KAERI/AR-594/2001

원전 안전 소프트웨어의 정량적 신뢰도 평가를 위한 Bayesian Belief Nets 기술 분석

Survey of Bayesian Belief Nets for Quantitative Reliability Assessment of Safety Critical Software used in Nuclear Power Plants

한 국 원 자 력 연 구 소

INTERNATIONAL ATOMIC ENERGY AGENCY

INIS Section

INFORMATION NOTE

Dear User,

The title of this document has erroneously entered the INIS Database as:
'**Survey of bayesian belif nets for quantitative reliability assessment of safety critical software used in nuclear power plants**'.
The correct full title is: '**Survey of bayesian belief nets for quantitative reliability assessment of safety critical software used in nuclear power plants**'.

We will rectify this error in the INIS Database accordingly.

We apologize for the inconvenience this has caused.

Inquiries should be mailed to:

International Atomic Energy Agency
INIS Section
P. O. Box 100
Wagramerstrasse 5
A-1400 Vienna
Austria

Fax: (+43) 1 26007 or (+43) 1 2600 29882
Phone: (+43) 1 2600 ext. 22866, 22869 or 22870
E-mail: chouse@iaea.org

제 출 문

한국원자력연구소장 귀하

본 보고서를 2000 연도 "차세대원자로 설계관련 요소기술 개발" 과제의 기술현황분석보고서로 제출합니다.

2001.03. .

부서명 : 종합안전평가팀

주 저 자 : 엄흥섭
공 저 자 : 성태용
정환성
박진균
강현국

부서명 : 동력로 기술개발팀
이기영

요 약 문

원전 안전계통 디지털 시스템의 확률론적 안전성평가(Probabilistic Safety: PSA) 연구를 위하여 안전 소프트웨어의 신뢰도 측정과 평가에 관련된 방법들을 조사하였다. 디지털 시스템의 PSA를 위해서는 소프트웨어의 신뢰도에 대한 고려가 필수적이다. 이를 위해 현재 일반적으로 사용되고 있는 소프트웨어의 신뢰도 평가 기술과 방법론들을 안전 소프트웨어의 평가에 적용시 나타나는 문제점 중심으로 정리하였다. 조사된 방법들은 (1)시험 및 신뢰도 성장 모델에 근거한 직접적 평가 방법들, (2)개발 과정의 품질이나 설계의 다양성 그리고 정형적 방법론에 근거한 간접적 평가 방법들, (3)신뢰도에 영향을 주는 여러 증거들을 종합한 방법들이다. 그리고 조사 분석된 기술들 중 현재 가능성 있는 방안 중 하나로 대두되고 있는 Bayesian Belief Nets(BBN) 기술과 동 기술의 디지털 시스템 평가 응용 사례를 조사 하였고 이를 바탕으로 안전 소프트웨어의 신뢰도 정량평가 가능성을 검토하였다.

조사된 대부분의 신뢰도 측정 방법들은 제 2장에서 기술된 바와 같이 단독적으로는 안전 소프트웨어의 평가에 사용하는데 모두 한계성 또는 현실적 제약점을 지닌 것으로 나타났다. 한편, BBN 방법은 아직까지는 실제 문제에 적용이 많지 않은 새로운 방법론이지만, 기존의 안전 소프트웨어의 평가시 핵심적 역할을 하는 전문가의 판단과정을 정형적으로 나타낼 수 있고 또 신뢰도 평가시 고려되는 정성적 판단 증거를 포함한 다양한 증거들을 한 모델에 결합 시켜 그 결과를 정량화 할 수 있다는 점과 같은 BBN 고유의 특징으로 인해 다른 방법론에 비하여 안전 소프트웨어의 정량적 신뢰도 평가에 있어서 보다 가능성을 가진 방안으로 나타났다.

예를들면, 소프트웨어의 시험 결과 값을 단독으로 사용하는 것보다는 그에 더하여 해당 소프트웨어의 요구명세서의 적절성, 완벽성이나 개발 과정/방법론, 시험 과정 그리고 확인 및 검증 활동에 대한 전문가의 판단을 정량화하여 이를 모델에 반영하는 것이 보다 일반적으로 인정받는 정량적 신뢰도 값을 얻을 수 있는데 이러한 것이 BBN에서는 가능하다는 점이다. 한편, BBN 기술이 실용적 방안이 되기 위해서는 아직 해결해야 할 문제들이 있고 또 그 해결에는 시간이 요할 것으로 보인다. 그러나 이런 문제점들을 해결하는 과정에서 지금까지는 묵시적 가정으로 되어있던 평가에 관련된 요소들과 과정이 명시적으로 정형화되어

개발자와 평가자 및 규제자 간의 논의에 도움을 주고 의사결정의 투명도와 감사도(auditability)를 높일 수 있다는 점도 나타났다.

SUMMARY

Measures and methodologies applicable to quantitative reliability assessment of safety critical software were surveyed for the research of " Probabilistic Safety Assessment of safety grade digital systems used in Nuclear Power Plants". It is inevitable for the PSA of digital system to consider the reliability of software included in it. Among the techniques proposed in the literature we selected those which are in use widely and investigated their limitations in assessing the reliability of safety critical software. Surveyed techniques are (1) Direct methods including test based and reliability growth based ones, (2) Indirect methods including the ones based on process quality, design diversity, and formal verification, (3)evaluating method using multiple evidence sources. One promising methodology from the survey is Bayesian Belief Nets (BBN) which has a formalism and can combine various disparate evidences relevant to reliability into final decision under uncertainty. Thus we analyzed BBN and its application cases in digital systems assessment area and finally studied the possibility of its application to the quantitative reliability assessment of safety critical software.

Surveyed techniques mostly have limitations in assessing the reliability of safety critical software at this time. And BBN method has a promise with its features that could formalize expert's judgement process and could combine multiple evidences which are relevant to the reliability of software into its model. But BBN method has some difficulties in applying it to the real problem solving and it seems to take some time for its solutions. But it is also pointed out that the process of solving such difficulties could facilitate discussions between experts(including developers, assessment experts, and regulators) because it makes explicit those assumptions that were previously hidden, which leads to visibility and auditability in the decision making process.

PLEASE BE AWARE THAT
ALL OF THE MISSING PAGES IN THIS DOCUMENT
WERE ORIGINALLY BLANK

목 차

표 목차

그림 목차

제 1 장 서론

원자력발전소의 보호계통과 같은 안전등급 기기에 사용되는 소프트웨어는 고 신뢰도를 요구하나 고장의 원인이 설계 결함에 주로 기인하고 또 입력에 대해 비 선형적 출력을 가지는 소프트웨어의 특성으로 인하여 시험이나 신뢰도 성장 모델과 같은 기존의 정량적 평가 기법들을 사용해서는 이와 같은 고 신뢰도의 정량적 평가가 거의 불가능한 실정이다. 또 개발 과정의 적절성 평가나 정형적 방법의 채용 그리고 설계 다양성의 구현과 같은 고 신뢰도 개발 방법론에 근거하여 신뢰도를 평가하는 것도 역시 문제점을 내포하고 있다.

따라서 디지털 시스템이 사용되는 여러 산업분야에서는 이들 안전 소프트웨어의 신뢰도 평가를 설계 방법론과 과정 그리고 시험과 같은 소프트웨어의 개발 전 단계에 걸쳐 신뢰도에 영향을 주는 여러 가지 다양한 증거들을 종합하여 전문가가 최종적으로 판단하고 있다. 그러나 이 또한 평가시의 전문성 정도나 전문가의 정형화되지 않은 판단 과정과 같은 어려운 점을 내포하고 있어서 관련 분야의 규제 방향과 동 분야의 국제 표준들은 신뢰도의 정량적 평가를 검토하고 있고 일부에서는 이미 도입된 상태이다[32][33][34].

이러한 동향에 맞추어 개발된 소프트웨어의 안전성과 신뢰도 달성에 대한 검증 및 증명을 위한 여러 가지 연구가 진행되고 있는데 그 중 가장 가능성이 있는 기술의 하나로 여러 분야에서 연구되고 있고 또 실제 업무에 적용이 시도되고 있는 것이 Bayesian Belief Nets(BBN) 기술인데, BBN이 이처럼 소프트웨어의 안전성이나 신뢰도 정량평가에 적용되는 가장 큰 이유는 시험결과와 같은 정량적 수치에 더하여 시험과정의 품질이나 확인 및 검증과 같은 정성적 평가 내용을 함께 모델링하여 정량화 할 수 있다는 특징 때문이다.

본 보고서에서는 "원전 안전계통 디지털 시스템의 확률론적 안전성 평가(PSA)" 연구의 한 부분으로 수행되었던 고 신뢰도 소프트웨어의 평가기술 조사에 대한 내용을 기술하였다. 고 신뢰도 소프트웨어의 평가 문제는 현재 원전 안전계통 디지털 시스템의 PSA에 관련된 주요 쟁점 중의 하나이다[1]. 지금까지 문헌에 나타난 소프트웨어의 신뢰도 측정과 평가 기법 및 모델들과 이들의 분석은 본 과제의 1차년도 결과물 중의 하나인 "소프트웨어 신뢰도의 정량적 평가 기법에 대한 고유 현안 분석"[2]에 정리되어 있으며 본 보고서에서는 이 중 현재 소프트웨어 분야에서 많이 사용되고 있는 방법론들을 안전 소프트웨어의 신

뢰도 평가시 발생하는 문제점 중심으로 정리 분석하였다. 그리고 조사된 방법 중 가능성 있는 방안으로 나타난 BBN 기술과 동 기술의 응용 사례 및 분석 내용을 기술하였다.

제 2장에서는 기존에 사용되고 있는 직접적 간접적, 신뢰도 평가 방법과 여러 가지 증거에 근거한 종합적 평가 방법을, 제 3장에서는 조사된 기술 중 가장 가능성 있는 방안으로 나타난 BBN기술에 대하여, 제 4장에서는 BBN이 디지털 시스템의 안전성 및 신뢰도 평가 관련 분야에 적용된 사례를, 제 5장에서 결론을 기술하였다. 그리고 부록에는 i) BBN을 이용한 소프트웨어 및 디지털 시스템의 안전성/신뢰도 평가 사례에 나타난 BBN의 그래프 및 노드의 샘플과 ii) BBN 구축을 위해 분야 전문가의 지식을 추출하는 일반적 방법을 수집 및 정리하였다.

제 2 장 기존 소프트웨어 신뢰도 평가 기법

제 1 절 기존 평가기법 개요

소프트웨어의 신뢰도를 분석하고 평가하기 위한 측정 방법이나 모델에 대하여는 많은 연구가 수행되어 지금까지 70여 가지에 이르는 신뢰도 관련 측정 기법과 모델들이 발표되었고 또 이들에 대한 검증이 시도되어 왔다. 소프트웨어의 신뢰도를 포함한 품질 전반에 걸친 각종 측정방법과 모델은 본 과제에서 발간한 기술현황분석보고서[1]와 IEEE 표준[9] 그리고 소프트웨어 메트릭스 관련 서적[30]에 종합적으로 정리되어 있고 신뢰도의 분석과 평가 그리고 엔지니어링에 초점을 둔 기법과 모델들은 소프트웨어 신뢰도 공학 핸드북[3]에 정리되어 있다. 그리고 디지털 시스템의 확률론적 안전성 평가(Probabilistic Safety Assessment: PSA)에 적용하기 위한 소프트웨어 신뢰도의 정량적 평가 기법 고유 현안에 대해서는 본 과제에서 기 발간한 기술현황 분석보고서에 정리되어 있다[1][2]. 이들 문헌에 나타난 기법과 모델들은 일반적으로 다음과 같은 문제에 초점을 맞추고 있다[4].

- o 시스템에 존재하는 결함의 수
- o 다음 번의 고장이 일어날 때까지의 시간으로 표현되는 신뢰도 추정
- o 설계와 시험 과정이 결함의 수와 고장 밀도(density)에 미치는 영향의 이해

하지만 이런 문제를 해결하기 위해 지금까지 나온 대부분의 기법과 모델들은 그 방법론이나 이론적 측면에서 오류가 있거나 또는 모델의 잘못된 명세나 신뢰도에 직접적인 연결성을 가지지 않은 부적절한 데이터의 사용과 같은 여러 가지 형태의 결함을 가지고 있으며 또 문제의 전체가 아니고 일부분만을 다루었다는 점과 같은 여러 가지 문제를 내포하고 있어서 소프트웨어의 실제 신뢰도와 품질을 평가하기에는 부적절하다는 것이 현재까지의 일반적 정설이다[4][5][6]. 특히 원자력발전소의 보호계통이나 항공, 운수, 군사 분야의 안전 시스템에 사용되는 소프트웨어와 같이 고 신뢰도와 안전성을 요구하는 경우에는 기존의 어떤 기법이나 모델도 규제나 표준에서 요구하는 신뢰도의 정량적 증명이 불가능하다고 알려져 있다[5][6]. 이것은 설계 결함과 같은 시스템적 결함이 소프트웨어의 주요 고장 원인이라는 데 기인한다. 보다 일반적으로는, 설계 결함은 복잡성이 높고 새롭게 개발되는 모든 시스템에 공통적으로 나타나는 문제점으로 볼 수 있

으나 최근 디지털 시스템의 발달과 사용 증가로 소프트웨어에 이 문제가 집중되고 있는 실정이다. 일반적으로 고 신뢰도 및 안전성을 요구하는 소프트웨어의 신뢰도 평가 시 문제점은 다음과 같이 들 수 있다[5].

- o 소프트웨어의 고장은 설계 결함에 의해 주로 발생하는데 이들 설계 결함은 회피하거나 내구성(tolerance)을 갖추기 어렵다.
- o 소프트웨어는 주로 새로운 시스템에 구현되기 때문에 과거의 성공적인 설계로부터의 경험이나 지식을 활용하기 어렵다.
- o 일반적으로 디지털 시스템은 일반적으로 불연속적인 입출력 관계를 가지고 있어서 단순한 수학적 모델링으로는 입출력 관계에 대한 사상을 만들 수 없다. 따라서 연속성 가정은 소프트웨어의 신뢰도 측정에 사용될 수 없다. 고장은 특정적이고 불명확한 사건들의 조합에 의해 발생되며 하드웨어의 경우와는 달리 확인 될 수 있는 스트레스 요인에 의해 생기지 않는다.

위에서 기술된 문제점 관점에서 현재 많이 사용되고 있는 기법과 방법론에 대하여 제 2절에서 요약 정리하였다.

제 2 절 기존 평가 방법의 한계성

지금까지 문헌에 나타난 70여 가지의 신뢰도 평가 방법 중 실제로 사용되고 있는 것은 몇 가지에 불과하다. 본 절에서는 시험이나 신뢰도 성장 모델에 근거하여 직접적으로 평가하는 방법들과 개발 과정의 품질에 의하여 평가하거나 설계의 다양성 또는 정형적 방법론에 의거 간접적으로 평가하는 방법들 그리고 신뢰도에 영향을 주는 여러 증거를 종합하여 평가하는 방법론들을 살펴보고 이들을 사용하여 안전 소프트웨어의 신뢰도를 평가 할 때의 한계점에 대해 기술하였다.

1. 직접 측정 방법

직접적으로 소프트웨어의 신뢰도를 측정하는 유일한 방법은 대상 시스템이 실제 운전 환경에서 운전되는 것을 일정시간 동안 관찰 한 다음 나타난 고장 빈도 또는 무 고장 시간을 가지고 통계적 방법을 사용하여 신뢰도를 추정하는 것이다. 그러나 이러한 방법이 가진 근본적인 문제점 중 하나는 그 적용 영역에 따

라 다르기는 하지만 실제와 동일하거나 유사한 환경을 구축하여 시험을 수행하는 것은 매우 어렵다는 것이다. 직접적 측정 방법은 시험 기반(test-based) 신뢰도 추정과 신뢰도 성장(reliability growth)의 두 가지 범주로 나누어 볼 수 있다.

가. 시험에 근거한 신뢰도 추정

개발이 완료된 최종 시스템의 시험 결과들을 사용하여 신뢰도를 추정하는 방법으로 통계학의 표본 시험(sample testing)과 동일한 원리를 사용한다. 즉 시험 과정에서 나타난 결과들을 시스템의 모든 가능한 행위 공간의 표본으로 간주한 다음 통계적 추론 기법을 사용하여 고장율(failure rate) 또는 요구 고장(failure per demand) 확률과 같은 신뢰도 추정을 위한 파라미터(parameter)를 구하는 것이다. 이 방법은 직접적으로 신뢰도를 추정할 수 있는 숫자를 얻을 수 있다는 장점이 있는 반면, 문제점으로 지적되고 있는 것은;

(a)소프트웨어의 고장을 통계적으로 처리 가능한 무작위성 고장(random failure)으로 보는 데는 아직 많은 논란이 있으며

(b)적절한 시험이 되기 위한 여러 가지 요인들(시험 횟수, 입력 자료 분포, 시험 중 발견된 고장, 결함을 수정하는 전략, 프로그램의 크기와 복잡도, 시험 결과를 판정하는 오라클)의 신뢰성이 필요하고

(c)요구되는 신뢰도가 높은 경우 현실적으로는 실현 불가능한 시험 시간을 요한다는 점이다.

나. 신뢰도 성장 모델(reliability growth model)에 의한 신뢰도 예측

소프트웨어의 개발 마지막 단계인 시험 단계에서는 디버깅을 통해 소프트웨어에 존재하는 결함이 점차로 줄어드는 새로운 버전의 소프트웨어들이 생성된다. 신뢰도 성장 모델에 근거한 방법들은 이런 디버깅(결함 수정) 활동에서 누적 되어진 증거(결함이 발견될 때까지의 시험 운전 시간, 또는 일정 시험 시간동안 발견된 결함의 수 등)를 사용하여 신뢰도가 향상되는 추세(trend)를 관측하고, 그 다음 이 추세로부터 보외법(외삽법, extrapolation)을 근간으로 하는 모델(reliability growth model)을 만들어 내어 이로부터 신뢰도를 예측하는 것이다. 따라서 이 방법에 의한 예측은 증거와 모델을 통해 만들어진 추세의 정확성에 의존하게 되는데 이를 위해 요구되는 것은 신뢰도 추세에 영향을 주는 각종 요인들(시험 팀의 변화, 새로운 기능의 추가 등)에 대한 고려, 디버깅 중 사용한 데이터와 시험환경이 실제 소프트웨어가 적용될 상황과 동일하거나 매우 유

사해야 한다는 점, 그리고 마지막 결함 수정이 신뢰도에 미치는 효과가 최소한 그 이전의 결함 수정 활동들의 평균적 신뢰도 상승 효과보다 나쁘지 않아야 한다는 점이 있다.

이 마지막 항목은 고 신뢰도 또는 안전성을 요구하는 소프트웨어의 평가에서는 간과할 수 없는 중요한 점이나 현재까지의 모델들은 이 부분을 간과하거나 단순화하여 처리하고 있으며 또한 현실성 있는 연구는 아직 진행되지 않고 있는 실정이다. 신뢰도 성장 모델은 소프트웨어의 신뢰도 엔지니어링과 측정에서 현재 널리 사용되는 방법론으로 AT&T, Bellcore, Ericsson Telecom, Hewlett Packard, Hitachi, IBM, NASA' s Jet Propulsion Laboratory, Lockheed-Martin, Lucent Technologies, Microsoft, the U.S. Air Force, and the U.S. Marine Corps.등에서 사용되고 있으며 동 방법론의 사용 경험들에 대하여 많은 문헌이 발표되어 있다[31]. 신뢰도 성장 모델들은 여러 가지가 있는데 각각 특정 소프트웨어(예. 통신용 소프트웨어)의 시험과 운전경험을 통해 만들어진 것으로서, 평가 대상 소프트웨어가 모델의 바탕이 되는 소프트웨어와 특성이 유사하고 또 중간 정도의 신뢰도를 평가하는 경우에는 유용한 것으로 나타나 있다[3][7][8].

