2000 추계학술발표회 논문집
한국원자력학회

원전 안전소프트웨어의 정량적 신뢰도 평가를 위한 Bayesian Belief Networks 기술 분석

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

엄흥섭, 성태용, 박종균

한국원자력연구소
대전광역시 유성구 덕진동 150

요약

원전 안전계통 디지털 시스템의 확률론적 안전성평가 연구의 한 부분으로 고 신뢰도를 요구하는 안전 소프트웨어의 정량적 신뢰도 측정 및 평가에 적용 가능한 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.

Abstract

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 currently and investigated their limitations in quantitative reliability assessment. One promising methodology from the survey is Bayesian Belief Nets (BBN) which has a formalism and can combine various disparate evidence 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.

1. 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 성장모델과 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 산업분야에서 이들 안전 소프트웨어의 신뢰도 평가는 설계 방법론과 과정 그리고 시험과 같은 소프트웨어의 개발 전 단계에 걸쳐 신뢰도에 영향을 주는 여러 가지 다양한 증거들을 종합하여 전문가가 최종적으로 판단하고 있으나 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. 이러한 동향에 맞추어 개발된 소프트웨어의 안전성과 신뢰도 달성에 대한 검증 및 증명을 위한 여러 가지 연구가 진행되고 있는데 그 중 가장 가능성이 있는 기술로 여러 분야에서 연구되고 있고 또 실제 업무에 적용이 시도되고 있는 것이 Bayesian Belief Nets(BBN) 기술이다. 본 논문에서는 원전 안전계통 디지털 시스템의 확률론적 안전성 평가 연구의 한 부분으로 수행된 안전 소프트웨어의 정량적 신뢰도 평가 기술 조사에 대한 내용을 기술하였다. 제 2 장에서는 기존에 사용되고 있는 직접적 간접적 신뢰도 평가 방법과 여러 가지 증거에 근거한 종합적 평가 방법을, 제 3 장에서는 조사된 기술 중 가장 가능성 있는 방안으로 나타난 BBN 기술에 대하여, 제 4 장에서는 BBN 을 소프트웨어의 평가 분야에 적용한 사례를, 그리고 제 5 장에서 분석 결과 및 결론을 기술하였다.

2. 기존의 소프트웨어 정량적 신뢰도 평가 기법

소프트웨어의 신뢰도를 분석하고 평가하기 위한 측정 방법이나 모델에 대하여는 많은 연구가 수행되어 지금까지 70 여 가지에 이르는 신뢰도 관련 측정 기법과 모델들이 발표 되었고 또 이들에 대한 검증이 시도 되어 왔다. 소프트웨어어의 신뢰도를 포함한 품질 전반에 걸친 각종 측정방법과 모델은 [1][9]에 종합적으로 정리되어 있고 신뢰도의 분석과 평가 그리고 엔지니어링에 초점을 둔 기법과 모델들은 [2]에 정리되어 있다. 그리고 디지털 시스템의 확률론적 안전성 평가(Probabilistic Safety Assessment: PSA)에 적용하기 위한 소프트웨어 신뢰도의 정량적 평가 기법 고유 현안에 대해서는 [3]에 정리되어 있다. 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 있고 모델의 잘못된 명세나 부적절한 데이터의 사용과 같은 여러 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 [3][4]. 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 있으나 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 내구하기 어렵다.
- o 소프트웨어는 주로 새로운 시스템에 구현되기 때문에 과거의 성공적인 설계로부터의 경험이나 지식을 활용하기 어렵다.
- o 디지털 시스템은 일반적으로 불연속적인 입력 출력 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.

이와 같은 관점에서 현재 많이 사용되고 있는 기법과 모델에 대하여 정리해 보면 다음과 같다.

