration 노드의 하위 레벨 네트워크

hw_sw_integration		good	bad
testplan_review	yes	0.9	0.1
	no	0.1	0.9

hw_sw_integration		good	bad
testprocedure_review	yes	0.9	0.1
	no	0.1	0.9

hw_sw_integration		good	bad
system_CM_test	yes	0.9	0.1
	no	0.1	0.9

hw_sw_integration		good	bad
integrationQA_review	yes	0.8	0.2
	no	0.2	0.8

hw_sw_integration		good	bad
memory_review	yes	0.8	0.2
	no	0.2	0.8

2.9 validation_review 노드의 하위 레벨 네트워크

validation_review		good	bad
detailed_testprocedure	good	0.9	0.1
	bad	0.1	0.9

detailed_testprocedure		good	bad
procedure_results	yes	0.99	0.01
	no	0.01	0.99

detailed_test_procedure		good	bad
testcondition_review	yes	0.9	0.1
	no	0.1	0.9

validation_review		good	bad
validation_testing	good	0.9	0.1
	bad	0.1	0.9

validation_testing		good	bad
test_organization	yes	0.9	0.1
	no	0.1	0.9

validation_testing		good	bad
big_configuration_test	yes	0.8	0.2
	no	0.2	0.8

validation_testing		good	bad
testreport_available	yes	0.9	0.1
	no	0.1	0.9

validation_testing		good	bad
testreport_assess	yes	0.9	0.1
	no	0.1	0.9

validation_testing		good	bad
error_finding	yes	0.99	0.01
	no	0.01	0.99

validation_testing		good	bad
propagation_effect	yes	0.8	0.2
	no	0.2	0.8

2.10 operation_history_record 노드의 하위 레벨 네트워크

operation_history_record		good	bad
sufficient_op_history	yes	0.99	0.01
	no	0.01	0.99

operation_history_record		good	bad
reliable_history_record	yes	0.9	0.1
	no	0.1	0.9

operation_history_record		good	bad
formal_errorreport_procedure	yes	0.9	0.1
	no	0.1	0.9

operation_history_record		good	bad
errortracking_mechanism	yes	0.8	0.2
	no	0.2	0.8

operation_history_record		good	bad
systemetic_problemsolving	yes	0.7	0.3
	no	0.3	0.7

operation_history_record		good	bad
unsolvederror_os	yes	0.9	0.1
	no	0.1	0.9

operation_history_record		good	bad
notification_process	yes	0.7	0.3
	no	0.3	0.7

operation_history_record		good	bad
mgt_error_report	yes	0, 8	0, 2
	no	0, 2	0, 8

operation_history_record		good	bad
similarity_of_application	yes	0, 99	0, 01
	no	0, 01	0, 99

서 지 정 보 양 식

수행기관보고서번호	위탁기관보고서번호	표준보고서번호	INIS 주제코드
KAERI/TR-2035/2002			

제목 / 부제	확률론적 안전성평가를 위한 BBN 기반의 소프트웨어 정량적 평가 방안: COTS Case Study
연구책임자 및 부서명 (주저자)	엄홍섭 (종합안전평가팀)
연 구 자 및 부 서 명	성태용 (종합안전평가팀), 정환성 (하나로운영팀), 박진균 (종합안전평가팀), 강현국 (종합안전평가팀) 이기영 (동력로기술개발팀), 박종균(동력로기술개발팀)

출 판 지	대전	발행기관	KAERI	발행년	2002. 3.
페 이 지	71 p.	도 표	있음(○), 없음( )	크 기	21×29.7cm
참고사항					
비밀여부	공개(○), 대외비( ), __ 급비밀		보고서종류	기술보고서	
연구위탁기관			계약 번호		

초록

현재 원전 안전 계통에 사용되는 소프트웨어의 신뢰도는 규칙기반의 정성적 평가에 의하고 있으나 원자력발전소의 안전성 평가를 위한 중요한 수단으로 사용되고 있는 확률론적 안전성 평가(PSA)에 디지털 시스템을 포함시켜야 하는 현실적 요구를 충족시키기 위해서는 소프트웨어 신뢰도의 정량화가 요구된다. 그러나 현재 각 산업분야에서 사용되고 있는 소프트웨어의 정량적 신뢰도 평가 방법들은 원전의 안전계통에 사용되는 고신뢰도 소프트웨어를 평가하기에 불충분하여 이러한 시스템의 정량적 신뢰도 분석에는 소프트웨어 부분을 제외시키거나 또는 임의로 특정한 값을 지정하여 사용하고 있는 실정이다. 본 보고서에서는 규제기관이나 산업체 등에서 현재 채용하고 있는 소프트웨어의 정성적인 평가 방법을 Bayesian Belief Nets을 이용하여 정형적으로 모델링하고 PSA에서 요구하는 정량화 된 결과를 구하는 한 가지 방안에 대하여 논의하였으며 원자력연구소에서 연구중인 "원전 상용소프트웨어 인정 프로세스"를 제안된 방안을 사용하여 동 방안의 PSA 활용 가능성을 검토하였다.

주제명키워드 (10단어내외)	PSA, Bayesian Belief Nets, BBN, 소프트웨어, 신뢰도, 정량적 평가

BIBLIOGRAPHIC INFORMATION SHEET

Performing Org. Report No.	Sponsoring Org. Report No.	Standard Report No.	INIS Subject Code
KAERI/TR-2035/2002			

Title / Subtitle	A Bayesian Belief Nets Based Quantitative Software Reliability Assessment for PSA : COTS Case Study
Project Manager and Department	H.S. Eom (Integrated Safety Assessment team)
Researcher and Department	T.Y. Sung (ISA), H.S. Jeong (Hanaro), J.H. Park (ISA) H.G. Kang (ISA), K.Y. Lee (ARTD), J.K.Park(ARTD)

Publication Place	Taejon	Publisher	KAERI	Publication Date	2002. 3.
Page	71 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)

Current reliability assessments of safety critical software embedded in the digital systems in nuclear power plants are based on the rule-based qualitative assessment methods. Then recently practical needs require the quantitative features of software reliability for Probabilistic Safety Assessment (PSA) that is one of important methods being used in assessing the whole safety of nuclear power plant. But conventional quantitative software reliability assessment methods are not enough to get the necessary results in assessing the safety critical software used in nuclear power plants. Thus current reliability assessment methods for these digital systems exclude the software part or use arbitrary values for the software reliability in the assessment. This reports discusses a Bayesian Belief Nets (BBN) based quantification method that models current qualitative software assessment in formal way and produces quantitative results required for PSA. Commercial Off-The-Shelf (COTS) software dedication process that KAERI developed was applied to the discussed BBN based method for evaluating the plausibility of the proposed method in PSA.

Subject Keywords (About 10 words)	PSA, Bayesian Belief Nets, BBN, Software, Reliability, Quantitative assessment