2. 간접적 측정 방법

위에서 살펴 본 직접적 평가 방법들을 사용해서 증명할 수 있는 소프트웨어의 신뢰도 레벨에는 한계가 있기 때문에 이를 극복하기 다른 방법들이 모색되었다. 이들 중 현재 산업계에서 많이 사용되고 있고 또 연구가 활발하게 진행 중인 방법들은 i) 소프트웨어 개발 과정 및 방법론에 근거한 신뢰도 증명 ii) 설계 다양성에 기반을 둔 고장 내구성(fault tolerance) 기법으로 신뢰도를 증명 iii) 정형적 방법에 의한 확인(Formal Verification)을 들 수 있다.

가. 개발 과정과 방법론에 근거한 신뢰도 증명

이 방법은 적절한 설계와 개발 기술(good practice)을 사용하면 고 신뢰도의 소프트웨어를 만들 수 있을 뿐만 아니라 이 방법의 사용 자체가 요구되는 고 신뢰도를 달성하였다는 것을 보증할 수 있다는 것에 근거하며 여러 산업 분야의 일부 표준들이 이 방법을 채택하고 있다[13]. 고 신뢰도를 요구하는 소프트웨어의 개발에 good-practice가 반드시 필요하다는 것은 분명하고 또 그런 소프트웨어가 만들어 질 수 있지만 문제는 목표한 신뢰도를 달성했다는 증거가 없고 증명할 수 없다는 점이다. 이런 기준에만 의거하여 신뢰도를 평가하는데 따르는 문제점은;

o 특정 개발과정이나 방법론을 사용하여 개발된 소프트웨어가 실제 운전 되면서 얻어진 신뢰도 자료가 없다는 점과

o 설사 그런 자료가 충분하다고 하더라도 그것은 평균적 신뢰도 자료이므로, 서로 다른 여건에서 개발된 각 소프트웨어어의 실제 신뢰도와 같을 수 없기 때문에 필연적으로 불확실성이 포함된다는 점이다.

나. 설계의 다양성을 사용한 고장 내구성에 의한 신뢰도 증명

하드웨어의 신뢰도 엔지니어링에서는 설계의 다양성에 의해 고 신뢰도 시스템의 개발이 가능하다. 이것은 동일한 기능을 가진 다른 기기의 고장 프로세스가 서로 독립적이라는 가정이 현실성이 있고 경험적 자료에서 증명되어 있기 때문이다. 그러나 소프트웨어의 경우에는 이런 설계 다양성에 의해 개발된 각 버전들의 고장 독립성 가정은 아직까지 명확한 답이 나와 있지 않다. 이 문제에 대한 논의를 보면 A. Avizienis와 J.P.J. Kelly가 자체 실험 결과에 근거하여 설계 다양성에 의한 고장내구성 방안을 제안하였고 이에 대하여 J.C. Kinght과 N.G. Leveson은 소프트웨어 버전들의 고장 프로세스가 서로 독립적이 아니라는 것을 역시 실험을 통해 보여주면서 설계 다양성에 의한 고장내구성 방안에 대하여 부정적 의견을 제시하였다[14]. 이들 두 가지 의견에 대한 논의가 계속 진행되고 있지만, 설계의 다양성에 의해 어느 정도 신뢰도가 향상된다는 것이 현재까지의 일반적 견해이다[15][16]. 그러나 이 경우에도 신뢰도가 향상된 정도를 측정하는 문제와 전체의 신뢰도를 구하기 위해 필요한 각 버전의 신뢰도를 별도로 구해야 하는 문제는 여전히 남아있게 된다.

다. 정형적 방법에 의한 확인(Formal Verification)을 통해 신뢰도 증명

이 방법은 정형적 방법론을 사용하여 처음부터 완전한 소프트웨어를 개발한다는데 근거를 두고 있다. 즉 정형적 명세가 정확하게 작성되었다면 프로그램이 그 명세에 따라 정확하게 구현되었다는 증거에 의해 그 소프트웨어는 설계에 의한 결함이 발생하지 않을 것이라는 것을 보증하는 것이다. 실제로 이 방법은 각 분야에서 활용되고 있고 연구가 활발하게 진행되고 있는 유용한 기술이나 다음과 같은 문제점이 지적되고 있다.

o 증명에는 실수나 결함이 도입되기 쉽다는 점이다. 예를 들면 증명 작업을 수행하는 인간의 오류나 지원 도구의 결함 그리고 자동화된 도구의 결함 등이 존재한다.

o 실용적 측면에서 보면, 현재까지는 이 기술을 사용하여 해결 할 수 있는 대상의 크기와 복잡성에 제한이 있다.
o 모든 경우에 정형적 명세가 완전하다는 가정은 현실적으로 어려우며 따라서 잘못된 명세에 따른 구현의 정확성 증명은 무 고장을 보증 할 수 없다.

3. 여러 증거의 종합에 근거한 신뢰도 평가 및 증명

위에서 살펴본 기술과 방법론은 그 단독으로는 신뢰도의 평가와 증명을 하는데 한계를 가지고 있다. 따라서 대부분의 산업분야에서 안전시스템에 사용되는 소프트웨어를 평가 할 때는 여러 가지 증거를 결합하여 관련 표준이나 가이드에 따라 종합적으로 최종 결정을 내리고 있는데 여기에는 전문가의 판단이 핵심적 역할을 담당한다. 이와 같은 평가 방법에서 문제점으로 지적되고 있는 것은;

o 평가에 사용되는 각 증거에 대해 값을 부여하는 문제와
o 각 증거들은 서로 다른 특성과 성질을 가지고 있어서 이들을 결합하는 것이 어렵다는 점이다. 특히 신뢰도에 영향을 미치는 증거들 간의 종속성 관계가 대부분 공식적으로 밝혀지지 않고 있어서 더욱 그렇다.

이들 문제점은 현재 전문가의 판단에 의존하여 처리되고 있는데 각 전문가들이 이런 문제들을 처리하는 과정은 거의 정형화 되어있지 않다. 다양한 증거에 의거하여 제품의 안전성이나 신뢰도를 평가 할 때 전문가의 판단(expert judgement or engineering judgement)의 역할에 대하여는 조사된 연구의 내용을 요약하면 다음과 같다[24].

[전문가 또는 공학적 판단(judgement)의 역할]

높은 수준의 안전성 또는 고 신뢰도를 요구하는 제품이나 시스템의 평가에 있어서 현재의 기술 수준에서는 완전하고 결정적인 증거를 얻을 수 없기 때문에 전문가의 판단(예를 들면 다양하고 복잡한 증거들에 근거한 비공식적 추론 등)이 의사결정에 있어서 중요한 역할을 담당한다. 이런 경향은 복잡도가 높은 시스템에서 설계 결함을 고려한 신뢰도를 평가 할 때 두드러지게 나타나는데 소프트웨어의 신뢰도 평가가 그 대표적인 경우에 속한다.

비록 전문가의 판단이 평가에 있어서 가장 핵심적인 역할을 수행하고는 있지만 그 판단 과정에는 문제의 소지가 있다. 전문가의 사고 방법, 증거를 종합하는 방법, 그리고 수행 능력에 대한 실험적 연구는 전문가의 능력이 과대 평가되는 경우가 종종 있다는 것을 보여주고 있다.

일반적으로 알려져 있는 전문가의 본질(nature)은;

- o 전문가는 해결 대상 문제에 대하여 보다 많은 사실들을 알고 있고,
- o 전문가는 이런 문제에 대한 추론에 익숙하기 때문에 그들이 알고 있는 사실들을 보다 잘 활용할 수 있다는 점이다.

이와 같은 근거에서 전문가의 판단이 중요한 역할을 하고 있지만 두 번째의 근거에 대해서는 아직 논란이 많다. 즉, 전문가는 비전문가에 비해 숙련기반(skill based) 또는 규칙 기반(rule based) 업무(이들 업무는 낮은 수준의 지적 활동을 요구함)에 대해서는 뛰어난 능력을 보이나 지식 기반(knowledge based) 업무(높은 수준의 지적 활동을 요구하는 업무)의 처리에 있어서는 비전문가와 동일한 실수를 하기 쉽다는 연구결과가 나와있다[25].

위에서 지적된 문제를 해결하기 위해 최근에 제안되고 있는 방법중 하나로 Bayesian Belief Nets 방법론이 있다. BBN은 인공지능 분야에서 이론적으로 연구되어 현재는 여러 분야에서 활용되고 있으며 불확실성 하에서 다양한 증거를 결합하여 의사결정을 하는데 가장 유용한 방법으로 알려져 있는데 다음의 제 3 장에서 동 기술에 대하여 기술하였다.

제 3 장 Bayesian Belief Nets(BBN)

제 1 절 배경

BBN은 Belief Nets, Causal Probabilistic Networks, Causal Nets, Graphical Probability Networks, Probabilistic Cause-Effect Models, Probabilistic Influence Diagrams라고도 하며 지식(knowledge)의 확률적 표현으로서 여러 분야에 적용되어 왔고[10] 최근에는 불확실성 하에서의 의사 결정 지원 문제를 해결하는 방법으로 많은 연구가 진행되고 있다. BBN의 근간이 되는 베이시안 확률 이론(Bayesian Probability Theory)은 예전부터 있었지만 현실적 문제의 해결에 본격적으로 적용되기 시작한 것은 확률계산의 알고리즘[11]이 발견되고 또 그것을 구현한 소프트웨어 도구[12]가 만들어지고 난 이후이다. 의료, 원유 가격 예측, 기계 고장 진단 분야에서는 90년대 초부터 실제 문제의 해결에 적용되어 왔고 특히 의학적 진단과 기계적 고장 진단 분야에서는 그 유용성이 입증되었는데 마이크로 소프트의 오피스에 내장된 Answer wizard와 지능형 프린터 고장 진단 시스템이 최근의 가장 성공적인 응용사례로 알려져 있다. 소프트웨어 엔지니어링 분야에서 사용되기 시작한 것은 Sabre airplane reservation system의 디버깅 단계에서 결함을 발견하고 진단하는데 사용한 것이며 90년대 중반 이후부터 소프트웨어의 신뢰도와 안전성 분석 및 평가에 적용이 시도되고 있다.

제 2 절 BBN 모델

1. BBN의 정의

BBN은 노드(Node), 노드들 사이를 연결하는 연결선(arcs 또는 directed edges) 그리고 각 노드에 속한 확률 테이블(Node Probability Tables: NPT, or Conditional Probability Table: CPT)로 구성된다. 노드는 모델에 포함된 변수들을 나타내며 노드 연결선은 노드간의 인과관계를 나타낸다. BBN상의 노드는 무작위 변수를 나타내고 이들 무작위 변수는 몇 개의 상태(예: 낮음, 중간, 높

음 등)를 가지고 있으며 각 상태의 확률 값의 합은 1이 된다. 각 노드에 연결된 노드 확률 테이블은 노드간의 연결 강도 즉 종속성 정도를 결정하며 모 노드(parent node)의 각 상태에 대한 조건부 확률로 표현된다. BBN의 구성은 다음과 같다.[19]

(a) 변수들의 집합과 이들 변수들을 연결하는 방향성을 가진 연결선들

(b) 각 변수는 유한한 상호 배타적 상태 집합을 가지고 있다

(c) 변수들과 연결선은 서로 합쳐져서 유도 비순환 그래프(Directed Acyclic Graph:DAG)를 형성한다. 유도 비순환 그래프는 A1 -> A2 -> ... An -> A1 으로되는 직접적 경로가 없는 그래프를 의미한다. 따라서 BBN에서는 그래프 내에 순환 연결이 포함되어서는 안 된다.

(d) 모 변수 B1, ... Bn을 가진 각 변수 A는 조건부 확률 테이블(conditional probability table: CPT) 또는 노드 확률 테이블(node probability table: NPT) P(A|B1, ..., Bn)을 가지고 있다. 그러나 변수 A가 모 변수를 가지지 않는 경우는 무조건(unconditional) 확률을 가지게 된다. 이 경우의 변수 A는 prior 확률 값을 가지게 되어 모델에 편견이 도입될 우려가 있다는 주장이 있었으나 이런 선 확실성 평가 (prior certainty assessment)는 인간이 확실성을 추론할 때의 중요한 요소이기 때문에 필요하다.

(e) BBN에 d-separation을 적용하기 위해서는 어떤 변수든지 자신의 Markov blanket 내의 다른 변수들에 대해 독립적이어야 한다. d-separation은 노드들 간의 조건부 종속성을 의미하는 것으로 한 노드의 값이 다른 노드에 영향을 미치는 여부를 결정하며 노드들간의 연결 형태에 따라 다르게 적용된다. d-separation에 대한 상세한 내용은 노드들의 연결 형태 설명 부분에 기술되어 있다. 그리고 어떤 변수 A의 Markov blanket은 A의 모 노드와 A의 자노드 그리고 그 자노드를 A와 공유하는 변수들을 의미한다.

2. 용어 및 개념

가. 증거(evidence)

노드(또는 변수)의 증거는 그 노드 상태의 확실성에 대한 진술이며 확정증거(hard evidence)와 추정증거(soft evidence)의 두가지 종류가 있다.

1) 확정증거

증거가 그 노드의 확률 값을 정확한 상태로 나타낼 수 있을 경우, 즉 명확하고 특정한 값을 가지는 증거를 확정증거라고 하고 또 이를 실증(instantiation)이라고도 한다. 예를 들어 100번의 테스트가 모두 성공적으로 수행된 경우 이것이 어떤 노드의 증거로 입력될 때 이는 확정증거라고 할 수 있다.

2) 추정증거

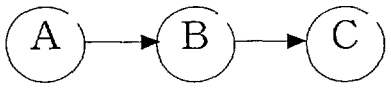
확정증거에 해당되지 않는 증거를 추정증거라고 한다. 여기에는 노드의 기존 확률 값을 변경시킬 수 있는 모든 증거들을 포함한다.

나. 노드의 연결 형태

BBN에는 직렬 연결(serial connection), 분기 연결(diverging connection), 수렴 연결(conversing connection)의 세 가지 노드간 연결 상태가 있다.

1) 직렬 연결

직렬 연결은 노드 A,B,C가 다음과 같이 연결되어 있는 경우이다.

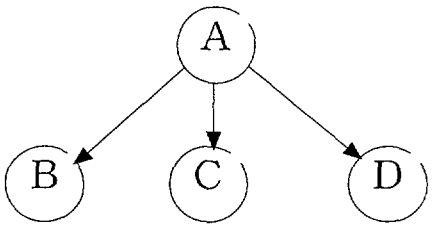


[그림 3.1 직렬 연결]

직렬연결에서는 노드 A가 노드 B에 영향을 주고 차례로 노드 B가 노드 C에 영향을 미친다. 즉 노드 A의 증거는 노드 B의 확실성을 변경시키고 뒤이어서 노드 C의 확실성을 변경시킨다. 그리고 노드 C의 증거가 노드 B와 노드 A에 영향을 미치는 것도 마찬가지이다. 하지만 노드 B의 상태가 분명하게 알려지면 연결 채널이 차단되어 노드 A와 노드 C는 독립적이 된다. 이럴 때 노드 A와 노드 C는 주어진 노드 B에 대해 종속 분리되었다고 한다.(A and C are *d-separated given B*). 따라서 직렬연결에서는 연결 내의 노드 상태가 알려지지 않은 한 증거는 직렬연결을 통해 전달된다. 각 노드들은 노드 확률 테이블을 가지고 있는데 직렬연결에서 이들 노드의 확률 변화를 보면, 노드 A나 노드 C에 증거(확정 증거 또는 추정 증거)를 입력하면 이는 노드 B의 확률 분포를 변경시킨다. 하지만 노드 B에 확정 증거가 입력되면 노드 A와 노드 C는 조건부로 독립적 상태가 되어 서로간 영향을 주지 않게 된다.

2) 분기 연결

분기연결은 노드들이 다음 그림과 같이 연결되어 있는 경우이다.

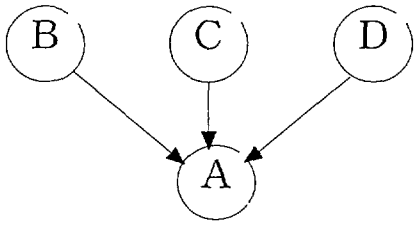


[그림 3.2 분기 연결]

분기연결에서는 노드 A의 증거가 A의 자 노드(child node)들 B,C,D에 영향을 준다. 그리고 모 노드인 A의 상태가 알려져 있지 않은 상태에서는 자 노드들이 서로 의존 상태로 되어 있어 이들 자식 노드들 중 하나에 증거가 입력되면 이는 다른 자식 노드들에 영향을 미친다. 하지만 모 노드 A의 상태가 알려져 있으면(즉, 모 노드 A에 확정 증거가 입력되면) 자식 노드들 중 하나(예: B 노드)에 입력된 증거는 다른 자식 노드들(예: C, D)에 영향을 미치지 않는다. 이를 노드 B,C,D가 주어진 노드 A의 상태에 대하여 d-separated 되었다고 한다(B,C,D are d-separated given A). 정리하면 분기연결에서 각 노드들의 증거들은 모 노드 A의 상태가 알려져 있지 않으면(노드 A에 확정 증거가 입력되지 않는 한) 서로 전달되고 노드 A의 상태가 알려져 있으면(확정 증거가 입력되면) 자 노드들 간의 연결을 차단한다.

3) 수렴 연결

수렴연결은 노드들이 다음 그림과 같이 연결되어 있는 경우이다.



[그림 3.3 수렴 연결]

위의 그림과 같은 수렴연결에서 모 노드들 B,C,D의 증거들이 자 노드 A에 영

향을 미치는 것은 당연하다. 문제가 되는 것은 모 노드들 B,C,D 간에 서로 영향을 주는가 인데 그것은 다음과 같이 정의된다.

- o 만약 자 노드 A의 상태가 알려져 있지 않다면 모 노드들은 서로 독립적 상태(independent)가 되어 한 노드의 증거가 다른 노드의 확실성에 영향을 미치지 않는다.
- o 만약 어떤 증거(노드 A에 직접 들어온 증거 또는 A의 자식노드로부터 영향을 받은 증거)가 자 노드 A의 확실성에 영향을 주었다면 노드 A의 모 노드 B,C,D는 서로 의존적 상태로 변하여 이들 노드들 중 하나의 확실성 변화는 다른 모 노드들에게 영향을 미치게 된다.

위에서 본 이들 3가지 형태의 연결에서 d-separation의 정의는 다음과 같이 내릴 수 있다.

[*d-separation의 정의*]

- o 두 개의 노드 A와 B가 있고 이 두 노드 사이의 경로에 중간 노드 V가 있을 때 다음과 같은 경우 두 노드 A, B는 d-separated 되었다고 한다.
  - 직렬 연결과 분기 연결 경우, 노드 V의 상태가 알려 질 경우 (즉, 노드 V에 확정 증거가 입력 된 경우)
  - 수렴 연결의 경우, 노드 V와 V의 자 노드들에게 어떤 증거(확정증거 또는 추정 증거)도 입력되지 않은 경우

그리고 두 노드 A,B가 "d-separated" 상태가 아닌 경우 이를 "d-connected" 라고 한다.

다. 조건부 독립성(conditional independence)

위의 노드 연결 상태 및 d-separation에 의한 증거의 전달 차단은 조건부 독립성의 개념에 반영되어 있다. 즉 변수 A, C는 주어진 변수 B에 대하여 다음과 같을 경우 독립적이 된다.

$$P(a_i|b_j) = P(a_i|b_j,c_k) \text{ for all } i,j,k$$

이것은 B의 상태가 알려지면 C의 어떤 지식도 A의 확률을 변경시키지 않는다는

것을 의미한다.