2.1 직접적 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 적용 영역에 따라 다르기는 하지만 실제와 동일하거나 유사한 환경을 구축하여 시험을 수행하는 것은 매우 어려운 일로 이 부류의 방법이 가진 근본적 문제점 중 하나이다. 직접적 측정 방법은 시험 기반(test-based) 신뢰도 추정과 신뢰도 성장(reliability growth based)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 고장/요구(failure per demand) 확률과 같은 신뢰도 추정을 위한 파라 미터(parameter)를 구하는 것이다. 이 방법은 직접적으로 신뢰도를 추정할 수 있는 숫자를 얻을 수 있다는 장점이 있는 반면, 문제점으로 지적되고 있는 것은;

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

(2) 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

(3) 요구되는 신뢰도가 높은 경우 그것을 만족하려면 현실적으로는 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 추세를 관측하고, 그 다음 이 추세로부터 보외법(extrapolation)을 근간으로 하는 모델(reliability growth model)을 만들어 내어 신뢰도 증가 추세(trend)를 만들고 이를 미래까지 확장한 후 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 유사해야 한다는 점, 그리고 마지막 결함 수정이 신뢰도에 미치는 효과가 최소한 그 이전의 결함 수정 활동들의 평균적 신뢰도 상승 효과보다 나쁘지 않아야 한다는 점이 있는데 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 가장 널리 사용되는 방법론으로 여기에 속한 각종 모델과 특정 경우에서 만들어진 고장 데이터에 의한 실험들은 [7][8][2]에 나타나 있다.

2.2 간접적 측정 방법

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

가. 개발과정과 방법론에 근거한 신뢰도 증명

이 방법은 충분히 훌륭한 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-practice”가 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 운전되면서 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 다양성(diversity)을 사용한 fault tolerance 에 의한 신뢰도 증명

하드웨어의 신뢰도 엔지니어링에서는 설계의 다양성에 의해 고 신뢰도 시스템의 개발이 가능하다. 이것은 동일한 기능을 가진 서로 다른 버전의 고장 프로세스가 서로 독립적이라는 가정이 현실성이 있고 경험적 자료에서 증명되어 있기 때문이다. 그러나 소프트웨어의 경우에는 이런 설계 다양성에 의해 개발된 버전들의 고장 독립성 가정은 아직까지 명확한 답이 나와있지 않고 있다. 이 문제에 대한 실험을 보면 [14]의 경우에는 버전들의 고장 프로세스가 서로 독립적이 아니라는 결과를 보여주고 있고 [15][16]의 경우에는 설계의 다양성에 의해 어느 정도 신뢰도가 향상된다는 결과를 보여주고 있다. 그러나 이 경우에도 향상된 정도를 측정하는 문제와 전체의 신뢰도를 구하기 위해 필요한 각 버전의 신뢰도를 별도로 구해야 하는 문제가 생긴다.

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 작성 되었다면 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 증명은 에러에 취약하다는 점. 작업을 수행하는 인간의 오류나 지원 도구의 결함 그리고 자동화된 도구의 결함 등 이 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.

2.3 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 틀에서 문제점으로 지적되고 있는 것은;

o 평가에 사용되는 각 증거에게 값을 부여하는 문제와

o 각 증거들은 서로 다른 특성과 성질을 가지고 있어서 이들을 결합하는 것이 어렵다는 점이다. 특히 신뢰도에 영향을 미치는 증거들 간의 종속성(dependency) 관계가 대부분 공식적으로 밝혀지지 않고 있어서 더욱 그러하다.

이들 문제점은 현재 전문가의 판단에 의존하여 처리되고 있는데 각 전문가들이 이 문제들을 처리하는 과정은 매우 비공식적(Informal)이다.

위에서 지적된 문제를 해결하기 위해 최근에 제안되고 있는 방법으로 Bayesian Belief Nets 방법론이 있다. BBN 은 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.