라. 전달(propagation)

구축된 BBN의 어느 노드에 증거(값)를 입력하고 그것을 사용하여 다른 노드들의 값을 갱신하는 것을 말한다. 특정 노드에 의해 다른 노드의 값이 갱신되는 원리는 기본적으로 베이스 정리(Bayes theorem)와 d-separation/conditional probability에 의한다. BBN의 계산은 매우 복잡하고 많은 양의 계산을 필요로 하기 때문에 그 동안 BBN 실용화의 걸림돌이 되어왔으나 J. Pearl[10]과 S.L. Lauritzen[11]이 계산을 효과적으로 빨리 할 수 있는 알고리즘을 발견한 이후 여러 가지 효과적인 확률 갱신 알고리즘들이 발표되었고 이들 중 일부는 현재 발표되어 있는 HUGIN[12]과 같은 소프트웨어 도구에 구현되어 있다[19].

마. 변수(노드)의 형태

BBN 모델을 구축할 때 사용되는 변수는 다음의 3가지 형태가 있다.[19]

- o 가설 변수(Hypothesis variables)
  모델에서 구하고자 하는 상태를 가진 노드이다. 일반적으로 이들은 관찰하기 불가능하거나 비용이 많이 소요되는 경우가 대부분이다.
- o 정보 변수(Information variables)
  현실적 제약 조건 내에서 관측 가능한 노드들이다.
- o 조정 변수(Mediating variables)
  특정 목적을 위해 도입된 노드들로서 가설 변수와 정보 변수에 해당하지 않는 모든 변수를 지칭한다. 주로 정보 노드와 가설 노드의 연결을 위해 사용되는데 확률 획득(acquisition)을 용이하게 하고 모델의 정밀도를 높이는 효과가 있다. 이 형태의 변수 사용은 모델을 정련(refine)하고 유용하게 만드는데 적합하지만 부작용으로 BBN 구축시 추가 노력을 필요로 하고 또 복잡성을 증가시켜 성능(performance)을 저하하게 할 가능성이 있다.

3. 노드 확인 및 그래프 작성

목표 노드의 값을 구하기 위해 관련된 모든 사건이나 증거를 결합하는 과정이

다. 이를 위해서는 모델 하고자 하는 영역의 모든 변수들이 확인되어야 하고 그 다음 이들을 연결하여 망을 구성하는 것이다. 연결하는 방법에 대한 명확한 답은 아직 까지는 없으나 목표 노드로부터 시작하여 이에 영향을 미치는 노드를 작성하는 것과 상위 레벨의 정보를 가진 망을 구성한 후 이들을 세분화하여 작성하는 방법 등이 있다. 노드간의 연결 방향을 결정하는 것은 인과관계에 따르는 것이 일반적 방법이다. 하지만 정성적 노드들의 연결에서는 인과관계가 명확하지 않은 경우가 종종 있는데 이럴 때는 보다 일반적 개념을 지닌 노드에서 구체적이고 상세한 내용을 지닌 노드로 방향을 정하거나 또는 상위 단계의 추상적 개념을 가진 노드에서 하위 단계의 추상적 개념을 가진 노드로 방향을 정하는 방법을 따른다. 이 작업이 끝나면 마지막으로 각 노드의 상태 공간을 설정하게 된다. 한 노드에 연결된 모노드가 많은 경우 상태 공간이 많아져서 노드의 확률값을 구하는 것이 어렵게 되는 경우에는 noisy-or 나 divorcing[19]과 같은 기법이 사용된다.

4. 노드 확률 테이블

그래프가 작성되면 다음 단계는 각 노드별 확률 테이블(NPT)을 작성하게 된다. 이 과정은 구두 또는 문서를 통하여 전문가의 판단을 얻고 이를 확률 값으로 변환하는 것이다. 이런 지식의 추출은 개별 면접, 그룹 회의, 그리고 Delphi situations과 같은 형태로 수행되고 일반적 순서는 다음과 같다.

(a) 문제 영역과 특정 질문의 작성
(b) 질문의 정련(refinement)
(c) 전문가(들)의 선정
(d) 추출 구성품의 선택
(e) 적용분야에 알맞게 선택된 구성품 정련
(f) 추출 연습 및 필요한 교육 수행
(g) 전문가의 판단을 추출하고 문서화

노드 확률 테이블을 작성하는 과정에서 주요한 점은 편향(bias)의 회피이다. 이것은 BBN 방법론 뿐 아니라 전문가의 지식을 추출하는 모든 작업에서 나타나는 현상인데 주로 지식 공학 분야에서 이에 대한 연구가 수행되고 있다[17][18]. 본 보고서의 부록 2에 BBN 구성을 위한 전문가의 지식과 판단 추출과 관련된 일반적인 절차와 방법의 개략을 정리 수록하였다.

5. 계산(computation)

그래프의 작성과 NPT 작성이 끝나면 관찰된 증거를 입력하고 계산을 수행한다. 계산은 특정 노드에 자료를 입력했을 때 이것이 구성된 망의 구조와 NPT에 따라 전체 노드의 값을 갱신시키는 것을 의미한다. 이와 같은 노드 값의 갱신 계산은 베이스 확률규칙(Bayes rule)과 조건부 확률(conditional probability) 정리 그리고 종속성 차단(d-separation) 정리를 기반으로 수행된다. 이와 같은 계산 작업은 수작업으로는 불가능할 정도로 복잡한데 80년대 후반에 발견된 효과적인 계산 알고리즘과 이를 구현한 소프트웨어 도구는 BBN을 실용적 문제에 적용할 수 있게 하는 중요한 역할을 하였다.

6. BBN용 소프트웨어 도구

BBN 모델의 복잡한 확률 계산을 효과적으로 수행 할 수 있는 알고리즘들이 발견된 이후 이를 구현한 소프트웨어들이 개발되었고 이것은 BBN을 그 동안의 이론적 연구에서 벗어나 실제 문제의 해결에 적용하는데 큰 역할을 하였다 [10][11][19]. 현재 나와있는 BBN용 소프트웨어들과 관련된 웹 주소(또는 연락처)는 다음과 같다.

가. 상용 소프트웨어

상용 소프트웨어의 대부분은 데모 버전을 받을 수 있다.

- o Demos : http://www.lumina.com/
- o DXpress : http://www.kic.com/
- o Ergo : http://www.noeticsystems.com/ergowin.exe
- o GRAPHICAL-BELIEF 2.0 : (gmellman@statsci.com StatSci (MathSoft, Inc.)
- o HUGIN : http://www.hugin.dk/
- o Netica : http://www.norsys.com/netica.html
- o + Strategist : Prevision 1947 NW Garryanna St. Corvallis, OR 97330 USA
  541-754-0569 FAX: 541-757-0976, dambrosi@prevision.com

나. Free 소프트웨어

o BAYES

: wuarchive.wustl.edu:/mirrors/msdos/neural-nets/bayes.zip

o BELIEF 1.2 (and version 1.1)

: http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/belief

o IDEAL : Request (ideal-request@rpal.rockwell.com)

o MacEvidence : Prakash P. Shenoy, School of Business University of Kansas Summerfield Hall Lawrence, KS 66045-2003 USA

o + MSBN32 : http://www.research.microsoft.com/research/dtg/msbn/

o Pulcinella : http://iridia.ulb.ac.be/pulcinella/Welcome.html

o SPI : http://www.spaces.uci.edu/thiery/elimbel/elimbel.scm

o TresBel : ftp://iridia.ulb.ac.be/pub/hongxu/software/TresBel.tar.Z

o XBaies : ftp://egon.stats.ucl.ac.uk/xbaies.zip

위의 소프트웨어 중 HUGIN과 Netica 두 가지가 사용의 용이성과 좋은 사용자 인터페이스를 가지고 있어서 널리 사용되고 있다.

7. 소프트웨어 신뢰도 정량 평가에의 적용 가능성

BBN의 최대 특징은 인과관계에 의한 모델링과 불확실성의 명시적 모델링 그리고 다양한 유형의 증거(정성적 평가 증거와 정량적 평가 증거 모두)를 일관된 프레임 안에서 정형적으로 처리하여 정량화 시킬 수 있다는 점이다. 안전소프트웨어의 신뢰도나 안전성 평가에는 제2절에서 본 바와 같이 인지적 한계와 현실적 제약이 수반되므로 필연적으로 불확실성이 포함되고 또 평가에 필요한 모든 증거를 얻을 수 없는 것이 일반적 상황이다. BBN은 이런 모든 상황을 포함한 모델링을 할 수 있다는 점에서 기존 방법에 비하여 장점을 가진다. BBN을 안전 소프트웨어의 신뢰도나 안전성 평가에 적용시 유용한 점들을 요약해 보면 다음과 같다.

(a) 안전 소프트웨어의 평가 시에는 필연적으로 불확실성과 인지적 한계가 포함되는데 BBN은 이를 명시적으로 모델링 할 수 있다.

(b) 소프트웨어의 신뢰도에 영향을 미치는 변수들의 원인과 결과 관계를 명시적으로 나타내어 신뢰도의 값에 영향을 미치는 중요 변수의 확인과

그 변화 과정을 관찰 할 수 있다.

(c) 정량적으로 평가 가능한 증거들(시험 결과 등) 뿐 아니라 신뢰도에 영향을 미치는 정성적 평가 항목들(품질, 과정, 전문가의 판단 등)을 모델링 할 수 있다.

(d) 기존의 방법에 의한 평가 시에는 함축적으로 감추어져 있던 여러 가지 가정들을 명시적으로 표현함으로서 의사결정 과정의 투명성을 높인다.

(e) 모델의 직관적 그래프 형태는 복잡한 연결상태와 표면적으로 모순된 추론들을 이해하기 쉽게 만들어 준다.

(f) 결여된 자료가 있어도 예측 가능하다.

(g) "what-if" 분석의 사용으로 과정이 변화하는 효과를 예측할 수 있다.

(h) 모델을 위한 엄격하고 수학적인 의미론(semantics)을 가지고 있다.

(i) 복잡한 확률 계산을 쉽게 해 주는 도구가 개발되어 있다.

(j) 예측이나 평가를 위해 새로운 측정방법을 개발하거나 사용할 필요가 없이 기존의 다양한 기법(예를 들면, 개발과정이나 개발자의 평가 메트릭스 들 또는 확인 및 검증의 결과나 시험 등)들을 활용하여 모델링 할 수 있다.

위와 같은 BBN의 특징들은 안전 소프트웨어 신뢰도/안전성의 정량적 평가 문제를 포함해서 불확실성이 내포되고 정성적, 정량적 증거를 동시에 고려해야 하며 또 전문가의 판단이 중요한 역할을 하는 모든 형태의 문제 해결에 적용 가능한 것으로 보여진다.

제 4 장 BBN 적용 사례

제 1 절 개요

90년대 초반부터 의료, 농업, 기계 등의 분야에서는 BBN이 활용되기 시작했으나 디지털 시스템의 안전성이나 신뢰도 그리고 품질에 동 기술이 적용되기 시작한 것은 90년대 중반 이후부터이며 아직까지는 사례가 많지 않다. 본 보고서에서는 소프트웨어와 디지털 시스템의 신뢰도나 안전성 평가 그리고 이와 관련된 분야에 적용된 다음과 같은 사례를 조사하였고 그 내용은 제 2절에서 기술하였다.

- o 소프트웨어 기반 시스템의 안전성 평가[21]
- o BBN을 이용한 안전성/위험성 평가 방법론/도구 개발[22]
- o 컴퓨터 시스템 명세의 안전성 평가[20]
- o 외부 개발 디지털 시스템의 승인[23]

제 2 절 사례

1. 소프트웨어 기반 시스템의 안전성 평가[21]

가. 개요

주목적은 프로그램 가능한 안전관련 시스템의 소프트웨어에 대한 정량적인 신뢰성과 안전성 평가 방법론들에 대한 조사(investigation)이다. 중점을 둔 것은 평가에 관련된 다양한 종류의 정보들과 서로 다른 정량적, 정성적 방법론들을 어떻게 단일화된 평가 체계로 통합 할 수 있는지를 검토한 것이다.

세부적 연구 목표들은;

- o 확률적 방법론들을 비롯한 기존의 신뢰도와 안전성 평가 방법론 및 모델 검토와 소프트웨어의 평가에 적용성 여부
- o 평가에 관련된 다양한 형태의 정보에 대한 개요
- o 신뢰도와 안전성 평가에 관련된 서로 다른 증거들을 결합하는 방안에 대한

조사 및 신뢰도 평가를 위한 프로토타입 BBN 구축

o 동 분야에서의 추후 연구 항목 도출이다.

나. 주요 결과

기존 신뢰도와 안전성 평가 방법론 및 모델들이 검토되었고 신뢰도 및 안전성의 평가에 관련된 다양한 증거들을 결합하는 방안으로 BBN 기술에 대한 조사가 수행되었다. 그리고 "안전 중요 프로그래머블 시스템의 승인 과정(acceptance process of safety critical programmable system)"을 위한 BBN 구성방법에 대한 연구가 이루어졌다. 동 과제의 결과 중 BBN 구축과 관련된 주요 내용은 다음과 같다.

1) BBN 구축

가) 그래프 작성

BBN을 구성하는 노드들은 안전 시스템의 승인 과정에서 사용되고 있는 여러 정보들을 기반으로 하여 작성되었으며 이들 정보들은 다음의 4가지 출처에서 구해졌다.

(a) 개발자와 개발 과정에 관련된 정보
: 개발 방법, 코딩 표준, 품질 관리

(b) 프로그램에 관련된 정보
: 코드, 기능, 운영 모드, 프로세스 입력자료의 수와 형태, 고장 모드, 복잡성, 이해도, 시간 관점

(c) 확인 및 검증(V&V)과 시험에 관련된 정보
: 고장 모드, 복잡성, 이해도, 시간 관점, 디버깅 보고서, 시험 데이터, 시험,코드 검토

(d) 사용 이력
: 설치된 수, 전체 사용 시간, 특정 모듈의 용도, 고장 보고서, 이전 평가 결과

노드들이 확인된 다음에는, 그래프의 작성을 위해 노드들과 관련된 변수들은 다음의 세 가지 그룹으로 나누었다.

(a) 목표 노드(target nodes)
평가의 목적이 되는 노드(들)로서 정량적 변수로 표현된다.

(b) 관측가능 노드(observable nodes)
직접 관찰 가능한 노드로서 측정 가능하거나 정량화가 가능해야 한다. 객관적 측정이나 정량화가 불가능할 경우에는 전문가의 판단에 근거하여 이루어진다.

(c) 중간 노드(intermediate nodes)
한정된 정보나 "믿음(belief)"에 관련된 변수를 나타내는 노드이다.

그래프를 작성하는데 있어서 특정한 기법이나 방법론이 사용되지는 않았으며 다음과 같은 일반적 방법을 사용하였다.

(a) 목표노드에서 시작하여 이에 영향을 주는 노드를 연결하고 이들 노드에서 다시 새로운 노드로 연결하여 전체적 그래프를 완성

(b) 상위 레벨의 정보를 지닌 노드를 포함하는 BBN을 우선 만들고 이 BBN에 속한 각각의 노드에 대한 상세한 BBN을 작성

(c) 노드간 연결은 인과관계에 따른 방향으로 하고 이것이 곤란할 경우에는 추상적 개념의 노드에서 구체적 개념의 노드로 연결한다.

나) NPT 생성

작성된 노드 변수들의 NPT는 예시용으로는 적절하지만 실용적이 될 정도의 충분한 분석에 근거하지는 않았다고 되어있고 문제의 크기를 제한하기 위해 노드의 최대 상태 개수는 3으로 정하였다. 작성된 NPT(일부) 및 이들의 초기 값은 부록 1에 수록되어 있다.

다) BBN 계산

몇 가지 시험 케이스(best, worst states 등)에 대한 평가가 수행되었다. 평가 목적은 관측가능 노드에 입력된 증거가 중간 노드들의 값을 어떻게 변화시키는 지에 대한 조사였다. 그리고 BBN의 계산에는 HUGIN 도구가 사용되었다.

2) BBN 방법론에 대한 자체 평가

BBN은 소프트웨어의 신뢰도나 안전성 평가와 같이 필연적으로 불확실성을 내포하고 따라서 다양한 정량적, 정성적 증거에 의해 전문가가 판단을 내릴 수밖에 없는 분야에 적합한 방법론으로 평가되었다. 또 이 방법론은 최종 시스템의 평가 뿐 아니라 개발 과정 전 단계에 걸쳐서 여러 가지 중간 목표를 달성하는데

도 유용한 것으로 나타났다. 동 방법론의 세부적 유용성은 다음과 같다.

- o 다양한 종류의 증거를 하나의 평가 프레임으로 종합할 수 있다.
- o 측정할 수 없는 특성에 대한 정성적 평가를 정량적 평가에 이용할 수 있다.
- o 소프트웨어와 하드웨어의 다양한 관점에 적용 가능하다.
- o 시스템 개발시 개발 주기의 각 단계에 적용 가능하다.
- o 시스템 개발 초기에 설계의 입력 자료로 사용될 수 있다.
- o 개발과 설계의 최적화를 위해 사용 가능하다.
- o 안전 소프트웨어의 승인을 위한 의사결정에 사용 가능하다.
- o PSA 분석시 하나의 입력 자료로 사용 가능하다.

3) 추후 연구 항목 도출

과제의 연구 결과는 Reliability Assessment of Programmable Protection Systems(RAPPS) 과제의 입력으로 사용되었으며 따라서 RAPPS 과제는 도출된 추후 연구 항목들을 주요 연구내용으로 하고 있다. ABB-Atom, VTT와의 공동 과제인 RAPPS의 목적은 BBN 방법론을 완전한 실제 시험 케이스에 적용하므로서 그 유용성을 조사하려는 것인데 RAPPS의 주요 연구항목은;

- o Commercial Off The Shelf(COTS) 시스템의 평가에 적용. 이는 기존 COTS 시스템의 평가에 관련된 정보를 수집하고 또 COTS 시스템이 특정 품질 요건을 충족시키는지 여부를 판단하는데 BBN 방법론이 유용한지를 조사하는데 그 목적이 있다.
- o 설계 방법론에 적용. 이는 소프트웨어 시스템 공급자가 사용한 설계원칙 문서에 근거하여야 한다. 이 활동의 목적은 실제 시험케이스 평가 전 단계에서 설계가 목표한 품질을 달성할지를 평가할 때 BBN이 어떻게 사용될 수 있는지를 조사하는 것이다.
- o 시험케이스 개발에 적용. 시험 케이스는 원자력발전소의 안전계통에 사용되는 실제 프로그램을 대상으로 하며, 목적은 안전등급 시스템의 개발 각 단계에서 목표 측정치를 평가하는데 BBN이 어떻게 사용될 수 있는지 조사하는 것이다.
- o 시험 케이스의 안전성 평가에 적용. 목적은 BBN 방법론이 안전성 평가자나 규제 기관에서 어떻게 사용될 수 있는지를 조사하는 것인데 주 용도는 안전등급 디지털 시스템의 승인/거부 의사결정에 대한 지원이다.

o PSA에 사용. 이는 BBN 방법론에 의해 발견된 신뢰도 관련 자료(figure)들이 어떻게 시스템의 PSA에 사용될 수 있는지를 평가하는 것이다. 예를 들면, PSA 분석에서 고 신뢰도 요건을 필요로 하는 중요한 소프트웨어 부분을 확인할 수 있고 BBN 분석에서는 어느 정도로 이 요건이 충족되었는지를 검사 할 수 있다.

2. BBN을 이용한 안전성/위험성 평가 방법론/도구 개발[22]

가. 개요

현재 일반적으로 채택되고 있는 표준-기반(standards-based) Programmable Electronic System(PES)의 안전성 평가 접근방법에서는 결여된 안전성의 정량화를 위해 시도된 것으로 BBN을 사용하여 PES의 안전성을 추론하는 것이 SERENE 과제의 총체적인 목표이다. 이를 위해 BBN 기반의 PES 안전 변수(safety arguments)를 나타내는 방법(SERENE Method)과 이 방법을 지원하기 위해 기존의 BBN 도구를 개선한 새로운 도구(SERENE Tool)를 개발하였다. SERENE 방법에서의 핵심은 BBN 기반의 안전변수인데 이 안전변수는 안전에 관련된 증거(evidence)와 주장(claim) 사이의 연결을 제공한다. 즉 안전변수는 안전에 관련된 증거들을 결합하는 "safety case"의 한 부분으로 그 증거가 시스템이 요건을 충족할 수 있을 만큼 안전하다는 것을 증명하는데 충분하다는 것을 보여주는 것이다. IEC 61508이나 CASCADE 과제에서 개발된 GAM과 같은 평가 틀(frameworks)에는 안전에 영향을 미치는 증거들(개발 과정과 최종 제품 모두에 대하여)이 나와 있지만 이들 다양한 증거들을 전체적인 안전성 정당화(justification)로 결합하는 방법론은 어떤 표준에서도 언급되지 않고 있는데 SERENE 방법론은 이를 해결하기 위한 하나의 시도였다. 그리고 동 과제의 부가적 목적은 다음과 같다.

o 서로 다른 출처와 형태를 가진 다양한 증거들을 한 모델에 합리적으로 결합
o 안전변수 내의 약점을 확인하여 그것을 개선
o 제품과 개발 과정의 약점을 확인하여 과정 개선을 지원
o 예측과 관련된 확신의 정도를 명시화
o 시스템의 개발과 설치에 대하여 합리적인 토론과 협상이 가능한 확실한 기반 제공

나. 주요 결과

동 과제의 결과물은 복잡한 시스템의 안전성을 정량화하기 위한 의사결정 지원 방법론(serene method)과 이 방법론을 지원하는 도구(serene tool)이다.

1) SERENE 방법론

SERENE 방법론은 i) 안전변수 준비(argument preparation) ii) 안전변수 구축(argument construction) iii) 안전변수 사용(argument use)의 세가지 활동으로 구성되며 각 활동의 내용을 요약하면 다음과 같다.

가) 안전변수 준비

안전변수를 구축하는데 필요한 모든 관련된 증거를 확인하는 과정이다. 여기에는 개발과정, 최종 제품, 소요자원, 그리고 그들의 속성이 모두 포함된다. 확인된 각 증거 아이템은 안전변수를 표현하기 위한 BBN의 한 노드가 된다. 이 단계에서는 아직 안전변수들 간의 관계를 고려할 필요가 없고 단순히 중요한 항목을 리스팅하면 된다. 주의할 사항으로 관련된 표준이나 생명주기가 반드시 고려되어야 한다. 이 단계에서 수행되어야 할 내용은 다음과 같다.

(a) 목표의 확인: 예측할 대상의 확인
(b) 선정된 목표에 영향을 주는 실체와 속성의 확인
(c) 예측에 포함될 다른 부속 목표, 실체, 속성의 확인. 여기에는 개발과정이나 제품의 특징 등이 포함된다.
(d) 개발 생명주기를 검사하고 핵심 과정을 평가
(e) 문제의 여지가 있는 비공식적(informal) 변수에 대한 논의 수행
(f) 관련된 표준, 절차, 가이드를 검사

나) 안전변수 구축

이 단계는 BBN의 그래프를 구축하는 과정으로 요약하면 다음과 같다.

(1) BBN을 구성할 노드들의 확인. 이들의 후보는 변수 준비단계에서 만들어진 것들이다.
(2) 각 노드의 형태를 정의. SERENE 방법론에서 정의된 노드 형태는 Discrete labelled, Boolean, Discrete Numbered, Interval, Continuous Gaussian 이다.

(3) BBN 그래프 작성

이디엄(idiom)과 템플릿(template)을 정해진 결합규칙에 따라 사용하여 전체 BBN 그래프를 구축한다. 이디엄은 불확실성 추론의 일반적 형태를 표현하는 BBN 그래프의 특정 단위를 의미하는데, 수개의 노드가 결합되어 (자주 사용되고 일반적으로 인정되는) 추론 단위 형태(예: 측정, 통합)의 이디엄을 구성한다. 템플릿은 특정 목적을 위해서 만들어지는 하나의 BBN 조각으로 한번 작성 된 후 다른 곳에서 실증화하여 재사용 할 수 있는 형태로서 객체지향 개념에서의 크래스(class)와 유사하다.

SERENE 방법론에서는 BBN을 구축 할 때 사용하는 이디엄을 적절히 선택하는 기준과 이들을 결합하여 완전한 안전변수를 구축하는 지침을 제공한다. 이디엄의 사용은 그동안 다른 BBN 구축 작업에서 나타났던 다음과 같은 경험과 문제를 해결하기 위한 것이다.

(a) 전문가들은 서로 다른 예측 문제에 걸쳐서 매우 비슷한 추론 형태를 적용한다.

(b) 전문가들이 BBN 모델을 통해 그들의 아이디어를 표현할 때 다음과 같은 비슷한 어려움을 공통적으로 겪는다.

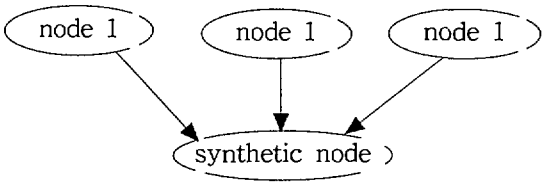
- o 연결선(edge) 방향의 결정
- o 그들이 만들려고 하는 진술(statements)이 실제로 불확실한 것인지, 그리고 그것이 BBN으로 표현될 수 있는지의 여부
- o 노드를 확인 할 때 필요한 레벨의 상세 정도
- o 상충되는 모델들이 하나의 BBN 모델로 조정될 수 있는지 여부.

이미 만들어진 이디엄을 사용하므로서 만들고자 하는 BBN의 모든 부분을 새로 작성하는 노력이 줄어들고 또 잘 정의된 이디엄의 사용으로 모델의 정확성을 높일 수 있게 되는 효과가 있다. 이러한 이디엄의 사용 결과 BBN의 개발 과정이 빨라졌고 또 품질이 향상되었다고 평가하고 있으며 SERENE에서 만들어진 이디엄은 다음의 5가지가 있다.

(가) 정의/통합 이디엄(definitional/synthesis idiom)

여러 노드를 하나의 노드로 결합/통합하는 모델링, 또는 노드들 간의 결정적/불확실한 정의를 모델링 할 때 사용하며 다음 그림과 같은 그래프 구조를 가지

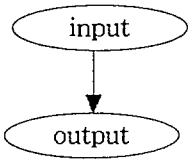
고 있다.



[그림 4.1 정의/통합 이디엄]

(나) 원인-결과 이디엄(cause-consequences idiom)

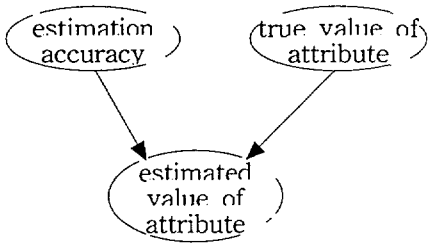
인과관계 과정과 관측된 결과들의 불확실성을 모델 할 때 사용하며 다음과 같은 그래프 구조를 가지고 있다.



[그림 4.2 원인-결과 이디엄]

(다) 측정 이디엄(measurement idiom)

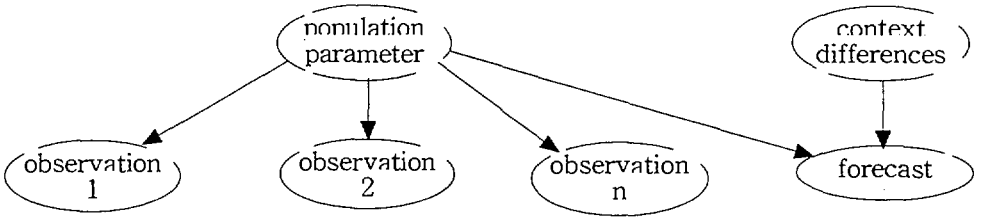
측정의 정밀도에 대한 불확실성을 모델 할 때 사용하며 다음과 같은 그래프 구조를 가지고 있다.



[그림 4.3 측정 이디엄]

(라) 귀납 이디엄(induction idiom)

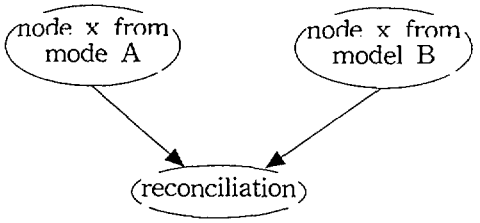
귀납적 추론과 관련된 불확실성을 모델 할 때 사용하며 다음과 같은 그래프 구조를 가지고 있다.



[그림 4.4 귀납 이디엄]

(마) 조정 이디엄(reconciliation idiom)

서로 상반되는 측정이나 예측 시스템으로부터 결과를 조정할 때 사용하며 다음과 같은 그래프 구조를 가지고 있다.



[그림 4.5 조정 이디엄]

(4) 각 노드의 NPT 정의

전문가의 지식 추출이나 실제 데이터로부터 나온 확률을 이용해 수작업으로 작성하거나 SERENE 도구에서 제공하는 "expression editor"의 기능을 사용한다. "expression editor"는 사용자 정의의 분포 함수나 정상 수학 함수를 사용하여 결정론적 함수를 정의할 수 있게 되어있으며 여기에서 다룰 수 있는 표현들은 다음과 같다.

- o Constant values
- o Discrete distribution functions
- o Continuous distribution functions
- o Arithmetic functions
- o Boolean functions
- o Comparison operators
- o If-then-else function

다) 안전변수 사용

완성된 BBN의 특정 노드에 데이터를 입력하여 이 노드가 관련 노드 및 모델 전체에 미치는 영향을 분석하는 단계이다. 이 단계에서 특히 유용한 점은 어떤 특정한 증거가 전반적 예측 결과에 가장 중요한 영향을 주는가를 시험하는 것이다. 여러 가지 시나리오와 의문이 있는 노드들의 값을 변경시켜 가면서 이를 검증 할 수 있다.

2) SERENE 도구(Tool)

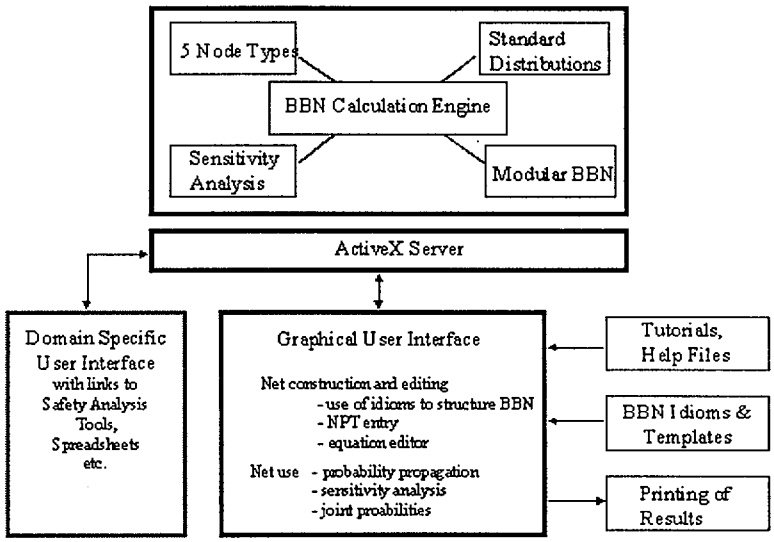
SERENE 방법을 지원하기 위한 BBN 모델링 도구로서 BBN 그래프 작성시 필요한 사용자 인터페이스, NPT의 작성, 베이시안(Bayesian) 추론 계산을 자동화 해 주고 또 SERENE 방법론에서 만들어진 idiom, templates, arguments를 포함하고 있다. SERENE 도구의 구성요소는 다음과 같다.

- o BBN 계산 엔진
- o Graphical User Interface : BBN의 준비, query, 프린트 기능
- o SERENE 방법과 도구의 도움 파일
- o Templates : 샘플 BBN으로 사용자가 BBN을 구축할 때 개조하거나 그대로 사용할 수 있다.

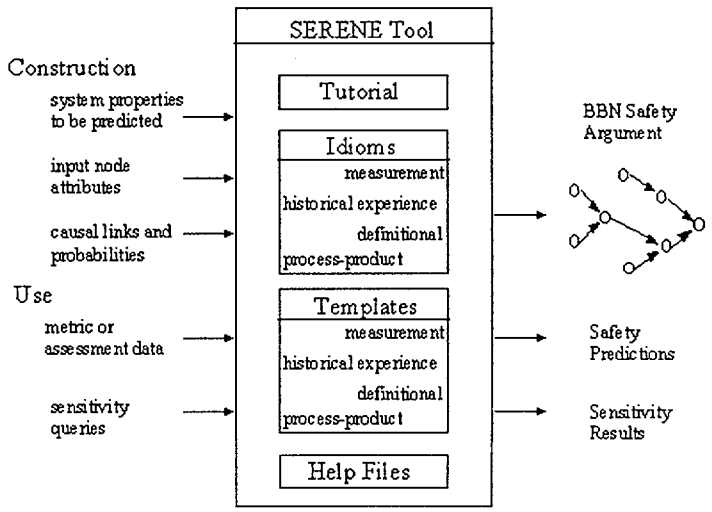
또한 이 도구의 특징은;

- o SERENE 이디엄(idiom)과 결합기능(join operation)을 사용하여 BBN을 구성' 할 수 있도록 BBN을 계층적 구조화
- o 불연속 및 연속 변수 모두를 사용할 수 있는 표준 분산
- o 민감도 분석(sensitivity analysis) 기능이다.

SERENE 도구의 구조는 그림 4.6과 같고 이 도구가 SERENE 방법론에서 수행하는 역할은 그림 4.7과 같다[22].



[그림. 4.6 SERENE 도구 구조]



[그림 4.7 안전변수 구축과 사용 활동에서의 SERENE 도구 역할]

3. 컴퓨터 시스템 명세의 안전성 평가[20]

가. 개요

컴퓨터 시스템의 검증에 관련된 문제를 해결하고 "확인을 위한 설계" 패러다

임을 제공하기 위해 만들어진 Deva(Design for Validation) 과제의 한 부분으로 "Dependability evaluation using disparate sources of evidence"가 수행되었는데 여기에서 BBN을 사용한 원전 컴퓨터 시스템의 안전성 평가연구가 부분적으로 이루어졌다. 동 연구에서 BBN 모델링의 대상이 된 부분은 원전 안전 시스템의 생명주기 초기 단계의 활동인 "컴퓨터 시스템 명세 문서의 안전성 평가" 및 이에 관련된 다른 활동과 생산된 문서들이다. 모델링 작업은 BBN 전문가와 원전 컴퓨터 시스템의 안전성 평가 전문가가 공동으로 수행하였다.

나. 주요 결과

컴퓨터 시스템 명세문서의 안전성 평가에 대한 BBN 그래프를 만들었고 "Design Process Performance" 변수에 필요한 NPT 값을 생성하였다. 생성된 그래프와 변수들 그리고 NPT 값(일부 노드들에 대하여)은 부록 1에 정리되어 있다.

1) 노드 확인 및 그래프 작성

BBN을 구성하는 노드의 확인은 안전성 평가자가 실제 평가 작업을 수행하는 활동과 일반적 소프트웨어 생명 주기의 초기 개발단계의 활동들을 근간으로 하여 이루어졌고 그래프는 "요건의 품질" "설계 과정의 이행" 그리고 목표 노드인 "컴퓨터 시스템 명세의 적절성" 노드를 각각 중심으로 하는 3개의 서브 그래프로 작성되었다. 작성된 노드와 그들의 상태 공간 그리고 그래프는 부록 1의 2에 수록되었다.

2) NPT 작성

대부분의 NPT 값은 함께 작업을 수행한 독립된 평가 전문가가 그의 전문 영역의 지식과 경험을 참조로 직접 작성하도록 하였는데 이 방법은 BBN의 규모가 작고 NPT의 차원(dimension)이 작았던 동 과제의 경우에는 실용적이라고 판명되었다. 그리고 일부 노드의 NPT는 선형 보간법(linear interpolation) 방법을 사용하여 테이블에 값을 채운 다음 전문가의 검토를 받는 방식으로 수행되었다. NPT 작성은 동 과제에서 BBN을 만들면서 가장 어려움을 겪었던 부분으로 되어있으며 이 과정에서 나타난 문제점들은 다음과 같다.

전문가들이 그들의 지식과 경험을 바탕으로 NPT의 값을 추출할 때 어려움이 있

는데 이는 방법론에 익숙하지 않다는 점과 문제의 복잡성 그리고 인간 마음의 근본적 한계성에서 주로 기인하게 되며 구체적 내용은 다음과 같다.

- o 불확실성을 확률로 표현하는데 따르는 어려움. 이 점은 다른 분야에서 확률을 다루고 있는 전문가들에게서도 나타난다. 누구나 주관적 확률 믿음(belief)을 가지고는 있으나 그들을 명시적으로 만드는데는 훈련이 되어 있지 않은 것이 주된 이유이다.
- o 모델 대상의 의존상태가 너무 복잡하고 값을 추출해 작성해야 할 테이블의 수가 너무 많기 때문에 생기는 긴장으로 인해 어려움이 있다.
- o BBN을 사용하면 전문가의 추론과정을 상세하고 명시적으로 구체화시키게 되는데, 전문가들은 이전에 이런 형태의 모델링을 한 경험이 없었기 때문에 그들의 복잡한 추론 구조와 지식을 BBN 형태로 표현하는데 따르는 어려움.
- o 기술해야 할 상황이 너무 복잡하기 때문에 해당 지식 전문가는 잠재적으로 모델을 단순화하려는 경향이 생김.
- o 전문가의 경험으로부터 올바른 추론을 끌어내는데 따르는 어려움.

3) 자체 평가

동 과제에 대한 자체 평가를 요약해 보면 다음과 같다.

- o 만들어진 BBN의 검증이 중요하다고 지적되었고 검증의 대상은 다음과 같다.
  - 일관성 문제와 같이 모델 자체에 관련된 검증
  - 만들어진 BBN이 대상 시스템에 대한 전문가의 믿음을 확실하게 포착하고 있는가에 대한 전문가의 확신에 대한 검증
  - 최종 BBN이 모델 대상이 되는 외부 세계에 대한 사실(truth)을 포착했는가에 대한 검증
- o BBN을 구축하는 전 단계에 걸쳐 모델의 품질 향상과 전문가의 노력 절감을 위해 일반적 지침과 도구의 필요성 제기 됨.
- o 복잡한 NPT의 추출이 BBN 구축의 가장 큰 문제로 나타남.
- o 장기적 관점에서 BBN의 유용성은 현재 가장 중요한 일로 다루고 있는 BBN의 결과(정량적 수치)에 한정 시켜서는 안됨. BBN의 주요 잠재적 강점은 이 기술이 전문가들의 추론에 대한 정형적 틀(formal frame)을 제공하는 것임.