3. BBN

BBN은 Belief Nets, Causal Probabilistic Networks, Causal Nets, Graphical Probability Networks, Probabilistic Cause-Effect Models, Probabilistic Influence Diagrams 라고도 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 입증되었다. 최근의 가장 성공적인 응용은 마이크로 소프트의 오피스에 내장된 "Answer wizard"와 지능형 프린터 고장 진단 시스템으로 알려져 있다. 소프트웨어 엔지니어링 분야에서 사용되기 시작한 것은 Sabre airplane reservation system의 디버깅 단계에서 결함을 발견하고 진단하는데 사용한 것이며 소프트웨어의 신뢰도와 안전성 분석 및 평가에 적용되기 시작한 것은 90년대 중반부터 이다.

3.1 BBN 개요

BBN은 관련된 변수들을 인과관계에 의해 구성하여 모델링하고 변수들 간의 종속성 정도를 조건부 확률로 나타낸 다음 주어진 여러 가지의 증거에 근거해 베이시안 확률정리를 사용하여 정량적으로 결론을 이끌어 내는 방법론이다.

가. BBN 구성 요소

BBN은 노드(Node), 노드들 사이를 연결하는 연결선(arcs 또는 directed edges) 그리고 각 노드 에 속한 확률 테이블(Node Probability Tables: NPT 또는 Conditional Probability Table: CPT)로 구성된다. 노드는 모델에 포함된 변수들을 나타내며 노드 연결선은 노드간의 인과관계를 나타낸다. BBN 상의 노드는 무작위 변수를 나타낸다. 이들 무작위 변수는 몇 개의 상태를 가지고 있으며(예: "yes" or "No") 각 상태의 확률 값의 합은 1이 된다. 각 노드에 연결된 노드 확률 테이블은 노드간의 연결 강도를 결정하며 부모 노드의 각 상태에 대한 조건부 확률로 표현된다.

나. 용어 및 개념

BBN의 주요 용어와 개념을 정리하면 다음과 같다.

1) 증거(evidence)

노드(또는 변수)의 증거는 그 노드 상태의 확실성에 대한 진술이며 확정증거와 추정증거의 두 가지 종류가 있다.

(i) 확정증거(hard evidence)

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

(ii) 추정증거(soft evidence)

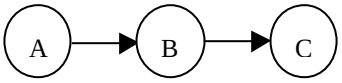
확정증거에 해당되지 않는 증거를 추정증거(soft evidence)라고 한다. 여기에는 노드의 기존 확률 값을 변경시킬 수 있는 어떠한 증거들도 포함된다.

2) 세가지 형태의 d-connection(dependency connection, 의존상태 연결)과 d-separation

BBN 에서는 3 가지의 노드 연결 상태가 있는데 serial connection(직렬 연결), diverging connection(분기 연결), conversing connection(수렴 연결) 이다. 그리고 이 연결 상태에 따라 종속성 연결 상태(d-separation)가 정의되는데 이는 BBN 계산의 근간이 된다.

(i) serial connection(직렬 연결)

직렬 연결은 노드 A,B,C 가 다음과 같이 연결되어 있는 경우이다.

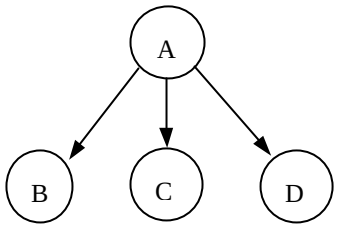


[그림 1. 직렬 연결]

직렬연결에서는 노드 A 가 노드 B 에 영향을 주고 차례로 노드 B 가 노드 C 에 영향을 미친다. 즉 노드 A 의 증거는 노드 B 의 확실성을 변경시키고 뒤이어서 노드 C 의 확실성을 변경시키는데 노드 C 에서 노드 B 와 노드 A 로의 영향도 마찬가지이다. 하지만 노드 B 의 상태가 분명하게 알려지면 연결 채널이 차단되어 노드 A 와 노드 C 는 독립적이 된다. 이럴 때 노드 A 와 노드 C 는 주어진 노드 B 에 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 확률 테이블(NPT)을 가지고 있는데 직렬연결에서 이들 확률의 변화를 보면, 노드 A 나 노드 C 에 증거(hard evidence)를 입력할 경우 이는 노드 B 의 확률분포를 변경시킨다. 하지만 노드 B 에 증거를 입력하면 노드 A 와 노드 C 는 노드 B 에 주어진 증거에 의해 조건부로 독립적 상태가 되어 서로 간 영향을 주지 않게 된다.