4. 외부 개발 디지털 시스템 승인[23]

가. 개요

Edf 내부 평가 전문가들이 외부에서 개발된 I&C 시스템을 평가하고 승인할 때 사용되는 평가 방법론 중 하나를 BBN으로 모델링하였다. BBN 모델의 초점은 평가 전문가가 최종 결정에 도달할 때까지 여러 가지 증거에 대한 가중치를 어떻게 배분하느냐에 두어졌다. BBN은 시스템의 개발 생명주기와 Edf 자체의 I&C 시스템 구입 절차를 기반으로 전체적 구조가 설정되었는데 개발 생명주기는 폭포수(waterfall) 모델을 채택하였고 Edf의 I&C 시스템 구입 절차는 다음과 같다.

o Edf의 I&C 시스템 구입 절차

(1) 시스템 요구 명세서에 그 필요성을 작성

(2) 입찰 절차에 착수(하나 또는 2명의 구매자 선정)

(3) 선정된 공급자가 수행하는 제품 개발을 모니터: 피어 리뷰, 감사, 기타 다양한 평가 방법을 사용함

(4) 납품된 시스템에 대한 승인 시험 수행

(5) 실제 운전 상태에 들어가지 전에, 사이트에서 현장 시험 수행을 통해 전체 시스템에 대한 검증을 수행

나. 주요 결과

1) BBN 그래프 작성

구축된 BBN의 전반적 구조와 서브 그래프 "Development is suitable"은 부록1에 기술되어 있다. 안전변수 구축에서는 Edf에서 내부 평가시 사용되고 있는 방법, 기법, 정보들에 근거하였는데 그들은 다음과 같다.

- o 감사(audits)
- o 소스 코드 분석
- o 코드의 가장 민감한 부분들을 모델링
- o 공급자에 의해 수행된 시험과 확인을 검토(시험 기준의 선택, 시험 데이터 선정 등)
- o 다양한 환경 조건하에서 시스템 시험
- o 기능적 검증 시험(functional validation test) : 성능시험, 다른 I&C

시스템과 인터페이스 만족여부 시험 등

o 제조업체가 획득한 피드백 경험에 대한 문서. 특히 실제 사용되고 있는 다른 시스템에서 이미 사용되었던 소프트웨어 모듈에 대한 정보 등.

한편, BBN 구축시 크기와 복잡성으로 인하여 모델의 범위를 시스템의 기능적 안전성(functional safety)에 한정하였으며 다음과 같은 항목은 제외되었다.

o 하드웨어 논점(issues): 무작위 고장, 방사선, 고온, 진동과 같은 가혹한 환경에 대한 저항성

o 인적요인과 관련된 논점

o 다른 시스템과 장비를 포함하는 포괄적 논점

o 이미 운전중인 시스템의 변경

o I&C 시스템의 기술적 특성을 어떻게 평가하는가 하는 문제

2) NPT 작성

구축하려는 BBN의 크기와 복잡성 특히 NPT의 상태 공간과 그 값을 구하는 어려움으로 인하여 각 노드는 "yes", "no"의 두 가지 상태만을 가지도록 하였다. 작성된 각 노드의 구조는 다음과 같고 만들어진 NPT(일부)는 부록 1에 수록되어 있다.

o 모든 노드는 2개의 상태 공간을 가진다.
  : good or poor, acceptable or not acceptable, yes or no etc.

o 모든 노드는 "object x is y"와 같은 문장과 일치시킨다.
  : (예) Requirements specification is suitable

o 노드 상태에 부여할 확률 값은 {0, 0.25, 0.5, 0.75, 1}의 집합에서 선택.
  : 이유는 이보다 더 상세한 값에 대한 정당화가 어려웠기 때문. 또한 이들 값은 다음과 같은 정성적 판단 기준을 표현한다.
  {impossible, improbable, probable, quite probable, certain}

이와 같이 만들어진 NPT는 다음 표와 같은 모양을 하고 있다.

[표 4.1 NPT for verification by Edf is appropriate(for Development)]

Verif. by EDF is appropriate (for Development)	This verification assesses some key properties of the Dev: o Is complexity of developed components mastered and justified(checked by static analysis of the source code?) o Will it lead to a maintainable system?

3) 구축된 BBN 활용

5개의 시나리오를 만들어서 Edf에서 구축한 BBN의 계산을 수행하였다. 계산은 HUGIN 도구를 사용하였으며 각 시나리오별 계산 결과는 다음 표와 같다.

[표 4.2 시나리오별 계산 결과]

시나리오 번호	1	2	3	4	5
System requirements specifications are suitable	.576	.9999907	0	1	0
Specification is suitable	.419	.9999483	0	1	0
Design is suitable	.265	.9996823	0	1	0
Development is suitable	.173	.9979557	0	.9999598	0
System confirms to SRS	.138	.9952124	.113	.9983177	.243
System can ensure safety(System confirms to real needs)	.112	.9670761	.138	.9822346	.279

(주)

시나리오 1 : 아무런 증거도 입력하지 않은 경우

시나리오 2 : 증거가 입력되지 않고, 확인(verification)은 좋은 결과로 가정한 경우

시나리오 3 : 부정적 증거가 입력되고, 확인과 시험은 좋은 결과로 가정한 경우

시나리오 4 : 긍정적 증거가 입력된 경우

시나리오 5 : 부정적 증거와 긍정적 증거가 함께 입력된 경우

Edf 자체의 분석에 따르면 시나리오 별 계산 결과는 모두 일관성을 보여주고 있으며 전문가의 직관과 일치하는 것으로 나타났다[23]. 그러나 이들 결과가 비록 숫자로 나타나기는 하였으나 이들은 단지 전문가의 믿음에 대한 정교한 표현에 지나지 않으며 이를 실제 확률로 받아들여서는 안 된다는 점과 이들 숫자는 단지 비교 목적으로만 사용되어야 한다는 점도 지적되었다.

4) 자체 평가

수행된 과제에 대한 Edf의 평가는 다음과 같다.

- o 이해하기 쉽고 전문가들 사이의 의사소통 질을 향상시켰음
- o 평가 전문가들 사이의 토론을 용이하게 만들었음.
- o 프로토타입 BBN의 구축은 컴퓨터 기반 안전중요 시스템의 평가에 있어서 전문가의 지식을 명시화하고 평가하는데 좋은 기회가 되었음.
- o 시스템의 설계 과정에서 가장 중요한 변수를 결정하는데 도움을 줄 수 있을 것으로 봄
- o 동 방법론은 신뢰할만한 예측을 가능하게 할 것이지만 이러한 목표를 달성하기 위해서는 많은 보정(calibration) 시간을 필요로 할 것임(예를 들면 방법론에 의하여 예측된 값과 평가된 시스템의 실제 관찰된 안전성 레벨의 비교에 의한 보정).

제 3 절 사례에 대한 분석

사례 조사에서 나타난 연구들의 주목적은 (i)안전성과 신뢰도의 평가 케이스에 BBN을 시험적으로 적용해 보고 그 가능성을 검토하는 것과 (ii) 프로토타입 BBN 구축을 통해 적용과정에서 나타난 문제점들을 확인하고 이들을 해결하기 위한 방안을 모색하는 것이 주로 되어있다. 이들 가능성과 문제점들은 다음과 같다.

1. 안전성과 신뢰도의 정량적 평가에 적용 가능성

BBN은 고신뢰도를 요구하는 안전 소프트웨어의 신뢰도/안전성 평가와 같이 필연적으로 불확실성을 내포하고 또 의사결정을 위한 충분한 증거를 얻는 것이 현실적으로 불가능한 문제의 해결에 적합한 것으로 나타났다[20][21][22][23][28][29].

현재, 이와 같은 문제에 대한 해결은 전문가가 다양한 형태(정성적 형태 및 정량적 형태)의 관련 증거들을 근거로 하여 정성적으로 판단을 하는 것이 유일한 방안으로 되어있는데, BBN은 이들 다양한 형태의 증거들을 정형적으로 결합하고 평가시 수반되는 불확실성들을 명시적으로 도입하여 그 결과를 정량화하는

방안을 제시할 수 있다는 점이 가장 큰 가능성으로 나타났다. 또 실제로 전문가에 의하여 평가시 사용되고 있으나 기존의 정량적 평가 기법들로는 구현하기 어려운 내용들, 예를 들면 측정할 수 없는 신뢰도 관련 특성들(개발팀의 품질 등급 등)에 대한 정성적 평가를 일관된 평가 프레임 안에서 정량적 평가에 도입할 수 있다는 점도 타 방법론에 비하여 장점으로 나타났다. 이 외에도 (i) 복잡하고 애매함이 함축된 평가 내용들이 BBN의 그래프와 NPT를 통하여 명시적이고 이해하기 쉽게 나타나므로 평가 전문가들이나 관련 전문가(개발팀, 시험팀)들 사이의 의사소통과 토론이 용이하게 되고, (ii) 기존의 소프트웨어 공학에서 사용되는 각종 메트릭스 또는 확인 및 검증 결과나 시험 결과를 활용하여 모델링 할 수 있어 새로운 측정방법을 추가로 개발할 필요가 없으며, (iii) 결여된 자료가 있어도 결과 값을 구할 수 있어 개발 초기단계부터 사용이 가능하며, (iv) 모델에 사용된 각 변수들이 상황에 따라 어떻게 변화하는지 쉽게 알 수 있어 신뢰도에 중요한 영향을 주는 변수들을 확인할 수 있다는 점과, (v) 복잡한 확률 계산을 용이하게 해주는 도구들이 있다는 점등이 사례에서 나타난 BBN의 유용성들이다.

2. 적용시 나타난 문제점과 이들의 해결 전망

사례의 각 과제에서는 BBN을 규모가 어느 정도 이상 되고 복잡한 실제의 문제해결에 적용할 때 어떤 문제점들이 발생하는가를 확인하고자 하였고 또 이들 나타난 문제점들을 차후 과제의 입력 사항으로 하여 그 해결 방안을 모색하였다. 각 사례에서 프로토타입 BBN을 구축하면서 발표된 BBN 구축시 문제점들을 정리해 보면 다음과 같다.

가. 그래프 작성

그래프 작성에 필요한 노드의 생성은 주로 전문가의 지식과 관련 표준에 나타난 항목에 근거하여 이루어졌는데 전체적 구성은 현재 인정되고 있는 소프트웨어 엔지니어링의 여러 원칙을 기반으로 하였기 때문에 유사한 형태를 지니고 있으나 노드가 구체적으로 될수록 케이스별로 달라지는 모습을 보였다. 이는 케이스별 전문가의 전문성 유형이나 또는 전문가가 속한 집단의 내부 가이드가 상이한데서 발생한 것으로 만들어진 그래프의 공식적 검증이 필요하다는 데는 모두 같은 의견을 제시하고 있다. 또 하나의 나타난 문제점은 노드간의 종속성을 밝혀내는 것으로 BBN 상의 노드간 연결은 주로 인과관계에 의하는 것이 위주가 되

지만 실제 상황에서는 모든 변수가 이런 인과관계로 설명되지 않기 때문에 발생하는 것으로 각 사례를 보면 이 부분도 역시 전문가의 판단에 의해 처리되었고 일관되고 공식적인 방법이 아직 정립되지 않고 있었다. SERENE 과제에서 이 문제를 해결하기 위한 시험적 시도를 하였는데 여러 경우의 안전성 평가에 공통적으로 나타나는 원소단위의 노드 결합을 생성하고 이를 이용하여 몇 가지 결합규칙과 객체지향 개념을 활용하여 전체 BBN 그래프를 구성하는 방안을 모색하였는데 그래프 작성 시간의 단축과 모델의 품질 향상에 어느 정도 기여한 것으로 나타났다[22].

나. 노드 확률 테이블 생성

모델상의 변수(노드)가 확인되면 각 노드들에 대하여 상태 공간을 설정하고 이들에 대하여 조건부 확률 값을 부여해야 하는데 이 부분은 조사된 모든 사례에서 가장 어려움이 많았던 것으로 나타나 있다. 이런 어려움의 주된 이유로는 분야 전문가가 BBN 방법론에 익숙하지 않다는 점, 문제의 복잡성 그리고 인간의 근본적인 지적 한계성에서 주로 기인하는 것으로 지적되었다[21].

NPT의 작성 과정에서 나타난 또하나의 어려운 점은 과도한 작업 양인데 예를 들면 많은 수의 상태를 가진 노드의 NPT 작성을 들 수 있다. 이들 많은 상태 노드에 대하여 전문가가 일일이 그 상태 값을 작성하는 데 많은 작업시간이 소요된 것으로 나타났고 또 제한된 자원(시간, 인력 등)내에서 작업을 완료하기 위해 노드의 상태 수를 간략화 시켜 모델의 정확성을 저하시키는 경향도 나타났다. SERENE 과제에서는 이 부분을 해결하기 위해 자주 사용되는 일반 통계 분포나 범위 값의 자동 분배 등의 기능을 NPT 편집기와 같은 도구에 구현하는 시도를 하였는데 이와 같은 기능은 NPT 전체의 작성시간을 줄일 수 있는 방안으로 나타났다. 이러한 기능 개선에 관련된 SERENE 과제 결과의 일부는 과제의 공동참여자에 의해 만들어진 HUGIN 도구에 반영되었으며, 앞으로도 이러한 작업의 용이성, 편리성 기능은 각종 BBN 구축 소트트웨어 도구들이 업그레이드할 내용의 주요부분으로 되어있다.

다. 계산

BBN의 계산은 매우 복잡하나 효과적인 계산 알고리즘의 발견과 이를 구현한 소프트웨어의 개발로 실용적 측면에서 문제가 없는 것으로 나타났고 군사용 운송장비의 신뢰도 평가 개선 연구에서는 현재 개발되어 사용되고 있는 BBN용 소

프트웨어 도구가 현실적 문제를 해결하는데 충분한 기능을 갖춘 것으로 나타났다[12][21][22][27].

라. BBN 검증

최종적으로 만들어진 BBN에 대한 검증의 중요성이 지적되었다[20]. 일반적으로 BBN의 검증은 다음의 세 가지 관점에서 이루어진다.

(i) 일관성 문제와 같은 모델 자체에 관련된 검증

(ii) 만들어진 BBN이 대상 시스템에 대한 전문가의 지식을 확실하게 포착하고 있는가에 대한 검증

(iii) BBN이 모델의 대상이 되는 외부 세계에 대한 사실을 적절하게 표현하고 있는가에 대한 검증

모델 자체에 대한 검증 문제는 이미 이론적으로 확립되어 있으며 또 일관성이나 상태 확률의 합과 같은 많은 자체 검증 부분을 BBN 구축 도구에서 자동으로 지원하기 때문에 어려움이 없는 것으로 나타났다. 그러나 BBN이 전문가의 지식을 확실하게 포착했는가에 대한 검증과 외부 세계에 대한 사실을 적절하게 표현했는가에 대한 검증은 아직까지는 적절한 방안이 마련되지 못하고 있는 실정이다. (ii)의 검증 부분은 BBN을 작성하지 않은 다른 전문가가 만들어진 BBN을 평가하는 방법과 (iii)의 검증을 통해 간접적으로 검증하는 방안이 가능하다. 다른 전문가가 BBN을 평가하는 경우에는 BBN 작성의 규칙성, 정형성이 많은 도움을 줄 수 있는데, SERENE 과제에서는 이를 위해 전문가들이 어느 정도 규칙적이고 정형적 방법으로 BBN을 구축할 수 있도록 여러 가지 방안을 제시하였다. 이 부분에 대한 연구가 더 진행된다면 일반적으로 인정되는 BBN 평가 지침이나 기준이 가능할 것으로 보여진다. (iii)의 검증 경우에는 BBN의 대상이 된 시스템이 실제로 운영되어 그 운전 결과와 BBN 모델의 계산 결과를 비교하는 것이 최상의 방안이나 현재까지는 그런 시스템이 없기 때문에 이 부분의 검증 문제는 단기간 내에 해결되기는 어려울 것으로 보여진다.

제 5 장 결론

소프트웨어가 가지는 고유 특성으로 인하여 시험이나 신뢰도 성장 모델과 같은 직접적 신뢰도 평가 방법들 그리고 개발 과정의 품질에 의하거나 설계 다양성에 기초한 fault tolerance 방법 및 정형적 방법론과 같은 간접적 평가 방법론들로는 고 신뢰도를 요구하는 안전 소프트웨어의 신뢰도 평가 및 증명이 어렵다. 또한 이런 결과를 바탕으로 현재 채택되고 있는 방법론 즉, 다양한 증거를 결합하여 전문가가 종합적으로 판단을 내리는 방안에도 문제가 내포되어 있어 여러 분야의 규제나 표준에서 정량적 평가를 도입하기 위한 시도를 하고 있는 것이 현재의 실정이다. 이와 같은 상황에서 BBN기술은 아직 초기 단계이기는 하나 소프트웨어의 신뢰도와 안전성을 정량적으로 평가하는 분야에서 몇 가지 유용한 결과를 보여주고 있다. 지금까지 연구 결과를 토대로 소프트웨어의 정량적 신뢰도 평가에 유용한 BBN 방법론의 장점 및 효용성을 요약해 보면 다음과 같다. [26-30]

- o 평가에 있어서 필연적으로 수반되는 불확실성과 무지를 명시적으로 모델링하고 소프트웨어의 신뢰도에 영향을 미치는 변수들의 원인과 결과 관계를 명시적으로 나타낸다.
- o 다양한 유형의 정보를 결합할 수 있게 해 준다. BBN이 다룰 수 있는 다양한 유형의 증거들은 다음과 같은 것들이 있다.
  - 다양한 과정(process)과 제품(product) 변수들
  - 경험적 증거와 전문가의 판단
  - 실제의 인과관계
  - 불확실성
  - 불완전한 정보 그리고 결여된 정보
- o 감추어져 있던 여러 가지 가정들을 명시적으로 표현함으로서 의사결정 과정에 투명도와 감사도(auditability)를 높인다.
- o 모델의 직관적 그래프 형태는 복잡한 연결상태와 표면적으로 모순된 추론들을 이해하기 쉽게 만들어 준다.
- o 결여된 자료가 있어도 예측할 수 있는 능력이 있다.
- o "what-if" 분석의 사용으로 과정이 변화하는 효과를 예측할 수 있다.
- o 주관적 또는 객관적으로 파생된 확률 분포들을 사용할 수 있다.

o 모델을 위한 엄격하고 수학적인 의미론(semantics)을 가지고 있다.
o 복잡한 확률 계산을 쉽게 해 주는 도구가 개발되어 있다.
o 예측이나 평가를 위해 새로운 측정방법을 개발하거나 사용할 필요가 없이 기존의 다양한 기법(예를 들면, 개발과정이나 개발자의 평가 메트릭스 들 또는 확인 및 검증의 결과나 시험 등)들을 활용하여 모델링 할 수 있다.

물론 사례 분석에서 나타난 바와 같이 실용적 문제 해결을 위해 모델링할 때 그래프의 작성과 NPT 작성 부분에 발견된 문제점들이 있기는 하지만 이런 문제점들을 해결하기 위한 연구가 각 분야에서 현재 활발히 진행 중이고 실제 케이스에의 적용도 계속 진행되고 있으므로 실용성 있는 결과가 나올 것으로 기대된다.

또한 BBN은 소프트웨어의 평가 및 검증 분야 뿐 아니라 소프트웨어 엔지니어링을 비롯한 다른 분야에서도 유용한 도구로 인식되고 있으며 (i) 소프트웨어의 위험도 평가에 적용 (ii) COTS 시스템의 평가에 적용, (iii) 설계 방법론에 적용, (iv) 시험 케이스 개발에 적용, (v) PSA와의 연계(PSA의 입력 데이터 생성) 등과 같은 다양한 목적으로 연구가 시도되고 있다.

결론적으로 디지털 시스템 및 안전 소프트웨어의 정량적 신뢰도 평가 영역에서 BBN 기술의 현 위치와 전망을 정리하면 다음과 같다.

(1) 중 단기적 관점에서 보면, 고 신뢰도 소프트웨어의 평가를 위한 확립된 기술이 없는 현 상태에서는 전문가의 판단(규제/표준의 틀에 의해 신뢰도에 영향을 주는 여러 증거를 종합한 판단)에 의존하여 안전 소프트웨어의 신뢰도를 평가하는 체제가 계속 될 것이다. 따라서 이런 체제에서는 전문가의 정성적인 판단과정을 정량적으로 정형화 할 수 있고 여러 형태의 증거들을 단일 모델 안으로 체계적으로 결합 할 수 있는 BBN과 같은 방법론이 인허가에 도움을 줄 수 있다.
(2) 장기적 관점에서는, 신뢰도에 영향을 주는 요인의 인과관계와 영향 강도(정량화)가 규명된다면 기존의 하드웨어 신뢰도 분석 적용되는 고장수목(fault tree) 분석법이나 RBD(Reliability Block Diagram)와 같이 소프트웨어 신뢰도 평가 분야에서 하나의 정립된 기법으로 될 가능성이 높다.

(3) 본 보고서에 언급되었던 BBN 모델 작성시 발생하는 제반 어려운 점들은 그 해결에 어느 정도 시간을 필요로 할 것으로 보인다. 그러나 이를 해결하는 과정에서 지금까지는 묵시적, 정성적 또는 비 정형적으로 처리되던 소프트웨어의 신뢰도 평가에 관련된 여러 가정들이 명시적으로 정형화되어 개발자와 평가자 또는 규제자 간의 논의에 도움을 주고 의사결정의 투명도와 감사도를 높일 수 있을 것으로 본다.

참고문헌

[1] 강현국외, 확률론적 안전성 평가에서의 디지털 계측제어 계통 고유 현안 분석,KAERI/AR-560/2000, 한국원자력연구소, 2000.

[2] 박진균 외, 소프트웨어 신뢰도의 정량적 평가 기법에 대한 고유 현안 분석, KAERI/AR-565/00, 한국원자력연구소, 2000

[3] M.R. Lyu, Handbook of Software Reliability Engineering, Computer Society Press, 1995

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

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

[6] 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

[7] A.A. Abdel-Ghaly and Chan et al., Evaluation of Competing Software Reliability Predictions, IEEE Transactions on Software Engineering, 12(9), 1986

[8] S.Brocklehurst and B. Littlewood, New Ways to get Accurate Reliability Measures, IEEE Software, 9(4), 1992

[9] IEEE Std. 982.1, IEEE Standard Dictionary of Measures to Produce Reliable Software, IEEE, 1988

[10] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks for Plausible Inference, Morgan Kaufman, 1988

[11] S.L. Lauritzen and D.J. Spiegelhalter, Local computation with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion), J.R. Statis. Soc. Series B, 50, No2, 1988

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

[13] RTCA. Software considerations in airborne systems and equipment certification, DO-178B, Requirements ad Technical Concepts for Aeronautics, 1992.

[14] J.C. Knight and N.G. Leveson, Experimental evaluation of the assumption of independence in multiversion software, IEEE Trans Software Engineering, 12(1), 1986

[15] J.C. Knight and N.G. Leveson, An Empirical Study of Failure Probabilities in Multi-version Software, in Proc. 16th Int. Symp. On Fault-Tolerant Computing, 1986

[16] P.A. Anderson, An Evaluation of Software Fault Tolerance in a Practical System, Proc. 15th Int. Symp on Fault-Tolerant Computing, 1985

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

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

[19] F.V.Jenson, An Introduction to Bayesian Networks, UCL Press, 1996

[20] B. Littlewood and L. Strigini, Examination of Bayesian Belief Networks for Safety Assessment of Nuclear Computer-based Systems, ESPRIT DeVa Project 20072, 1998

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

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

[23] Marc Bouissou, Assessment of a Safety-Critical System Including Software: A Bayesian Belief Network for Evidence Sources, Reliability and Maintainability Symposium, 1999[23]

[24] Lorenzo Strigini, Engineering judgement in reliability and safety and its limits: what can we learn from research in psychology, CSR tech-report,1996

[25] J. Reason, Human error, Cambridge University Press, 1990

[26] N. Fenton, M. Neil, Software Metrics and Risk, FESMA 99, 2nd European Software Measurement Conference, 1999.

[27] Bayesian Belief Networks, www.agena.co.uk

[28] N. Fenton, Predicting software quality using Bayesian Belief Networks Proceedings of 21at Annual Software Engineering Workshop, 1996

[29] M. Neil, B. Littlewood, N. Fenton, Applying Bayesian Belief Networks to System Dependability Assessment, Proceedings of Safety Critical Systems Club Symposium, 1996

[30] N.E. Fenton and S.L. Pfleeger, Software Metrics: A Rigorous & Practical Approach, (2nd Edition), International Thomson Computer Press, 1996.

[31] http://members.aol.com/JohnDMusa/users.htm

[32] D.M. Karydas, A.C. Brombacher, Reliability certification of programmable electronic systems, Reliability Eng. & Systems Safety, Vol.66, 1999.

[33] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, IEX, 1997.

[34] ANSI/ISA SP84, Applications of safety unstrumented systems for the precess industry, instrument society of America, 1996.

부록 1. 컴퓨터 및 소프트웨어의 신뢰도/안전성 평가와 관련된 BBN

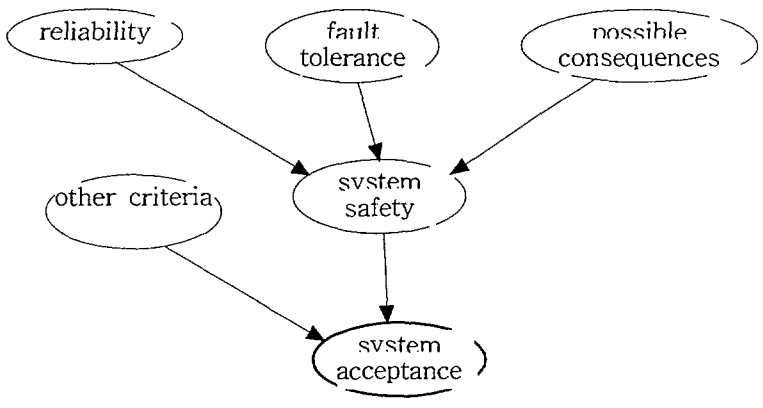
부록 1에서는 사례에 나타난 소프트웨어 및 디지털 시스템의 안전성/신뢰도 평가 관련 BBN 자료를 수록하였다.

1. 소프트웨어 품질(Quality) 모델 - from HALDEN project

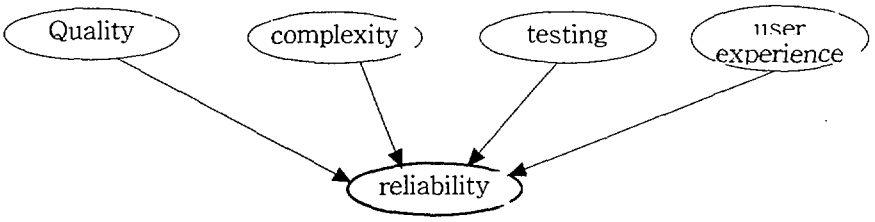
1.1 노드 정의

목표 노드인 "system acceptance"에 중요한 영향을 주는 요소 중 정치적 압력이나 여론 등과 같은 요인(아래 그림의 "other criteria")을 제외하면 시스템의 안전성(system safety)이 가장 중요한 요소로 확인되었다. 그 다음 단계로 이 시스템 안전성에 중요한 영향을 주는 요인들을 확인(아래 그림에서는 reliability, fault tolerance, possible consequences의 3개 요인)하여 초기 그래프를 작성하고, 그 다음 단계로 초기 그래프에 있는 신뢰도(reliability) 노드에 영향을 주는 요인들을 파악하는 방식으로 전체 망에 포함될 노드들을 확인하였다. 초기 BBN 그래프 중 신뢰도 노드에 대해서는 관련된 모든 그래프가 작성되었고 NPT는 신뢰도 노드의 모 노드 중 하나인 품질(Quality)에 속한 노드들에 대하여 작성되었다.

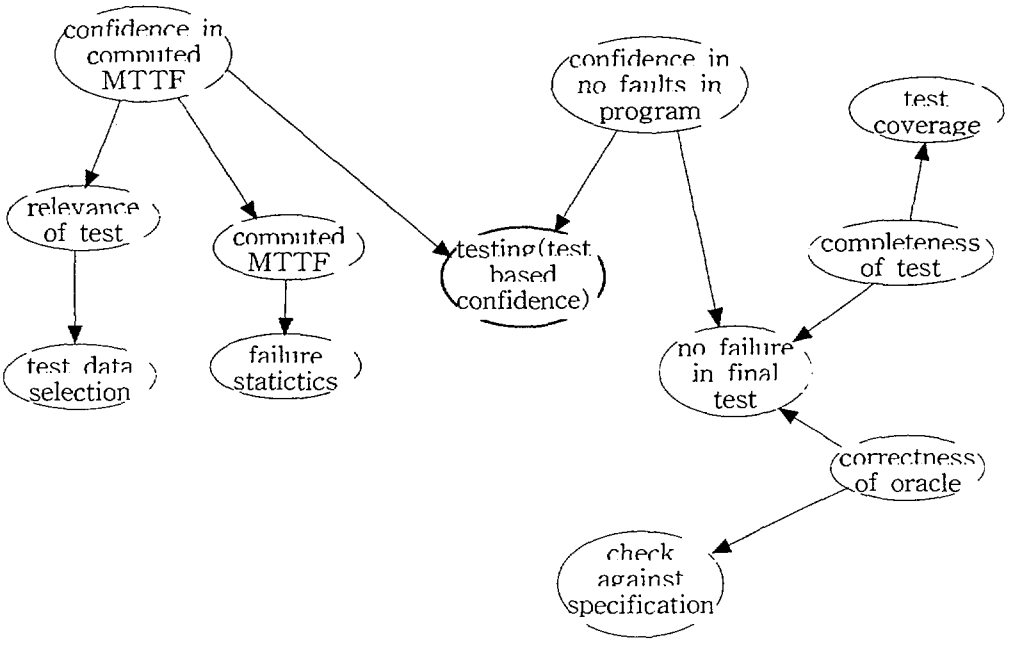
1.2 그래프(graph, topology)



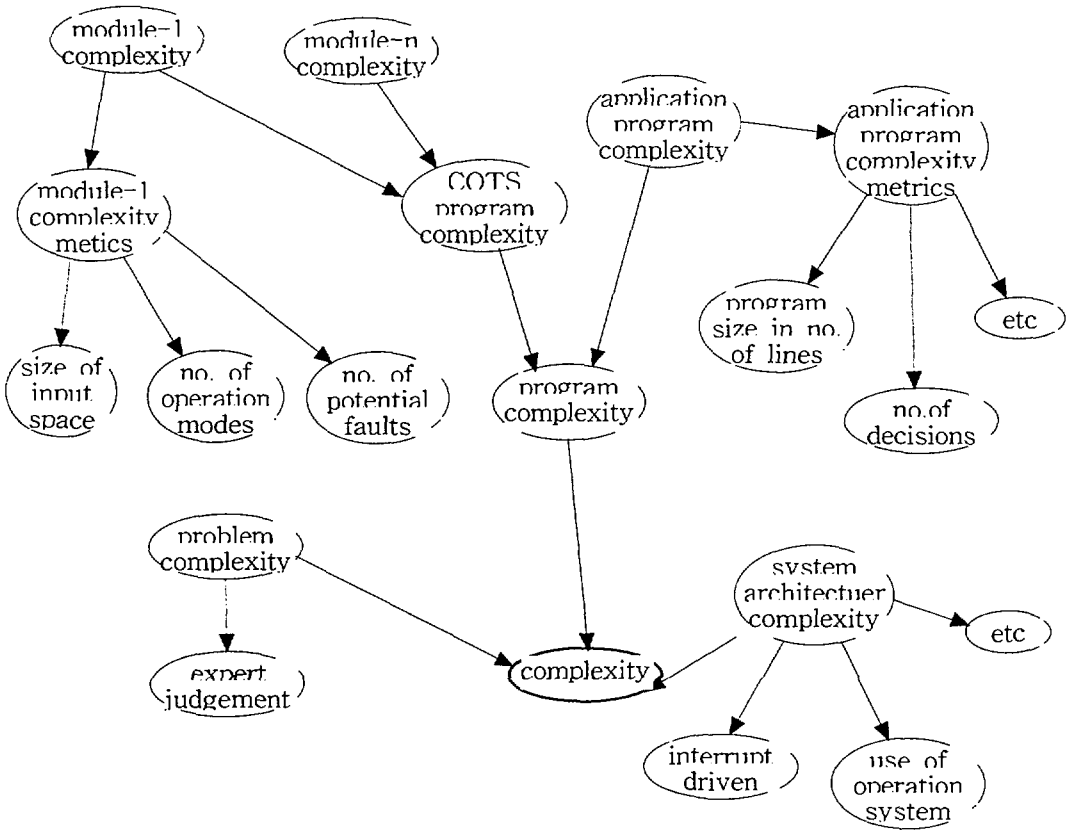
[안전성 평가를 위한 초기 BBN]



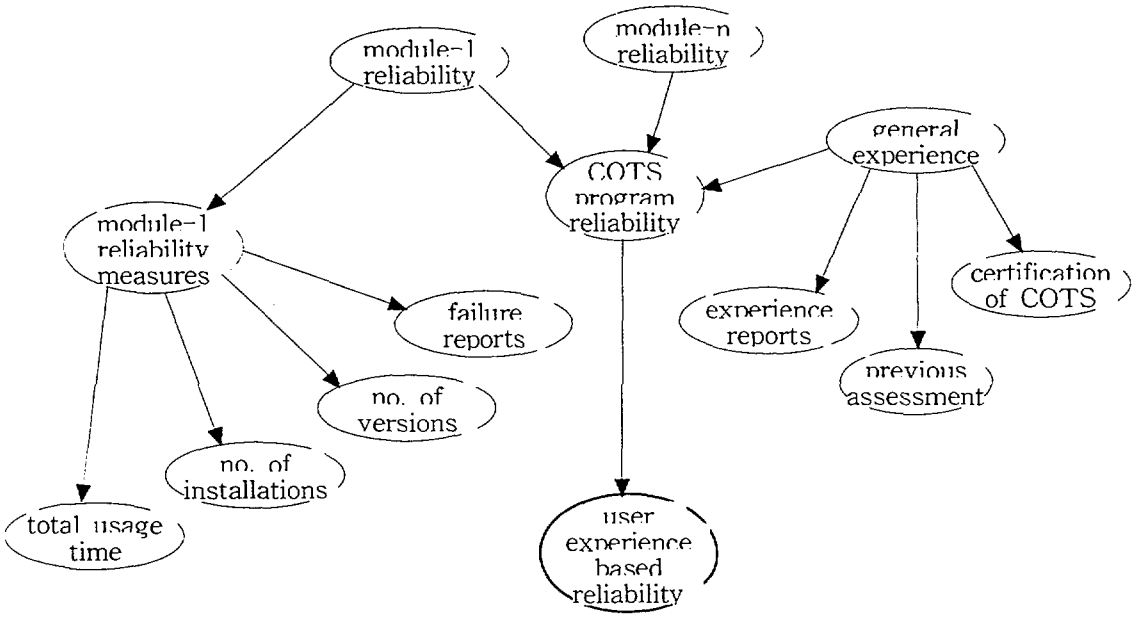
[BBN for Reliability]



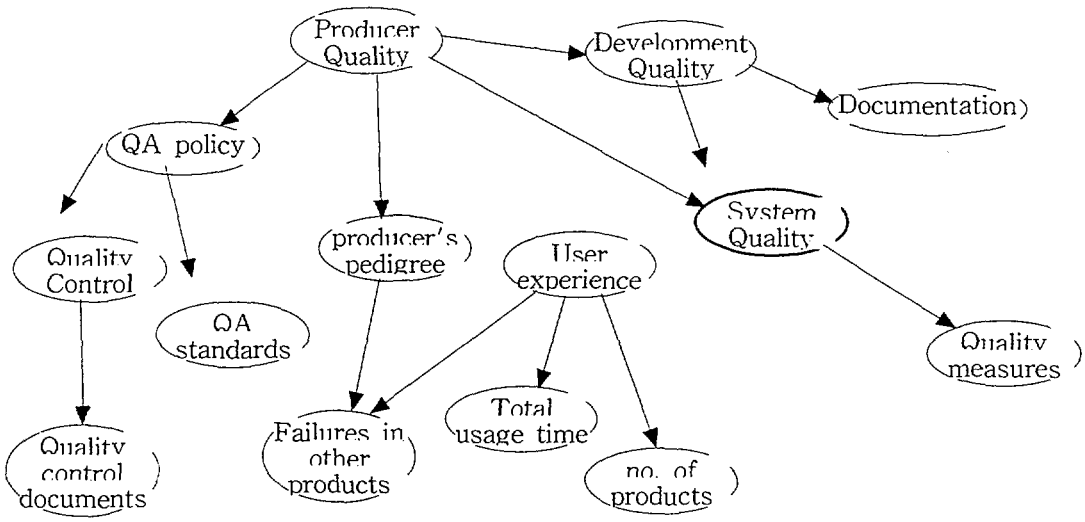
[BBN for Testing]



[BBN for System Complexity]



[BBN for reliability assessment based on user experience]



[BBN for System Quality]

1.3 노드 확률 테이블

1) 노드 변수 상태(states)

Node variables	States of variables		
Quality-control-documents	none	partly	completed
QA-standards	none	generic	detailed
Failure-in-other-products	>50%	10% - 50%	<10%
Number-of-products	<10	10 - 100	>100
Usage-time	100 h	100 - 10000 h	>10000 h
Documentation	bad	acceptable	excellent
Quality-measures	<0.1	0.1 - 0.8	>0.8
Quality-control	strict	lousy	
QA-policy	bad	acceptable	excellent
Producer's-pedigree	low	medium	high
System-quality	low	medium	high
Development-quality	low	medium	high
User-experience	low	medium	high
Producer-quality	low	medium	high

2) 각 노드 별 prior 값

Producer Quality	
Low	0.25
Medium	0.5
High	0.25

User Experience	
Low	0.3
Medium	0.4
High	0.3

Producer Pedigree given the Producer Quality			
	Producer Quality		
	low	medium	high
low	0.9	0.2	0.0910
medium	0.1	0.6	0.1818
high	0.0	0.2	0.7272

Development Quality given the Producer Quality			
	Producer Quality		
	low	medium	high
low	0.9	0.2	0.0910
medium	0.1	0.6	0.1818
high	0.0	0.2	0.7272

QA policy given the Producer Quality			
	Producer Quality		
	low	medium	high
low	0.9	0.2	0.0910
medium	0.1	0.6	0.1818
high	0.0	0.2	0.7272

System Quality given the Producer Quality and the Development Quality									
	Producer Quality								
	low			medium			high		
	Development Quality								
	low	medium	high	low	medium	high	low	medium	high
low	1.0	0.9	0.7	0.9	0.2	0.1	0.7	0.2	0.0
medium	0.0	0.1	0.2	0.1	0.6	0.2	0.2	0.6	0.1
high	0.0	0.0	0.1	0.0	0.2	0.7	0.1	0.2	0.9

Usage Time given the User experience			
	User Experience		
	low	medium	high
<100 h	0.8	0.1	0.1
100-10000h	0.1	0.8	0.1
<10000 h	0.1	0.1	0.8

Number of Products given the User experience			
	User Experience		
	low	medium	high
<100 h	0.8	0.1	0.1
100-10000h	0.1	0.8	0.1
<10000 h	0.1	0.1	0.8

Failures in other Products given the Producer's Pedigree and the Use Experience									
	Producer's Pedigree								
	low			*medium*			*high*		
	User Experience								
	low	medium	high	low	medium	high	low	medium	high
low	0.3	0.2	0.0	0.9	0.2	0.1	0.9	0.2	0.0
medium	0.4	0.2	0.1	0.1	0.6	0.2	0.1	0.6	0.1
high	0.3	0.6	0.9	0.0	0.2	0.7	0.0	0.2	0.9

Quality measures given the System Quality			
	System Quality		
	low	medium	high
< 0.1	0.8	0.1	0.1
0.1 - 0.8	0.1	0.8	0.1
> 0.8	0.1	0.1	0.8

Documentation given the Development Quality			
	Development Quality		
	low	medium	high
bad	0.9	0.2	0.0910
acceptable	0.1	0.6	0.1818
excellent	0.0	0.2	0.7272

Quality Control given the QA Policy			
	QA Policy		
	low	medium	high
strict	0.1	0.5	0.8
lousy	0.9	0.5	0.2

Quality Control Document given the Quality control		
	Quality Control	
	low	lousy
none	0.0	0.5
partly	0.2	0.5
completed	0.8	0.5

QA Standards given the QA Policy			
	QA Policy		
	bad	acceptable	excellent
none	0.5	0.3	0.1
generic	0.3	0.5	0.2
detailed	0.2	0.2	0.7

2. Design Process Performance 모델 - from Deva project

2.1 노드(variable) 정의

3개의 주 노드(변수)들은 다음과 같이 정의.