(ii) diverging connection(분기 연결),

분기연결은 노드들이 다음 그림과 같이 연결되어 있는 경우이다.



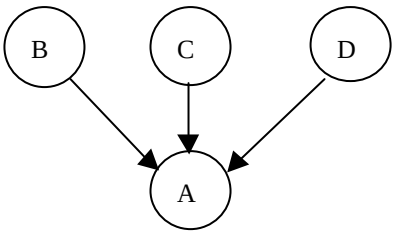
[그림 2. 분기 연결]

분기연결에서는 노드 A 의 증거가 A 의 자식 노드들 B,C,D 에 영향을 준다. 그리고 부모 노드인 A 의 상태가 알려져 있지 않은 상태에서는 자식 노드들이 서로 종속된 상태로 되어 있어 이들 자식 노드들 중 하나에 증거가 입력되면 이는 다른 자식 노드들에 영향을 미친다. 하지만 부모 노드 A 의 상태가 알려져 있으면 자식 노드들 중 하나(예: B 노드)에 입력된 증거는 다른 자식 노드들(예: C, D)에 영향을 미치지 않는다. 이를 노드 B,C,D 가 주어진 노드 A 의 상태에 대하여 종속상태 분리 되었다고 한다(B,C,D are d-deparated given A). 결론적으로 분기연결에서 각 노드들의 증거들은 부모 노드 A 의 상태가 알려져 있지 않은 한 서로 전달되고 노드 A 의 상태가 알려져 있으면 그

것은 자식 노드들 간의 연결을 차단한다.

(iii) conversing connection(수렴 연결)

수렴연결은 노드들이 다음 그림과 같이 연결되어 있는 경우이다.



[그림 3. 수렴 연결]

위의 그림과 같은 수렴연결에서 부모 노드들 B,C,D의 증거들은 자식 노드 A에 영향을 미치는 것은 당연하다. 문제가 되는 것은 부모 노드들인 B,C,D 간에 서로 영향력을 미치는가 인데 그것은 다음과 같이 정의된다.

(i) 만약 자식 노드 A의 상태가 알려져 있지 않다면 부모 노드들은 서로 독립적 상태(independent)가 되어 한 노드의 증거가 다른 노드의 확실성에 영향을 미치지 않는다.

(ii) 만약 어떤 증거(노드 A에 직접 들어온 증거 또는 A의 자식 노드로부터 영향을 받은 증거)가 자식노드 A의 확실성에 영향을 주었다면 노드 A의 부모 노드 B,C,D는 서로 의존적 상태로 변하여 이들 노드들 중 하나의 확실성 변화는 다른 부모 노드들에게 영향을 미치게 된다.

3.2 BBN 모델링

BBN 모델링은 관련 변수(노드)의 확인 및 그래프(topology) 작성, 노드의 상태 공간(state space) 설정과 조건부 확률 값 부여, 그리고 증거의 입력과 이에 따른 계산으로 이루어진다.

가. 노드 확인 및 그래프 작성

목표 노드의 값을 구하기 위해 관련된 모든 사건이나 증거를 결합하는 과정이다. 이를 위해서는 모델하고자 하는 영역의 모든 변수들이 확인 되어야 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. 노드 간의 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 상위의 추상적 노드에서 하위의 추상적 노드로 방향을 정하는 방법을 따른다. 이 작업이 끝나면 마지막으로 각 노드의 상태 공간을 설정하게 된다. 한 노드에 연결된 부모 노드가 많은 경우 상태 공간이 많아져서 노드의 확률 값을 구하는 것이 어렵게 되는 경우에는 noisy-or 나 divorcing[19]과 같은 기법이 사용된다.

나. 노드 확률 테이블(NPT) 작성

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.

(1) 문제 영역과 특정 질문의 작성

(2) 질문의 정련

(3) 전문가(들)의 선성

(4) 추출 구성품의 선택

(5) 적용분야에 알맞게 선택된 구성품 정련

(6) 추출 연습 및 필요한 교육 수행

(7) 전문가의 판단을 추출하고 문서화

NPT를 작성하는 과정에서 주요한 점은 편향(bias)의 회피이다. 이것은 BBN방법론 뿐 아니라 전문가의 지식을 추출하는 모든 작업에서 나타나는 현상인데 이에 대한 처리 방법은 [17],[18]에 정리되어 있다.

다. 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 갱신 시키는 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. BBN의 계산 방법들은 [19]에 정리되어 있다. 이와 같은 계산 작업은 수작업으로는 불가능할 정도로 복잡한데 80년대 후반에 개발된 효과적인 계산 알고리즘과 이를 구현한 소프트웨어 도구는 BBN을 실용적 문제에 적용할 수 있게 하는 중요한 역할을 하였다.

4. BBN 적용 사례

BBN의 적용 사례 중 소프트웨어의 신뢰도나 안전성 평가 그리고 이와 관련된 분야에 적용된 연구 사례는 다음과 같다.

- 컴퓨터 시스템 명세 문서의 안전성 분석을 위한 BBN[20]
- 안전중요(safety critical) 디지털 시스템의 승인 과정을 위한 BBN[21]
- 대형 BBN 구성을 위한 방법론과 도구 개발[22]
- 외부 개발 I&C 시스템의 검증 및 승인을 위한 BBN[23]

이들 연구의 주 목적은 안전성과 신뢰도의 평가 케이스에 BBN을 시험적으로 적용해 보고 그 가능성과 적용과정에서 나타난 문제를 도출하는 것과 나타난 문제점들을 해결하기 위한 방안을 모색하는 것이 주로 되어있다. BBN 모델링 작업 순서대로 이들 사례의 주요 결과를 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 되지만 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. [22]에서 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 모색하는 것이 시도 되었다.

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 점과 문제의 복잡성 그리고 인간 마음의 근본적 한계성에서 주로 기인하는데 사례별로 나타난 내용을 요약하면 다음과 같다.

(1) 불확실성을 확률로 표현하는데 따르는 어려움. 이 점은 다른 분야에서 확률을 다루고 있는 전문가들에게서도 나타난다. 누구나 주관적 확률적 믿음(belief)을 가지고는 있으나 그들을 명시적으로 만드는 데는 훈련이 되어있지 않은 것이 주된 이유이다.
(2) 모델 대상의 의존 상태가 너무 복잡하고 또 값을 추출해 작성해야 할 테이블의 수가 많다.
(3) BBN 과 같은 언어가 없었기 때문에, 이전에는 상세하고 명시적으로 나타내 보지 않았던 복잡한 구조를 기술하는데 따르는 어려움이 있다.
(4) 기술해야 할 상황이 너무 복잡하기 때문에 해당 지식 전문가는 잠재적으로 모델을 단순화 하려는 경향이 생겨 모델링이 부정확하게 된다.
(5) 전문가의 경험으로부터 올바른 추론을 끌어내는데 따르는 어려움이 있다.

다. 계산

BBN 의 계산은 매우 복잡하나 효과적인 계산 알고리즘의 발견과 이를 구현한 소프트웨어의 개발로 실용적 측면에서 문제가 없는 것으로 나타났고 군사용 운송장비의 신뢰도 평가 개선 연구에서는 개발된 소프트웨어 도구가 현실적 문제를 해결하는데 충분한 기능을 갖춘 것으로 발표되었다.