(1) Quality of requirements

(2) Design process performance

(3) Adequacy of computer systems specification(goal node)

이들 세 개의 주 노드들은 각각 서브 그래프를 가지고 있으며 이들 서브 그래프의 중심이 된다. 각 서브 그래프의 노드들은 시간적 또는 인과적 순서에 의해 정돈된 노드들로 구성되어 있다. 3개의 서브 그래프에 속한 노드들의 정의는 다음과 같다.

1) Requirement document sub-graph

- o Quality of requirements (Poor, OK, Good)
- o Anticipation of plant & system failure mode & hazards
  (Sketchy, Satisfactory, Detailed)
- o Independent hazard analysis report (Superficial, Average, Thorough)
- o Adequacy with respect to application safety requirements
  (Unsatisfactory, Satisfactory)
- o Plant Experts' safety assessment report
  (Superficial, Average, Thorough)
- o Completeness & Correctness (No, Yes)
- o Licensee Verification Thoroughness (Superficial, Average, Thorough)
- o Understandability by manufacturer - Absence of ambiguity
  (Inadequate, Satisfactory, Good)
- o Manufacturer Verification report (Superficial, Average, Thorough)

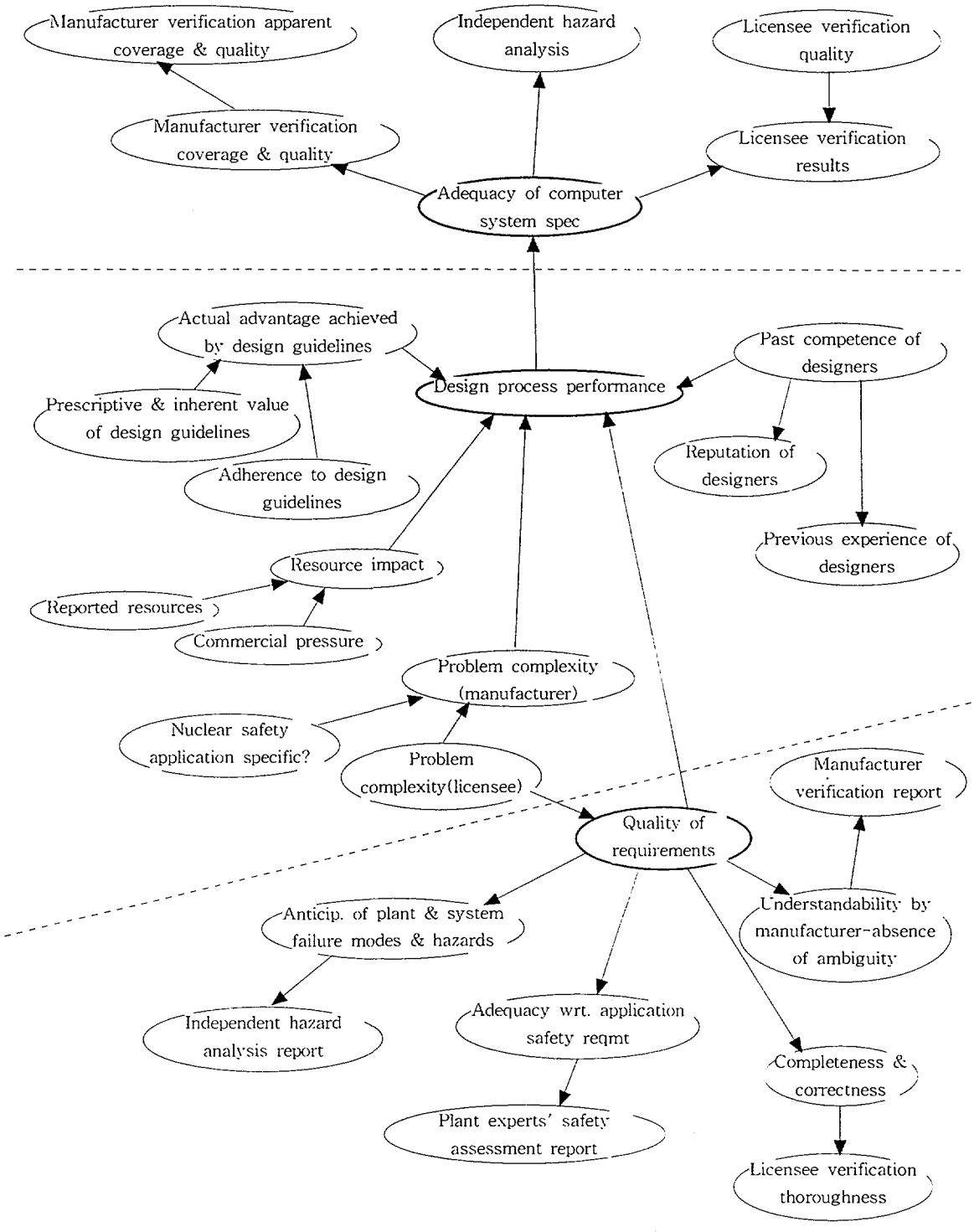
2) Design Process sub-graph

- o Design process performance (Unsatisfactory, OK, Good)
- o Actual advantage achieved by design guidelines (No, Yes)
- o Prescriptiveness & Inherent value of design guidelines(Low, Good)
- o Adherence to design guidelines (No, Yes)

o Problem complexity(manufacturer)
(Complex/difficult, Moderate, Simple/Easy)
o Past competence of designers (Low, Average, Good)
o Previous experience of designers
(0 Similar systems licensed, 1 Similar system licensed,
>1 Similar system licensed)
o Reputation of designers(Doubtful, Average, Good)
o Resource impact(Inadequate, Adequate)
o Commercial pressure(High, Low)
o Reported resources(Inadequate, Adequate, More than adequate)

3) Computer system specification sub-graph
o Adequacy of computer system specification
(Awful, Unsatisfactory, OK, Good, Wonderful)
- Awful: 명세서에 문제가 있다는 것이 분명하게 드러남
- Unsatisfactory: 명세서에 문제가 있는 것처럼 보이나 그것을 찾아내거나 진단하는 것이 쉽지 않음.
- OK: 명세서에 안전성에 관련된 문제는 없는 것으로 확신됨. 그러나 명세서의 표현 형태 부적절로 인하여 이를 분명하게 증명하거나 높은 확신도로 기술 하기는 어려움.
- Good: 안전성과 관련된 문제가 없음. 그리고 고 품질의 표현형태에 의해 이 사실이 분명함
- Wonderful: 명세서의 표현 형태가 분명하고 안전성에 관련된 문제가 전혀 없다는 것이 분명함
o Manufacturer verification coverage & quality
(Unsatisfactory, OK, Good)
o Manufacturer verification apparent coverage & quality
(Unsatisfactory, OK, Good)
o License verification results (0 issues, A few issues, Many issues)
o License verification quality (Low, OK, high)
o Independent hazard analysis (No unresolved issues, Minor unresolved issues, Serious unresolved issues)

2.2 그래프 (Structure of the network topology)



2.3 노드 확률 테이블

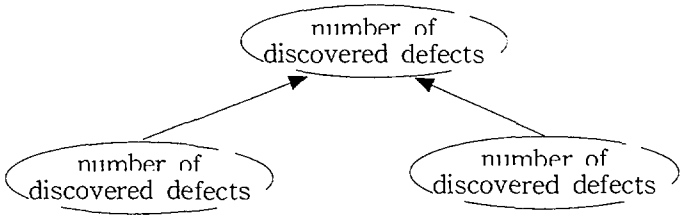
"Licensee Verification Results" 노드의 NPT

Licensee verification quality		Low				
Adequacy of comp. system spec.		Awful	Unsatisfact.	OK	Good	Wandeful
Licensee verification results	0 issues	0.1	0.22	0.9	0.9425	0.95
	A few issues	0.2	0.182	0.08	0.046	0.04
	Many issues	0.7	0.598	0.02	0.015	0.01

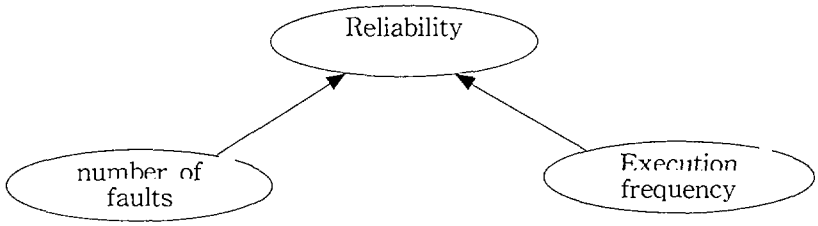
Licensee verification quality		OK				
Adequacy of comp. system spec.		Awful	Unsatisfact.	OK	Good	Wandeful
Licensee verification results	0 issues	0.05	0.065	0.15	0.7875	0.9
	A few issues	0.15	0.2475	0.8	0.1965	0.09
	Many issues	0.8	0.6875	0.05	0.016	0.01

Licensee verification quality		High				
Adequacy of comp. system spec.		Awful	Unsatisfact.	OK	Good	Wandeful
Licensee verification results	0 issues	0.02	0.0245	0.05	0.6875	0.8
	A few issues	0.08	0.203	0.9	0.2965	0.19
	Many issues	0.9	0.7725	0.05	0.016	0.01

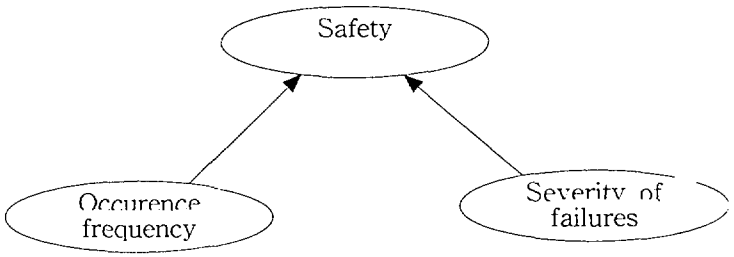
3. SERENE instantiation of idiom/templates



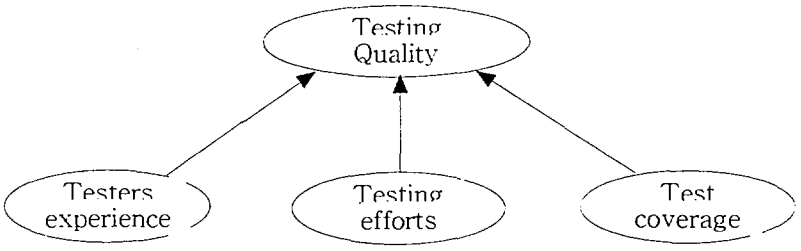
[측정 이디엄의 실증(Instantiation) : 시험]



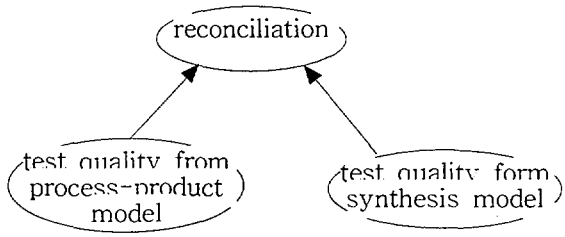
[정의/통합 이디엄의 실증 : 신뢰도]



[정의/통합 이디엄의 실증 : 안전성]

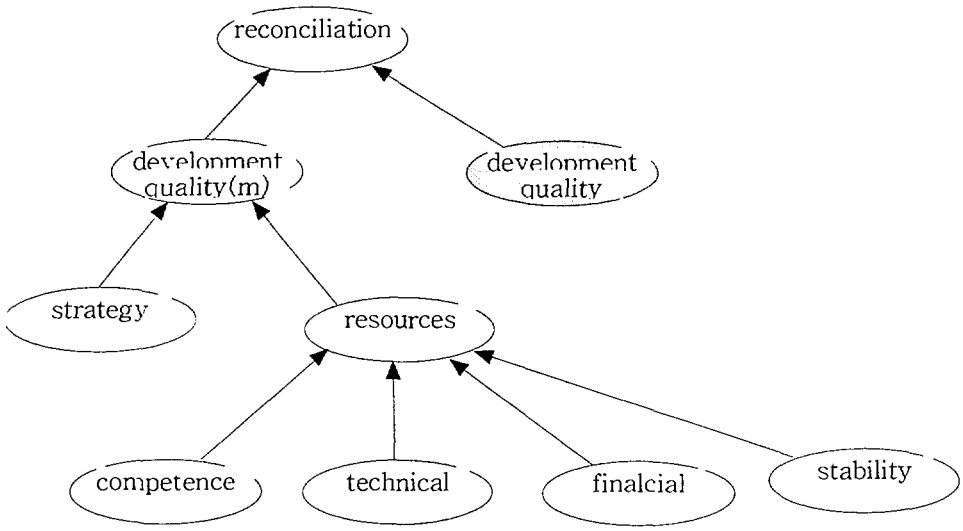


[정의/통합 이디엄의 실증 : 시험 품질]

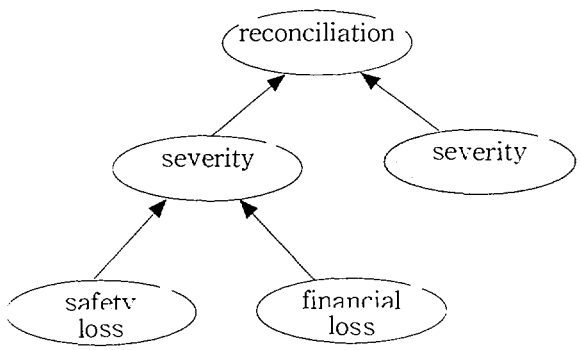


[조정 이디엄의 실증 : 시험 품질]

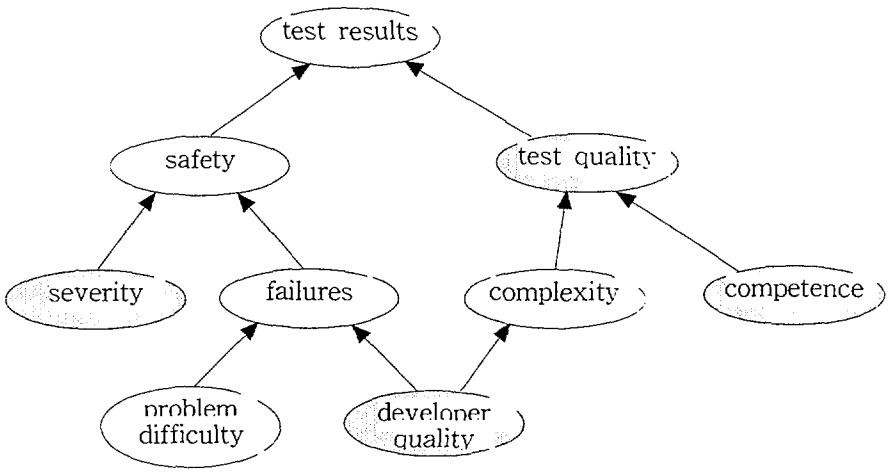
* 아래의 탬플릿에서 회색으로 표시된 노드들은 다른 BBN 모듈과 공유되고 있다는 것을 표시함.



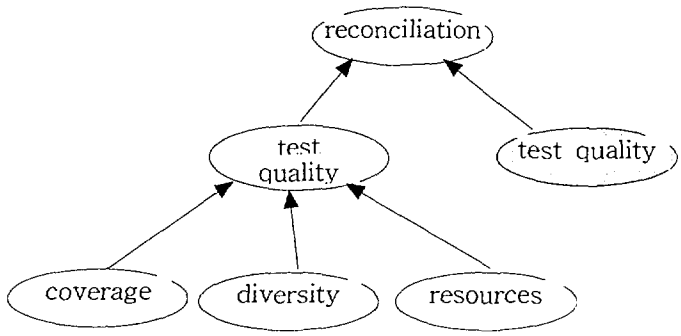
[개발자 품질 탬플릿]



[심각도(Severity) 탬플릿]



[안전성 템플릿]



[시험 품질 템플릿]

4. 외부 개발 시스템 평가 모델 from Edf Project

4.1 노드 정의

Node: "Verif. by EDF is appropriate(for Dev)"

Verif. by EDF is appropriate (for Dev)	This verification assesses some key properties of the Dev: o Is complexity of developed components mastered and justified(checked by static analysis of the source code?) o Will it lead to a maintainable system?

* Dev: Development

Node: "Verif. by supplier is appropriate(for Dev)"

Verif. by supplier is appropriate (for Dev)	This verification assesses some key properties of the Dev: o Is complexity of developed components mastered and justified(checked by static analysis of the source code?) o Will it lead to a maintainable system? o Are the programming rules respected?

(주) 같은 확인 활동(verification)에 대하여 두 번째가 더 엄격하게 만들어진 것은 공급자에 의한 확인 활동에는 다 많은 시험이 필요하고 따라서 자원(resources)의 양에 매우 민감하기 때문임.