조사된 사례 중 특기 할만한 것은 대형 BBN 구성을 위한 방법론과 도구 개발[22] 경우인데 여기서는 실제 문제에 BBN 적용할 때 나타날 수 있는 문제점들 특히 그 동안 주먹구구식으로 행해진 BBN 의 그래프 작성을 체계적으로 수행하기 위한 방법론과 모델링상의 위험을 관리하기 위한 방안 그리고 이 방법론을 지원하기 위한 도구를 개발하였다. 동 사례의 내용을 요약하면 다음과 같다.

o 개요

현재 일반적으로 채택되고 있는 표준-기반(standards-based) Programmable Electronic System(PES)의 안전성 평가 접근방법에서는 결여된 안전성의 정량화를 해결하기 위한 것으로 BBN 을 사용하여 PES 의 안전성을 추론하는 것이 SERENE 과제[22]의 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 안전 변수인데 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 61508 이나 CASCADE 과제에서 개발된 GAM 과 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.

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

o 연구 결과

위에서 기술된 목적을 위해 SERENE 방법론을 개발하였다. 이 방법론은 안전변수 준비(argument preparation), 안전변수 구축(argument construction), 안전변수 사용(argument use)의 세가지 활동으로 구성되며 동 방법론을 지원하기 위한 BBN 모델링 도구로서 SERENE 도구를 개발하였는데 그 구성요소는 다음과 같다.

(i) BBN 계산 엔진
(ii) Graphical User Interface : BBN 의 준비, query, 프린트 기능
(iii) SERENE 방법과 도구의 도움 파일
(iv) Templates : 샘플 BBN 으로 사용자가 BBN 을 구축할 때 개조하거나 그대로 사용할 수 있다.

그리고 동 도구의 특징은;

(i) SERENE 이디엄(idiom)과 결합기능(join operation)을 사용하여 BBN 을 구성할 수 있도록 BBN 을 계층적 구조화
(ii) 불연속 및 연속 변수 모두를 사용할 수 있는 표준 분산
(iii) 민감도 분석(sensitivity analysis) 기능 이다.

5. 분석 결과 및 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 BBN 기술은 아직 초기 단계이기는 하나 소프트웨어의 신뢰도와 안전성을 정량적으로 평가하는 분야에서 유용한 결과를 보여주고 있다. 지금까지 연구 결과를 토대로 소프트웨어의 정량

적 신뢰도 평가에 유용한 BBN 방법론의 장점 및 효용성을 요약해 보면 다음과 같다.

- o 평가에 있어서 필연적으로 수반되는 불확실성과 무지를 명시적으로 모델링하고 원인과 결과의 관계를 명시적으로 나타낸다.
- o 다양한 유형의 정보를 결합할 수 있게 해 준다. BBN 이 다룰 수 있는 다양한 유형의 증거들은 다음과 같은 것들이 있다.
  - 다양한 과정(process)과 제품(product) 변수들
  - 경험적 증거와 전문가의 판단
  - 실제의 인과관계(원인-결과 관계)
  - 불확실성
  - 불완전한 정보 그리고 결여된 정보
- o 감추어져 있던 여러 가지 가정들을 명시적으로 표현 함으로서 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 사용과 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 기법들을 활용하여 모델링 할 수 있다.

물론 사례 분석에서 나타난 바와 같이 그래프의 작성과 NPT 작성시 발견된 많은 문제점이 있기는 하지만 이런 문제점들을 해결하기 위한 연구가 각 분야에서 현재 활발이 진행 중이고 실제 케이스에의 적용도 계속 진행되고 있으므로 실용성 있는 결과가 나올 것으로 기대된다. 또한 BBN 은 소프트웨어의 평가 및 검증 분야 뿐 아니라 소프트웨어 엔지니어링의 다른 분야에서도 유용한 도구로 인식되고 있으며[20] 그 주요 내용은 i) COTS 시스템의 평가에 적용, ii) 설계 방법론에 적용, iii) 시험 케이스 개발에 적용, iv) 시험 케이스의 안전성 평가에 적용, v) PSA 에 적용 등이다.

참고문헌

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

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

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

[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 n 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, http://www.hugin.dk/SERENE, 1999

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