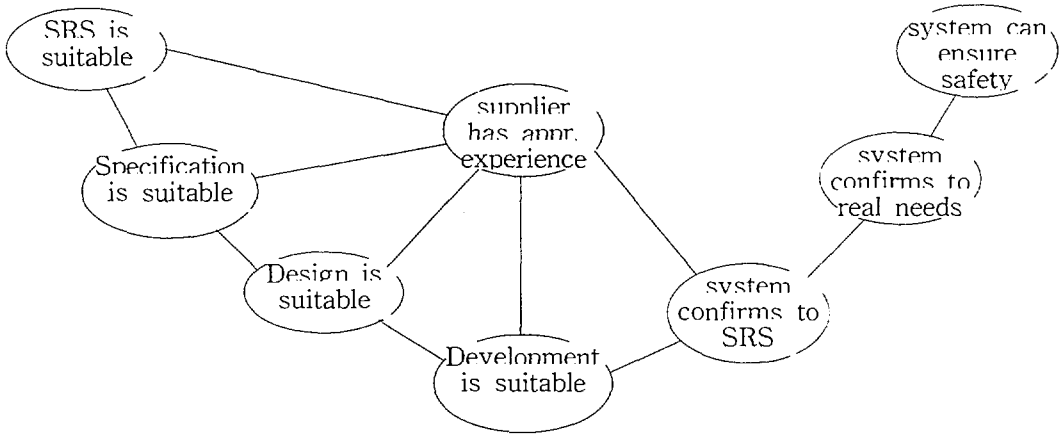
4.2 노드 확률 테이블

o 모든 노드는 2개의 상태 공간을 가진다.
  : good or poor, acceptable or not acceptable, yes or no etc.

o 모든 노드는 "object x is y"와 같은 문장과 일치시킨다.
  : (예) Requirements specification is suitable

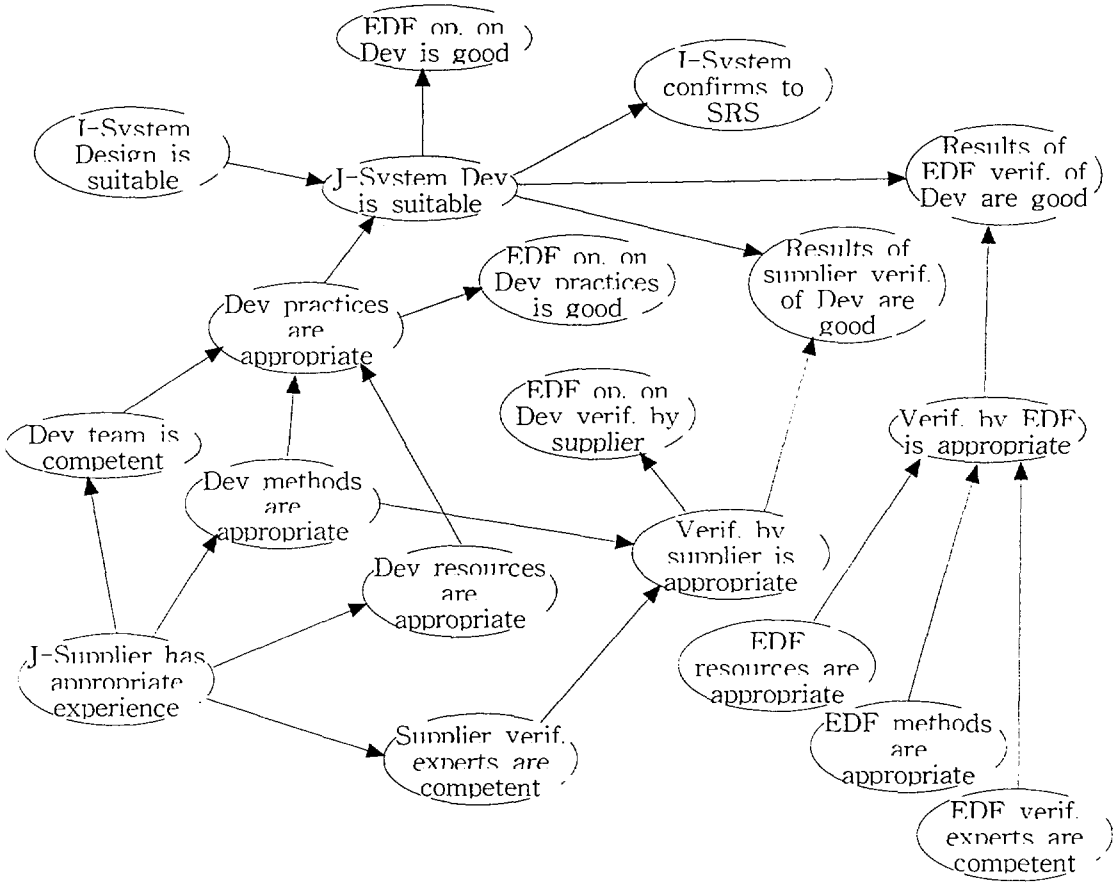
o 노드 상태에 부여할 확률 값은 {0, 0.25, 0.5, 0.75, 1}의 집합에서 선택한다. 이유는 이보다 더 상세한 값에 대한 정당화가 어려웠기 때문이다. 또한 이들 값은 다음과 같은 정성적 판단 기준을 표현한다.
  {impossible, improbable, probable, quite probable, certain}

4.3 그래프



[Global structure of the BBN]

SRS: System Requirement Specification



[Sub-net "Development is suitable"]

J- : Global structure 상에 있는 외부 sub-net

EDF op. on x : EDF opinion on x

부록 2. BBN 구성을 위한 전문가의 지식과 판단 추출 방법

BBN을 구성하는 것은 그래프의 작성과 NPT의 작성 두 부분으로 나누어지는데 이 모두에서 전문가의 지식과 판단은 중요한 역할을 한다. 그래프의 작성 부분에서는 결과 노드 또는 관심 노드에 영향을 미치는 각 노드들을 확인하고 이를 연결 구성하는데 전문가의 경험과 지식이 필요하고 NPT의 작성 부분에서는 각 노드의 상태 종류와 개수 그리고 각 상태의 확률 값을 정하는데 전문가의 지식과 판단이 필요하다. 정량적 측정이 가능한 노드의 경우에는 측정된 값을 그대로 사용 가능하나 이 경우에도 그 값이 내포하고 있는 불확실성이나 기타 사항을 고려하는 노드가 필요할 경우가 있고 이 경우에도 전문가의 판단이 추가된다. 따라서 전문가의 지식과 판단은 BBN의 구성에 가장 중요한 역할을 담당하며 따라서 이들 지식과 판단을 추출하는 방법이나 기술 그리고 절차는 만들어진 BBN 모델의 정확성에 많은 영향을 미치나 현재까지의 연구들 대부분은 아직 이 부분에 대하여 부족하다. SERENE 방법론에서는 이 부분에 대한 연구를 어느 정도 시도하였는데 이를 부록 2에서 요약 정리하였다. 그러나 SERENE의 연구 내용도 전문가의 지식과 판단 추출에 관련된 여러 가지 기법의 나열에 그치고 있으며 그래프의 작성이나 NPT의 추출 작업시 어떤 기법이 어떤 절차로 적용되어야 한다는 것과 같은 구체적이고 체계적인 내용은 되지 못하고 있다. 본 과제에서는 이 부분에 대하여 보다 구체적이고 상세한 가이드 또는 절차를 개발할 예정이다.

1. 전문가의 지식 추출 과정

구두 또는 문서로 작성된 특별히 설계된 방법을 통해 전문가의 판단을 모으는 과정으로 BBN을 개발하기 위해 전문가의 지식은 두 단계로 나누어서 추출되어져야 한다.

- o BBN의 위상(topology, 그래프)을 포함하는 개념과 인과관계에 의한 연결을 추출하는 단계
- o 각 노드의 NPT를 포함하는 확률을 추출하는 단계

추출 과정은 BBN을 구축할 대상에 따라 많은 다양성이 존재하고 각 경우마다 적절한 추출 과정이 되기 위한 요건은 다르다. 하지만 일반적으로 고려해야 할

사항들은 다음과 같다.

- o 전문가들 간의 상호 작용의 횟수와 그 정도
- o 그룹의 조정자나 인터뷰 담당자가 추출 과정에 부과해야 할 구조 (structure)의 양
- o 모임 횟수
- o 문제를 구조화하고 전문가의 판단을 추출하는데 할당할 시간
- o 직무 수행자(분석가 또는 전문가)의 선정
- o 전문가의 추론이 추출되는 대답 모드(response mode)
- o 전문가의 추론이 요청되는지 여부
- o 전문가의 판단이 추출되는 상세 정도
- o 전문가의 판단이 어느 정도 변환(translation)을 겪는지 여부 그리고 다음 단계를 위해 전문가에게 회신되는지 여부
- o 추출 작업이 직접 면접 또는 메일이나 전화를 통해 수행되는지 여부.

이런 다양한 요소에도 불구하고 기본이 되는 추출 환경은 다음의 세 가지이다.

- o 개별 면접
  인터뷰 담당자가 전문가와 개별 면접 형태로 작업 수행.
- o 그룹에 의한 상호 토론
  전문가의 그룹과 세션 조정자가 함께 모여 작업 수행.
- o Delphi 환경
  서로 떨어져 있는 전문가들이 그들의 판단을 조정자에게 제출하고 조정자는 이 내용을 익명으로 다시 각 전문가에게 배포함. 전문가들은 이 자료를 다시 검토하여 조정자에게 제출하고 이 순환 작업은 합의가 될 때까지 계속 한다.

분석가가 전문가에게 사용하는 질문 기법은 직접적 방법과 간접적 방법이 있다.

- o 직접적 방법
  여기에는 인터뷰, 질문 리스트, 사례 분석 인터뷰, interruption 분석, 프로토콜 분석, laddered grids 방법, 개념 분류 방법이 있다.
- o 간접적 방법
  개념 리스팅, 스무고개 질문법, 모사(transcription) 직무법, chapter listing 등이 있다.

일반적 추출 과정의 순서는 다음과 같다.

(1) 문제 분야와 특정 질문의 작성
(2) 질문의 상세 논술 및 개량
(3) 전문가(들)의 선정
(4) 추출 구성품(components, or building blocks)의 선택
(5) 적용분야에 알맞게 선택된 구성품 정련
(6) 추출 연습 및 필요한 교육 수행
(7) 전문가의 판단을 추출하고 문서화

추출 작업, 특히 확률 평가의 추출 작업에서 나타나는 문제는 편향(bias)인데 이것은 다음의 두 범주로 나눌 수 있다.

o 동기적 편향(motivational bias)
  추출 과정이 전문가의 생각이나 답의 내용을 바꿀 때 나타나는데 사회적 압력같은 것이 원인이 될 수 있다.
o 인지적 편향(cognitive bias)
  여기에는 전문가의 판단이 객관적 규칙이나 표준을 따르지 않는 것들이 포함된다.

이와 같은 편향을 다루는데 도움이 되는 방안들은 다음과 같다.

o 계획된 추출 과정에서 일어날 가능성이 있는 편향들을 예상한다.
o 예상된 편향에 덜 영향을 받도록 계획된 추출 작업을 재 설계한다.
o 전문가로 하여금 특정 편향의 잠재적 도입 가능성을 알게 하고 추출 과정에서 그것들에 익숙해지게 한다.
o 추출 과정에서 편향이 일어나는 것을 모니터 한다.
o 실시간으로 편향이 발생하는 것을 조정한다.
o 특정 편향이 발생했던 자료를 분석한다.

BBN 및 확률 평가와 관련되어 가장 흔하게 나타나는 편향들은 다음과 같다

o 대표성(Representativeness)
o 불확실성의 부정(Denial of uncertainty)
o 유효성(Availability)

o 조정 및 고착(Adjustment and Anchoring)
o 결합 오류(Conjunction fallacy)
o Hindsight bias
o 편차, 공분산, 상관성을 평가할 때의 어려움(Difficulties in assessing variance, covariance and correlation)
o 보수적 경향(Conservatism)
o 과신(Overconfidence)

2. 확률 추출 과정을 향상시키는 방안

추출 과정을 개선하기 전에 결정해야 할 것은 확률 평가의 유효성(goodness)을 실제로 어떻게 평가하는가를 결정하는 것으로 다음과 같은 기준이 있다.

o 확률 이론의 공리에 합치하는지 여부
o 평가자의 믿음을 정확하게 반영하는지 여부
o 평가가 반복 가능하고 안정적이며 추출 전반에 걸쳐 일관성이 있는지 여부
o 여러번의 시도에 걸쳐 예측했던 사건의 경험적 상대 빈도가 평가된 확률과 일치하는지 여부

확률 추출을 향상시키는데 영향을 주는 인자로는 다음의 세 가지를 들 수 있다.

o 작업의 특성
o 그룹 평가의 활용
o 피드백과 교육

가. 추출 작업 특성에 의해 생기는 영향들

1) 빈도 표현의 사용

전문가의 지식을 추출하는 과정에서 확률 표현을 사용하지 않고 빈도(frequency) 표현을 사용하면 "대표성", "결합 오류", "과신"과 같은 편향을 없앨 수 있다.

2) 일의 복잡도

작업의 복잡도가 추출 작업에 주는 영향에 대한 것인데 이 경우 재정적 동기(보상)를 부여하는 것이 도움이 된다고 한다. 즉 성공-보수 동기는 "조정 및 고착" 편향을 극복할 수 있다는 것이다.

3) 대답 모드(response mode)

대답 모드 역시 확률의 추출에 영향을 미치는 것으로 나타나 있다.

4) 리스트의 길이

여러 사건에 관련된 확률을 평가할 때 그 항목에 대하여 명시적으로 나열된 사건의 개수에 얽매이게 되는 경우이다. 예를 들면 N개의 대체안을 가진 사건의 확률을 구할 때 1/N의 확률에 잠재적으로 이끌려서 충분한 조정이 이루어지지 않는 것이다.

5) 측정 단위(measurement scale)

측정하는 단위가 추출 작업에 영향을 미친다. 엔지지어의 경우 대수적 (logarithmic) 단위에 익숙해져 있어서 이 단위를 사용할 때 보다 용이하게 확률을 추정할 수 있다.

나. 그룹 평가의 활용

다수 전문가의 판단을 모으는 것이 유용한 이유는;

- o 모여진 분산(distribution)은 개별적 분산에 비해 보다 나은 지식의 평가 내용을 제공한다.(즉 하나의 관찰보다는 표준 평균이 좋다)
- o 모여진 분산은 여론(consensus)의 대표로 간주될 수 있다.

그룹 평가자들로부터 추출된 확률들을 결합하는 방법에는 두 가지 형태가 있다.

- o 행위적 집합

  전문가 그룹의 멤버들 간에 상호 의견교환과 절충을 거쳐 의견일치에 도달하는 형태이다. 세션 중에 합쳐진 결과를 도출 할 수 있고 익명성에 대한 방어가 가능하다는 장점과 이런 형태에서는 전문가들 사이에 극단적 의견 편차가 종종 발생하고 또 시간을 많이 소요한다는 단점이 있다.

- o 기계적 접근법

  수학적 통계적 방법을 사용하여 복수 전문가의 자료를 하나의 추산 또는 단일 추산 분포로 결합하는 형태이다. 간단한 평균 방법부터 베이시안 기법까지 여러 가지 방법을 사용할 수 있다. 전문가들 간의 토론이 없어 쉽게 구하고자 하는 결론이 얻어지지만 "공통 편견(shared bias)"이 존재할 가능성이 높고 이런 수학적 방법으로 구해진 결론은 모든 전문가들이 거부하는 그런 답이 될 가능성이 높다.

다. 피드백과 교육의 영향

피드백과 교육에 의해서 보정(calibration) 작업은 향상될 수 있다. 이들 각각의 영향을 보면 다음과 같다.

1) 피드백

결과 피드백(outcome feedback)은 업무의 구조(환경이나 예측할 변수에 존재하는 단서들(cues) 간의 관계 등)를 강조하는 피드백에 비해 보정(calibration)이나 과신 오류를 개선하는데 덜 효과적이다. 과신의 가능성이 있는 주제를 지적해 내는 것은 보다 넓은 분포를 가진 결과를 나오게 만들지만 그래도 좋은 조정 결과를 얻거나 분포의 극단에 위치한 답을 회피하는데는 유용하지 않다. 하지만 결과에 의한 피드백은 결과의 해상도(resolution)를 높일 수 있다.

2) 교육

다음과 같은 이유로 평가자는 그들이 확률 추출 작업을 수행하기 전에 교육을 받아야 한다.

- o 주제(subjects)의 동기 부여와 추출된 확률이 어떻게 사용되는지를 포함한 그 과정의 개요를 알기 위해서.
- o 전문가가 그의 판단을 확률로 표현하는 그의 능력에 대한 확신을 개발하기 위해서. 여기서는 확률 추출과정에서 나타나는 편견에 대해서도 인지되어야 한다.
- o 추출 작업에 영향을 미치는 여러 가지 작업의 특성에 대한 이해를 위해서.
- o 평가자가 관련된 배경 정보나 관심의 대상이 되는 특정 증거에 접근할 수 있다는 것을 보증하기 위해서.

또한 좋은 확률 추출 작업이 되기 위해서는 판단 평가에 대한 기준(표준)이 있어야 하는데 다음과 같은 것들이 고려될 수 있다.

- o 작업은 평가자가 친숙한 영역과 관련된 것이어서 그에게 의미가 있어야 한다.
- o 확률 판단은 가장 유용한 통계적 모델을 사용해서 얻을 수 있는 결론에 추가하여 어느 정도의 예측 정확성을 부가할 수 있어야 한다.

o 확률적 형태로 표현된 판단들은 정상적(결정론적)으로 표현된 것에 비해 보다 정확하고 유용해야 한다.

3. 확률 평가시 사용되는 기술

확률의 추출 단계는 다음의 3 단계로 나누어진다.

o 결정론적 단계(deterministic phase)
  : 관련된 변수들이 확인되고 값이 가능한 결과에 할당한다.
o 확률적 평가 단계(probabilistic phase)
  : 주관적 확률 평가가 이루어진다.
o 정보 제공 단계(informational phase)
  : 확인 점검작업이 이루어진다.

확률적 평가 단계에서 사용되는 기술들은 그 질문 방법과 대답 모드에 따라 분류된다.

o 질문 방법
  (1) P-method : 고정된 값의 확률을 질문하는 방법이다.
    예: A 상품의 내년도 매출 개수가 2000개 이상일 확률은?
  (2) V-method : 고정된 확률의 값을 질문하는 방법이다.
    예: 동전 던지기에서 앞면이 10번 계속해서 나올 확률과 비슷하려면
      A 상품이 내년도에 몇 개가 팔려야 하는가?
  (3) PV-method : 위의 두 가지 스케일 모두에 대한 질문 방법이다.

o 대답(response) 모드
  (1) 직접 모드 : 숫자로 직접 대답하는 방식이다.
  (2) 간접 모드 : 두 개 이상의 대체안 중에서 선택하는 방법이다.

이들 질문 방법과 대답 모드별 조합에 의해 다음과 같은 기술이 있다.

o P-method/직접 모드
o P-method/간접 모드
o V-method/직접 모드
o V-method/간접 모드
o PV-method/직접 모드

서 지 정 보 양 식

수행기관보고서번호	위탁기관보고서번호	표준보고서번호	INIS 주제코드
KAERI/AR-594/2001			

제목 / 부제	원전 안전 소프트웨어의 정량적 신뢰도 평가를 위한 Bayesian Belief Nets 기술 분석
연구책임자 및 부서명 (주저자)	엄흥섭 (종합안전평가팀)
연 구 자 및 부 서 명	성태용 (종합안전평가팀), 정환성 (하나로운영팀), 박진균 (종합안전평가팀), 강현국 (종합안전평가팀) 이기영 (동력로기술개발팀)

출 판 지	대전	발행기관	KAERI	발행년	2001. 3.
페 이 지	78 p.	도 표	있음(○), 없음( )	크 기	21×29.7cm
참고사항					

비밀여부	공개(○), 대외비( ), __ 급비밀	보고서종류	기술현황분석보고서
연구위탁기관		계약 번호	

초록

원전 안전계통 디지털 시스템의 확률론적 안전성평가 연구의 한 부분으로 고 신뢰도를 요구하는 안전 소프트웨어의 정량적 신뢰도 측정 및 평가 기술과 방법론을 조사하였다. 현재 사용되고 있는 신뢰도의 직접적 간접적 평가 방법과 다양한 증거에 의거하여 종합적으로 평가하는 방법론을 조사하고 그들을 안전 소프트웨어의 신뢰도 평가에 적용할 때 생기는 문제점들을 분석하였다. 그리고 조사 분석된 기술들 중 현재 가장 가능성 있는 방안으로 대두되고 있는 Bayesian Belief Nets(BBN) 기술과 동 기술의 디지털 시스템 평가 응용 사례를 조사 분석 하고 이를 바탕으로 안전 소프트웨어의 정량적 신뢰도 평가 적용 가능성을 검토하였다.

주제명키워드 (10단어내외)	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/AR-594/2001			

Title / Subtitle	Survey of Bayesian Belief Nets for Quantitative Reliability Assessment of Safety Critical Software used in Nuclear Power Plants
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)

Publication Place	Taejon	Publisher	KAERI	Publication Date	2001. 3.
Page	78 p.	Ill. & Tab.	Yes(○), No ( )	Size	21 × 29.7cm
Note					

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

Abstract(15-20 Lines)

As part of the Probabilistic Safety Assessment of safety grade digital systems used in Nuclear Power plants research, measures and methodologies applicable to quantitative reliability assessment of safety critical software were surveyed. Among the techniques proposed in the literature we selected those which are in use widely and investigated their limitations in quantitative software reliability assessment. One promising methodology from the survey is Bayesian Belief Nets (BBN) which has a formalism and can combine various disparate evidences relevant to reliability into final decision under uncertainty. Thus we analyzed BBN and its application cases in digital systems assessment area and finally studied the possibility of its application to the quantitative reliability assessment of safety critical software.

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